

SOLPS-ITER

November 2, 2020

Preface

This manual is built from the following code repository versions:

```
SOLPS-ITER version : 3.0.7-8-g2558728
B2.5 version : 3.0.7-2-ga905cd6
Eirene version : 3.0.7-5-ga5646dd
Carre version : 3.0.7
DivGeo version : 3.0.7-1-gbfdc2f9
```

This is the combined work of many authors. Started by D.P. Coster, it includes contributions from

- D.P Coster and O. Wénisch, the original **b2plot** manual (Appendix **I**)
- K. and A.S. Kukushkin, the **DG** manual (Appendix **K**), with revisions by A. Chin and D. Stotler
- M. Stanojevic, Appendix **D** on boundary conditions
- X. Bonnin, for many contributions and the **SOLPS-ITER** updates
- S. Voskoboinikov, E. Kaveeva, and I. Senichenkov, for the **SOLPS-ITER** model equations (Appendix **C**)

All errors, omissions *etc.* should be e-mailed to `Xavier.Bonnin@iter.org`.

Contents

Contents	3
List of Figures	13
List of Tables	15
1 Introduction	17
1.1 Obtaining and updating the code	18
1.2 Requirements	20
1.3 Initial set-up	21
1.4 Before every SOLPS-ITER session	26
1.5 Examples	26
1.6 Using a central installation	27
2 The structure	29
2.1 Setup and configuration files	30
2.2 modules	32
2.2.1 modules/B2.5	33
2.2.2 modules/Eirene	37
2.2.3 modules/amds	39
2.2.4 modules/adas	39
2.3 lib	40
2.4 runs	40
2.5 scripts	41
2.5.1 scripts/commands	54
2.5.2 scripts/matlab	54
2.5.3 scripts/palettes	54
2.5.4 scripts/Balance	54
2.5.5 scripts/nc2text_simple	54

2.5.6	scripts/MatlabPostProcessing	54
2.5.7	scripts/Matlab_SPb	54
2.6	data	55
3	Running the standalone version of B2.5	56
3.1	b2ag, setting up the geometry	60
3.1.1	b2ag.dat	60
3.2	b2ah, setting up the default physics parameters	61
3.2.1	b2ah.dat	61
3.3	b2ar, setting up the atomic physics rates	66
3.3.1	b2ar.dat	66
3.4	b2ai, the default initial plasma state	67
3.4.1	b2ai.dat	67
3.5	b2mn, the main program	69
3.5.1	b2mn.dat	69
3.5.2	b2.boundary.parameters	70
3.5.3	b2.transport.parameters	77
3.5.4	b2.numerics.parameters	79
3.5.5	b2.transport.inputfile	80
3.5.6	b2.sources.profile	83
3.5.7	b2.neutrals.parameters	88
3.5.8	b2.atomic_physics_rescale.parameters	88
3.5.9	b2.user.parameters	88
3.5.10	b2.XXX.parameters	89
3.5.11	While the code is running	89
3.6	b2pl, the plotting program	90
3.7	b2yg, plotting the geometry	107
3.8	b2yh, plotting the run parameters	109
3.9	b2yi & b2yp, plotting the plasma state	110
3.10	b2yr, plotting the atomic physics rates	113
3.11	b2ym, producing movie output	114
3.12	b2yn, display the progress of inner B2.5 iterations	115
3.13	b2yq, display the evolution data	116
3.14	b2yt, changing from one grid size and species set to another	118
3.15	b2co, recovering from a crash	121
3.16	b2uf & b2fu, converting the plasma output file	122

3.17	b2rd, fetching an archived MDSPLUS run	123
3.18	b2ymb, post processing for monoblock heat transfer model	124
3.19	b2yrp, post processing for reflector plates heat transfer model	125
3.20	b2ydm, post processing of dome wall loading	127
3.21	b2_ual_write, writing last state to IDS	128
3.22	b2xx, the others	130
4	Running the standalone version of Eirene	131
4.1	Specific SOLPS-ITER blocks in the Eirene input file	131
4.1.1	Block 14. Data for interfacing routine INFUSR	131
4.1.2	Block 15. Data for interfacing routine GEOUSR	132
4.1.3	Information for MPI	134
5	Running the coupled version of B2.5-Eirene	135
5.1	Input files	135
5.1.1	b2ag.dat	135
5.1.2	b2ai.dat	135
5.1.3	b2ar.dat	136
5.1.4	b2ah.dat	136
5.1.5	b2mn.dat	144
5.1.6	b2.boundary.parameters	145
5.1.7	b2.numerics.parameters	146
5.1.8	b2.transport.parameters	146
5.1.9	b2.neutrals.parameters	147
5.2	Output files	148
5.3	Changing dt	152
5.4	Averaging scheme	153
5.5	b2ye, statistical error assessment	154
6	Using MDSPLUS	158
6.1	SOLPS tree structure	159
6.1.1	SOLPS	159
6.1.2	IDENT	159
6.1.3	SNAPSHOT	160
6.1.4	TALLIES	164
6.1.5	TIMEDP	166
6.1.6	MOVIES	170

6.1.7	DIAG	170
6.2	Accessing data from the MDSPLUS server	171
6.2.1	Using standard experimental data viewing programs	171
6.2.2	Using IDL	173
6.2.3	Using MATLAB	174
6.2.4	Using Python	174
6.3	Querying the database	181
6.4	Coupling to other codes via MDSplus	183
6.5	Some information about the server	187
7	Running DivGeo	188
8	Running Carre	190
9	Running agg	194
A	Internal B2.5 documentation	198
A.1	b2cdca.F	198
A.2	b2cdcr.F	201
A.3	b2cdcv.F	216
A.4	b2cdci.F	243
A.5	b2cdcn.F	290
B	The physics of B2.5	315
B.0.1	Derivation of parallel viscosity term in momentum equation	322
B.1	Boundary conditions with drifts and currents	323
B.2	Problems for a fluid description of the edge plasma	324
B.3	Neutral models	326
B.3.1	Fluid neutrals	326
B.3.2	First-flight model	328
B.3.3	Kinetic neutrals: B2.5-Eirene coupling	328
B.4	Classical transport: Braginskii, Balescu and Zhdanov formulations	331
C	Model equations for B2.5	334
C.1	Sign conventions	334
C.2	Continuity equations for particle conservation	334
C.2.1	Source terms in density equations	337
C.2.2	Velocities	338

C.2.3	Particle transport coefficients	339
C.3	Parallel momentum conservation equations	340
C.3.1	Sources for parallel momentum equations	340
C.3.2	Parallel momentum fluxes	343
C.3.3	Parallel momentum transport coefficients	344
C.4	Electron energy equations	345
C.4.1	Electron energy sources	346
C.4.2	Poloidal electron energy fluxes	346
C.4.3	Electron energy transport	347
C.4.4	Radial electron energy fluxes	350
C.5	Ion energy equations	351
C.5.1	Ion energy sources	351
C.5.2	Poloidal ion energy fluxes	352
C.5.3	Poloidal ion energy transport coefficients	353
C.5.4	Radial ion energy fluxes	355
C.5.5	Radial ion energy transport coefficients	355
C.6	Current continuity equation	355
C.6.1	Poloidal currents	356
C.6.2	Radial currents	357
C.6.3	Conductivities	358
C.7	Typical boundary conditions	358
C.7.1	Particle balance equations	358
C.7.2	Parallel momentum balance for ions	363
C.7.3	Current continuity equation	367
C.7.4	Energy balance equation for electrons	368
C.7.5	Energy balance equation for ions	371
C.8	Total energy balance	374
C.9	Parallel viscosity	378
C.10	Scaling of residuals	379
D	Boundary conditions in B2.5	380
D.1	Introduction	380
D.2	Analytical Expressions for the Standard ("Old") Boundary Sources in the B2.5 Code	381
D.2.1	Definitions of Physical Parameters Used in the Analytical Expressions for the Standard Boundary Sources	381
D.2.2	Standard Boundary Sources	383

E	Quantities stored in b2time.nc	390
F	Quantities stored in b2tallies.nc	397
G	Quantities printed by 'b2wdat_iout' = '4'	400
H	Region definitions	411
H.1	Regions for 1-D cases	412
H.2	Regions for limiter cases	413
H.3	Regions for SN	414
H.4	Regions for CDN	416
H.5	Regions for DDN	418
H.6	Regions for periodic BC stellarator island case	420
I	b2plot manual	422
I.1	Introduction	422
I.2	Commands	423
I.3	Common errors	447
I.4	Available Functions and Operators in b2plot	448
I.4.1	Data plotting commands	448
I.4.2	Data manipulating functions	452
I.4.3	Arithmetic commands	458
I.4.4	Stack manipulating commands	466
I.4.5	Page format parameters	472
I.4.6	Graph format commands	472
I.4.7	Graph appearance commands	474
I.4.8	Overlays to contour and line plots	477
I.4.9	Changing contour parameters	482
I.4.10	miscellaneous	484
I.4.11	Still need to do	490
I.5	Commands taking a string parameter	490
I.5.1	<extra label> extl	490
I.5.2	<palette file> cltb	491
I.5.3	<chord file> chor	492
I.5.4	<chord file> echor	492
I.5.5	<chord file> chpl	492
I.5.6	<command file> cmd	494

I.5.7	<CDF command file> cmdcdf	494
I.5.8	<chord file> chbx	495
I.5.9	<position file> chvl	495
I.5.10	<flux info> flux	495
I.5.11	<chord file> chmp	495
I.5.12	<chord file> cam	495
I.5.13	<chord file> echbx	495
I.5.14	<position file> echvl	495
I.5.15	<chord file> echmp	495
I.5.16	<chord file> ecam	495
I.5.17	<CDF file name> movi	495
I.5.18	<istart,I> <iend,I> <varname, S> movie	496
I.5.19	<istart,I> <iend,I> <filename, S> <varname, S> omovie	496
I.5.20	<sonnetfile> sonn	496
I.5.21	<output_sonnet file> oson	496
I.5.22	<z _{shift} ,R> <UEDGE grid file> uedgegrd	496
I.5.23	<filename> <H123> <reac> <crc> amju	496
I.5.24	<reac> adas	496
I.6	Environment variables	496
I.7	Examples	497
I.7.1	Mesh plot	497
I.7.2	Plotting plasma solution data	497
I.7.3	Chord plots	497
I.7.4	Wall plots	498
I.7.5	Target plots	499
I.7.6	Radiation plots	500
I.7.7	Fluxes and values plot	501
J	Miscellaneous	504
J.1	DIMENSIONS.F	504
J.2	Making movies	506
J.3	Compiling B2.5	506
J.4	Defines	527
J.4.1	SOLPS-ITER	527
J.4.2	Carre	528
J.4.3	DivGeo	529

J.4.4	B2.5-Eirene	529
J.4.5	B2.5	530
J.4.6	Eirene	538
J.5	NAG calls in SOLPS	540
J.5.1	B2.5	540
J.5.2	EIRENE	541
J.6	Performance monitoring	541
K	DG manual: Version 2.10	542
K.1	Introduction	543
K.1.1	Purpose of DG	543
K.1.2	DG objects	543
K.2	Using DG	545
K.2.1	Starting DG	545
K.2.2	Main window	545
K.2.3	Using tools	546
K.2.4	Using Undo	547
K.3	Working with models	547
K.3.1	Opening an existing model	547
K.3.2	Creating a new model	547
K.3.3	Saving a model	547
K.3.4	Quitting DG	548
K.4	Working with elements	548
K.4.1	Converting a template	548
K.4.2	Creating elements manually	548
K.4.3	Editing elements	549
K.4.4	Changing external normals	549
K.4.5	Renumbering elements	549
K.5	Controlling how objects are displayed	550
K.5.1	Changing the zoom factor and location	550
K.5.2	Hiding and displaying objects	550
K.5.3	Rotating the view	550
K.5.4	Stretching and shrinking the view	550
K.5.5	Top-down view	551
K.6	Working with magnetic surfaces and grid points	552
K.6.1	Equilibrium conversion and manipulation	552

K.6.2	Importing equilibrium	554
K.6.3	Output mode	555
K.6.4	Creating and editing surfaces	555
K.6.5	Defining the X-point	558
K.6.6	Limiter configurations	558
K.6.7	Creating and editing grid points	559
K.7	Working with chords	560
K.7.1	Creating chords	560
K.7.2	Modifying chords	560
K.8	Working with sources	561
K.8.1	Creating and editing sources	561
K.9	Working with meshes	561
K.9.1	Importing a SONNET format mesh	561
K.9.2	Modifying a mesh	561
K.9.3	Repairing mesh cells using rotated view and variable aspect ratio	562
K.9.4	Defining cells	562
K.10	Working with variables	563
K.10.1	Basics	563
K.10.2	Working with layers	564
K.10.3	Setting variables	564
K.11	Creating output files	566
K.11.1	Types of output files	566
K.11.2	Producing output files	566
K.12	Customizing DG	566
K.12.1	Installing DG	566
K.12.2	Files needed by DG	566
K.12.3	Specifying directory masks	567
K.12.4	The configuration file	567
K.12.5	Creating or modifying the configuration file	567
K.13	File formats	569
K.13.1	Simple template file format	569
K.13.2	Structure file format	569
K.13.3	Targets file format	569
K.13.4	Output file format	570

L	Some information from the SOLPS 2002 Course Notes	572
L.1	Breakdown of B2.5 programs	572
L.2	B2.5 ".dat" and ".prt" files	574
L.3	B2.5 communication files	575
M	Subversion repository logs	576
M.1	SOLPS-ITER	576
M.2	Changes in output for the AUG_16151_D test case	633
	Bibliography	635
	Index	638

List of Figures

3.1	Region numbers for single-null geometries.	81
3.2	Region numbers for double-null geometries.	81
3.3	Residuals.	89
3.4	Power crossing various surfaces.	90
3.5	Upstream (outer midplane) electron temperature.	90
3.6	Monoblock tile cross section	125
3.7	Reflector plate tile cross section	126
5.1	Upstream (outer midplane) electron temperature for a range of time-steps.	152
5.2	Upstream (outer midplane) electron temperature for a range of time-steps with a core speed-up factor.	152
5.3	Upstream (outer midplane) electron temperature for a range of time-steps with and without a core speed-up factor.	153
5.4	Downstream (outer target) electron temperature for a range of time-steps.	153
5.5	Downstream (outer target) electron temperature for a range of time-steps with a core speed-up factor.	154
5.6	Downstream (outer target) electron temperature for a range of time-steps with and without a core speed-up factor.	154
5.7	Upstream (outer midplane) electron temperature.	155
5.8	Upstream (outer midplane) electron temperature.	155
5.9	Upstream (outer midplane) electron temperature.	156
5.10	Upstream electron temperature for several timesteps after puff b.c. applied.	156
5.11	Upstream electron temperature for several timesteps after puff b.c. applied.	157
6.1	Using CVIEW to access SOLPS MDSplus data.	172
6.2	Using JETDSP to access SOLPS MDSplus data.	173
6.3	Comparison of profiles for a range of shots.	177
6.4	Comparison of profiles for a range of shots.	183
6.5	Comparison of profiles for a range of shots.	183
6.6	Comparison of profiles for a range of shots.	184

B.1	Coordinate system: x is the poloidal coordinate, y is the radial coordinate orthogonal to the flux surfaces. The coordinate $\perp$ corresponds to the direction perpendicular both to the magnetic field and to the y -axis. The directions of magnetic field and plasma current correspond to normal operation conditions of ASDEX–Upgrade (ion ∇B drift directed towards the X-point).	316
B.2	Impact of ion flux limits for a 10^{19} m^{-3} upstream density case.	326
B.3	Impact of ion flux limits for a $2 \times 10^{19} \text{ m}^{-3}$ upstream density case.	327
B.4	Dependence of the various models for the $\kappa_{e\parallel}$ prefactor as a function of Z_{eff} .	332
B.5	Dependence of the various models for the parallel electron conductivity $\sigma_{\parallel}$ prefactor as a function of Z_{eff} .	332
B.6	Dependence of the various models for the thermo-electric coefficient $\alpha_{e\parallel}$ prefactor as a function of Z_{eff} .	333
I.1	Ion and atomic densities for DIII–D.	448
I.2	Trajectory plot.	451
I.3	Demonstration of various color palettes available in b2plot	492
I.4	BLS, BOL, DCN and DIV chords.	493
I.5	HAL chords.	494
I.6	Mesh plots.	497
I.7	Electron temperature plots.	498
I.8	Other solution data plots.	498
I.9	Chord plots applied to JET.	499
I.10	Wall plots.	499
I.11	Target plots.	500
I.12	C and Ne radiation.	501
I.13	Total radiation for a D+He+C example case.	502
I.14	Line plots of fluxes and other quantities.	503
K.1	Demonstration of a rotated view with associated dialog box and tear-off menu.	551
K.2	Demonstration of the variable aspect ratio capability along with the associated tear-off menu.	552
K.3	Create Surface(s) . . . dialog with highlighted Structures.	556
K.4	Zoom of the PFR showing the highlighted "bounding element".	557
K.5	Zoom of the twisted mesh cells near the inner divertor target of an NSTX geometry.	562
K.6	After stretching and rotation, the overlapping of the cells in the upper part of the region cells depicted in Fig. K.5 becomes clear.	563
M.1	Residuals of the AUG_16151_D standard case for different code versions and options.	634
M.2	Relative changes for the considered options.	634

List of Tables

3.1	Changes of quantity names from SOLPS 4.x b2plot to SOLPS 5.x b2pl.exe .	106
B.1	Comparison of classical transport coefficients according to the Braginskii, Balescu and Zhdanov formulations	331
E.1	Scalar time dependent quantities stored in b2time.nc	392
E.2	Vector time dependent quantities stored in b2time.nc. Note that the conversion to "per unit area" for fluxes is done in the script "2d_profiles"; the actual data stored in b2time.nc for fluxes are the fluxes per cell. There is a similar conversion from J to eV for temperatures.	393
E.3	Double-null vector time dependent quantities stored in b2time.nc	394
E.4	Multi-dimensional time-dependent quantities stored in b2time.nc	395
E.5	Scalar batch-averaged time dependent quantities stored in b2time.nc	396
F.1	Quantities stored in b2tallies.nc	399
G.1	Files produced by means of 'b2wdat_iout = 4', the variable they contain, and the equation to which they are related	408
I.1	Available quantities	430
I.2	Commands acting on the "matrix" stack together with their stack requirements.	432
I.3	Commands acting on the "matrix" stack plotting together with their stack requirements.	435
I.4	Commands assigning real variables from stack for use in plotting	436
I.5	Commands taking character strings as parameters. Note that the string should be delimited by single quotes.	437
I.6	Quantities that can be stored in the "Eirene" stack.	438
I.7	Commands related to the "Eirene" stack and triangular grid data plots.	440
I.8	Quantities that can be stored in the "wall" stack.	442
I.9	Commands related to the "wall" stack and wall plots.	443
I.10	Available target quantities for plotting.	444
I.11	Commands related to the "target" stack and target plots.	446
I.12	Available b2plot constants.	446

I.13	Additional commands.	447
J.1	Defines for Carre and where they are used.	529
J.2	Defines for DivGeo and where they are used.	529
J.3	Defines for B2.5 and where they are used.	538
J.4	Defines for Eirene and where they are used.	540
J.5	NAG calls from B2.5	541
J.6	NAG calls from Eirene	541
K.1	DG external objects	544
K.2	DG internal objects	545

Chapter 1

Introduction

This document describes the **SOLPS-ITER** code package and its components. Following this introduction is a chapter (2) describing the structure of the code. Then chapter 3 describes the standalone version of B2.5. This is followed by a description of the standalone version of Eirene in chapter 4 and then of the coupled B2.5-Eirene in chapter 5. Then follow chapters on DivGeo (7), Carre (8) and agg (9). Appendix A contains copies of the B2.5 internal documentation files, B describes the physics in B2.5, C gives the full model equations used in B2.5, D gives details about some B2.5 boundary conditions, E information about the quantities stored in `b2time.nc`, F information about the quantities stored in `b2tallies.nc`, H gives the region definitions used in the code, I is an updated version of the **b2plot** manual, J contains information about array dimensions and compilation options, K is the DG manual, and lastly L includes a listing of the programs and files included in the B2.5 part of the **SOLPS-ITER** package.

Historically, there have been many implementations of the SOLPS (Scrape-Off Layer Plasma Simulation) code package, according to which version of the fluid plasma code (B2 or B2.5) and of the Eirene Monte Carlo neutral transport code are used. We distinguish between **SOLPS 4.x** which uses B2 and **SOLPS 5.x** which uses B2.5. This document is devoted solely to **SOLPS-ITER** although several features are common to multiple versions. **SOLPS-ITER** contains the B2.5 plasma kernel and a recent version of Eirene. Most of the current development and support is done exclusively on **SOLPS 5.x** and **SOLPS-ITER** versions (with the important exception of the Forschungszentrum Jülich and their collaborators where **SOLPS 4.x** is still in use). The main differences of note between B2 and B2.5 are:

1. For ions, B2 solves a *total* energy equation, while B2.5 solves an *internal* energy equation. See section B.3.3 for an example of the consequences.
2. The primary momentum variable is the *poloidal* velocity in B2 and the *parallel* velocity in B2.5.
3. The velocities are computed on cell faces in B2 and on cell centres in B2.5.
4. The current continuity equation is solved by default in B2.5, but not in B2.
5. The treatment of drifts and currents is much more complete in B2.5 than in B2.
6. Classical transport in B2.5 follows the Zhdanov and Balescu treatments by default, while it uses Braginskii in B2 (see B.4 for a comparison).
7. Some variables names and indexing have changed. See section 3.6 for a partial list.
8. The run parameters to B2.5 are split into several input files. See sections 3.5, A.5 and L.3 for a detailed description.

Some effort has been undertaken for the user to be able to toggle between the **SOLPS 4.x** and **SOLPS 5.x** numerical treatments. The interested reader is referred to the listing of run switches in section A.4 for the complete list. Two converters also exist to transfer run output from **SOLPS 4.x** or **SOLPS 5.x** to **SOLPS-ITER**:

From 4 to 5 and ITER `b2sxdx`

From 5 and ITER to 4 b2xd

From 5 to ITER b2yt

Additional documentation on other code versions can be found in the `${SOLPSTOP}/doc` directory. Namely, the original report by Bas Braams is:

`BRAAMS_-_A_Multi-Fluid_Code_for_the_Simulation_of_the_Edge_Plasma_in_Tokamaks_-_NET_Report_1987.pdf` ,

and an excellent review of **SOLPS 4.x** can be found in:

`Martine_Baelmans_-_Juelich_Report_1994.pdf` .

When referring to **SOLPS-ITER** in scientific papers, please use [1, 2] and references therein. You will also find these references in the `${SOLPSTOP}/doc` directory.

Several of the **SOLPS4.2** features are detailed in `Kotov_SOLPS42_Report.pdf`.

The next section discusses obtaining and updating the code.

1.1 Obtaining and updating the code

To obtain the code a request should be sent to Xavier Bonnin (`Xavier.Bonnin@iter.org`) at ITER, the code Responsible Officer (RO).

The first formality will then be for the prospective users to obtain an IDM (ITER Document Management) account, in case they do not have one already. This account request will be done by the RO, and the requesting user will receive an automated electronic message asking for confirmation of the provided e-mail address. With this account, the user will have access to the **SOLPS-ITER** GIT repository (<https://git.iter.org/projects/BND/repos/solps-iter/browse>) maintained at the IO, and the **SOLPS-ITER** SharePoint web site hosted by ITER (<https://portal.iter.org/departments/POP/CM/IMAS/SOLPS-ITER>, works best with Internet Explorer). The manual to the SOLPS-GUI can be found online at <https://static.iter.org/imas/assets/solps-gui/html/>. Links to the IDM-controlled documents related to **SOLPS-ITER** can be found in a centralized folder (<https://user.iter.org/?uid=Q92BAQ>). The account request will also serve for registration of the user to the **SOLPS-ITER** mailing list (`SOLPS-ITER@iter.org`). Additionally, there is a Confluence Wiki page located at <https://confluence.iter.org/> for the purpose of collecting user contributions to the body of knowledge about the code. Lastly, a Slack discussion forum has been set up, for which the invitation link is: https://join.slack.com/t/solps/shared_invite/zt-ctpoedhs-KZYjYeBDDH7I3pH1M6eGcg .

The **SOLPS-ITER** Git repository also includes the repositories for B2.5, EIRENE, Carre and DivGeo individually as submodules. Some service libraries (such as the GGD, MSCL and GR service libraries), can be found in the repository in the `IMAS Extra and Libraries` section. Users can also have access to the standalone `SOLPS4.3` repository.

One can generally browse the ITER Git repositories here: <https://git.iter.org> and **SOLPS-ITER** specifically here: <https://git.iter.org/projects/BND/repos/solps-iter/browse>, but since these repositories are restricted users will first need to authenticate with their ITER (IDM) username and password - possibly twice, once for the intranet, and once for the Git web front end in the top right hand corner in the web browser window.

To do this, one must upload a copy of their public ssh key here: <https://git.iter.org/plugins/servlet/ssh/account/keys> (by following links under "Manage Account"). Then one should be able to simply clone a copy of **SOLPS-ITER** to a local machine by doing:

```
> git clone ssh://git@git.iter.org/bnd/solps-iter.git
> cd solps-iter
> git submodule init
> git submodule update
```

It is best afterwards to position oneself on the `master` branches, which is achieved by the `switch_to_master` script or, alternatively:

```

> git checkout master
> cd modules/B2.5
> git checkout master
> cd ../Eirene
> git checkout master
> cd ../DivGeo
> git checkout master
> cd ../Carre
> git checkout master
> cd ../adas
> git checkout master

```

For later updates, it is recommended you use the `solps-iter_update` script. If you are planning to do code development, your starting point should be the `develop` branches, which you can switch to by using the `solps-iter_update_devel` script instead.

Please note that because of the way some of the **SOLPS-ITER** utility scripts function, the path to the **SOLPS-ITER** top directory must not contain any of the strings `.debug`, `.mpi`, `.openmp`, or `.ig`.

Since the user-id is immutably encoded into any changes, it is always a good idea to properly configure a local installation of Git on all the systems on which one plans on making code revisions with, e.g.:

```

> git config --global user.name "Your Name"
> git config --global user.email your.name@anywhere.you.are

```

Users all have write access to these repositories although not to the `develop`, `master` or `svn/*` branches. This basically means one should create `feature/[topic]`, `bugfix/[topic]`, or `hotfix/[topic]` branches and raise a "Pull Request" (<https://git.iter.org/projects/BND/repos/solps-iter/pull-requests>) to have them merged into the `develop` branch after pushing them back to the ITER repository.

Issues are handled using JIRA (<https://jira.iter.org/browse/IMAS>) and users should feel encouraged to create new issues associated with **SOLPS-ITER** (selected from the drop-down list of components) if they have any problems or suggestions. (Please ignore the options associated with resource estimates and sprints.)

For more general help with Git, we can recommend the documentation here (<http://git-scm.com/>) and the Pro Git book (linked from there) in particular. Another useful resource for novice Git users is the tutorial from Codecademy available at <https://www.codecademy.com/learn/learn-git>.

When using Git, the command

```
> git status
```

will check if there are any local changes to your local copies of the version-controlled files from the repository, using as a reference the state of the repository from when you last cloned or updated it. To update these references to the current state of the `origin` repository, please use

```
> git fetch
```

If that command returns no output, then your references are up-to-date. Otherwise, if someone else has been committing changes to the repository, Git will indicate the tags of the updated repository. You can then import these latest changes with the command

```
> git pull
```

followed by

```
> git submodule update
```

which will also update the files contained in the various submodules **SOLPS-ITER** is linking to. It is best to do this with your local directory up-to-date and your local changes committed, otherwise Git will ask you to resolve any

conflicts before you may proceed. If you wish to update your entire **SOLPS-ITER** distribution, the recommended method is to use the automated script:

```
> solps-iter_update
```

which will fetch and pull the main development branches for the main **SOLPS-ITER** repository and all of its submodules, provided you are in a clean state, i.e. with no locally modified files under version control, unless these are stashed or committed to local branches. After the distribution files have been updated, the script will attempt to recompile the code.

SOLPS-ITER is a complicated code package and it would be difficult to learn how to install, compile and run it from this document and the internal documentation alone. In the past we have found that an initial visit of at least two weeks to a research team already proficient with **SOLPS-ITER** use with subsequent visits, or about two months and no further visits, are desirable.

The Git **SOLPS-ITER** repository is structured as a self-contained whole, but some effort was also devoted to be able to check out, as individual standalone modules, the main pieces of the SOLPS suite, i.e. DivGeo, Carre, Eirene and B2.5, although this is not recommended.

1.2 Requirements

To best use the **SOLPS-ITER** code suite, the user will need, locally installed:

Fortran 90/95 Compiler the main programs are written in FORTRAN (see Section 1.3 for a list of supported compilers)

C Compiler the DivGeo code and a few utility routines are written in C

MPI Parallelization framework (optional)

NAG mathematical library (optional) (available from <http://www.nag.com/content/nag-library>)

BLAS/LAPACK mathematical libraries (available from <http://www.netlib.org/>)

NCAR plotting library (available from <http://library.ucar.edu>)

GR plotting library (optional, developed at FZ-Jülich, available from the IO Git repository at <https://git.iter.org/projects/LIB/repos/gsoftware/browse>)

GKS graphical kernel library (available from <http://ngwww.ucar.edu>)

HDF5 data storage library (available from <http://support.hdfgroup.org/HDF5/>)

NetCDF storage/archival library (optional) (available from <http://www.unidata.ucar.edu/netcdf/>) (**SOLPS-ITER** needs at least version 4)

MSCL utility library (available from the IO Git repository at <https://git.iter.org/projects/LIB/repos/mscl/browse>)

makedepend to build the dependencies needed by the Makefiles

gnuplot for many analysis scripts

nc2text for obtaining time-dependent data traces (optional)

plotmtv for obtaining graphs of the time evolution of various profiles (optional)

LaTeX interpreter to produce this manual

GIT version control software

python scripting language (**SOLPS-ITER** needs at least version 2.7)

/bin/bash used by some scripts

/bin/csh used by some scripts

/bin/ksh used by some scripts

/bin/sh used by some scripts

/bin/tcsh used by some scripts

For all the various libraries, it is recommended that they be installed as modules, so that they may be shared amongst multiple users on the same machine, so as not to duplicate large files and to centralize their management. Most, but not all, of these libraries are publicly available from the Internet.

If one uses local copies of these libraries, a copy or link to them should be put in the `${SOLPSTOP}/lib/${HOST_NAME}.${COMPILER}` directory.

For fuller code capability, it is also recommended to load the IMAS (Integrated Modelling and Analysis Suite) environment, provided by the IO, whenever possible.

If the `nc2text` utility program is not present on the system, but the `NetCDF` library is available, the code will then automatically build an alternative skeleton version and add it to the scripts path.

1.3 Initial set-up

For each local implementation, environment variables are used to designate the location where the various libraries are to be found, as well as the preferred compilers to be used. These are propagated throughout the **SOLPS-ITER** code structure and used by the Makefiles and the various scripts. It is therefore important to carefully assign their values.

These environment variables are found in files `config.${HOST_NAME}.${COMPILER}`.

The current list of recognized hostnames, deduced by means of the `whereami` script, includes:

ASIPP Academica Sinica, Hefei: `*.ipp.cas.cn`

CCFE Culham Centre for Fusion Energy: `*.ccfe.ac.uk`

FZJ Forschungszentrum Jülich: `*.fz-juelich.de`, `*.ipp.kfa-juelich.de`

GA General Atomics: `*.fusion.gat.com`

IAEA International Atomic Energy Agency: `*.iaea.org`

IFERC IFERC supercomputer: `*.iferc-csc.org`

IN-DA ITER-India: `*.iter-india.org`

IPP IPP-Garching: `*.ipp.mpg.de`, `*.ipp-garching.mpg.de`, `*.rzg.mpg.de`

IPP-HGW IPP-Greifswald: `*.ipp-hgw.mpg.de`

IPR Institute for Plasma Research, Gandhinagar: `*.ipr.res.in`

ITER ITER domain: `*.iter.org`

ITM, MARCONI ITM Gateway machine: `*.eufus.eu`, `*.efda-itm.eu`, `*.marconi.cineca.it`

JET JET domain: `*.jet.uk`

KEIO Keio University of Yokohama: `*.keio.ac.jp`

LEUVEN KU-Leuven: `*.mech.kuleuven.be`

MEPHI Moscow Engineering Physics Institute: `*.mephi.ru`

NERSC DOE supercomputer center: `*.nersc.gov`

ORNL Oak Ridge National Laboratory: *.ornl.gov

PPPL PPPL domain: *.pppl.gov

SPBSTU St. Petersburg State PolyTechnical University: *.spbstu.ru

SWIP SouthWestern Institute of Plasma Physics, Chengdu: *.swip.ac.cn

WM College of William and Mary: *.wm.edu

AU_UNKNOWN All other domain names from Australia and associated territories

CA_UNKNOWN All other domain names from Canada

CN_UNKNOWN All other domain names from China and associated territories

EU_UNKNOWN All other domain names from EURATOM Member countries and their associated territories

IN_UNKNOWN All other domain names from India

JP_UNKNOWN All other domain names from Japan

KR_UNKNOWN All other domain names from the Republic of Korea

KZ_UNKNOWN All other domain names from Kazakhstan

RF_UNKNOWN All other domain names from the Russian Federation

UK_UNKNOWN All other domain names from the United Kingdom and associated territories

US_UNKNOWN All other domain names from the United States and associated territories, including *.edu and *.gov

UNKNOWN everywhere else

If your specific domain name is not on this list, please add it to `whereami`. In cases where the `hostname` and `nslookup` commands used by `whereami` do not provide the desired `$HOST_NAME` value, you can specify it explicitly by means of a `$SOLPSTOP/SETUP/setup.csh.HOST_NAME.local` file, which should contain:

```
setenv HOST_NAME <your_desired_HOST_NAME_value>
```

At the moment the following compiler platforms are supported, assuming a Linux work environment:

cray The Cray compiler, in use at the JFRS supercomputing center

pgf90 Portland Group 32-bit Fortran compiler, in use at JET and ORNL

ifort64 Intel 64-bit Fortran compiler, in use at ITER and IPP

gfortran The gfortran compiler, in use at FZ-Jülich, KU-Leuven

g95 The g95 compiler, in use on the Marconi Gateway

nag_f90 The NAG Fortran compiler

For each hostname domain, a default compiler can be defined by means of the `default_compiler` file. This default can be overridden by giving the desired compiler name as an argument following the `setup.csh` command, i.e.:

```
source setup.csh ifort64
```

If your domain name does not yet have a default compiler associated with it or you want to change it, please edit the `default_compiler` script accordingly.

Then, if this is the first installation in your file space, please do:

```
first_setup
```

This will, if they are not already present in the distribution, create the various `config.${HOST_NAME}.${COMPILER}` files for each of the **SOLPS-ITER** submodules. Feel free to modify them afterwards according to your local requirements. As with `whereami` and `default_compiler`, if those modifications are worth sharing with the other users, please commit them back to the repository using a pull request.

For each combination of hostname and compiler, the set of corresponding environment variables are loaded from the `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}` file and, subsequently, the `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}.local` file. The compiler switches are given through the configuration file `${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}`. This last file should contain the location of all the main libraries with which the **SOLPS-ITER** executables will be linked and their linking instructions. It also contains the shared compiler directives for all the **SOLPS-ITER** object files (see Section J.4.1 for a full listing) and the name of the debugger program to be used, if any. Users should make sure the settings within these files fit their run environment and modify them as needed. If changes are needed, please forward them to the **SOLPS-ITER** RO so that they may be included in the set of distributed files to help other future users.

For the case where single submodules are cloned from the **SOLPS-ITER** repository, without the full regalia of programs, additional sets of config files can be found in the `config` directory in each module. The compiler settings specific to this module are included there. Users are encouraged to modify these as needed. If local changes are required that should not be archived in the repository, then users should create a file named `config.${HOST_NAME}.${COMPILER}.local` which the `Makefile` will load and use to overwrite the more general settings from the distributed `config` file.

Some Windows implementations have been attempted, and require an additional `-DWINDOWS` compiler switch, but are not currently being supported.

If you wish your domain name to be included in the list of hostnames above, please contact the **SOLPS-ITER** RO and send the hostname you would like to have and the Internet address range to which it would apply. Please also send the relevant compiler config files so they may be distributed to future users from the same domain or serve as examples for other users using the same compiler.

If the NAG library is not available (it requires a pay licence), then the BLAS and LAPACK packages should be used instead, and links to them added to the corresponding `config` files.

To compile the coupled version of the code you should begin by checking that the default array dimensions provided fit your problem size. These are stored in the file `$SOLPSTOP/modules/B2.5/src/include/DIMENSIONS.F`. If the dimensions are not right for you, then do the following:

```
sb2
cd src
mkdir -p include.local
cp include/DIMENSIONS.F include.local
```

and edit the file `include.local/DIMENSIONS.F` so that the dimensions (mainly used for the namelists in B2.5) are big enough (see the description of the relevant variables in section J.1). Once you have edited that file, you will need to recompile with:

```
stop
gmake listobj
gmake tags
gmake depend
gmake
```

so that your local file will be used. The last `gmake` command may of course be replaced by `gmake solps(_debug)`, `gmake solps_mpi(_debug)`, `gmake all(_debug)` or `gmake all_mpi(_debug)` as needed.

It is possible to pass down `Makefile` options with the syntax

```
gmake MAKE_OPTIONS=-j8
```

for instance to multi-thread on the compilation. It is not recommended to give compilation options directly in the compilation line (say, with `gmake -j8 solps`) because the **SOLPS-ITER** compilation includes many parts which must be done in order and sometimes serially.

Once the config files and dimensions have been filled in with the appropriate information for the local implementation desired, one proceeds with the compilation of the code. For this purpose, a master **SOLPS-ITER** Makefile is provided, located at `${SOLPSTOP}/Makefile`. The options supported by this Makefile are:

solps (Default) Compiles the necessary components for a full **SOLPS-ITER** implementation, including:

libs Installs the needed libraries according to config file

carre The Carre grid generator and associated translators

divgeo The DivGeo graphical input file builder and accompanying equilibrium and template converters

b25eirene The B2.5-Eirene main code and accompanying programs

uinp The Uinp translator from DivGeo output to B2.5-Eirene input

sonnet-light The Sonnet library

triang The Tria and Triageom triangulation utilities

amds The AMDS utility for building consistent reaction sets

manual Builds this manual

all The same as **solps** above, but in addition:

b25 The B2.5 standalone plasma fluid solver and accompanying programs

eirene The Eirene standalone Monte Carlo neutral transport code

nox The same as **solps** above, but without linking to any graphical libraries:

carre_nox The Carre grid generator will not generate graphs

divgeo_nox Equilibrium and template converters for DivGeo

b25eirene_nox Only the B2.5-Eirene executable will be built, without graphics

triang_nox The tria.eps and triageom.eps files will not be produced

uinp_nox The Uinp translator from DivGeo output to B2.5-Eirene input

manual Builds this manual

libs, divgeo, amds, sonnet-light Will not be built

all_nox The same as **nox** above, but in addition:

b25_nox The B2.5 standalone plasma fluid solver and accompanying programs

eirene_nox The Eirene standalone Monte Carlo neutral transport code

nox_build The same as **nox** above and used for compilation tests on the Continous Integration server:

carre_nox The Carre grid generator will not generate graphs

divgeo_nox Equilibrium and template converters for DivGeo

b25eirene_nox Only the B2.5-Eirene executable will be built without graphics

triang_nox The tria.eps and triageom.eps files will not be produced

uinp_nox The Uinp translator from DivGeo output to B2.5-Eirene input

libs, divgeo, amds, sonnet-light, manual Will not be built

carre The Carre grid generator and associated translators only

divgeo The DivGeo graphical input file builder and accompanying equilibrium and template converters only

divgeo_nox Equilibrium and template converters for DivGeo

b25 The B2.5 standalone plasma fluid solver and accompanying programs only

b25_nox The B2.5 standalone plasma fluid solver and accompanying programs, not using NCAR Graphics

b25.all The B2.5 standalone plasma fluid solver and accompanying programs only, plus b2xd the **SOLPS 5.x** to **SOLPS 4.x** converter

eirene The Eirene standalone Monte Carlo neutral transport code only

eirene_nox The Eirene standalone Monte Carlo neutral transport code only, without graphics

b25eirene The B2.5-Eirene main code and accompanying programs only

b25eirene.all The B2.5-Eirene main code and accompanying programs only, plus b2xd the **SOLPS 5.x** to **SOLPS 4.x** converter

uinp The Uinp translator from DivGeo output to B2.5-Eirene input only (requires **b25eirene** previous compilation)

sonnet-light The Sonnet library only

triang The Tria and Triageom triangulation utilities only

nc2text_simple The nc2text_simple replacement program for systems where nc2text is not available (requires the NetCDF library)

b2sxdx The b2sxdx converter from **SOLPS 4.x** to **SOLPS-ITER**

amds The AMDS utility for building consistent reaction sets only

manual This manual only

local Creates the local directories in B2.5 and Eirene used for local code modifications

listobj Builds the list of object files for the various code components

depend Builds the dependencies files for the various code components

depend_nox Builds the dependencies files for the various code components when not linking to graphics libraries

tags Builds etags files for easy source code searched using emacs

VERSION Updates the GIT version number included in the source code and output

clean Alias to **clean_solps**

Adding the `_mpi` suffix at the end of any build target will execute the command with MPI parallelization options, when relevant, and put the executables in a separate `builds/${TOOLCHAIN}.mpi` directory.

Adding the `_openmp` suffix at the end of any build target will execute the command with OpenMP parallelization options, when relevant, and put the executables in a separate `builds/${TOOLCHAIN}.openmp` directory.

Adding both the `_openmp` and `_mpi` suffixes, in either order, i.e. `_openmp_mpi` or `_mpi_openmp`, at the end of any build target will execute the command with both OpenMP and MPI parallelization options, when relevant, and put the executables in a separate `builds/${TOOLCHAIN}.openmp.mpi` directory.

Adding the `_debug` suffix at the end of any build target will execute the command in debug mode, and put the executables in a separate `builds/${TOOLCHAIN}.debug` directory.

Adding the `clean_` prefix at the beginning of any build target will remove that build target from its directory, as well as all associated pre-processed source files, object files, module files, dependencies and libraries.

For compilation of the Sonnet library, the user should first do:

```
fs sysname
```

and copy the result in `${SOLPSTOP}/modules/Sonnet-light/Makefile`.

1.4 Before every SOLPS-ITER session

At the beginning of every **SOLPS-ITER** session, the user must load up the **SOLPS-ITER** environment, which will contain the aliases and environment variables needed for smooth code operation (See also Section 2.1 for full details). **SOLPS-ITER** is expected to be run within the TC-shell. This is achieved by the command:

```
tcsh
source setup.csh
```

Among other things, these setup files establish some aliases that are useful for navigating the **SOLPS-ITER** directory structure (see details in Chapter 2). Those aliases are:

stop Goes to `${SOLPSTOP}`, the **SOLPS-ITER** top directory

sb2, sbb Go to `${SOLPSTOP}/modules/B2.5`, the B2.5 top directory

sei Goes to `${SOLPSTOP}/modules/Eirene`, the Eirene top directory

ssc Goes to `${SOLPSTOP}/modules/Carre`, the Carre top directory

ssd Goes to `${SOLPSTOP}/modules/DivGeo`, the DivGeo top directory

sst Goes to `${SOLPSTOP}/modules/Triang`, the Tria top directory

ssu Goes to `${SOLPSTOP}/modules/Uinp`, the Uinp top directory

ssw Goes to `${SOLPSTOP}/modules/Sonnet`, the Sonnet top directory

slib Goes to `${SOLPSTOP}/lib/${HOST_NAME}.${COMPILER}`, the directory where libraries are stored

sbr Go to `${SOLPSTOP}/runs/`, the top run directory

scr Goes to `${SOLPSTOP}/scripts`, the directory where the scripts are stored

\$DG The location of the DivGeo top directory

It is also possible to toggle between versions of the code compiled with and without debug options (array bounds checking, traceback, etc...) by use of the `set_debug` and `unset_debug` commands. Similarly, the `set_mpi` and `unset_mpi` will turn on or off, respectively, compilation with MPI parallelization flags. Same for `set_openmp` and `unset_openmp`, for OpenMP parallelization. By default, neither debug nor parallelization (MPI and OpenMP) flags are turned on.

After activating one of these flags and upon succesful compilation, it is recommended to do:

```
rehash
```

to refresh the list of executables to be found with the search path.

1.5 Examples

On the **SOLPS-ITER** web site (<https://portal.iter.org/departments/POP/CM/IMAS/SOLPS-ITER>), in the folder *Examples*, one can find a few example cases for **SOLPS-ITER** runs, in both B2.5 standalone or B2.5-Eirene coupled modes.

There is also a dedicated directory `$SOLPSTOP/runs/examples` which contains links to a series of sample cases. These are actually stored in the `/work/imas/shared/external/assets/solps-iter/MD5` directory hosted at the IO, and the means to retrieve these cases is detailed in the `$SOLPSTOP/runs/examples/README.md` file. Following the instructions therein will extract an archived file in gzipped tar format containing a `baserun` and one or more run directories and associated files. In some cases, a `DIMENSIONS.F` file is also provided, which shows the dimensions necessary to run those cases (see Appendix J.1 for full details about the contents of such files).

The cases stored for retrieval include:

test_slab_ortho_standalone An orthogonal slab test case for B2.5 standalone

AUG_16151_D ASDEX-Upgrade D-only benchmark cases, including B2.5 standalone, coupled and single Eirene call run directories

AUG_16151_D+C+He ASDEX-Upgrade multi-fluid benchmark cases, including B2.5 standalone, coupled and single Eirene call run directories

AUG_28903_D+N_drifts ASDEX-Upgrade D+N case including drifts

AUG_28903_D+Ne_drifts ASDEX-Upgrade D+Ne case including drifts

ITER_2171_D+He+Be+Ne ITER full performance case with Ne injection

ITER_2264_0.6Ne_100MW Another ITER full performance case with Ne injection

ITER_2275_Only_20MW ITER low power case at low density (D-fuelled)

ITER_2292_Only_20MW ITER low power case at low density (H-fuelled)

ITER_2294_Only_20MW Same as above, but increasing density

ITER_2296_Only_20MW Same as above, but continuing density scan

ITER_2297_Only_20MW Same as above, continuing scan

ITER_2298_Only_20MW Same as above, continuing scan

ITER_2299_Only_20MW Same as above, continuing scan, reaching detachment

ITER_2308_Only_20MW Same as above, continuing scan, detached conditions

ITER_2360_0.95He+0.05H_40MW ITER Helium plasma case at mid-power

ITER_2588_D+He+N ITER full power case with N injection and gas puff feedback on SOL D particle content

ITER_2588_drifts ITER B2.5 standalone D-only demonstration case with and without drifts

ITER_535_D+He+Ar ITER full power case with Ar injection

ITER_Be-W_D+T+He+Ne ITER full power case with 98 species

as well as additional partial cases used in tutorials when learning how to build a case using the `DivGeo` and `SOLPS-GUI` utilities. Other cases can be added at the suggestion of users, if available.

After the cases have been downloaded to your system, make sure that the timestamps of the various input files are in the proper chronological order (using the `correct_baserun_timestamps` and `touch bf2tstat` instructions as necessary) before attempting to launch any actual runs. Also, the links to database files are not provided within these examples, and they can be set up using the `setup_baserun_eirene_links` script.

1.6 Using a central installation

For systems with several users it is advisable to set up a central **SOLPS-ITER** installation that can be maintained by an administrator. The scripts have been set up to allow this by specifying a `$SOLPSWORK` directory that is different

from the default `${SOLPSTOP}/runs` and setting the environment variable `SOLPS_CENTRAL` to 'yes'. Each user will then have a directory for their runs, `$SOLPSWORK`, which should contain a file named `SOLPSTOP` which contains the value for `$SOLPSTOP`. Using an example where each user has their runs in `${HOME}/solps-runs` then the central installation could have a `setup.csh.local` with the following lines:

```
setenv SOLPSWORK /home/$USER/solps-runs
alias sbr 'cd ${SOLPSWORK}'
echo $SOLPSTOP >! $SOLPSWORK/SOLPSTOP
if (-e $SOLPSWORK/setup.csh.local) then
    source $SOLPSWORK/setup.csh.local
endif
setenv SOLPS_CENTRAL "yes"
```

This will set `$SOLPSWORK`, create the appropriate `SOLPSTOP` file, and load a `setup.csh.local` for each user if it exists. The scripts such as `dg`, `carre`, and `triang` then check if `SOLPS_CENTRAL` is set and will create the meshes, etc under `${DEVICE}/${USER}` instead of just `${DEVICE}`. It is therefore important that the `DivGeo/device`, `Carre/meshes` and `Triang/triangles` and each `${DEVICE}` subdirectory have the permissions set such that each user can write to it. This can be achieved by having a `solps` group, for example, or by providing write access for all users. Users should also have read and execute access throughout the arboresence emanating from the central installation directory. To avoid issues where the first user to create a `${DEVICE}` directory will do so without the correct permissions, an `add_central_device` script has been provided which will create the appropriate directories with write access for all users. The administrator should run this and add each `DEVICE` that will be used by the group.

Before each session, each user will then set up the **SOLPS-ITER** environment by running

```
tcsh; source $(SOLPS_central_dir)/setup.csh [optional_compiler_name]
```

from their `$SOLPSWORK` directory, and the rest of the usage would be standard, where `$(SOLPS_central_dir)` is the location of the central installation.

Chapter 2

The structure

The **SOLPS-ITER** directory has the following global structure:

```
SOLPSTOP
+---doc
|   +---SOLPS_2002_Course
|   +---solps
+---scripts
|   +---Balance
|   +---commands
|   +---matlab
|   +---MatlabPostProcessing
|   +---Matlab_SPb
|   +---nc2text_simple
|   +---palettes
|   +---ITER.ifort64
+---scripts.local
+---modules
|   +---Sonnet-light
|   +---fxdr
|   +---Triang
|   +---Uinp
|   +---Carre
|   +---amds
|   +---adas
|   +---B2.5
|   +---solps4-5
|   +---DivGeo
|   +---Eirene
+---lib
|   +---ITER.ifort64
+---runs
+---SETUP
```

In order to set up the appropriate environment variables and aliases, the command

```
source setup.csh
```

should be executed in the top level directory.

Three important environment variables are `SOLPSTOP` which points to the top level directory, `HOST_NAME` which

tells **SOLPS-ITER** in which environment it is running and `COMPILER` which identifies the chosen Fortran compiler (e.g. `ifort64`).

2.1 Setup and configuration files

The `setup.csh` file and its subsidiaries `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}` and `${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}` are used to load any modules packages useful for the local **SOLPS-ITER** run environment and the following environment variables:

1. `B25_DEFINES`, an environment variable containing the additional "defines" to use during compilation of `B2.5`
2. `DBX`, the name of the debugger to be used
3. `DEVICE` the name of the fusion device most likely to be simulated for this `HOST_NAME` (see Chapter 7 for available choices)
4. `EIRENE_DEFINES`, an environment variable containing the additional "defines" to use during compilation of `Eirene`
5. `FFLAGSEXTRA`, for providing system-wide Fortran compilation flags
6. `GKS_ROOT`, the root of the `GKS` tree, and `LD_GKS`, the appropriate loader instruction to load the `GKS` library
7. `GLI_HOME`, the location of the `gksfont.dat` file (this definition must appear in both the `setup.csh.*.*` and `config.*.*` files as it is used both during compilation and at runtime)
8. `GR_ROOT`, the root of the `GR` tree, and `LD_GR`, the appropriate loader instruction to load the `GR` library (a graphics package produced by FZ-Jülich and used by `Eirene`)
9. `H5DIR`, the root directory for the `HDF5` utility library
10. `LD_CATALYST`, the appropriate loader instruction to load the `Paraview Catalyst` utility
11. `LD_MSCL`, the appropriate loader instruction to load the `MSCL` library
12. `LD_NETCDF`, the appropriate loader instruction to load the `NetCDF` library
13. `LD_SONNET`, the appropriate loader instruction to load the `Sonnet-light` library
14. `LD_UAL`, the appropriate loader instruction to load the `Integrated Modelling and Analysis Suite (IMAS) Universal Access Layer (UAL)` library
15. `MDSPLUS_DIR`, the root of the `MDSplus` tree, and `LD_MDSPLUS`, the appropriate loader instruction to load `MDSplus`
16. `MSCL_ROOT`, the root of the `mscl` tree, and `LD_MSCL`, the appropriate loader instruction to load the `mscl` library
17. `NAGDIR`, the root of the `NAG` tree, and `LD_NAG`, the appropriate loader instruction to load the `NAG` library
18. `NCARG_ROOT`, the root of the `NCAR` tree, and `LD_NCARG`, the appropriate loader instruction to load `NCAR` (It is best if the definition of `NCARG_ROOT` appears in both the `setup.csh.*.*` and `config.*.*` files as some implementations of `NCAR` use it both during compilation and at runtime)
19. `NCDIR`, the root of the `NetCDF` tree, and `LD_NETCDF`, the appropriate loader instruction to load `NetCDF`
20. `PERFMON`, an environment variable used by the compiler to know whether to enable the performance monitoring libraries
21. `SOLPS_CPP`, an environment variable containing the global "defines" to use during compilation of `SOLPS` code components
22. some aliases to move around the tree (`sbb`, `sei`, *etc.*)

- 23. some aliases for plotting (xyplot, etc...)
- 24. two aliases to swap between the debug and no_debug versions of the code (set_debug and set_nodebug)
- 25. two aliases to swap between the MPI and no_MPI versions of the code (set_mpi and set_nompi)
- 26. two aliases to swap between the OpenMP and no_OpenMP versions of the code (set_openmp and set_noopenmp)

Of course, any additional desired definitions can be added. The preferred place for those is, for those variables that need not be shared by other users, the `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}.local` and `${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}.local` files. Variables that are needed at compilation time need to appear in the `config.*.*(.local)` file, while variables needed at runtime must appear in `setup.csh.*.*(.local)` file.

However, the **SOLPS-ITER** GIT server holds several examples in the `${SOLPSTOP}/SETUP/` directory.

The command

```
stop
```

will take you to `${SOLPSTOP}`.

2.2 modules

```
+---modules
|   +---Sonnet-light
|   +---fxdr
|   +---Triang
|   +---Uinp
|   +---Carre
|   +---adas
|   +---amds
|   +---B2.5
|   +---solps4-5
|   +---DivGeo
|   +---Eirene
```

The `modules/B2.5` subtree contains the source directories for `B2.5`; `modules/Eirene` for the Monte Carlo code, `modules/Carre` and `modules/fxdr` the grid generator and associated converter, `modules/DivGeo` for the DG graphical interface to the grid preprocessor, `modules/Triang` for the Eirene triangular mesh generator, `modules/Uinp` for the Uinp input file generator, `modules/adas` for the ADAS atomic data files, `modules/amds` for the automated generator of atomic physics blocks for Eirene, `modules/solps4-5` for the `b2sxdx` converter from **SOLPS 4.x** to **SOLPS-ITER**, and `modules/Sonnet-light` for the Sonnet library in use for **SOLPS 4.x** grids.

2.2.1 modules/B2.5

The different versions of B2.5 are shown below:

```
SOLPSTOP
+---modules
|   +---B2.5
|   |   +---config
|   |   +---src
|   |   +---src.local (optional)
|   |   +---builds
```

config contains compiler option files

src.local is an optional location which can contain locally modified source files (not version-controlled)

src contains the B2.5 source code files

builds contains the compiled object files for building the B2.5 executables.

A more varied selection of B2.5 historical versions is available within the **SOLPS5.0** distribution.

The commands

```
ssb
```

or

```
sb2
```

will take you to `${SOLPSTOP}/modules/B2.5`.

Local modifications to the B2.5 source

With the installation instructions from Section 1.1, users get a local clone of the B2.5 GIT submodule, and **this is the only supported option!** The files in `src` will be version-controlled and updated whenever the user will pull in new code version. If, however, one wishes to make local changes to a file and protect those from GIT updates, the procedure for that purpose is to copy the file you want to change to `src.local` and then edit that file. You will then need to do

```
sbb
gmake local
```

to create the `src.local` directory. Once you have copied and edited the file(s) you need to change in `src.local`, you should recompile with:

```
stop
gmake listobj
gmake tags
gmake depend
gmake
```

so that the local file is used. The last `gmake` command may of course be replaced by `gmake solps(_debug)`, `gmake solps_mpi(_debug)`, `gmake all(_debug)` or `gmake all_mpi(_debug)` as needed.

If this local change should no longer be needed, then it should be removed from `src.local` and the following performed:

```
stop
gmake clean
gmake listobj
```

```
gmake tags
gmake depend
gmake
```

If you want to avoid a long recompilation, you need to remove from the `builds/*/` directories, the object and module files corresponding to your locally changed file(s), and then ask for a normal recompilation with a single `gmake` command.

If the user wishes to modify the compiling options used, the creation of a `config.$HOST_NAME.$COMPILER.local` file in the `config` subdirectory is recommended. Since the `Makefile` reads those after the `config.$HOST_NAME.$COMPILER` file, they need only contain the modified definitions.

If you just want to compile `b2mn.exe`, you can do

```
gmake MAIN
```

or to compile all but the graphics programs

```
gmake NOPLOT
```

To compile everything, including the converters from **SOLPS 5.x** to and from **SOLPS 4.x** (which might not function as they require a complicated combination of options), you can

```
gmake ALL
```

The default compiles all but the **SOLPS 5.x** to **SOLPS 4.x** converters.

If you wish to submit code changes for a pull request so that they may be inserted in the general **SOLPS-ITER** distribution or to have them version-controlled within your own topical branches, the locally modified files need to remain in the `src` version-controlled directories.

modules/B2.5/src

Within the `modules/B2.5/src` source directory, we have:

```
SOLPSTOP
+---modules
|   +---B2.5
|   |   +---src.local
|   |   +---src
|   |   |   +---b2aux
|   |   |   +---solvers
|   |   |   +---sources
|   |   |   +---include
|   |   |   +---include.local (optional)
|   |   |   +---documentation
|   |   |   +---postprocessing
|   |   |   +---user
|   |   |   +---driver
|   |   |   +---utility
|   |   |   +---output
|   |   |   +---input
|   |   |   +---preprocessing
|   |   |   +---b2plot
|   |   |   +---modules
|   |   |   +---modules.local (optional)
|   |   |   +---convert
|   |   |   +---transport
|   |   |   +---common
|   |   |   +---equations
|   |   |   +---catalyst
|   |   |   +---ids
|   |   |   +---mpi_dummy
|   |   +---Database
|   |   |   +---Bulk_properties
|   |   |   +---Surface_properties
|   |   |   +---trim.data
|   |   |   |   +---range.data
|   |   |   |   +---refl.data
|   |   |   |   +---sputter.data
|   |   +---builds
```

Similarly to the `src.local` directory, two additional local directories are provided `src/include.local` and `src/modules.local` for locally modified include files and modules, respectively, that the user does not wish to have updated by the version control software. The same compilation instructions as in previous Section 2.2.1 apply.

The user wanting to delve deeper into the source code is directed to the detailed routine listing in Appendix A.2.

modules/B2.5/Database

The `Database` directory contain the data files which B2.5 relies on to compute things such as reaction rates, wall element surface temperatures, etc...

These data files include:

ionization_potentials is a B2 and B2.5 input file containing ionization potentials

incatm is a B2.5 input file containing data for atomic mass, surface binding energy and density of all elements

ratadas.filelist is a B2.5 input file containing information used to index into the ADAS files when the "adas" option is used in `b2ar` (see section 3.3)

stra.dat is a file containing STRAHL atomic data results for use by `b2ar` when building `b2frates` files where the 'strahl' data option is chosen. Must be copied to the local `baserun` directory.

weis.dat is a file containing atomic data results for hydrogen by Weisheit *et al.* for use by `b2ar` when building `b2frates` files. Must be copied to the local `baserun` directory.

modules/B2.5/Database/Bulk_properties

Contains data for bulk properties of different target materials. A sample file to be filled by the user for additional materials is contained in `blank`.

modules/B2.5/Database/Surface_properties

Contains data for surface properties of different target materials. A sample file to be filled by the user for additional materials is contained in `blank`.

modules/B2.5/Database/trim.data

Subdivided into `range.data`, `refl.data` and `sputter.data` directories. They contain the TRIM range, reflection and sputtering data used by B2.5. Files ending in `e` are for energy, in `n` are for particles.

2.2.2 modules/Eirene

```
SOLPSTOP
+---modules
|   +---Eirene
|   |   +---builds
|   |   +---config
|   |   +---Database
|   |   |   +---AMdata
|   |   |   +---SurfaceData
|   |   |   |   +---TRIM
|   |   |   |   |   +---10_bins
|   |   +---scripts
|   |   +---src.local
|   |   +---src
|   |   |   +---assistant
|   |   |   +---broadcast
|   |   |   +---broadcast_dummy
|   |   |   +---diagno
|   |   |   +---file-handling
|   |   |   +---geometry
|   |   |   |   +---time-routines
|   |   |   +---grsoft-dummy
|   |   |   +---interfaces
|   |   |   |   +---couple_B2
|   |   |   |   +---couple_B2.5
|   |   |   |   +---couple_Dummy
|   |   |   |   +---couple_SOLPS-ITER
|   |   |   |   +---couple_Tria
|   |   |   +---ioflush
|   |   |   +---iterate
|   |   |   +---main-routines
|   |   |   +---mathematics
|   |   |   +---modules
|   |   |   +---output
|   |   |   +---particle-tracing
|   |   |   +---photons
|   |   |   +---plot_dummy
|   |   |   +---plotting
|   |   |   +---sampling
|   |   |   +---scoring
|   |   |   +---startup-routines
|   |   |   +---surface-processes
|   |   |   +---tetrahedra
|   |   |   +---user-routines
|   |   |   |   +---user_default
|   |   |   |   +---user_EDGE2D
|   |   |   |   +---user_generalized_interface
|   |   |   |   +---user_iter
|   |   |   |   +---user_solps4.3
|   |   |   |   +---user_solps5.0
|   |   |   +---volume-processes
```

The command

sei

will take you to `${SOLPSTOP}/modules/Eirene`.

Because Eirene is distributed as a standalone submodule and is being maintained by FZ-Jülich, some of the directories found above correspond to Eirene implementations other than that used in **SOLPS-ITER**. These are the subdirectories within `interfaces` and `user-routines`. **SOLPS-ITER** uses only the code found in the `interfaces/couple_SOLPS-ITER` and `user-routines/user_iter` subdirectories. The other are included for compatibility with the FZ-Jülich master Eirene repository.

Additionally, the `Database` directory contain the data files which Eirene relies on to compute things such as backscattering coefficients, atomic reactions, etc...

As with B2.5, a `local` directory for the user that wishes to make code changes sheltered from GIT updates. The same procedures apply.

Note that if you want to use the coupled version of the code, you should compile Eirene first.

If routines have been added or removed, if the dependencies have been altered or if this is the first compilation in a new tree, and you are not compiling using the **SOLPS-ITER** master `Makefile`, you should do

```
gmake depend
gmake listobj
gmake tags
gmake depend
gmake
```

The first `gmake depend` is only needed for a brand new compilation. The `gmake listobj` command is needed for a first-time compilation or if a new source file has been added. The `gmake tags` command is needed if this is the first compilation or new files are added or a new routine has been added within an existing source file. The second `gmake depend` is needed if this is the first compilation or if the dependencies have been altered.

If the user wishes to modify the compiling options used, the creation of a `config.$HOST_NAME.$COMPILER.local` file in the `config` subdirectory is recommended. Since the `Makefile` reads those after the `config.$HOST_NAME.$COMPILER` file, they need only contain the modified definitions.

If you are having problems (you've tried to be clever with altering files, or the SOLPS administrator has), you might need to do

```
gmake clean
rm builds/*/LISTOBJ
gmake depend
gmake listobj
gmake tags
gmake depend
gmake
```

modules/Eirene/Database

Data files needed by Eirene are stored in this directory. The most important of these data files are:

AMJUEL is an Eirene input file containing atomic physics rates

H2VIBR is an Eirene input file containing atomic physics rates for the vibrational states of hydrogen molecules

HYDHEL is an Eirene input file containing atomic physics rates for hydrogen and helium

METHANE is an Eirene input file containing atomic physics rate for methane

SPUTER is an Eirene input file containing sputtering data

TRIM is an Eirene input file containing sputtering data

modules/Eirene/Database/Surfacedata/TRIM

Contains TRIM data used by Eirene, pointed to by a cpath instruction in the Eirene input file.

This directory is now located in modules/Eirene/Database/Surfacedata.

2.2.3 modules/amds

Allows for the provision of chemistry set files to be used by Eirene, through selection in DG and pre-processing by Uinp. See the dedicated manual \$SOLPSTOP/modules/amds/doc/manual.pdf for full details.

The AMDS program can be called by use of the amds script.

2.2.4 modules/adas

The modules/adas subdirectory contains files which are used to tell various programs which ADAS files to use.

source.location indicates which class of lines to use ("pjr" currently) and the name of the other index file (ratadas.filelist currently).

ratadas.filelist indicates the global source of the ADAS files, and which "years" to use.

2.3 lib

The `lib` subtree contains libraries used during compilation.

```
SOLPSTOP
+---lib
|   +---ITER.ifort64
```

Since most of the libraries are now expected to be provided as modules, only the Sonnet library, `libsonnet.a`, needs to be stored here. However, local copies of the required libraries can also be put there at the user's convenience.

The command

```
slib
```

will take you to `${SOLPSTOP}/lib/${HOST_NAME}.${COMPILER}`.

2.4 runs

The runs for **SOLPS-ITER** are done in `runs`.

```
SOLPSTOP
+---runs
```

The command

```
sbr
```

will take you to `${SOLPSTOP}/runs`.

Usually a "case series" would be run in a sub-directory of `${SOLPSTOP}/runs`. Sometimes, further subdirectories are created to separate categories of runs — the depth of sub-directories is a matter of taste! In each of the directories, a `baserun` sub-directory is created, and then, at the same level, run directories. The `baserun` directory contains the defaults for all of the other directories at its same level, and typically the grid generation steps and the input build programs `b2ag`, `b2ah`, `b2ar` and `b2ai` are run in this directory (see chapter 3).

2.5 scripts

The command

```
scr
```

will take you to $\${SOLPSTOP}/scripts$.

The number of scripts developed for various **SOLPS** versions is very large. Efforts are currently underway to translate all the relevant ones for use within **SOLPS-ITER**. Any of these scripts can be copied to the $\${SOLPSTOP}/scripts.local$ directory for local user modifications as desired. There also are $\${HOST\_NAME}.\${COMPILER}$ subdirectories where compiler-dependent script programs will reside. A non-exhaustive listing of these scripts, with some explanation as to their purpose, follows:

- the residuals can be plotted with

resall_D all residuals for a D -only case. A python version of this script also exists (Python 2.7 and 3.5 compatible).

resall_D+_only all residuals for a D^+ -only case

resall_D+D all residuals for test cases where we treat two separate D sequences

resall_D+T all residuals for a $D + T$ case

resall_H all residuals for a H only case

resrest residuals for non-species dependent equations

res_D+C non-species dependent residuals for a $D + C$ case

res_D+O non-species dependent residuals for a $D + O$ case

res_D+C+He non-species dependent residuals for a $D + C + He$ case

res_D+C+Ar+Ne+He non-species dependent residuals for a $D + C + Ar + Ne + He$ case

resco continuity residuals

resco_reg continuity residuals per region

resco_reg_region #1 continuity residuals per region for species #1 (default 0)

resco_D+C continuity residuals for a $D + C$ case

resco_D+O continuity residuals for a $D + O$ case

resco_D+C+He continuity residuals for a $D + C + He$ case

resmo parallel momentum residuals per species

resmo_reg #1 parallel momentum residuals per species for region #1 (default 0)

resmo_reg_region #1 parallel momentum residuals per region for species #1 (default 0)

resmo_D+C parallel momentum residuals for a $D + C$ case

resmo_D+O parallel momentum residuals for a $D + O$ case

resmo_D+C+He parallel momentum residuals for a $D + C + He$ case

resmt_reg residuals of the total parallel momentum equation

reshe_reg residuals of the electron heat equation per region

reshi_reg residuals of the ion heat equation per region

respo residuals of the potential equation, including the internal iterations

respo_num the number of internal potential equation iterations

respo_reg residuals of the potential equation per region

resopt Internal script to apply options to above residuals scripts (see below)

resmod #1 Selects residuals *modulo* #1 (default all)

all of the above can take "-m #" to plot the residuals *modulo* "#" occurrences and/or "-l #" to plot the last "#" residuals and/or "-a #" to plot the average residual over "#" occurrences. The options are applied in that order independently of the order in which they were typed.

- power/particle losses (see section [H](#) and Appendix [3.5.4](#) for the region definitions)

b2stbc_sna_reg #1 Boundary particle sources for region #1 (default 0)

b2stbr_sna_reg #1 Recycling particle sources for region #1 (default 0)

b2stbm_sna_reg #1 Additional particle sources for region #1 (default 0)

b2sext_sna_reg #1 External particle sources for region #1 (default 0)

b2stbc_smo_reg #1 Boundary parallel momentum sources for region #1 (default 0)

b2stbr_smo_reg #1 Recycling parallel momentum sources for region #1 (default 0)

b2stbm_smo_reg #1 Additional parallel momentum sources for region #1 (default 0)

b2sext_smo_reg #1 External parallel momentum sources for region #1 (default 0)

rcxhireg #1 charge exchange energy for region #1 (default 0)

rcxmoreg #1 charge exchange parallel momentum for region #1 (default 0)

rcxnareg #1 charge exchange rate for region #1 (default 0)

rdneureg #1 radiation rate from *Eirene* neutrals for region #1 (default 0)

rqahereg #1 electron heat loss for region #1 (default 0)

rqbrmreg #1 bremsstrahlung losses for region #1 (default 0)

rgradreg #1 total radiation rate for region #1 (default 0)

rrahireg #1 recombination energy for region #1 (default 0)

rramereg #1 recombination parallel momentum for region #1 (default 0)

rranareg #1 recombination rate for region #1 (default 0)

rsahireg #1 ionization energy for region #1 (default 0)

rsamereg #1 ionization parallel momentum for region #1 (default 0)

rsanareg #1 ionization rate for region #1 (default 0)

b2stbc_she_reg_region #1 Boundary electron heat sources, by region, for species #1 (default 0)

b2stbc_shi_reg_region #1 Boundary ion heat sources, by region, for species #1 (default 0)

b2stbc_sne_reg_region #1 Boundary electron particle sources, by region, for species #1 (default 0)

b2stbc_sch_reg_region #1 Boundary charge sources, by region, for species #1 (default 0)

b2stbc_sna_reg_region #1 Boundary particle sources, by region, for species #1 (default 0)

b2stbc_smo_reg_region #1 Boundary parallel momentum sources, by region, for species #1 (default 0)

b2stbr_she_reg_region #1 Recycling electron heat sources, by region, for species #1 (default 0)

b2stbr_shi_reg_region #1 Recycling ion heat sources, by region, for species #1 (default 0)

b2stbr_sne_reg_region #1 Recycling electron particle sources, by region, for species #1 (default 0)

b2stbr_sch_reg_region #1 Recycling charge sources, by region, for species #1 (default 0)

b2stbr_sna_reg_region #1 Recycling particle sources, by region, for species #1 (default 0)

b2stbr_smo_reg_region #1 Recycling parallel momentum sources, by region, for species #1 (default 0)

b2stbm_she_reg_region #1 Additional electron heat sources, by region, for species #1 (default 0)

b2stbm_shi_reg_region #1 Additional ion heat sources, by region, for species #1 (default 0)

b2stbm_sne_reg_region #1 Additional electron particle sources, by region, for species #1 (default 0)

b2stbm_sch_reg_region #1 Additional charge sources, by region, for species #1 (default 0)

b2stbm_sna_reg_region #1 Additional particle sources, by region, for species #1 (default 0)

b2stbm_smo_reg_region #1 Additional parallel momentum sources, by region, for species #1 (default 0)

b2sext_she_reg_region #1 External electron heat sources, by region, for species #1 (default 0)

b2sext_shi_reg_region #1 External ion heat sources, by region, for species #1 (default 0)

b2sext_sne_reg_region #1 External electron particle sources, by region, for species #1 (default 0)

b2sext_sch_reg_region #1 External charge sources, by region, for species #1 (default 0)

b2sext_sna_reg_region #1 External particle sources, by region, for species #1 (default 0)

b2sext_smo_reg_region #1 External parallel momentum sources, by region, for species #1 (default 0)

rcxhireg_region #1 charge exchange energy, by region, for species #1 (default 0)

rcxmoreg_region #1 charge exchange parallel momentum, by region, for species #1 (default 0)

rcxnareg_region #1 charge exchange rate, by region, for species #1 (default 0)

rqahereg_region #1 electron heat loss, by region, for species #1 (default 0)

rqbrmreg_region #1 bremsstrahlung radiation loss, by region, for species #1 (default 0)

rqradreg_region #1 total radiation rate, by region, for species #1 (default 0)

rrahireg_region #1 recombination energy, by region, for species #1 (default 0)

rramereg_region #1 recombination parallel momentum, by region, for species #1 (default 0)

rranareg_region #1 recombination rate, by region, for species #1 (default 0)

rsahireg_region #1 ionization energy, by region, for species #1 (default 0)

rsamereg_region #1 ionization parallel momentum, by region, for species #1 (default 0)

rsanareg_region #1 ionization rate, by region, for species #1 (default 0)

these require the `SOLPS_FLUIDS` and `NREG` variables to be set correctly (see section 3.5.4 and Appendix H for the definitions of the regions). If the `SOLPS_FLUIDS` or `NREG` files are not found, the necessary information is extracted from the `run.log` file.

- plotting

plot The various `xyplot#`, `lxyplot#`, `xlyplot#` and `lxlyplot#` call `plot` passing their name as the 1st parameter; `plot` parses the 1st parameter to determine if $\log x$ (`lx`) or $\log y$ (`ly`) and the number of curves (`#`); args 1..`#` specify labels for curves; the next two arguments specify *x*- and *y*-axis labels, the last specifies a header. The system normally uses `gnuplot.cmd` for the `gnuplot` commands, and

`gnuplot.data` for the data. A number of environment variables can be used to modify the behaviour of plot:

GNUPLOT_BATCH used at the start of the process of specifying a batch of plots that should all be processed together into one PostScript file. The basic structure for using this feature would be of the following form:

```
setenv GNUPLOT_BATCH yes
setenv GNUPLOT_POSTSCRIPT_OPTIONS 'color solid'
setenv PRINT_PLOT b2time2d.ps
rm -f GNUPLOT_COUNT gnuplot.cmd
echo 0 > GNUPLOT_COUNT

2dt nesepm
2dt tesepm tisepm
2dt nesepi nemxip nesepa nemxap
...
2dt tmne
2dt tmte tmti

gnuplot gnuplot.cmd
rm GNUPLOT_COUNT gnuplot.*data
```

where the `2dt` could be any series of commands that use `plot` for their graphics. See the script `2d_plots` (2.5 on page 46) for an example

GNUPLOT_CMD changes the command file used from `gnuplot.cmd` to the filename specified by this environment variable

GNUPLOT_COUNT internal environment variable used for batch plots

GNUPLOT_DATA changes the data file used from `gnuplot.data` to the filename specified by this environment variable

GNUPLOT_DELAY allows the user to change the delay set at the bottom of `gnuplot.cmd`

GNUPLOT_EXTRA adds the contents of the file whose name is stored in `GNUPLOT_EXTRA` to the `gnuplot.cmd` file

GNUPLOT_LINEWIDTH changes the line width being used

GNUPLOT_MAX_POINTS sets the maximum number of points that will be plotted; if more points are present, the data will be averaged so that the limit will not be exceeded

GNUPLOT_MAX_POINTS_FACTOR integer factor by which the number of points is reduced to be below that set by `GNUPLOT_MAX_POINTS` (defaults to 2)

GNUPLOT_NOTIMESTAMP removes the timestamp from the plots

GNUPLOT_POINTSIZE changes the size of points plotted

GNUPLOT_POSTSCRIPT_OPTIONS `gnuplot` options for the PostScript driver

GNUPLOT_TERM provides the "term" information for `gnuplot`

GNUPLOT_TMP causes the `gnuplot.cmd`, `gnuplot.data` and other temporary files to be written to the directory specified by this environment variable. Works for most scripts. Also moves the location of the `GNUPLOT_EXTRA` and `GNUPLOT_COUNT` files to being located in `GNUPLOT_TMP`.

PLOTPOINTS just plot points

PRINT_PLOT if set to a filename, then PostScript is written to that file; else if set but empty, PostScript is sent to the default printer; otherwise the plot is done to the X terminal

If the file `GNUPLOT_COUNT` exists, then an internal counter `GNUPLOT_COUNT` is set by reading the number from the file; incremented; and then saved to the file; instead of using `gnuplot.data` for the data, the file `gnuplot.$GNUPLOT_COUNT.data` is used and the commands for plotting are concatenated to `gnuplot.cmd` allowing for multiple pages of plots.

plot.png produces a PNG output file instead

xyplot alias for simple " $x\ y$ " plot

xyplot2 alias for simple " $x\ y_1\ y_2$ " plot

xyplot3 alias for simple " $x\ y_1\ y_2\ y_3$ " plot

xyplot4 alias for simple " $x\ y_1\ y_2\ y_3\ y_4$ " plot

xyplot5 alias for simple " $x\ y_1\ y_2\ y_3\ y_4\ y_5$ " plot

xyplot6 alias for simple " $x\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6$ " plot

xyplot7 alias for simple " $x\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7$ " plot

xyplot8 alias for simple " $x\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7\ y_8$ " plot

xyplot9 alias for simple " $x\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7\ y_8\ y_9$ " plot

xlyplot alias for simple " $x\ \log(y)$ " plot

xlyplot2 alias for simple " $x\ \log(y_1)\ \log(y_2)$ " plot

xlyplot3 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)$ " plot

xlyplot4 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)$ " plot

xlyplot5 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)\ \log(y_5)$ " plot

xlyplot6 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)\ \log(y_5)\ \log(y_6)$ " plot

xlyplot7 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)\ \log(y_5)\ \log(y_6)\ \log(y_7)$ " plot

xlyplot8 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)\ \log(y_5)\ \log(y_6)\ \log(y_7)\ \log(y_8)$ " plot

xlyplot9 alias for simple " $x\ \log(y_1)\ \log(y_2)\ \log(y_3)\ \log(y_4)\ \log(y_5)\ \log(y_6)\ \log(y_7)\ \log(y_8)\ \log(y_9)$ " plot

lxyplot alias for simple " $\log(x)\ y$ " plot

lxyplot2 alias for simple " $\log(x)\ y_1\ y_2$ " plot

lxyplot3 alias for simple " $\log(x)\ y_1\ y_2\ y_3$ " plot

lxyplot4 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4$ " plot

lxyplot5 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4\ y_5$ " plot

lxyplot6 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6$ " plot

lxyplot7 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7$ " plot

lxyplot8 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7\ y_8$ " plot

lxyplot9 alias for simple " $\log(x)\ y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7\ y_8\ y_9$ " plot

lxlyplot alias for simple " $\log(x)\ \log(y)$ " plot

lxlyplot2 alias for simple " $\log(x)\ \log(y_1)\ \log(y_2)$ " plot

lxlyplot3 alias for simple " $\log(x)\ \log(y_1)\ \log(y_2)\ \log(y_3)$ " plot

lxlyplot4 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4)" plot

lxlyplot5 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4) log(y_5)" plot

lxlyplot6 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4) log(y_5) log(y_6)" plot

lxlyplot7 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4) log(y_5) log(y_6) log(y_7)" plot

lxlyplot8 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4) log(y_5) log(y_6) log(y_7) log(y_8)" plot

lxlyplot9 alias for simple "log(x) log(y_1) log(y_2) log(y_3) log(y_4) log(y_5) log(y_6) log(y_7) log(y_8) log(y_9)" plot

- Detailed run diagnostics (Many of these invoke the "plot" command described above and can be toggled to use a logarithmic y scale by use of an additional "l" argument while "L" also plots the absolute values of negative quantities on the same scale and "M" the absolute values of negative quantities in a separate plot)

2d plots time dependent quantities from `b2time.nc` (Appendix E), 2^{nd} , 3^{rd} etc. versus the 1^{st} argument.

2da prints time dependent quantities from `b2time.nc` (Appendix E), 2^{nd} , 3^{rd} etc. versus the 1^{st} argument.

2dt plots time dependent quantities from `b2time.nc` (Appendix E) versus time. A python version of this script also exists (Python 2.7 and 3.5 compatible).

2dt.tallies.py plots time dependent quantities from `b2tallies.nc` (Appendix F) versus time (Python 3.5 compatible).

2dt.av plots averaged time dependent quantities from `b2time.nc` (Appendix E) versus time.

2dvi prepares a **b2plot** command file `TDV/contr.b2p` for production of input files to start a `DIVIMP` run.

2d.plots prepares `b2time2d.ps` from `b2time.nc` (Appendix E). This uses the batch feature of the `plot` command.

2d.plots.av prepares `b2time2d.av.ps` from averaged values in `b2time.nc` (Appendix E). This uses the batch feature of the `plot` command.

2d.profiles creates `*.last10` files for plotting. When processing fluxes, divides them by the area of contact to the neighbouring cell.

2d.profiles.extended creates `*.last10` files for plotting. When processing fluxes, divides them by the area of contact to the neighbouring cell. The flux files ending in `P.last10` are divided by the perpendicular area of the cell face.

2d.summarize Outputs the time dependent quantities from the last time point in `b2time.nc`.

D+fluid.fluxes Plots the fluxes of D^+ ions across the grid boundaries for a B2.5 standalone case

D0.fluid.fluxes Plots the fluxes of D^0 neutrals across the grid boundaries for a B2.5 standalone case

D.coupled.conservation Plots the particle balance of D for coupled runs

D.coupled.conservation.core Plots the particle balance of D in the core plasma for coupled runs

D.fluid.conservation Plots the particle balance of D for B2.5 standalone runs

D.fluid.conservation.core Plots the particle balance of D in the core plasma for B2.5 standalone runs

D.fluid.fluxes Plots the fluxes of D across the grid boundaries for a B2.5 standalone case

He.coupled.conservation.core Plots the particle balance of He in the core plasma for coupled runs

He.fluid.conservation Plots the particle balance of He for B2.5 standalone runs

He.fluid.fluxes Plots the fluxes of He across the grid boundaries for a B2.5 standalone case

bremsstrahlung Obsolete. Plots bremsstrahlung losses in **SOLPS 4.x** cases. Use `rqbrmreg` instead.

check_b2_output Compares the two `b2fstate` files provided as arguments.

cpu Lists the CPU time per B2 . 5 timestep.

currents.py Plots the electric current across various surfaces. Currents into the domain are shown as negative values, fluxes out of the domain as positive values.

delta Plots the maximum fractional change of the primary variables per B2 . 5 timestep.

density Plots the time evolution of the density in a B2 . 5 run.

display_tallies Displays the B2 . 5 tallies in an easy-to-read format (argument "H" gives the first instance, and "L" the last instance)

energy_analysis Plots the energy flows across certain important surfaces in B2 . 5.

energy_analysis_fht Plots the total energy flows across certain important surfaces in B2 . 5.

energy_analysis_total Plots the total energy flows across certain important surfaces in B2 . 5.

energy_balance Plots the energy balance.

energy_balance_core Plots the energy balance for the core plasma.

energy_balance_core_extended Plots an extended version including volumetric sources of the energy balance for the core plasma.

energy_balance_coupled Plots the energy balance, including sources from `Eirene`.

energy_balance_extended Plots an extended version including volumetric sources of the energy balance.

fchxreg Plots the electric current flow across important x -directed surfaces in B2 . 5.

fchyreg Plots the electric current flow across important y -directed surfaces in B2 . 5.

fhexreg Plots the electron energy flow across important x -directed surfaces in B2 . 5.

fheyreg Plots the electron energy flow across important y -directed surfaces in B2 . 5.

fhiereg Plots the ion energy flow across important x -directed surfaces in B2 . 5.

fhiyreg Plots the ion energy flow across important y -directed surfaces in B2 . 5.

fhmereg Plots the parallel kinetic energy flow across important x -directed surfaces in B2 . 5.

fhmereg Plots the parallel kinetic energy flow across important y -directed surfaces in B2 . 5.

fhpereg Plots the ionization energy flow across important x -directed surfaces in B2 . 5.

fhpereg Plots the ionization energy flow across important y -directed surfaces in B2 . 5.

fjxreg Plots the electrostatic energy flow across important x -directed surfaces in B2 . 5.

fjyreg Plots the electrostatic energy flow across important y -directed surfaces in B2 . 5.

fhtxreg Plots the total energy flow across important x -directed surfaces in B2 . 5.

fhtyreg Plots the total energy flow across important y -directed surfaces in B2 . 5.

fluid_fluxes #1 #2 Plots the sum of fluid fluxes for species numbers varying from #1 to #2 (respective defaults are 0 and 1). Also exists as a Python script. Fluxes into the domain are shown as negative values, fluxes out of the domain as positive values.

fluxt #1 Gives the recycling fluxes for the first #1 `Eirene` strata. The default value is all the strata, as read from the `b2.neutrals.parameters` file, or if not present, the `$SOLPSTOP/modules/B2.5/src/include.local/DIMENSIONS.F` file, or if not present, the `$SOLPSTOP/modules/B2.5/src/include/DIMENSIONS.F` file.

internal_energy_balance Outputs the energy balance tallies.

internal_energy_balance_coupled Outputs the energy balance tallies for a coupled run.

joule_heating Plots the amount of Joule heating per region.

mtv_3d Plots the time evolution of a CDF quantity (Appendix E) profile.

mtv_3d_f Plots the time evolution of the profile of the flux of a CDF quantity (Appendix E) along the targets.

mtv_3d_nf Plots the time evolution of the profile of a CDF quantity (Appendix E) along both targets side by side.

mtv_3d_t Plots the time evolution of the profile of the a CDF quantity (Appendix E) along the targets.

nareg #1 Shows averaged density per region for species #1.

nareg_species #1 Shows averaged density per species for region #1.

nereg Shows averaged electron density per region.

nesepm_feedback Plots in `feedback.ps` the parameters of the *nesepm_sol* feedback scheme.

nesepm_feedback_by_core Plots in `feedback.ps` the parameters of the *nesepm* feedback scheme.

nesepm_feedback_by_pfr Plots in `feedback.ps` the parameters of the *nesepm_pfr* feedback scheme.

nesepm_feedback_fluid Plots in `feedback.ps` the parameters of the *nesepm_sol* feedback scheme for fluid runs.

nireg Shows averaged total ion density per region.

particle_conservation_coupled Plots the particle balance for multi-species coupled runs

print_species Prints the list of species included in the run.

reflection Shows reflected particle fluxes (for B2.5 standalone runs).

reflection_energy Shows reflected energy fluxes (for B2.5 standalone runs).

sputter Shows sputtering particle fluxes (for B2.5 standalone runs).

sputter_energy Shows sputtering energy fluxes (for B2.5 standalone runs).

sputter_RES Shows RES sputtering particle fluxes (for B2.5 standalone runs).

sputter_RES_energy Shows RES sputtering energy fluxes (for B2.5 standalone runs).

sputter_chemical Shows chemical sputtering particle fluxes (for B2.5 standalone runs).

sputter_chemical_energy Shows chemical sputtering energy fluxes (for B2.5 standalone runs).

sputter_physical Shows physical sputtering particle fluxes (for B2.5 standalone runs).

sputter_physical_energy Shows physical sputtering energy fluxes (for B2.5 standalone runs).

tflux #1 Gives the recycling fluxes for the first #1 strata from `b2.neutrals.parameters`. The default value is all the strata, as read from the `b2.neutrals.parameters` file, or if not present, the `$SOLPSTOP/modules/B2.5/src/include/local/DIMENSIONS.F` file, or if not present, the `$SOLPSTOP/modules/B2.5/src/include/DIMENSIONS.F` file.

thermal_evaporation Shows thermal evaporation particle fluxes (for B2.5 standalone runs).

thermal_evaporation_energy Shows thermal evaporation energy fluxes (for B2.5 standalone runs).

tereg Shows averaged electron temperature per region.

time_dep Script to obtain time traces of many useful quantities.

tireg Shows averaged ion temperature per region.

viscous_heating Shows the viscous heating term per region.

zeffreg Shows Z_{eff} per region.

many of the above require the SOLPS_FLUIDS and NREG variables to be set correctly (see section 3.5.4 and Appendix H for the definitions of the regions and surfaces). If the SOLPS_FLUIDS or NREG files is not found, the necessary information is extracted from the run.log file.

- Utility

add_central_device DEVICE Creates writeable directories for a given DEVICE for a centralized SOLPS-ITER installation.

ave.awk #1 Performs an average over the #1 rows (default all).

awk.first+even Selects the first and every even column from an array of data.

awk.integrate Integrates an array of data.

awk.mesh Outputs the mesh.extra file data in a form usable for xyplot.

awk.notfirst Removes the first column from an array of data.

awk.plot_mesh #1 Outputs the mesh specified as #1 in a form usable for xyplot.

awk.reverse Outputs a file in reverse order.

awk.transpose Transposes an array of data.

convert_from_solps4.3 #1 Converts run number #1 from \$DEVICE from the SOLPS4.3 ITER archive to a SOLPS-ITER run in directory \$SOLPSTOP/runs/\$DEVICE/#1.

fixbb A utility script to properly define the bounding box of PostScript files.

get_solps_fluids A utility script to obtain the list of fluid species labels from the run.log file.

grf A more effective version of "grep" when searching for use of a variable in source code.

inspect_plasmastate #1 This script will return the plasma time stamp at which the plasmastate savefile #1 was written.

plot_cps #1 Copies gnuplot.cmd to gnuplot_cps.cmd, modifying it so that colour plotting occurs to #1 or by default out.ps; executes gnuplot on new file. The location of these files can be modified by use of the GNUPLOT_TMP variable.

plot_it Utility script used to produce complex plots.

plot_ps #1 Copies gnuplot.cmd to gnuplot_ps.cmd, modifying it so that black-and-white plotting occurs to #1 or by default out.ps; executes gnuplot on new file. The location of these files can be modified by use of the GNUPLOT_TMP variable.

plot_rzpsi.dat Utility script using plotmtv to obtain a plot of the equilibrium data from the rzpsi.dat file used by Carre to build the computational grid.

plot.set Utility script preparing files for complex plots.

species Obtains the list of species from SOLPS_FLUIDS.

wavelength_to_energy Converts a wavelength (in m) to an energy (in eV) (and the reverse).

- Others

b2 Shortcut to B2.

b2favere_OK Checks that the b2favere file is of correct size and content.

b2fstate_OK Checks that the `b2fstate` file is of correct size and content.

b2fstate_OK_bool Checks that the `b2fstate` file is of correct size and content and returns 0 or 1 for use in scripting.

b2p Script to obtain a standard series of **b2plot** graphs.

b2pc Script to obtain a standard series of **b2plot** graphs.

b2pl Script to obtain a standard series of **b2plot** graphs.

b2plot Shortcut to **b2plot**. A `"-n"` option allows for dry-run tests. A `"-d"` option allows to run within the chosen debugger. A `"-s"` option allows to run **b2plot** without coupling to `Eirene` data. A `"-h"` option prints a help description.

b2pp Script to obtain a standard series of **b2plot** graphs.

b2pt Script to obtain a standard series of **b2plot** graphs.

b2run Shortcut to the `Makefile` launching `B2.5`. A `"-n"` option allows for dry-run tests. A `"-d"` option allows to run within the chosen debugger. A `"-s"` option allows to run `B2.5` in standalone mode. A `"-m"` option allows to run with MPI options. A `"-o"` option adds any other desired run options. The options can be grouped together inside quotes or given individually with multiple `"-o"` instances. A `"-t #"` option allows to request an OpenMP run with `#` threads. A `"-b"` option can be used when invoking the `b2co` program to obtain a provisional `b2fstate` file while the main code is running. A `"-h"` option prints a help description. (See 3 for fuller details)

b2run_hgw_cluster Shortcut to the `Makefile` launching `B2.5` for the IPP–Greifswald cluster. A `"-n"` option allows for dry-run tests.

b2sxdx Shortcut to the **SOLPS 4.x** to **SOLPS 5.x** converter. A `"-n"` option allows for dry-run tests. A `"-d"` option allows to run within the chosen debugger. A `"-s"` option allows to specify that the run to convert was not coupled to `Eirene`. A `"-o"` option adds any other desired run options. The options can be grouped together inside quotes or given individually with multiple `"-o"` instances. A `"-h"` option prints a help description.

chop_pwd #1 Truncates the name of the current directory to not exceed `#1` characters (default 60).

climb_and_look_for Script used to find a file in a directory above the current one.

copy_tracing_files Script to bring tracing files to temporary run directory, if present.

correct_baserun_timestamps Makes sure the timestamps for the start-up files in the `baserun` directory are in the right chronological order.

correct_b2yt_timestamps Makes sure the timestamps for the start-up files in the current directory are in the right chronological order for the `b2yt` converter.

count Basic counting with advanced options (`"count -h"` for help).

d2d Doubles the resolution of the DG equilibrium file (see Sec. K.6.1).

d2e Converts DG equilibrium files to `EFIT` format (see Sec. K.6.1).

d2v Converts DG equilibrium files to `Carre` format (see Sec. K.6.1).

datestamp Gets a date stamp from the machine.

e2d Converts `EFIT` equilibrium files to DG format (see Sec. K.6.1).

eirobj Alias to run `Eirene` standalone with graphics.

eirobjx Alias to run `Eirene` standalone without graphics.

first_setup Creates the `config.$HOST_NAME.$COMPILER` files in each of the **SOLPS-ITER** submodules.

get_git_email Finds the user's email address from the `.gitconfig` file.

get_git_name Finds the user's name from the `.gitconfig` file.

get_parm.bpl extracts certain useful quantities from a run input and output files for use in later scripts, particularly those building **b2plot** command files.

gtldr Script to make a directory tree local.

gtfl Replaces the specified links by the files they are linked to.

import_files_from_convert Imports and renames the files produced by `b2yt` in a `convert` directory at the same level as the current directory to prepare the new converted case for a run. Also updates some of the files in the `baserun` directory, so it is often prudent to use `correct_baserun_timestamps`, from the parent directory, afterwards.

input.dat_to_mesh.extra Converts standard input (usually directed from a `input.dat` file) into a `mesh.extra` file.

ITER_divertor_HTM Performs the thermal calculation of the divertor target monoblocks and reflector plate hypervapotron channels. It requires `b2ymb.dat` and `b2yrp.dat` to be present in the run directory. It runs **b2plot** to produce the heat load profile data of the targets and reflector plates. Then it runs `b2ymb` to produce the input files (see 3.18) for the `FreeFem++`[3] heat transfer model of the monoblock `Monoblock_HTM.edp` for the outer target. Afterwards, `b2ymb` is ran again to produce the input files to run `Monoblock_HTM.edp` for the inner target. When the inner target is completed, `b2yrp` (see 3.19) is run to produce the input files for the inner reflector plate heat transfer model `ReflectorPlate_HTM.edp`. Finally, `b2yrp` is executed one last time to produce the input files for the thermal analysis of the outer reflector plate.

itersubmit Command file to launch runs on the ITER HPC cluster. See Section 3 for details.

itmsubmit Command file to launch runs on the Marconi ITM Gateway machine. See Section 3 for details.

marconisubmit Command file to launch runs on the Marconi supercomputer. See Section 3 for details.

ippsubmit Command file to launch runs on the new (2019) IPP cluster. See Section 3 for details.

jetsubmit Command file to launch runs on the JET Analysis Cluster (JAC). See Section 3 for details.

jfrssubmit Command file to launch runs on the JFRS IFERC-CSC supercomputer. See Section 3 for details.

last10 Truncates the `run.log` file to contain only the output from the last 10 written code iterations and stores them in `run.log.last10`.

localsubmit Command file to launch runs on the local machine. See Section 3 for details.

lln Lists the chain of symbolic links leading to a file.

lndr routine to generate shadow trees.

mds_id Obtains certain data necessary for run identification when saving to the `MDSplus` database.

move_optional_files Script used by the run Makefile to check for the existence of optional output files and, upon succesful termination of a run, move these files to the run directory.

move_save_files Script used by the crash recovery program `b2co` to replace optional run output files with their recovered versions.

n2d Converts "Naka" format equilibrium files to DG format (see Sec. K.6.1).

nc2text_simple Replacement program providing the basic capabilities of the `nc2text` utility when not available on the system. A symbolic link to `nc2text` will be created for other scripts to be able to use it.

nn2d Converts new "Naka" format equilibrium files to DG format (see Sec. K.6.1).

noquote_pwd Returns the current working directory as a string without quotes (c.f. `quote_pwd`).

obtain_tracing_files Script to grab the last version of the tracing files for run recovery.

pipe_b2.log Prepares the B2 log file for use by other scripts.

pipe_run.log Prepares the B2.5 log file for use by other scripts.

plot_mesh #1 #2 Plots the mesh specified as #1 accompanied by the `mesh.extra` file specified as #2 as #1.ps.

preserve_scratch_to_work List of instructions to move run results from temporary scratch space to a backed-up filesystem

print_versions Prints the GIT repository versions of the various code executables

QSUB.postprocess List of instructions appended to the B2.5 standalone submission scripts for run post-processing

QSUB.postprocess_coupled List of instructions appended to the **SOLPS-ITER** submission scripts for run post-processing

quote_pwd Returns the current working directory as a string with quotes (c.f. `noquote_pwd`).

recover_initial_files For use to copy back the stored initial case files (`.i` extension) before a run restart, or when use of `b2co` is not possible because the run crash happened too early for `plasmastate` files to be written.

regression_test Utility script to run regression tests.

res_av Special script to reduce residuals in the finalization step of the `Eirene` sources averaging scheme (see Section 5.4 for details).

resave_mds Update an archived `MDSplus` case. A `"-n"` option allows for dry-run tests. A `"-d"` option allows to run within the chosen debugger. A `"-s"` option allows to update a B2.5 standalone case. A `"-r"` option allows to pass variable redefinitions to the underlying Makefile.

rlndr Replaces a link address by another

rps Shifts Ψ values by a given amount within a DG equilibrium file (see Sec. K.6.1).

run_av Special script to prepare the averaging step of the `Eirene` sources averaging scheme (see Section 5.4 for details).

save_initial_files Script used by runs Makefile at start of a run to save initial state files for eventual recovery.

save_mds Archive an `MDSplus` case. A `"-n"` option allows for dry-run tests. A `"-d"` option allows to run within the chosen debugger. A `"-s"` option allows to archive a B2.5 standalone case. A `"-r"` option allows to pass variable redefinitions to the underlying Makefile.

setup_baserun_eirene_links Prepares the run directory before a B2.5-Eirene run by settings up links to the proper data files.

sgesubmit Command file to launch runs on an SGE system. See Section 3 for details.

SLURM.postprocess Alias to `QSUB.postprocess`

SLURM.postprocess_coupled Alias to `QSUB.postprocess_coupled`

SOLPS4to5 Script to convert a **SOLPS 4.x** run directory into a **SOLPS 5.x** one.

solps-iter.update Script to update to the latest **SOLPS-ITER** master distribution.

solps-iter.update_develop Script to switch to the up-to-date **SOLPS-ITER** develop branch.

str2ogr Converts a `.str` structure file into a DG `.ogr` template file.

summarize_run List of instructions provided during job submission for obtaining a standard set of run diagnostics

switch_to_master Script to switch to the up-to-date **SOLPS-ITER** master branch.

triang Main script used to prepare run input files after the `DivGeo` case has been defined and the `Carre` grid has been built.

triang.conf Script containing utility functions used by the `triang` script.

transfer_solps-iter_runs Script for transferring a **SOLPS-ITER** case. If a run directory from machine X is to be transferred to machine Y then the script is to be run from machine X and just one level up from the directory which is to be transferred. The script will trace back and update the soft links in the run directory according to the remote environment. Running the script requires knowing the IP address of the remote machine and have a username and password to log on to it. This script makes use of the `sshpass` utility.

truncate_tracing_files Script used by the crash recovery program `b2co` to truncate the optional tracing files so that they stay synchronized with the recovered plasma state.

ustcsubmit Command file to launch runs on the USTC computer cluster. See Section 3 for details.

v2d Converts `Carre` format equilibrium files to `DG` format (see Sec. K.6.1).

- Some new Python based scripts have also been added:

calculate_sputter_inner_target.py Calculates the physical sputtering to be expected at the inner target in a simulation saved in MDSplus (chapter 6) on the `solps-mdsplus.aug.ipp.mpg.de` server (“-h” help is provided).

calculate_sputter_outer_target.py Calculates the physical sputtering to be expected at the outer target in a simulation saved in MDSplus (chapter 6) on the `solps-mdsplus.aug.ipp.mpg.de` server (“-h” help is provided).

compare_2_solps.py Compares two SOLPS shots with same dimensions saved in MDSplus (chapter 6) on the `solps-mdsplus.aug.ipp.mpg.de` server (“-h” help is provided).

energy_analysis.py Energy passing through various surfaces.

fluid_core_fluxes.py #1 #2 Total fluid flux entering the core region summed from species #1 to #2.

fluid_fluxes.py #1 #2 Fluid fluxes through various surfaces summed from species #1 to #2.

nareg.py #1 Average ion densities for region #1.

nareg_species.py #1 Average ion densities for region #1 summed over charge states for each species.

nereg.py Average electron densities.

rgradreg.py #1 Line radiation summed over each species’ charge states for region #1.

rranareg.py #1 Recombination rates for region #1.

rranareg_per_rsanareg.py #1 Recombination rates divided by ionization rates for region #1 (summed over charge states).

rsanareg.py #1 Ionization rates for region #1.

tereg.py Average electron temperatures.

tireg.py Average ion temperatures.

zeffreg.py Average Z_{eff} .

Still need to do : The following have not yet been documented — user contributions are welcome!

awk.derivative

blank

canonical

cnvtria

min_ave_max.awk

plot_bc_conpar13

plotdg

regxy_sum_over_species

2.5.1 scripts/commands

Contains various **b2plot** command files.

2.5.2 scripts/matlab

Contains various `Matlab` routines provided by users for data analysis and post-processing.

2.5.3 scripts/palettes

Contains the palettes that can be used by **b2plot** and `mtv_3d` scripts.

2.5.4 scripts/Balance

Contains the Matlab routines used for balance diagnostics, provided by David Moulton of CCFE. A detailed description of their use can be found in the document `${SOLPSTOP}/doc/Balance_routines_in_SOLPS-ITER.pdf`.

2.5.5 scripts/nc2text_simple

Contains the replacement skeleton program for the `nc2text` utility.

2.5.6 scripts/MatlabPostProcessing

Contains various Matlab routines used for postprocessing run results.

2.5.7 scripts/Matlab_SPb

Contains various Matlab routines used for postprocessing run results when activating the SPb code output switches (`b2wdat_iout` and the like).

2.6 data

Contrary to previous **SOLPS** versions, there is no more centralized version-controlled `${SOLPSTOP}/data` directory. The user may create one if needed, and use it a similar structure as in the past.

For data files that are still experimental in nature, the option to create a parallel `data.local` directory, with the same structure (only putting in the needed new files and subdirectories), at the level `${SOLPSTOP}/data.local` is also still available. The code routinely checks for the existence of local data before relying on the presence of Master copies.

Because the **SOLPS-ITER** distribution is split into submodules, the data files that are accessed exclusively by either `Eirene` or `B2.5` are now placed in directories named `Database` and placed in the corresponding `${SOLPSTOP}/modules` directory. Files that were needed by the post-processing or analysis scripts are now placed within the `${SOLPSTOP}/scripts` directory. Please refer to the appropriate sections above for more details.

Chapter 3

Running the standalone version of B2.5

The compilation of B2.5 is described in section 2.2.1 on page 33.

The run directory is described in section 2.4 on page 40. It is accessible with the `sbr` command.

The main code is `b2mn.exe` but it needs input files written by `b2ag.exe`, `b2ah.exe`, `b2ar.exe` and `b2ai.exe`. As described in section 2.4, these last four are usually run in the `baserun` directory, and then `b2mn.exe` is run in the actual run directory.

If one is running B2.5 in standalone mode, it is recommended to set the environment variable:

```
setenv STAND_ALONE yes
```

The general way of running any of these programs is by

```
gmake -f ${SOLPSTOP}/runs/Makefile b2XX.prt
```

or

```
gmake -f ${SOLPSTOP}/runs/Makefile b2fYYYYY
```

where `b2XX.exe` would be the program run, and `b2fYYYYY` would be one of the output files created by the program. The `Makefile` is best accessed via `${SOLPSTOP}/runs/Makefile` in the above examples and in the following sections. Alternatively, if you are running within the `$tt {SOLPSTOP}/runs` tree, the `Makefile` can be found as `../..../Makefile` where the number of `../`s would reflect how many levels the actual run directory is below the `${SOLPSTOP}/runs` directory.

A useful shortcut is to use

```
b2run b2XX
```

This automatically does the `gmake` for you, finding the `Makefile` for itself by climbing up the tree. You can add options to the `b2run` command:

- h** Print a help message.
- n** Will run the `Makefile` in dry-run mode. This is useful to check that all the needed files for the runs are in the right order of precedence. If they are not, the `correct_baserun_timestamps` script can be used to re-order the pre-requisite files found in the associated `baserun` directory
- m "MPI-launcher + options"** Allows to launch `b2XX` in MPI mode with the provided options (example: [the syntax is important, both the slashes and the quotes are needed] `b2run -m \"mpiexec -np 2\" b2mn`)
- s** Specifies that one wishes to run B2.5 in standalone mode, i.e. without any coupling to Eirene.
- o "run options"** Allows to launch `b2XX` with additional run options. The options can be grouped together inside quotes or given individually with multiple `"-o"` instances.

-t # Requests an OpenMP run using # threads.

-b Specifies, when calling the `b2co` program, that only the `b2fstate` file should be produced. To be used when invoking `b2co` while the main code is running.

-d "debugger_name" Allows to run `b2XX` with the named debugger (example: `b2run -d totalview b2mn`). If no debugger name is specified, the debugger defined with the `DBX` environment variable and specified in the `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}.local` or `${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}` files will be used.

The use of `b2run` allows for the run to be launched in a file space that is outside the `${SOLPSTOP}` tree, provided there is a file named `SOLPSTOP` in the run directory or above that contains the path to the `${SOLPSTOP}` tree. If the file named `SOLPSTOP` is missing, the submission script (`QSUB.*` or `SLURM.*`) will construct `${SOLPSTOP}` by truncating the path to actual run directory `${SOLPSTOP}/runs/<RUN_DIRECTORY>` to `${SOLPSTOP}`.

To stop an on-going run in a clean way, before it has reached the initially requested number of iterations, one can use, from the run directory:

```
touch b2mn.exe.dir/.quit
```

This will force the code to stop cleanly at the end of the current iteration, and produce all the output files in an orderly fashion, which then can be post-processed with the usual tools.

When submitting jobs in a cluster environment, one will make use of the job submission commands provided in the `${SOLPSTOP}/runs` directory (`QSUB.*`, `SBATCH.*` or `SLURM.*` files). Currently, the following commands are provided:

`ippsubmit` for submissions on the new (2019) IPP cluster

`itersubmit` for submissions on the ITER HPC cluster

`itmsubmit` for submissions on the Marconi ITM Gateway machine

`marconisubmit` for submissions on the Marconi supercomputer

`jetsubmit` for submissions on the JET Analysis Cluster (JAC)

`jfrssubmit` for submissions on the JFRS IFERC-CSC supercomputer

`localsubmit` for local submission

`sgesubmit` for submissions on a system running SGE

`ustcsbmit` for submissions on the USTC computer cluster

Users on other systems are encouraged to recopy these command scripts (found in `${SOLPSTOP}/scripts`) and adapt them to their local environment, along with the needed modifications to the `${SOLPSTOP}/runs/QSUB.*` or `${SOLPSTOP}/runs/SLURM.*` actual submission scripts. Each submission command file can handle five possible cases:

`QSUB.*`, `SLURM.*` Runs a standalone `B2.5` case.

`QSUB.*-gz`, `SLURM.*-gz` Runs a standalone `B2.5` case where the output log file has been compressed (activated with the `"-z"` option).

`QSUB.*_coupled`, `SLURM.*_coupled` Runs a coupled **SOLPS-ITER** case.

`QSUB.*_coupled-gz`, `SLURM.*_coupled-gz` Runs a coupled **SOLPS-ITER** case where the output log file has been compressed (activated with the `"-z"` option).

Any otherwise named file specified by the user and given as an argument to the submission command (in which case the `"-z"` option is ineffective). The specified file must be located in the `${SOLPSTOP}/runs` directory, or in the actual run directory, or one above it.

where "*" stands for "iter", "itm", "jet", "lsf", or "sge", referring to the machine system on which the submission script is to be used. The "_coupled" scripts are used when the presence of an Eirene input.dat input file is detected in the run directory. Looping scripts, with a "_loop" extension, are also available for some systems. These will automatically relaunch a case at the end of the time allotted to an individual job submission, provided the run appears to have finished cleanly. If the run appears to have failed, the looping script will stop. All the submission scripts make use of b2run so the same possibility of submitting runs from outside the \$SOLPSTOP file space is available.

If, for instance, one is running on the ITER cluster, then, the instructions to follow would be to run the script

```
itersubmit
```

or

```
itersubmit -z
```

or

```
itersubmit <<Submission_file>>
```

from your current run directory. This script climbs up the tree looking for <<Submission_file>>, and submits the job to the cluster by the command

```
sbatch <<Job_file>>
```

where <<Job_file>> is a temporary modified copy of <<Submission_file>> with changes corresponding to the particular job submission at hand (built automatically by itersubmit), or stops if the submission file is not found.

If <<Submission_file>> is not specified, itersubmit looks for SLURM.iter_coupled or for SLURM.iter depending on whether a coupled or a standalone run is required. To request a standalone run explicitly, the user can use the "-s" option. Otherwise, the script will assume a coupled run is requested if an Eirene input.dat file is present in the run directory, and a standalone run otherwise.

In the case one wishes to run the code with MPI and needs to pass MPI options to the launching program, this is achieved by means of the "-m" option, followed by the desired options inside double quotes, as follows:

```
itersubmit -m "<my MPI options>" <my_optional_submission_file_name>
```

The launching program is identified by the variable \$MPI_EXEC inside the submission file, and options can be by hand as well using the \$MPI_OPTS variable.

If the user wishes to submit an OpenMP job, this is done by use of the "-t" option, followed by the number of desired CPU threads.

The default job name is SOLPS-ITER for coupled jobs and B2.5 for standalone jobs. It can be changed by means of the

```
itersubmit -j <my_job_name>
```

command. Standalone jobs are requested with the "-s" option.

The number of CPU hours dedicated to the job (with possible consequences for the queue on which the submission will take place) can be modified by means of the

```
itersubmit -T 24
```

command, here to ask for a 24-hour long job. The various submission environments will have different default values for the job duration. Similarly, the requested memory size can be modified from the queue default with the

```
itersubmit -M 8GB
```

command, to obtain here a run using 8GB of memory. The "B" (for "bytes") is necessary in this syntax for SLURM submission, but should be omitted when using QSUB submission.

The partition on which the job will be run can be specified by means of the `-Q` option. For example:

```
itersubmit -Q gen9_ib
```

chooses the `gen9_ib` partition. By default on the ITER HPC cluster, jobs requiring 24 hours or less are submitted on `gen9`, while longer jobs go to `gen9_ib`.

A listing of the available options for each submission script can be obtained by the

```
itersubmit -h
```

command. All options (except `-h`) can be used simultaneously and in any order, provided the name of the (optional) submission file is given last.

The `SLURM.iter` script first of all changes directory to `${SOLPSTOP}` with the help of `SOLPSTOP` file and executes `setup.csh` there. This is necessary to set up environment variables for the cluster node which would actually execute the job, since that machine does not necessarily coincide with one from which the job was submitted. Then the `SLURM.iter` script returns back to the current run directory and executes the script `b2run`.

The examples presented below are for a pure D plasma.

3.1 b2ag, setting up the geometry

b2ag prepares the geometry file, b2fgmtry, reading the input file b2ag.dat

It is run by doing

```
gmake -f $SOLPSTOP/runs/Makefile b2ag.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2fgmtry
```

or

```
gmake -f $SOLPSTOP/runs/Makefile fort.30
```

or, best:

```
b2run b2ag
```

3.1.1 b2ag.dat

In the following example of b2ag.dat, we identify the mesh file as jet.50401.58000.geo, identify it as being of size 96×24 and require a 96×24 b2fgmtry file to be prepared.

```
*dimens          (nx, ny, nxl, nyl; free format)
  96  24   96  24
*param           (param(0:99); free format)
-1.0,  0.0,  0.75,  1.0, -0.010,  0.0,  0.040,
  5.0,  1.0,  0.02,  0.05,  0.05,  0.05,
  0.06,  1.0,
 100*0.0
'b2agfs_geometry'      'jet.50401.58000.geo'
```

***dimens** specifies the size of the grid

- first pair is NX & NY of the grid you want to produce
- second pair is the size of the grid that was originally created
 - each needs to be an integer multiple of the corresponding entry of the first pair. Note that for double-null cases, the interior guard cells corresponding to the top divertor boundaries should not be multiplied.

***param** at least 100 additional numbers, of which only the first is relevant for us

-1.0 read the mesh data using the "simplified" Carre format

-2.0 read the mesh data using the Sonnet format

b2agfs_geometry specifies the mesh file to be read

- searched for in (in order)
 1. the b2ag.exe.dir sub-directory of the current run directory
 2. the current run directory
 3. \$SOLPSTOP/data.local/meshes
 4. \$SOLPSTOP/data/meshes

... other options can be used to modify the mesh and magnetic field information that are read in from the mesh file (see Appendix A.4)

b2ag will also prepare the grid information in the format usable by Eirene, i.e. a fort.30 file.

3.2 b2ah, setting up the default physics parameters

b2ah prepares the default physics file, b2fpardf, reading the input file b2ah.dat

```
gmake -f $SOLPSTOP/runs/Makefile b2ah.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2fpardf
```

or, best:

```
b2run b2ah
```

3.2.1 b2ah.dat

The recommended way of specifying the actual values used for boundary conditions and transport coefficients is specified later.

```
*dimens          (ns; free format)
  2
*label           (lblps, character*60; free format)
  'Species D'
*b2cmpa          basic parameters
*specs          za      zn      am      za2
  '0'           0.0      1.0      2.0      0.0
  '1'           1.0      1.0      2.0      1.0
*b2cmpb          boundary conditions
*cbregs
  'south'       3        0.25     0.75     2.00
  'north'       1        2.0
  'west'        1        2.0
  'east'        1        2.0
*region
  0 'south, private flux'
*cbsna          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        0        -1.0e-02  0        0        0        0        0
  '1'           0        0        -1.0e-03  0        0        0        0        0
*cbsmo          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        0        0        -1.0e+10  0        0        -1.0      0
  '1'           0        0        0        0        0        0        -1.0      0
*cbshi          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           1.00e-18 -100.0  -1.0e-01  0        0        0        0        -2.5
  '1'           1.00e-18 -100.0  -1.0e-02  0        0        0        0        -2.5
*cbshe          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '-1'          1.00e-18 -100.0  -1.0e-04  0        0        0        0        -2.5
*cbsch          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '-1'          0        0        0        0        0        0        0        0
*cbrec          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        0        0        0        0        0        0        0
  '1'           1.0      0.3      0        0        0        0        0        0
*cbmsa          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '-1'          0        0        0        0        0        0        0        0
*cbmsb          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        0        0        0        0        0        0        0
  '1'           0        0        0        0        0        0        0        0
*region
  1 'south, core plasma'
*cbsna          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        -1.0e+10  0        0        0        0        0        0
  '1'           4.00e+29 -1.0e+10  0        0        0        0        0        0
*cbsmo          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
  '0'           0        0        0        -1.0e+10  0        0        0        0
  '1'           0        0        0        -1.0e+10  0        0        0        0
*cbshi          (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
```

'0'	1.00e+04	0	0	0	0	0	0	-2.5
'1'	1.00e+04	0	0	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e+04	0	0	0	0	0	0	-2.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*region								
2 'south, private flux'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	-1.0e-02	0	0	0	0	0
'1'	0	0	-1.0e-03	0	0	0	0	0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	-1.0	0
'1'	0	0	0	0	0	0	-1.0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'1'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	-1.0e-04	0	0	0	0	-2.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*region								
3 'north, SOL/DIV outer wall'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	-1.0e-02	0	0	0	0	0
'1'	0	0	-1.0e-03	0	0	0	0	0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	-1.0	0
'1'	0	0	0	0	0	0	-1.0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'1'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	-1.0e-04	0	0	0	0	-2.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*region								
4 'west, target plate'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	-1.0
'1'	0	0	0	0	0	0	0	-1.0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	-1.0e+10	0	0	0

'1'	0	0	1.0e+10	0	-1.0e+10	0	0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	1.0	0.5
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*region								
5 'east, target plate'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	-1.0
'1'	0	0	0	0	0	0	0	-1.0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	-1.0e+10	0	0	0
'1'	0	0	1.0e+10	0	-1.0e+10	0	0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	1.0	0.5
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*b2cmpt transport coefficients								
*cfdfo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.0e+02	0	1.0	5.0e-20	1.5e-18	0	0	0
'1'	0.5	0	0	0	0	0	0	0
*cfdna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	1.0	0	0	0	0	0
*cfdpa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*cfvla	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
*cfvsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.0	0	0	0	0	0
'1'	0	0	1.0	0	0	0	0	0
*cfhci	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.4	0	0	0	0	0
'1'	0	0	1.4	0	0	0	0	0
*cfhce	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0.7	0	0	0	0	0	0	0
*cfsig	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.0e-07	0	0	0	0	0	0	0
*cfalf	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cflim	cflme	cflmi	cflmv	cflal	cflab	...		
'-1'	0.15	0.15	0.5	0.5	0.5	0	0	0
*endphy								

The basic structure is

***dimens** the number of charge states

***label** specifies, on the next line, a label for the run

***b2cmpa** specifies a block of basic parameters

***b2cmpb** specifies a block of boundary conditions

***b2cmpt** specifies a block of transport coefficients

Taking each of the `b2cmp` blocks in turn:

b2cmpa, basic parameters

The `b2cmpa` block contains

***specs** atomic charge, nuclear charge, atomic mass and atomic charge squared; this data should match that given in `b2ai.dat`

Starting with code version 01.001.024, an alternative means of describing the plasma species and filling out the `b2cmpa` block is provided, in order to allow for bundling of charge states, when running cases with high- Z species. The relevant description is then

***specs** minimum atomic charge of the stage, maximum atomic charge of the stage, nuclear charge, and atomic mass; this data should match that given in `b2ai.dat`

The code can accept indifferently both types of input and makes the appropriate self-consistency checks. It is not permitted to bundle neutral and ionized species together.

b2cmpb, boundary conditions

The `b2cmpb` block contains

***cbregs** specifies the number of regions where boundary conditions will be specified

- the "south" boundary is broken into three pieces with one quarter of the cells in the first region, half of the cells in the second, and the remaining quarter in the third region (West target private flux, core, East target private flux). This is geometry-dependent information the code will check against the mesh connectivity and return an error if the two do not match
- the "north", "west" and "east" are not broken up, and correspond to the SOL boundary, inner (outer) target and outer (inner) target, respectively, for lower (upper) single-null configurations. For double-null cases, the north boundary should be split into two sections

***region** one block for each of the regions containing; the immediately following line being the region identifier

***cbsna** boundary conditions for density (1 line per species)

***cbsmo** boundary conditions for parallel momentum (1 line per species)

***cbshi** ion/neutral (or "atomic") temperature/energy boundary condition (1 line per species)

***cbshe** electron temperature/heat flux boundary condition

***cbsch** boundary condition for the electric potential equation

***cbrec** recycling coefficients (1 line per species)

***cbmsa** [unused]

***cbmsb** [unused] (1 line per species)

The exact form of the specification is somewhat arcane: users are referred to the code (`b2stbc_bas`) for the definitive meaning, or to the write-up in Appendix D.

Users are encouraged to use the "physics" formulation of the boundary conditions specified later (see section 3.5.2), rather than the "Bas" formulation listed here. For double-null cases, the "south" boundary must be split in six pieces, and the "north" boundary in two. However, the `b2ah` format does not allow for the specification of additional "east" or "west" boundaries, so double-null users **must** use the "physics" formulation in addition. The code enforces this requirement.

The boundary condition implemented arises from numerical combinations of the eight parameters with various quantities from the code.

Anything from this block can be overridden in `b2mn.dat` and, depending on switches in `b2mn.dat`, by `b2.boundary.parameters` (the recycling coefficients can be overridden by `b2.neutrals.parameters` (see section 3.5.7)).

b2cmpt, transport coefficients

The `b2cmpt` block contains

- ***cfdf0** auxiliary transport quantity (1 line per species)
- ***cfdna** density driven particle diffusion coefficient (1 line per species)
- ***cfdpa** pressure driven particle diffusion coefficient (1 line per species)
- ***cfvla** anomalous "pinch" velocity (1 line per species)
- ***cfvsa** anomalous transverse momentum diffusivity (1 line per species)
- ***cfhci** anomalous ion/neutral heat diffusivity (1 line per species)
- ***cfhce** anomalous electron heat diffusivity
- ***cfsig** diffusivity corresponding to the anomalous electrical conductivity
- ***cfalf** diffusivity corresponding to the anomalous thermo-electric coefficient
- ***cfllm** coefficients for flux limiting

Users are encouraged to use the "physics" formulation of the transport coefficients using the namelist formulation in `b2.transport.parameters` or `b2.transport.inputfile` as described in sections 3.5.3 and 3.5.5.

The transport coefficients implemented arise from numerical combinations of the seven parameters with various quantities from the code.

Anything from this block can be overridden in `b2mn.dat` and, depending on switches in `b2mn.dat`, by use of the `b2.transport.parameters` (section 3.5.3) and `b2.transport.inputfile` (section 3.5.5) files.

Fuller descriptions are found in `b2cdcv.F` in the `documentation` sub-directory of the B2.5 source code (see Appendix A.3), and the relevant subroutine is `b2tqna`.

It is important, for numerical stability reasons, to provide mechanisms by which some particle transport and radial current (if the electric potential equation is being solved) can be established. If the transport is given by `b2cmpt`, this means that, first, the `cfdna` and `cfdpa` arrays cannot be simultaneously zero when solving density equations. Second, the arrays `cfsig` and `cfalf` cannot be simultaneously zero when solving the potential equations without drift current terms on. The code will fail a transport coefficients test in `b2tral` otherwise.

3.3 b2ar, setting up the atomic physics rates

b2ar prepares the default atomic physics rates file, b2frates, reading the input file b2ar.dat

```
gmake -f $SOLPSTOP/runs/Makefile b2ar.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2frates
```

or, best:

```
b2run b2ar
```

3.3.1 b2ar.dat

```
*tlohi      (tlo, thi; real, free format)
  1.0e-1  1.0e4
*nlohi      (nlo, nhi; real, free format)
  1.0e18  1.0e22
*numnuc     (nnuc; integer, free format)
  1
*nucspec    (nz, izlo, izhi; integers, one triplet per line)
  1  0  1
*flag       (either 'adpak' or 'strahl', free format)
  'adas'
*tailep     (real, free format)
  0.08
```

tlohi range of temperatures for atomic physics table in eV

nlohi range of densities for atomic physics table in m^{-3}

numnuc number of distinct isonuclear sequences

nucspec nuclear charge, lowest charge state, highest charge state

flag which atomic physics database to use

tailep smoothing parameter for the temperature dependence of some rate coefficients

switches optional, described in b2cdci, (see Appendix A.4)

The allowable choices for the **flag** switch are:

"adpak" A complete atomic physics package with scaling law rates applicable to all charge states, but not always of high accuracy [4]

"strahl" A collisional-radiative package developed at IPP–Garching [5], stored in the `stra.dat` file

"adas" (Atomic Data and Analysis Structure) A complete collisional-radiative atomic physics database, actively being maintained and upgraded [6, 7]. Recommended. Available at <http://adas.phys.strath.ac.uk/>

"amns" (Atomic, Molecular, Nuclear, and Surface data) For IMAS and ITM environment runs only, uses access to the AMNS tools and database

Unless the user chooses to activate the `"b2ardr_no_weisheit"` switch (see section A.4 for description), the hydrogen rates come from [8] and [9] and overwrite those provided by the data packages above. Those data are stored in the `weis.dat` file which should be present in the `baserun` directory where `b2ar` is run. When running `b2ar.dat` using the dedicated `Makefile`, this is automatically done.

Supplemental information on atomic physics rates can be found in the file

`${SOLPSTOP}/doc/SOLPS_2002_Course/atomic_physics.pdf`.

3.4 b2ai, the default initial plasma state

b2ai prepares the default initial plasma state file, b2fstati, reading the input file b2ai.dat

```
gmake -f $SOLPSTOP/runs/Makefile b2ai.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2fstati
```

or, best:

```
b2run b2ai
```

3.4.1 b2ai.dat

```
*dimens          (ns; free format)
  2
*label           (lblps, character*60; free format)
  'Flat initial state for D, all charge states'
*specs           za      zn      am      za2
  '0'           0.0      1.0      2.0      0.0
  '1'           1.0      1.0      2.0      1.0
*naini           (na0(0:ns-1); free format)
  1.0e12  1.0e19  100*1.0e0
*ttini           (tt0; free format)
  1.0e2
```

The basic structure is

***dimens** the number of charge states

***label** a label

***specs** atomic charge, nuclear charge, atomic mass and atomic charge squared for each charge state

***naini** initial densities for each of the charge states, in m^{-3}

***ttini** initial ion and electron temperature, in eV

The initial potential is set to 3.1 times the initial temperature.

Starting with code version 01.001.024, an alternative means of describing the plasma species and filling out the b2cmpa block is provided, in order to allow for bundling of charge states, when running cases with high- Z species. The relevant description is then

***specs** minimum atomic charge of the stage, maximum atomic charge of the stage, nuclear charge, and atomic mass; this data should match that given in b2ah.dat

The code can accept indifferently both types of input and makes the appropriate self-consistency checks. It is not permitted to bundle neutral and ionized species together.

The initial state being created by b2ai.dat will be a homogeneous isothermal static plasma, which is very far from a realistic plasma edge solution. It is therefore almost always better and strongly recommended to start a case using an already converged case of the same topology and containing the same main species, even it does not have the same grid size, or corresponds to a different device, or if the species list is somewhat different. To convert such a solution to the parameters of the case at hand, the b2yt converter is provided. See Section 3.14 for details on how to use it.

It is possible, by adding the switch

```
'b2aidr_read_b2fstate'    '1'
```

at the end of the `b2ai.dat` file, to obtain a new starting point file which will include the data from the already computed solution in the locally present `b2fstate` file, and to which are added fields corresponding to a flat state for new species declared in the species list from `b2ai.dat`.

3.5 b2mn, the main program

The program can be run with one of the following

```
gmake -f $SOLPSTOP/runs/Makefile b2mn.prt
gmake -f $SOLPSTOP/runs/Makefile b2fparam
gmake -f $SOLPSTOP/runs/Makefile b2fstate
gmake -f $SOLPSTOP/runs/Makefile b2fmovie
gmake -f $SOLPSTOP/runs/Makefile b2ftrace
gmake -f $SOLPSTOP/runs/Makefile b2ftrack
gmake -f $SOLPSTOP/runs/Makefile b2fplasma
```

or, best:

```
b2run b2mn
```

With all of these the code output will come to the screen. It is recommended that it be re-directed to the file `run.log` and the program run in the background. Thus

```
b2run b2mn > run.log &
```

The code reads its data from `b2mn.dat`, and optionally from a number of other files, as described below. Options for monitoring the code run are described in section 3.5.11.

3.5.1 b2mn.dat

```
*label          (lblmn: character*60)
'JET : no drifts : D only : no drifts'
*b2cmpa  basic parameters
*b2cmpb  boundary conditions
*cbregs
'south'    3      0.25      0.75      2.00
'north'     1      2.0
'west'      1      2.0
'east'      1      2.0
*region
4 'west, target plate'
*cbrec      (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'0'         0        0        0        0        0        0        0        0
'1'         1.0      0.3      0        0        0        0        0        0
*region
5 'east, target plate'
*cbrec      (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'0'         0        0        0        0        0        0        0        0
'1'         1.0      0.3      0        0        0        0        0        0
*b2cmpt  transport coefficients
*cfvsa     (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'0'         0        0        1.0      0        0        0        0        0
'1'         0        0        1.0      0        0        0        0        0
*cfsig     (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'-1'        1.00E-07 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00
*cfalf     (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'-1'        0.00E+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00
*endphy
'b2mndr_ntim'      '1000'
'b2mndr_dtim'      '1.0e-3'
'b2mndt_nstg0'     '1'
'b2mndt_nstg1'     '1'
'b2mndt_nstg2'     '1'
'b2mndt_rxf'       '0.5'
'b2mndr_mvinc'     '1'
'b2mndr_mvnum'     '0'
'b2mndr_b2time'    '10'
```

```

'b2mndr_cpu'      '182800.0'
'b2mndr_savecpu'  '3600.0'
'b2ux5p_nltrsol'  '0'
'b2ux5p_style'    '2'
'b2stbc_sna0ep'   '1.0e-10'
'b2trcl_lluciani' '1'
'b2news_potit'    '100'
'b2news_potok'    '1e-2'
'b2trcl_lvis21'   '0'
'b2trcl_lthf21'   '0'
'b2tfhe_neutral'  '0.0'
'b2tfhe_vis_par'  '0.0'
'b2news_facdrift_target' '0.00'
'b2news_facdrift_start' '0.00'
'b2news_facdrift_inc'  '1.01'
'b2news_facdrift_dec'  '0.95'
'b2news_facExB_target' '0.00'
'b2news_facExB_start'  '0.00'
'b2news_facExB_inc'    '1.01'
'b2news_facExB_dec'    '0.95'
'b2stbc_fchy_dia_coreonly' '1'
'b2stbc_boundary_namelist' '1'
'b2srdt_numerics_namelist' '1'
'b2tqna_transport_namelist' '1'
'b2mwti_jxi'        '40'
'b2mwti_jxa'        '54'
'b2tlh0_alpha'      '1.0'
'b2tfnb_alpha'      '1.0'
'b2news_potitmin'   '5'
'b2mndr_eirene'     '0'
'b2stbc_feedback'   '1'
'b2stbc_nesepm_sol' '1e19'
'b2stbc_diagno'     '1'
'b2stbc_solregno'   '3'

```

The basic structure of `b2mn.dat` is

***label** specifies, on the next line, a label for the run

***b2cmpa** specifies a block of basic parameters, overriding those specified in the block of the same name in `b2ah.dat`

***b2cmpb** specifies a block of boundary conditions, overriding those specified in the block of the same name in `b2ah.dat`

***b2cmpt** specifies a block of transport coefficients, overriding those specified in the block of the same name in `b2ah.dat`

switches described in `b2cdci`, (see Appendix A.4)

3.5.2 b2.boundary.parameters

See also Appendix A.5 for a more up-to-date description.

If `'b2stbc_boundary_namelist'` is set to 1 in `b2mn.dat` then the boundary conditions are read in from `b2.boundary.parameters` instead of being taken from the file created by `b2ah` and possibly modified in `b2mn.dat`. For cases with a double-null topology, use of `b2.boundary.parameters` is required, and will be enforced irrespective of the value of the `'b2stbc_boundary_namelist'` switch.

The boundary conditions are specified by the boundary namelist which contains

NBC integer, number of boundary segments

BCCHAR character array, length NBC, specifying the nature of the boundary segment

N "North" boundary

- S** "South" boundary
- W** "West" boundary
- E** "East" boundary
- X** "X" boundary used for specifying a fixed value on a row of cells
- Y** "Y" boundary used for specifying a fixed value on a column of cells
- BCPOS** integer array, length NBC, for North, South or X boundary conditions, it specifies the row index; for West, East and Y boundary conditions, it specifies the column index
- BCSTART** integer array, length NBC, for North, South or X boundary conditions, it specifies the start column index; for West, East and Y boundary conditions, it specifies the start row index
- BCEND** integer array, length NBC, for North, South or X boundary conditions, it specifies the end column index; for West, East and Y boundary conditions, it specifies the end row index
- BCCON** integer array, length NS * NBC, specifying the type of density boundary condition for each segment and species (fastest varying index is species); makes use of CONPAR to specify additional information, as indicated
- 0** no boundary condition is applied (default)
 - 1** prescribe the value of the density, CONPAR (, , 1) specifies the required density in m^{-3}
 - 2** prescribe the gradient of the density, CONPAR (, , 1) specifies the required density gradient in m^{-4}
 - 3** sheath conditions, CONPAR (, , 1) not used (zero gradient is used)
 - 4** prescribe the value of the density, weakly a mixed boundary condition, CONPAR (, , 1) specifies the required density in m^{-3} and CONPAR (, , 2) specifies the "strength" of the boundary condition
 - 5** prescribe the particle flux per unit area, CONPAR (, , 1) specifies the required particle flux density in $m^{-2}s^{-1}$
 - 6** prescribe the total particle flux for a constant density, CONPAR (, , 1) specifies the particle flux in s^{-1}
 - 7** prescribe the density as a function of other plasma parameters [not yet available]
 - 8** prescribe the total particle flux with constant flux density, CONPAR (, , 1) specifies the particle flux in s^{-1}
 - 9** prescribe the decay length for the density, CONPAR (, , 1) specifies the gradient length in m
 - 10** leakage option for density, recommended for cases with drifts, CONPAR (, , 1) specifies the leakage factor, α in $\Gamma_{\text{loss}} = \alpha C_{s,a} n_a$
 - 11** particle flux feedback boundary condition, CONPAR (, , 1) not used, derived from CBSNA (0 , IS , IREG) . The species used must be declared using the b2stbc_isfeedback switch.
 - 12** particle density feedback boundary condition, as above, CONPAR (, , 1) not used, derived from CBSNA (0 , IS , IREG) . The species used must be declared using the b2stbc_isfeedback switch.
 - 13** particle density to achieve specified total flux, CONPAR (, , 1) is the specified radial flux crossing the surface b2stbc_type13_ref steps away from the boundary, to which is added the returning flux of ionised neutrals that crossed into the plasma core (via CONPAR (, , 3)) when running with Eirene and the 'ionising_core' switch is used, CONPAR (, , 2) is the strength of the feedback.
 - 14** sound speed velocity flux, CONPAR (, , 1) is a multiplier to the outgoing sound speed $C_{s,\text{collective}}$. Recommended sheath boundary condition for use with drifts, in conjunction with BCENE/I=15, BCPOT=11, and BCMOM=13.
 - 15** prescribe a radial leakage velocity, CONPAR (, , 1) specifies the leakage velocity in units of the local thermal velocity.

- 16 particle density to achieve specified total flux, used with ASTRA coupling. The total desired flux is summed over all BCCON=16 core boundaries. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
 - 18 prescribe total main ion particle flux, used with ASTRA coupling.
 - 19 particle flux feedback boundary condition, flux is summed over neutrals and ions, for coupling with ASTRA. The total desired flux is summed over all BCCON=19 core boundaries. This boundary condition applies to ions in their highest ionisation stage. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
 - 20 constant density feedback condition, CONPAR (, , 1) specifies the desired density in m^{-3} .
 - 21 prescribe the value of the density and add a density perturbation to get a solution which is as close as possible to neoclassical theory. It is recommended to use this boundary condition together with corresponding condition on ion temperature (BCENI=24). The total desired density is summed over all BCCON=21 core boundaries. CONPAR (, , 1) specifies the desired density in m^{-3} .
 - 22 Feedback boundary condition: given total particle flux with constant average density. The poloidal density variation is added to make the solution as close as possible to the neoclassical value derived for pure H/D/T plasma. It is recommended to use this boundary condition with corresponding boundary condition on ion temperature (BCENI=23, 24). The total desired flux is summed over all BCCON=22 core boundaries. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
 - 23 Feedback boundary condition: given sum of integrated neutrals and main ion particle fluxes with constant average density. The poloidal density variation is added to make the solution as close as possible to the neoclassical value derived for pure H/D/T plasma. It is recommended to use this boundary condition with corresponding boundary condition on ion temperature (BCENI=23, 24). The total desired flux is summed over all BCCON=23 core boundaries. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
 - 24 constant density feedback scaled by density on the ring bc.type21_ref away CONPAR (, , 1) specifies the desired density in m^{-3} . CONPAR (, , 2) is the strength of the feedback
 - 25 Feedback boundary condition: prescribe the average value of the density and add a density perturbation from neighbouring radial cell. This boundary condition is suitable for any species of a multi-species plasma (i.e. in the case when the condition used in BCCON=21 fails). It is recommended to use this boundary condition together with the corresponding condition on ion temperature (BCENI=26, 27). CONPAR (, , 1) specifies the desired average density in m^{-3} .
 - 26 Feedback boundary condition: prescribe the total ion flux and find the average density. A density perturbation is taken from the neighbouring radial cell. This boundary condition is suitable for any species of a multi-species plasma (i.e. in the case when the condition used in BCCON=22 fails). It is recommended to use this boundary condition together with a corresponding condition on the ion temperature (BCENI=26, 27). The total desired flux is summed over all BCCON=26 core boundaries. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
 - 27 Feedback boundary condition: prescribe the particle flux sum for all neutrals and ions belonging to a given isonuclear sequence and find the average density of the highest ionization stage. A density perturbation is taken from the neighbouring radial cell. It is recommended to use this boundary condition together with corresponding condition on ion temperature (BCENI=25, 26). The total desired flux is summed over all BCCON=27 core boundaries. CONPAR (, , 1) specifies the desired particle flux in s^{-1} .
- BCMOM** integer array, length NS * NBC, specifying the type of parallel momentum or velocity boundary condition for each segment and species (fastest varying index is species); makes use of MOMPARG to specify additional information, as indicated
- 0 no boundary condition is applied (default)
 - 1 prescribe the value of the parallel velocity, MOMPARG (, , 1) specifies the parallel velocity in $m.s^{-1}$
 - 2 prescribe the gradient of the parallel velocity, MOMPARG (, , 1) specifies the parallel velocity gradient in s^{-1}

- 3 sheath conditions, Mach number as input, if $\text{MOMPAR}(\text{ , , } 2) < \frac{1}{2}$, then the velocity is set to exactly $\text{MOMPAR}(\text{ , , } 1) \times C_{s,\text{collective}}$, otherwise the velocity is set to be at least $\text{MOMPAR}(\text{ , , } 1) \times C_{s,\text{species}}$
 - 4 prescribe the value of the velocity, weakly a mixed boundary condition, $\text{MOMPAR}(\text{ , , } 1)$ specifies the parallel velocity in $m.s^{-1}$ and $\text{MOMPAR}(\text{ , , } 2)$ specifies the "strength" of the boundary condition
 - 5 prescribe the parallel momentum flux per unit area, $\text{MOMPAR}(\text{ , , } 1)$ specifies the parallel momentum flux density in $N.m^{-2}$
 - 6 prescribe the total parallel momentum flux for a constant parallel velocity [not yet available]
 - 7 prescribe the parallel momentum as a function of other plasma parameters [not yet available]
 - 8 special : limited shear, imposes zero gradient for the Mach number. [[[Eventually intended to have $\text{MOMPAR}(\text{ , , } 1)$ specify the gradient of the Mach number in m^{-1}]]]
 - 9 prescribe the total parallel momentum flux with constant flux density, $\text{MOMPAR}(\text{ , , } 1)$ specifies the parallel momentum flux in N
 - 10 prescribe the decay length for the parallel momentum, $\text{MOMPAR}(\text{ , , } 1)$ specifies the decay length in m
 - 11 Rozhansky viscosity condition for the parallel momentum, $\text{MOMPAR}(\text{ , , } 1)$ is not used
 - 12 condition from `b2stbc_spb` for the parallel momentum
 - 13 sheath boundary condition from `b2stbc_spb` for the parallel momentum. Recommended boundary condition for use with drifts, in conjunction with `BCENE/I=15`, `BCPOT=11`, and `BCCON=14`.
 - 14 condition from `b2stbc_spb` for the parallel momentum
 - 15 prescribe the value of the parallel velocity as a function of poloidal coordinate, scaling with $\frac{\langle B \rangle}{B}$
 - 16 Prescribe the average value of the parallel velocity, $\text{MOMPAR}(\text{ , , } 1)$ specifies the parallel velocity in $m.s^{-1}$
 - 17 leakage option for parallel momentum, $\text{MOMPAR}(\text{ , , } 1)$ specifies the leakage factor, α in $\Gamma_{\text{loss}} = \alpha C_{s,a} m_a n_a u_a$
- BCENE** integer array, length NBC, specifying the type of electron energy or temperature boundary condition for each segment; makes use of `ENEPAR` to specify additional information, as indicated
- 0 no boundary condition is applied (default)
 - 1 prescribe the value of the electron temperature, `ENEPAR( , 1)` specifies the temperature in eV
 - 2 prescribe the gradient of the electron temperature, `ENEPAR( , 1)` specifies the temperature gradient in $eV.m^{-1}$
 - 3 sheath conditions, electron energy transmission, `ENEPAR( , 1)` specifies an additional contribution to the energy transmission coefficient in addition to that of the potential difference [the sound speed used depends on settings of `MOMPAR( , ISMAIN, 2)`]
 - 4 prescribe the value of the electron temperature, weakly a mixed boundary condition, `ENEPAR( , 1)` specifies the temperature in eV and `ENEPAR( , 2)` specifies the "strength" of the boundary condition
 - 5 prescribe the electron energy flux per unit area, `ENEPAR( , 1)` specifies the energy flux density in $W.m^{-2}$
 - 6 prescribe the total electron energy flux for a constant electron temperature, `ENEPAR( , 1)` specifies the energy flux in W (no longer supported, use types [16] or [17] instead)
 - 7 prescribe the electron temperature as a function of other plasma parameters [not yet available]
 - 8 prescribe the total electron heat flux with constant flux density, `ENEPAR( , 1)` specifies the energy flux in W
 - 9 prescribe the decay length for the electron temperature, `ENEPAR( , 1)` specifies the decay length in m (can also use type [19] instead)
 - 10 feedback option for core, `ENEPAR( , 1)` not used, derived from `cbshe(0, coreregno)`
 - 11 not used

- 12 sheath conditions, electron energy transmission coefficient, `ENEPAR(, 1)` specifies an energy transmission factor, δ_e in $Q_e = \delta_e \Gamma_e T_e$
- 13 prescribe the electron energy flux per unit area proportional to temperature, `ENEPAR(, 1)` specifies the energy flux density per temperature in $W.m^{-2}.J^{-1}$ (the temperature here in J)
- 14 leakage option for electron energy, `ENEPAR(, 1)` specifies the leakage factor, α in $\Gamma_{loss} = \alpha C_{s,collective} n_e T_e$
- 15 sheath boundary condition, from `b2stbc_spb`, recommended when using drifts (see Section C.7.4 for details). Linked to using `BCCON=14` and `BCMOM=13` for all ion species, `BCENI=15`, and `BCPOT=11`.
- 16 feedback boundary condition with constant temperature, `ENEPAR(, 1)` specifies the power flux in W across the flux surface with index `b2stbc_type16_ref` (default=-1), `ENEPAR(, 2)` should be something like 0.1 and specifies the strength of the feedback. Also see type [17] below. Available if `bcene_16_style=0` (default). If `bcene_16_style=1`, integrated electron heat flux with constant T_e , summed over all core boundaries with `BCENE=16`.
- 17 feedback boundary condition with constant shared temperature for both electrons and ions, with `ENEPAR(, 1) + ENIPAR(, 1)` giving the total power flux in W across the flux surface with index `b2stbc_type16_ref` (default=-1), `ENEPAR(, 2)` should be something like 0.1 and specifies the strength of the feedback. Replaces [16] for high densities and large values of `b2stbc_type16_ref`.
- 18 fractional drop condition. Not yet working.
- 19 same as [9] but to be used when simultaneously setting `BCCON=1` on the same boundary.
- 20 feedback boundary condition with constant temperature, as per type [16] but with `type20_` switches and a different feedback scheme.
- 21 constant temperature feedback scaled by temperature on the ring `bc_type21_ref` away. `ENEPAR(, 1)` specifies the desired electron temperature in eV . `ENEPAR(, 2)` is the strength of the feedback.
- 22 radial leakage condition for the electron temperature. `ENEPAR(, 1)` specifies the leakage velocity in units of the electron thermal velocity. A temperature gradient such that the diffusive flux is set to match this leakage is imposed.

BCENI integer array, length `NBC`, specifying the type of ion energy or temperature boundary condition for each segment; makes use of `ENIPAR` to specify additional information, as indicated

- 0 no boundary condition is applied (default)
- 1 prescribe the value of the ion temperature, `ENIPAR(, 1)` specifies the temperature in eV
- 2 prescribe the gradient of the ion temperature, `ENIPAR(, 1)` specifies the temperature gradient in $eV.m^{-1}$
- 3 sheath conditions, ion energy transmission, `ENIPAR(, 1)` specifies the contribution to the energy transmission coefficient [the sound speed used depends on settings of `MOMPAR(, ISMAIN, 2)`]
- 4 prescribe the value of the ion temperature, weakly a mixed boundary condition, `ENIPAR(, 1)` specifies the temperature in eV and `ENIPAR(, 2)` specifies the "strength" of the boundary condition
- 5 prescribe the ion energy flux per unit area, `ENIPAR(, 1)` specifies the energy flux density in $W.m^{-2}$
- 6 prescribe the total ion energy flux for a constant ion temperature, `ENIPAR(, 1)` specifies the energy flux in W (no longer supported, use types [16] or [17] instead)
- 7 prescribe the ion temperature as a function of other plasma parameters [not yet available]
- 8 prescribe the total ion heat flux with constant flux density, `ENIPAR(, 1)` specifies the energy flux in W
- 9 prescribe the decay length for the ion temperature, `ENIPAR(, 1)` specifies the decay length in m (can also use type [19] instead)
- 10 feedback option for core, `ENIPAR(, 1)` not used, derived from `cbshi(0, ISMAIN, coreregno)`

- 11 sheath conditions, ion energy transmission coefficient, `ENIPAR(, 1)` specifies an energy transmission factor, δ_i in $Q_i = \delta_i T_i \sum_a n_a C_{s,a}$
- 12 sheath conditions, ion energy transmission coefficient, `ENIPAR(, 1)` specifies an energy transmission factor, δ_i in $Q_i = \delta_i T_i \sum_a \Gamma_a$
- 13 prescribe the ion energy flux per unit area proportional to temperature, `ENIPAR(, 1)` specifies the energy flux density per temperature in $W.m^{-2}.J^{-1}$ (the temperature here in J)
- 14 leakage option for ion energy, `ENIPAR(, 1)` specifies the leakage factor, α in $\Gamma_{loss} = \alpha C_{s,collective} n_i T_i$
- 15 sheath boundary condition, from `b2stbc_spb`, recommended when using drifts (see Section C.7.5 for details). Linked to using `BCCON=14` and `BCMOM=13` for all ion species, `BCENE=15`, and `BCPOT=11`.
- 16 Feedback boundary condition with constant temperature, `ENIPAR(, 1)` specifies the power flux in W across the flux surface with index `b2stbc_type16_ref` (default=-1), `ENIPAR(, 2)` should be something like 0.1 and specifies the strength of the feedback. Also see type [17] below. Available if `bceni_16_style=0` (default). If `bceni_16_style=1`, integrated ion heat flux with constant T_i , summed over all core boundaries with `BCENI=16`.
- 17 Feedback boundary condition with constant shared temperature for both electrons and ions, see `BCENE=17` above for description.
- 18 Fractional drop condition. Not yet working.
- 19 same as [9] but to be used when simultaneously setting `BCCON=1` on the same boundary.
- 20 Feedback boundary condition with constant temperature, as per type [16] but with `type20_` switches and a different feedback scheme.
- 21 from `b2stbc_spb`
- 22 Radial leakage velocity for the ion temperature. `ENIPAR(, 1)` specifies the leakage velocity in units of the collective ion thermal velocity. A temperature gradient such that the diffusive flux is set to match this leakage is imposed.
- 23 Prescribe the poloidally averaged value of the ion temperature and introduce a poloidal variation as close as possible to neoclassical solution. It is recommended to use this boundary condition together with corresponding condition on ion density (`BCCON=21, 22, 23`). The average is taken over all core boundaries with `BCENI=23`. `ENIPAR(, 1)` specifies the temperature in eV
- 24 Feedback boundary condition with prescribed total ion flux, constant poloidally averaged ion temperature and a poloidal variation as close as possible to neoclassical solution. It is recommended to use this boundary condition together with corresponding condition on ion density (`BCCON=21, 22, 23`). The flux is summed over all core boundaries with `BCENI=24`. `ENIPAR(, 1)` specifies the energy flux in W
- 25 Constant temperature feedback scaled by temperature on the ring `bc_type21_ref` away `ENIPAR(, 1)` specifies the desired ion temperature in eV . `ENIPAR(, 2)` is the strength of the feedback
- 26 Prescribe the poloidally averaged value of the ion temperature and introduce a poloidal variation in a simplified manner. This boundary condition is suitable for any plasma composition (i.e. when `BCENI=23` fails). It is recommended to use this boundary condition together with corresponding condition on ion density (`BCCON = 25, 26, 27`). The average is taken over all core boundaries with `BCENI=26`. `ENIPAR(, 1)` specifies the temperature in eV
- 27 Feedback boundary condition with prescribed total ion heat flux, constant poloidally averaged ion temperature and a poloidal variation in a simplified manner. This boundary condition is suitable for any plasma composition (i.e. when `BCENI=24` fails). It is recommended to use this boundary condition together with corresponding condition on ion density (`BCCON = 25, 26, 27`). The flux is summed over all core boundaries with `BCENI=27`. `ENIPAR(, 1)` specifies the energy flux in W

BCPOT integer array, length NBC, specifying the type of electric potential or current boundary condition for each segment; makes use of POTPAR to specify additional information, as indicated

0 no boundary condition is applied (default)

1 prescribe the value of the potential, POTPAR (, 1) specifies the potential in V

2 prescribe the gradient of the potential, POTPAR (, 1) specifies the potential gradient in $V.m^{-1}$

3 sheath conditions, POTPAR (, 2) used for biasing [see code for details]

4 prescribe the value of the potential weakly a mixed boundary condition, POTPAR (, 1) specifies the potential in V and POTPAR (, 2) specifies the "strength" of the boundary condition

5 prescribe the current flux density per unit area, POTPAR (, 1) specifies the electric current flux density in $A.m^{-2}$

6 prescribe the total current flux density for a constant potential [not yet available]

7 prescribe the potential as a function of other plasma parameters [not yet available]

8 prescribe the total electric current with constant flux density, POTPAR (, 1) specifies the electric current in A

9 prescribe the decay length for the potential, POTPAR (, 1) specifies the decay length in m

10 feedback option for core [not yet tested!!!!!!!] (based on using cbsch (0, coreregno))

11 sheath conditions, electron energy transmission from b2stbc_spb, recommended when using drifts. Linked to using BCCON=14 and BCMOM=13 for all ion species, and BCENE/I=15. POTPAR (, 2) specifies the bias potential in V

12 imposes the currents due to drifts for the South core boundary. Must be used in conjunction with the switch setting istyle_cur_contr_on_S_and_N=2

13 imposes the currents due to drifts for the South private flux and North boundaries. Must be used in conjunction with istyle_cur_contr_on_S_and_N=2

16 feedback condition of constant electric potential for a given prescribed current. The prescribed current (in A) is given by the sum of the POTPAR (, 1) of all the boundary conditions with BCPOT=16. Still experimental, will not work for drift cases.

21 constant potential feedback scaled by potential on the ring bc_type21_ref away, POTPAR (, , 1) specifies the desired electric potential in V, POTPAR (, , 2) is the strength of the feedback.

CONPAR real array, length NS * NBC * 3, supplies the additional parameters needed for the option specified by BCCON

MOMPAR real array, length NS * NBC * 2, supplies the additional parameters needed for the option specified by BCMOM

ENEPAR real array, length NBC * 2, supplies the additional parameters needed for the option specified by BCENE

ENIPAR real array, length NBC * 2, supplies the additional parameters needed for the option specified by BCENI

POTPAR real array, length NBC * 2, supplies the additional parameters needed for the option specified by BCPOT

GAMMAI real, ratio of specific heats (adiabatic coefficient)

GAMMAE real, secondary electron emission coefficient

LBNDUSR logical, if true will also call Braams boundary condition routine after the end of the physics boundary condition routine

LFEEDBACK logical, if true enables various feedback mechanisms (needed for any feedback mechanisms)

NNISO integer, number of dead (or isolated) regions

NXISO1 integer array, length NNISO, column number of bottom left corner of the dead regions

NXISO2 integer array, length NNISO, column number of top right corner of the dead regions

NYISO1 integer array, length NNISO, row number of bottom left corner of the dead regions

NYISO2 integer array, length NNISO, row number of top right corner of the dead regions

NIISO real array, length NS, ion densities of the dead regions in m^{-3}

TEISO real, electron temperature of the dead regions in eV

TIISO real, ion temperature of the dead regions in eV

PHISO real, electric potential of the dead regions in V

BOUNDARY_FILENAME filename, the name of a new boundary input file that will be read based on a condition determined by $\text{mod}(\text{TIM}, \text{BOUNDARY\_TIME\_MOD}) > \text{BOUNDARY\_TIME\_SWITCH}$

BOUNDARY_TIME_MOD real, the basic time periodicity used to set time varying boundary conditions, in s

BOUNDARY_TIME_SWITCH real, the basic time at which a switch is to be made to a new boundary input file, in s

LCBS integer array, length NBC, indices of the core boundary segments in B2 and B2.5 for passing to Eirene

Additional boundary conditions modifications due to drifts are automatically included if drifts are turned on (the core current is adjusted to be its neo-classical value, and the Bohm criterion at the plates is adjusted to include drift velocities).

3.5.3 b2.transport.parameters

See also Appendix A.5 for a more up-to-date description.

If 'b2tqna_transport_namelist' is set to 1 in b2mn.dat then the transport coefficients are read in from the namelist found in b2.transport.parameters instead of being taken from the file created by b2ah and possibly modified in b2mn.dat. The transport coefficients are specified by the transport_namelist which contains

FLAG_DNA integer array, length NS, density driven particle diffusion model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_DNA (IS)

2 constant value specified by PARM_DNA (IS) scaled by $\frac{10^{20}}{n_e [m^{-3}]}$

3 constant value specified by PARM_DNA (IS) scaled by Bohm ($\frac{T_e [eV]}{16B [T]}$)

4 constant value specified by PARM_DNA (IS) (flux scaled)

FLAG_DPA integer array, length NS, pressure driven particle diffusion model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_DPA (IS)

2 constant value specified by PARM_DPA (IS) scaled by $\frac{10^{20}}{n_e [m^{-3}]}$

3 constant value specified by PARM_DPA (IS) scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_DPA (IS) (flux scaled)

FLAG_VLA integer array, length NS, radial pinch velocity model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_VLA (IS)

2 constant value specified by PARM_VLA (IS) scaled by $\frac{10^{20}}{n_e[m^{-3}]}$

3 constant value specified by PARM_VLA (IS) scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_VLA (IS) (flux scaled)

FLAG_VSA integer array, length NS, anomalous viscosity model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_VSA (IS)

2 constant value specified by PARM_VSA (IS) scaled by $\frac{10^{20}}{n_e[m^{-3}]}$

3 constant value specified by PARM_VSA (IS) scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_VSA (IS) (flux scaled)

FLAG_HCI integer array, length NS, anomalous ion thermal diffusivity model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_HCI (IS)

2 constant value specified by PARM_HCI (IS) scaled by $\frac{10^{20}}{n_e[m^{-3}]}$

3 constant value specified by PARM_HCI (IS) scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_HCI (IS) (flux scaled)

FLAG_HCE integer, anomalous electron thermal diffusivity model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_HCE

2 constant value specified by PARM_HCE scaled by $\frac{10^{20}}{n_e[m^{-3}]}$

3 constant value specified by PARM_HCE scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_HCE (flux scaled)

FLAG_SIG integer, anomalous current conductivity model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

1 constant value specified by PARM_SIG

2 constant value specified by PARM_SIG scaled by $\frac{10^{20}}{n_e[m^{-3}]}$

3 constant value specified by PARM_SIG scaled by Bohm ($\frac{T_e[eV]}{16B[T]}$)

4 constant value specified by PARM_SIG (flux scaled)

FLAG_ALF integer, anomalous thermo-electric current model

0 use value specified in b2cmpt block in b2ah.dat, perhaps overridden in b2mn.dat

- 1 constant value specified by PARM_ALF
- 2 constant value specified by PARM_ALF scaled by $\frac{10^{20}}{n_e [m^{-3}]}$
- 3 constant value specified by PARM_ALF scaled by Bohm ($\frac{T_e [eV]}{16B[T]}$)
- 4 constant value specified by PARM_ALF (flux scaled)

PARM_DNA real array, length NS, provides parameters for FLAG_DNA

PARM_DPA real array, length NS, provides parameters for FLAG_DPA

PARM_VLA real array, length NS, provides parameters for FLAG_VLA

PARM_VSA real array, length NS, provides parameters for FLAG_VSA

PARM_HCI real array, length NS, provides parameters for FLAG_HCI

PARM_HCE real, provides parameters for FLAG_HCE

PARM_SIG real, provides parameters for FLAG_SIG

PARM_ALF real, provides parameters for FLAG_ALF

CFLME real, multiplier to the electron flux limit

CFLMI real, multiplier to the ion flux limit

CFLMV real, multiplier to the viscosity flux limit

CFLAL real, multiplier to the thermo-electric coefficient flux limit

CFLAB real, multiplier to the friction force flux limit

TRANSPORT_FILENAME filename, the name of a new transport input file that will be read based on a condition determined by `mod (TIM,TRANSPORT_TIME_MOD) > TRANSPORT_TIME_SWITCH`

TRANSPORT_TIME_MOD real, the basic time periodicity used to set time varying transport coefficients, in s

TRANSPORT_TIME_SWITCH real, the basic time at which a switch is to be made to a new transport input file, in s

See also section 3.5.5 for an alternative description of the transport coefficients which allows for radial variation.

It is important to note that the settings from `b2.transport.parameters` will **NOT** be applied to neutral species!

As stated in the `b2cmpt` block description (Section 3.2.1), for numerical stability reasons, mechanisms must exist to provide some particle transport and establish radial currents. Therefore, the user must not, when determining the transport coefficients by use of the `b2.transport.parameters` file, set both `parm_dna` and `parm_dpa` simultaneously to zero when solving density equations. Similarly, `parm_sig` and `parm_alf` must not simultaneously be set to zero when solving the potential equations without drift current terms on. The code will fail a transport coefficients test in `b2tral` otherwise.

3.5.4 b2.numerics.parameters

See also Appendix A.5 for a more up-to-date description.

DTCO real array, length NS * (NREG+1), the timestep multiplier for the continuity equation of species IS in region IREG

DTMO real array, length NS * (NREG+1), the timestep multiplier for the parallel momentum equation of species IS in region IREG

DTEE real array, length (NREG+1), the timestep multiplier for the electron energy equation in region IREG

DTEI real array, length (NREG+1), the timestep multiplier for the ion energy equation in region IREG

SOLVECO logical array, length NS * (NREG+1), indicates whether the continuity equations are to be solved in each region

SOLVEMO logical array, length NS * (NREG+1), indicates whether the parallel momentum equations are to be solved in each region

SOLVEPO logical array, length (NREG+1), indicates whether the potential equation is to be solved in each region

SOLVEEE logical array, length (NREG+1), indicates whether the electron heat equation is to be solved in each region

SOLVEEI logical array, length (NREG+1), indicates whether the ion heat equation is to be solved in each region

TIME_FACTOR_REQUIRED real, related to local source re-scaling (see routine `b2srsm.F`)

CORR_CORE_DN real array, length NS, damping parameter for speed-up of the density equations in the core when drifts are activated (does not apply to neutral species)

CORR_CORE_DT real, damping parameter for speed-up of the temperature equations in the core when drifts are activated

CORE_DT_SUPPRESSION real, multiplier to the timestep for the innermost core ring of cells

CORE_DT_FACTOR real, multiplier to the timestep for successive core rings of cells

ADD_TE_CORR_TO_PO logical array, length (NREG+1), indicates whether the electron temperature correction is added to the electric potential correction (`.true.` by default)

When referring to array length (NREG+1) above, we mean range [0:NREG] in the code.

The region indices are defined as follows for a SN case:

- 0 Dead regions (unused cells)
- 1 Core
- 2 SOL
- 3 Western Divertor (Inner for LSN, Outer for USN)
- 4 Eastern Divertor (Outer for LSN, Inner for USN)

The extent of those regions in both computational and physical space is shown in Fig. 3.1. For double-null configurations, the regions follow a similar numbering scheme (modulo 4) as shown in Fig. 3.2. Details for other geometries can be found in the `init_region` routine of `b2agfs.F` and in Appendix H.

3.5.5 b2.transport.inputfile

See also Appendix A.5 for a more up-to-date description.

This allows for a radial variation ... for a simpler option see section 3.5.3.

If `'b2tqna.inputfile'` is set to 1 in `b2mn.dat` then the transport coefficients are read in from the namelist in `b2.transport.inputfile` instead of being taken from the file created by `b2ah` and possibly modified in `b2mn.dat`. This then allows one to set radially varying profiles of the transport coefficients, and to achieve time

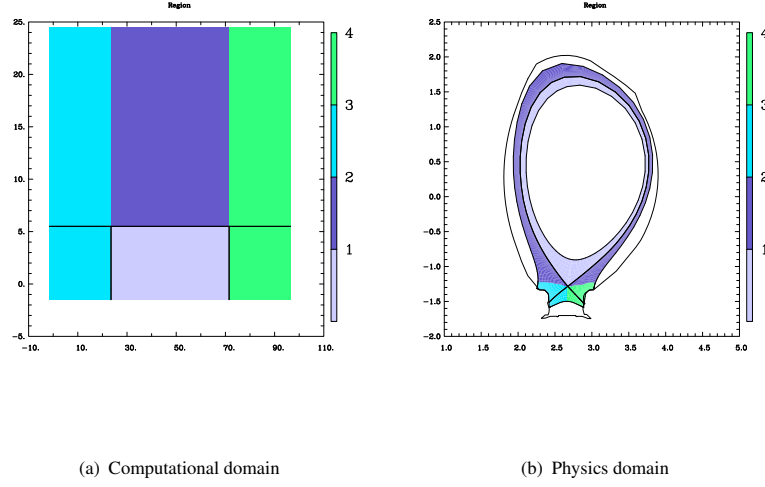


Figure 3.1: Region numbers for single-null geometries.

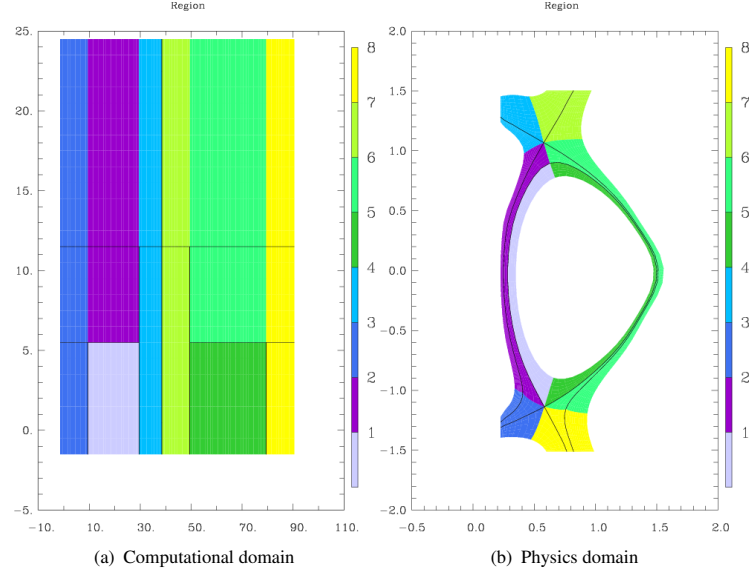


Figure 3.2: Region numbers for double-null geometries.

dependence in these transport coefficients by triggering a read of a new file (as for `b2.transport.parameters`). The transport coefficients are specified by the `transport` namelist as described below.

The input format for these files is somewhat arcane (the relevant subroutine is `transport_input`, to be found in `b2mod_input_profile.F`) and an example, without the modification for reading in a new file triggered by time, which is the same as for the `b2.transport.parameters`, is included below.

The basic idea is to specify pairs of numbers which give the position and value of the relevant transport coefficients.

`tdata` has four indices:

The value of the first index = 1 means position in space, = 2 means the value.

The second is used to index the various positions (6 positions in the example below, specified by `ndata`). The total number of positions should not exceed $ny + 2$, as having transport coefficient profiles finer than the plasma grid resolution is overkill.

The third index of `tdata` relates to the kind of transport coefficient being specified, where:

- kind_coeff.eq.1** $\Rightarrow$ **dna0** particle density-driven diffusivity $D_{\perp}^{na}$
- kind_coeff.eq.2** $\Rightarrow$ **dpa0** particle pressure-driven diffusivity $D_{\perp}^{pa}$
- kind_coeff.eq.3** $\Rightarrow$ **hcib** ion thermal anomalous diffusivity χ_i
- kind_coeff.eq.4** $\Rightarrow$ **hce0** electron thermal anomalous diffusivity χ_e
- kind_coeff.eq.5** $\Rightarrow$ **vla0_x** x -component of the anomalous "pinch" velocity
- kind_coeff.eq.6** $\Rightarrow$ **vla0_y** y -component of the anomalous "pinch" velocity
- kind_coeff.eq.7** $\Rightarrow$ **vsa0** anomalous viscosity
- kind_coeff.eq.8** $\Rightarrow$ **sig0** anomalous radial electrical conductivity
- kind_coeff.eq.9** $\Rightarrow$ **alf0** anomalous radial thermo-electric coefficient

The fourth index is the species index.

Again, as in Section 3.2.1, it is important, for numerical stability reasons, to provide mechanisms by which some particle transport and radial current can be established. If the transport coefficients are given here, this means that, first, the `dna0` and `dpa0` arrays cannot be simultaneously zero when solving density equations. Second, the arrays `sig0` and `alf0` cannot be simultaneously zero when solving the potential equations without drift current terms on. The code will fail a transport coefficients test in `b2tral` otherwise.

The relevant routine was written by

```
! *****
!*      subroutine for reading in experimental transport coefficients from *
!*      the midplane and calculating transport coefficients                *
!*      for B2.5 (written by Heimo Buerbaumer 1999)                      *
!*      Inputfile:  b2.transport.inputfile                                *
! *****
```

Some additional description of the relevant variables is as follows:

ndata(1,i,j) where i is an index for the type of transport coefficient being defined (see table above) and j is a species index (0 for atoms and 1 for ions in a D -only case, for instance, and irrelevant (use 1) for electron coefficients). `ndata` is the number of data points where the value of the transport coefficients is given.

addspec(k,i,j) are the indices of any additional species whose transport coefficients are also described by the same set of numbers. k is a dummy that runs from 1 to the number of additional species described.

tdata(1,m,i,j) are the radial positions (measured in metres) of the `ndata(1,i,j)` points, where m runs from 1 to `ndata(1,i,j)`. By default, the radial positions are measured from the separatrix outward, along the outer midplane, but this can be changed by making use of the `ixref` and `iyref` switches, respectively, to modify the poloidal location of the profile and the radial position from which the distances are measured.

tdata(2,m,i,j) are the corresponding values of the coefficients.

no_pflux is a logical that, if `.true.`, does not apply these transport coefficients to the private flux region(s).

no_div is a logical that, if `.true.`, does not apply these transport coefficients to the divertor regions.

So in the example file below, three transport coefficients profiles are given, i.e. in order, $D_{\perp}^{na}$, χ_i and χ_e , defined with 6 points each, and setting a transport barrier around the separatrix. Note that the radial locations of those points are quite independent of where the mesh is. The code interpolates the transport values onto the mesh, so one needs not

worry about making sure that a point is at an exact cell location. It is recommended that the first and last points be a large distance away from the separatrix, typically well outside the computational domain. This ensures that the code is still interpolating and not using (possibly different) default values when computing coefficients for cells near the edges of the mesh. The interpolation the code uses is a monotonicity-preserving cubic Hermite polynomial spline from the NAG library, or a simple linear interpolation if the NAG library is unavailable at compilation time. For computational cells located outside the range provided by the `tdata(1,m,i,j)` positions, the code will use whatever 'default' values were declared previously. As a reminder, the order of precedence of the transport coefficients settings is:

```
b2ah.dat < b2mn.dat < b2.transport.parameters < b2.transport.inputfile
```

The following contains an example where the flag `no_pflux` is used to switch off the setting of transport coefficients in the private flux region.

```
&TRANSPORT
ndata(1, 1, 1) = 6,
tdata(1, 1, 1, 1) = -4.00E-02, tdata(2, 1, 1, 1) = 0.50,
tdata(1, 2, 1, 1) = -1.50E-02, tdata(2, 2, 1, 1) = 0.50,
tdata(1, 3, 1, 1) = -1.40E-02, tdata(2, 3, 1, 1) = 0.05,
tdata(1, 4, 1, 1) = 5.00E-03, tdata(2, 4, 1, 1) = 0.05,
tdata(1, 5, 1, 1) = 5.50E-03, tdata(2, 5, 1, 1) = 0.50,
tdata(1, 6, 1, 1) = 2.50E-02, tdata(2, 6, 1, 1) = 0.50,
ndata(1, 3, 1) = 6,
tdata(1, 1, 3, 1) = -4.00E-02, tdata(2, 1, 3, 1) = 0.70,
tdata(1, 2, 3, 1) = -1.50E-02, tdata(2, 2, 3, 1) = 0.70,
tdata(1, 3, 3, 1) = -1.40E-02, tdata(2, 3, 3, 1) = 0.07,
tdata(1, 4, 3, 1) = 5.00E-03, tdata(2, 4, 3, 1) = 0.07,
tdata(1, 5, 3, 1) = 5.50E-03, tdata(2, 5, 3, 1) = 0.70,
tdata(1, 6, 3, 1) = 2.50E-02, tdata(2, 6, 3, 1) = 0.70,
ndata(1, 4, 1) = 6,
tdata(1, 1, 4, 1) = -4.00E-02, tdata(2, 1, 4, 1) = 0.70,
tdata(1, 2, 4, 1) = -1.50E-02, tdata(2, 2, 4, 1) = 0.70,
tdata(1, 3, 4, 1) = -1.40E-02, tdata(2, 3, 4, 1) = 0.07,
tdata(1, 4, 4, 1) = 5.00E-03, tdata(2, 4, 4, 1) = 0.07,
tdata(1, 5, 4, 1) = 5.50E-03, tdata(2, 5, 4, 1) = 0.70,
tdata(1, 6, 4, 1) = 2.50E-02, tdata(2, 6, 4, 1) = 0.70,
no_pflux=.true.
/
```

3.5.6 b2.sources.profile

See also Appendix A.5 for a more up-to-date description.

One can also set radially dependent source profiles, in much the same way as with `b2.transport.inputfile`, with the `b2.sources.profile` input file, accessed by setting 'b2sral.inputfile' to '1'. The logic is the same in terms of fixing radial positions, applicable species, etc...

The coefficient types go as:

1. Particle sources *sna*
2. Parallel momentum sources *smo*
3. Electron heat sources *she*
4. Ion heat sources *shi*
5. Electric charge sources *sch*
6. Electric particle source (non-ambipolar) *sne*

It is also possible to multiply these radial profiles by an axial form factor. This axial multiplier is set in the same way as the poloidal profile.

The description of the variables found in `b2.sources.profile` is as follows:

nsdata(1,i,j) where i is an index for the type of source being defined (see table above) and j is a species index (0 for atoms and 1 for ions in a D -only case, for instance, and irrelevant (use 1) for electron sources). **nsdata** is the number of data points where the strength of the source is given.

sdata(1,m,i,j) are the radial positions (measured in metres) of the **nsdata(1,i,j)** points, where m runs from 1 to **nsdata(1,i,j)**. By default, the radial positions are measured from the separatrix outward, along the outer midplane, but this can be changed by making use of the *ixref* and *iyref* switches, respectively, to modify the poloidal location of the profile and the radial position from which the distances are measured.

sdata(2,m,i,j) are the corresponding strengths of the sources.

nxdata(k,i,j) where i is an index for the type of source being defined (see table above) and j is a species index (0 for atoms and 1 for ions in a D -only case, for instance, and irrelevant (use 1) for electron sources). **nxdata** is the number of data points where the value of the axial multiplier to the source is given. If k is equal to 1, the data is to be understood as function of normalized connection length. If k is equal to 2, then the axial profile is given in terms of the normalized computational coordinate (running from 0 to 1 as ix runs from -1 to nx).

xdata(1,m,i,j) are the axial positions (normalised to the total length) of the **nxdata(1,i,j)** points, where m runs from 1 to **nxdata(1,i,j)**. By default, the connection length used is the one measured just inside the separatrix, with the origin located at the outer midplane, but this can be changed by making use of the *ixref* and *iyref* switches, respectively, to modify the poloidal location of the origin and the radial position along which the distances are measured. In case the reference location is chosen inside a closed field line region, the axial multiplier is set to zero everywhere on the open field lines.

xdata(2,m,i,j) are the corresponding values of the source strengths.

divheat is the total volumetric ion heat source (in W/m^3) spread equally throughout the divertor volume.

The results of these imported source profiles can be visualised in **b2plot** with the **exna**, **exmo**, **exhe**, **exhi**, **exch**, and **exne** commands. An example of input file is given below:

```
&profile
nsdata(1,1,0)=42,
sdata(1,1,1,0)=-0.4165, sdata(2,1,1,0)=0.9895E+20,
sdata(1,2,1,0)=-0.4029, sdata(2,2,1,0)=0.9582E+20,
sdata(1,3,1,0)=-0.3900, sdata(2,3,1,0)=0.9000E+20,
sdata(1,4,1,0)=-0.3775, sdata(2,4,1,0)=0.8314E+20,
sdata(1,5,1,0)=-0.3652, sdata(2,5,1,0)=0.7621E+20,
sdata(1,6,1,0)=-0.3532, sdata(2,6,1,0)=0.6930E+20,
sdata(1,7,1,0)=-0.3416, sdata(2,7,1,0)=0.6253E+20,
sdata(1,8,1,0)=-0.3300, sdata(2,8,1,0)=0.5615E+20,
sdata(1,9,1,0)=-0.3185, sdata(2,9,1,0)=0.5057E+20,
sdata(1,10,1,0)=-0.3071, sdata(2,10,1,0)=0.4570E+20,
sdata(1,11,1,0)=-0.2955, sdata(2,11,1,0)=0.4189E+20,
sdata(1,12,1,0)=-0.2835, sdata(2,12,1,0)=0.3876E+20,
sdata(1,13,1,0)=-0.2715, sdata(2,13,1,0)=0.3642E+20,
sdata(1,14,1,0)=-0.2594, sdata(2,14,1,0)=0.3455E+20,
sdata(1,15,1,0)=-0.2473, sdata(2,15,1,0)=0.3316E+20,
sdata(1,16,1,0)=-0.2352, sdata(2,16,1,0)=0.3205E+20,
sdata(1,17,1,0)=-0.2232, sdata(2,17,1,0)=0.3122E+20,
sdata(1,18,1,0)=-0.2112, sdata(2,18,1,0)=0.3054E+20,
sdata(1,19,1,0)=-0.1996, sdata(2,19,1,0)=0.3002E+20,
sdata(1,20,1,0)=-0.1881, sdata(2,20,1,0)=0.2960E+20,
sdata(1,21,1,0)=-0.1768, sdata(2,21,1,0)=0.2925E+20,
sdata(1,22,1,0)=-0.1658, sdata(2,22,1,0)=0.2895E+20,
sdata(1,23,1,0)=-0.1550, sdata(2,23,1,0)=0.2867E+20,
sdata(1,24,1,0)=-0.1445, sdata(2,24,1,0)=0.2838E+20,
sdata(1,25,1,0)=-0.1342, sdata(2,25,1,0)=0.2805E+20,
sdata(1,26,1,0)=-0.1243, sdata(2,26,1,0)=0.2769E+20,
sdata(1,27,1,0)=-0.1145, sdata(2,27,1,0)=0.2728E+20,
sdata(1,28,1,0)=-0.1050, sdata(2,28,1,0)=0.2683E+20,
sdata(1,29,1,0)=-0.0958, sdata(2,29,1,0)=0.2632E+20,
sdata(1,30,1,0)=-0.0868, sdata(2,30,1,0)=0.2576E+20,
sdata(1,31,1,0)=-0.0780, sdata(2,31,1,0)=0.2511E+20,
```



```

sdata(1,32,1,0)=-0.0694, sdata(2,32,1,0)=0.2438E+20,
sdata(1,33,1,0)=-0.0612, sdata(2,33,1,0)=0.2354E+20,
sdata(1,34,1,0)=-0.0530, sdata(2,34,1,0)=0.2258E+20,
sdata(1,35,1,0)=-0.0451, sdata(2,35,1,0)=0.2144E+20,
sdata(1,36,1,0)=-0.0374, sdata(2,36,1,0)=0.2007E+20,
sdata(1,37,1,0)=-0.0298, sdata(2,37,1,0)=0.1831E+20,
sdata(1,38,1,0)=-0.0224, sdata(2,38,1,0)=0.1605E+20,
sdata(1,39,1,0)=-0.0153, sdata(2,39,1,0)=0.1302E+20,
sdata(1,40,1,0)=-0.0082, sdata(2,40,1,0)=0.9238E+19,
sdata(1,41,1,0)=-0.0014, sdata(2,41,1,0)=0.4474E+19,
sdata(1,42,1,0)= 0.0035, sdata(2,42,1,0)=0.0000E+00,
nsdata(1,1,1)=42,
sdata(1,1,1,1)=-0.4165, sdata(2,1,1,1)=0.7723E+20,
sdata(1,2,1,1)=-0.4029, sdata(2,2,1,1)=0.7472E+20,
sdata(1,3,1,1)=-0.3900, sdata(2,3,1,1)=0.7003E+20,
sdata(1,4,1,1)=-0.3775, sdata(2,4,1,1)=0.6447E+20,
sdata(1,5,1,1)=-0.3652, sdata(2,5,1,1)=0.5872E+20,
sdata(1,6,1,1)=-0.3532, sdata(2,6,1,1)=0.5292E+20,
sdata(1,7,1,1)=-0.3416, sdata(2,7,1,1)=0.4712E+20,
sdata(1,8,1,1)=-0.3300, sdata(2,8,1,1)=0.4158E+20,
sdata(1,9,1,1)=-0.3185, sdata(2,9,1,1)=0.3667E+20,
sdata(1,10,1,1)=-0.3071, sdata(2,10,1,1)=0.3234E+20,
sdata(1,11,1,1)=-0.2955, sdata(2,11,1,1)=0.2894E+20,
sdata(1,12,1,1)=-0.2835, sdata(2,12,1,1)=0.2612E+20,
sdata(1,13,1,1)=-0.2715, sdata(2,13,1,1)=0.2400E+20,
sdata(1,14,1,1)=-0.2594, sdata(2,14,1,1)=0.2227E+20,
sdata(1,15,1,1)=-0.2473, sdata(2,15,1,1)=0.2094E+20,
sdata(1,16,1,1)=-0.2352, sdata(2,16,1,1)=0.1983E+20,
sdata(1,17,1,1)=-0.2232, sdata(2,17,1,1)=0.1898E+20,
sdata(1,18,1,1)=-0.2112, sdata(2,18,1,1)=0.1828E+20,
sdata(1,19,1,1)=-0.1996, sdata(2,19,1,1)=0.1772E+20,
sdata(1,20,1,1)=-0.1881, sdata(2,20,1,1)=0.1725E+20,
sdata(1,21,1,1)=-0.1768, sdata(2,21,1,1)=0.1684E+20,
sdata(1,22,1,1)=-0.1658, sdata(2,22,1,1)=0.1646E+20,
sdata(1,23,1,1)=-0.1550, sdata(2,23,1,1)=0.1609E+20,
sdata(1,24,1,1)=-0.1445, sdata(2,24,1,1)=0.1574E+20,
sdata(1,25,1,1)=-0.1342, sdata(2,25,1,1)=0.1536E+20,
sdata(1,26,1,1)=-0.1243, sdata(2,26,1,1)=0.1499E+20,
sdata(1,27,1,1)=-0.1145, sdata(2,27,1,1)=0.1460E+20,
sdata(1,28,1,1)=-0.1050, sdata(2,28,1,1)=0.1421E+20,
sdata(1,29,1,1)=-0.0958, sdata(2,29,1,1)=0.1381E+20,
sdata(1,30,1,1)=-0.0868, sdata(2,30,1,1)=0.1339E+20,
sdata(1,31,1,1)=-0.0780, sdata(2,31,1,1)=0.1294E+20,
sdata(1,32,1,1)=-0.0694, sdata(2,32,1,1)=0.1245E+20,
sdata(1,33,1,1)=-0.0612, sdata(2,33,1,1)=0.1191E+20,
sdata(1,34,1,1)=-0.0530, sdata(2,34,1,1)=0.1133E+20,
sdata(1,35,1,1)=-0.0451, sdata(2,35,1,1)=0.1071E+20,
sdata(1,36,1,1)=-0.0374, sdata(2,36,1,1)=0.9985E+19,
sdata(1,37,1,1)=-0.0298, sdata(2,37,1,1)=0.9087E+19,
sdata(1,38,1,1)=-0.0224, sdata(2,38,1,1)=0.7958E+19,
sdata(1,39,1,1)=-0.0153, sdata(2,39,1,1)=0.6455E+19,
sdata(1,40,1,1)=-0.0082, sdata(2,40,1,1)=0.4582E+19,
sdata(1,41,1,1)=-0.0014, sdata(2,41,1,1)=0.2220E+19,
sdata(1,42,1,1)= 0.0035, sdata(2,42,1,1)=0.0000E+00,
nsdata(1,2,0)=42,
sdata(1,1,2,0)=-0.4165, sdata(2,1,2,0)=0.7080E+00,
sdata(1,2,2,0)=-0.4029, sdata(2,2,2,0)=0.6405E+00,
sdata(1,3,2,0)=-0.3900, sdata(2,3,2,0)=0.5226E+00,
sdata(1,4,2,0)=-0.3775, sdata(2,4,2,0)=0.4044E+00,
sdata(1,5,2,0)=-0.3652, sdata(2,5,2,0)=0.3377E+00,
sdata(1,6,2,0)=-0.3532, sdata(2,6,2,0)=0.2951E+00,
sdata(1,7,2,0)=-0.3416, sdata(2,7,2,0)=0.2676E+00,
sdata(1,8,2,0)=-0.3300, sdata(2,8,2,0)=0.2466E+00,
sdata(1,9,2,0)=-0.3185, sdata(2,9,2,0)=0.2285E+00,
sdata(1,10,2,0)=-0.3071, sdata(2,10,2,0)=0.2115E+00,
sdata(1,11,2,0)=-0.2955, sdata(2,11,2,0)=0.1942E+00,
sdata(1,12,2,0)=-0.2835, sdata(2,12,2,0)=0.1772E+00,

```

```

sdata(1,13,2,0)=-0.2715, sdata(2,13,2,0)=0.1614E+00,
sdata(1,14,2,0)=-0.2594, sdata(2,14,2,0)=0.1464E+00,
sdata(1,15,2,0)=-0.2473, sdata(2,15,2,0)=0.1327E+00,
sdata(1,16,2,0)=-0.2352, sdata(2,16,2,0)=0.1205E+00,
sdata(1,17,2,0)=-0.2232, sdata(2,17,2,0)=0.1110E+00,
sdata(1,18,2,0)=-0.2112, sdata(2,18,2,0)=0.1023E+00,
sdata(1,19,2,0)=-0.1996, sdata(2,19,2,0)=0.9300E-01,
sdata(1,20,2,0)=-0.1881, sdata(2,20,2,0)=0.8348E-01,
sdata(1,21,2,0)=-0.1768, sdata(2,21,2,0)=0.7411E-01,
sdata(1,22,2,0)=-0.1658, sdata(2,22,2,0)=0.6554E-01,
sdata(1,23,2,0)=-0.1550, sdata(2,23,2,0)=0.5882E-01,
sdata(1,24,2,0)=-0.1445, sdata(2,24,2,0)=0.5295E-01,
sdata(1,25,2,0)=-0.1342, sdata(2,25,2,0)=0.4763E-01,
sdata(1,26,2,0)=-0.1243, sdata(2,26,2,0)=0.4266E-01,
sdata(1,27,2,0)=-0.1145, sdata(2,27,2,0)=0.3789E-01,
sdata(1,28,2,0)=-0.1050, sdata(2,28,2,0)=0.3327E-01,
sdata(1,29,2,0)=-0.0958, sdata(2,29,2,0)=0.2855E-01,
sdata(1,30,2,0)=-0.0868, sdata(2,30,2,0)=0.2438E-01,
sdata(1,31,2,0)=-0.0780, sdata(2,31,2,0)=0.2137E-01,
sdata(1,32,2,0)=-0.0694, sdata(2,32,2,0)=0.1900E-01,
sdata(1,33,2,0)=-0.0612, sdata(2,33,2,0)=0.1699E-01,
sdata(1,34,2,0)=-0.0530, sdata(2,34,2,0)=0.1498E-01,
sdata(1,35,2,0)=-0.0451, sdata(2,35,2,0)=0.1257E-01,
sdata(1,36,2,0)=-0.0374, sdata(2,36,2,0)=0.1022E-01,
sdata(1,37,2,0)=-0.0298, sdata(2,37,2,0)=0.8162E-02,
sdata(1,38,2,0)=-0.0224, sdata(2,38,2,0)=0.6258E-02,
sdata(1,39,2,0)=-0.0153, sdata(2,39,2,0)=0.4355E-02,
sdata(1,40,2,0)=-0.0082, sdata(2,40,2,0)=0.2603E-02,
sdata(1,41,2,0)=-0.0014, sdata(2,41,2,0)=0.1151E-02,
sdata(1,42,2,0)= 0.0035, sdata(2,42,2,0)=0.0000E+00,
nsdata(1,2,1)=42,
sdata(1,1,2,1)=-0.4165, sdata(2,1,2,1)=0.7080E+00,
sdata(1,2,2,1)=-0.4029, sdata(2,2,2,1)=0.6405E+00,
sdata(1,3,2,1)=-0.3900, sdata(2,3,2,1)=0.5226E+00,
sdata(1,4,2,1)=-0.3775, sdata(2,4,2,1)=0.4044E+00,
sdata(1,5,2,1)=-0.3652, sdata(2,5,2,1)=0.3377E+00,
sdata(1,6,2,1)=-0.3532, sdata(2,6,2,1)=0.2951E+00,
sdata(1,7,2,1)=-0.3416, sdata(2,7,2,1)=0.2676E+00,
sdata(1,8,2,1)=-0.3300, sdata(2,8,2,1)=0.2466E+00,
sdata(1,9,2,1)=-0.3185, sdata(2,9,2,1)=0.2285E+00,
sdata(1,10,2,1)=-0.3071, sdata(2,10,2,1)=0.2115E+00,
sdata(1,11,2,1)=-0.2955, sdata(2,11,2,1)=0.1942E+00,
sdata(1,12,2,1)=-0.2835, sdata(2,12,2,1)=0.1772E+00,
sdata(1,13,2,1)=-0.2715, sdata(2,13,2,1)=0.1614E+00,
sdata(1,14,2,1)=-0.2594, sdata(2,14,2,1)=0.1464E+00,
sdata(1,15,2,1)=-0.2473, sdata(2,15,2,1)=0.1327E+00,
sdata(1,16,2,1)=-0.2352, sdata(2,16,2,1)=0.1205E+00,
sdata(1,17,2,1)=-0.2232, sdata(2,17,2,1)=0.1110E+00,
sdata(1,18,2,1)=-0.2112, sdata(2,18,2,1)=0.1023E+00,
sdata(1,19,2,1)=-0.1996, sdata(2,19,2,1)=0.9300E-01,
sdata(1,20,2,1)=-0.1881, sdata(2,20,2,1)=0.8348E-01,
sdata(1,21,2,1)=-0.1768, sdata(2,21,2,1)=0.7411E-01,
sdata(1,22,2,1)=-0.1658, sdata(2,22,2,1)=0.6554E-01,
sdata(1,23,2,1)=-0.1550, sdata(2,23,2,1)=0.5882E-01,
sdata(1,24,2,1)=-0.1445, sdata(2,24,2,1)=0.5295E-01,
sdata(1,25,2,1)=-0.1342, sdata(2,25,2,1)=0.4763E-01,
sdata(1,26,2,1)=-0.1243, sdata(2,26,2,1)=0.4266E-01,
sdata(1,27,2,1)=-0.1145, sdata(2,27,2,1)=0.3789E-01,
sdata(1,28,2,1)=-0.1050, sdata(2,28,2,1)=0.3327E-01,
sdata(1,29,2,1)=-0.0958, sdata(2,29,2,1)=0.2855E-01,
sdata(1,30,2,1)=-0.0868, sdata(2,30,2,1)=0.2438E-01,
sdata(1,31,2,1)=-0.0780, sdata(2,31,2,1)=0.2137E-01,
sdata(1,32,2,1)=-0.0694, sdata(2,32,2,1)=0.1900E-01,
sdata(1,33,2,1)=-0.0612, sdata(2,33,2,1)=0.1699E-01,
sdata(1,34,2,1)=-0.0530, sdata(2,34,2,1)=0.1498E-01,
sdata(1,35,2,1)=-0.0451, sdata(2,35,2,1)=0.1257E-01,
sdata(1,36,2,1)=-0.0374, sdata(2,36,2,1)=0.1022E-01,

```

```

sdata(1,37,2,1)=-0.0298, sdata(2,37,2,1)=0.8162E-02,
sdata(1,38,2,1)=-0.0224, sdata(2,38,2,1)=0.6258E-02,
sdata(1,39,2,1)=-0.0153, sdata(2,39,2,1)=0.4355E-02,
sdata(1,40,2,1)=-0.0082, sdata(2,40,2,1)=0.2603E-02,
sdata(1,41,2,1)=-0.0014, sdata(2,41,2,1)=0.1151E-02,
sdata(1,42,2,1)= 0.0035, sdata(2,42,2,1)=0.0000E+00,
nsdata(1,3,1)=42,
sdata(1,1,3,1)=-0.4165, sdata(2,1,3,1)=0.1570E+07,
sdata(1,2,3,1)=-0.4029, sdata(2,2,3,1)=0.1418E+07,
sdata(1,3,3,1)=-0.3900, sdata(2,3,3,1)=0.1152E+07,
sdata(1,4,3,1)=-0.3775, sdata(2,4,3,1)=0.8862E+06,
sdata(1,5,3,1)=-0.3652, sdata(2,5,3,1)=0.7358E+06,
sdata(1,6,3,1)=-0.3532, sdata(2,6,3,1)=0.6396E+06,
sdata(1,7,3,1)=-0.3416, sdata(2,7,3,1)=0.5760E+06,
sdata(1,8,3,1)=-0.3300, sdata(2,8,3,1)=0.5254E+06,
sdata(1,9,3,1)=-0.3185, sdata(2,9,3,1)=0.4774E+06,
sdata(1,10,3,1)=-0.3071, sdata(2,10,3,1)=0.4298E+06,
sdata(1,11,3,1)=-0.2955, sdata(2,11,3,1)=0.3804E+06,
sdata(1,12,3,1)=-0.2835, sdata(2,12,3,1)=0.3331E+06,
sdata(1,13,3,1)=-0.2715, sdata(2,13,3,1)=0.2943E+06,
sdata(1,14,3,1)=-0.2594, sdata(2,14,3,1)=0.2616E+06,
sdata(1,15,3,1)=-0.2473, sdata(2,15,3,1)=0.2369E+06,
sdata(1,16,3,1)=-0.2352, sdata(2,16,3,1)=0.2167E+06,
sdata(1,17,3,1)=-0.2232, sdata(2,17,3,1)=0.2008E+06,
sdata(1,18,3,1)=-0.2112, sdata(2,18,3,1)=0.1871E+06,
sdata(1,19,3,1)=-0.1996, sdata(2,19,3,1)=0.1757E+06,
sdata(1,20,3,1)=-0.1881, sdata(2,20,3,1)=0.1651E+06,
sdata(1,21,3,1)=-0.1768, sdata(2,21,3,1)=0.1547E+06,
sdata(1,22,3,1)=-0.1658, sdata(2,22,3,1)=0.1450E+06,
sdata(1,23,3,1)=-0.1550, sdata(2,23,3,1)=0.1362E+06,
sdata(1,24,3,1)=-0.1445, sdata(2,24,3,1)=0.1278E+06,
sdata(1,25,3,1)=-0.1342, sdata(2,25,3,1)=0.1192E+06,
sdata(1,26,3,1)=-0.1243, sdata(2,26,3,1)=0.1107E+06,
sdata(1,27,3,1)=-0.1145, sdata(2,27,3,1)=0.1022E+06,
sdata(1,28,3,1)=-0.1050, sdata(2,28,3,1)=0.9360E+05,
sdata(1,29,3,1)=-0.0958, sdata(2,29,3,1)=0.8460E+05,
sdata(1,30,3,1)=-0.0868, sdata(2,30,3,1)=0.7630E+05,
sdata(1,31,3,1)=-0.0780, sdata(2,31,3,1)=0.6990E+05,
sdata(1,32,3,1)=-0.0694, sdata(2,32,3,1)=0.6460E+05,
sdata(1,33,3,1)=-0.0612, sdata(2,33,3,1)=0.6010E+05,
sdata(1,34,3,1)=-0.0530, sdata(2,34,3,1)=0.5490E+05,
sdata(1,35,3,1)=-0.0451, sdata(2,35,3,1)=0.4780E+05,
sdata(1,36,3,1)=-0.0374, sdata(2,36,3,1)=0.4030E+05,
sdata(1,37,3,1)=-0.0298, sdata(2,37,3,1)=0.3350E+05,
sdata(1,38,3,1)=-0.0224, sdata(2,38,3,1)=0.2690E+05,
sdata(1,39,3,1)=-0.0153, sdata(2,39,3,1)=0.1950E+05,
sdata(1,40,3,1)=-0.0082, sdata(2,40,3,1)=0.1210E+05,
sdata(1,41,3,1)=-0.0014, sdata(2,41,3,1)=0.5500E+04,
sdata(1,42,3,1)= 0.0035, sdata(2,42,3,1)=0.0000E+00,
nsdata(1,4,1)=42,
sdata(1,1,4,1)=-0.4165, sdata(2,1,4,1)=0.1300E+07,
sdata(1,2,4,1)=-0.4029, sdata(2,2,4,1)=0.1179E+07,
sdata(1,3,4,1)=-0.3900, sdata(2,3,4,1)=0.9645E+06,
sdata(1,4,4,1)=-0.3775, sdata(2,4,4,1)=0.7452E+06,
sdata(1,5,4,1)=-0.3652, sdata(2,5,4,1)=0.6223E+06,
sdata(1,6,4,1)=-0.3532, sdata(2,6,4,1)=0.5476E+06,
sdata(1,7,4,1)=-0.3416, sdata(2,7,4,1)=0.5071E+06,
sdata(1,8,4,1)=-0.3300, sdata(2,8,4,1)=0.4841E+06,
sdata(1,9,4,1)=-0.3185, sdata(2,9,4,1)=0.4703E+06,
sdata(1,10,4,1)=-0.3071, sdata(2,10,4,1)=0.4604E+06,
sdata(1,11,4,1)=-0.2955, sdata(2,11,4,1)=0.4513E+06,
sdata(1,12,4,1)=-0.2835, sdata(2,12,4,1)=0.4421E+06,
sdata(1,13,4,1)=-0.2715, sdata(2,13,4,1)=0.4286E+06,
sdata(1,14,4,1)=-0.2594, sdata(2,14,4,1)=0.4104E+06,
sdata(1,15,4,1)=-0.2473, sdata(2,15,4,1)=0.3869E+06,
sdata(1,16,4,1)=-0.2352, sdata(2,16,4,1)=0.3636E+06,
sdata(1,17,4,1)=-0.2232, sdata(2,17,4,1)=0.3458E+06,

```

```

sdata(1,18,4,1)=-0.2112, sdata(2,18,4,1)=0.3279E+06,
sdata(1,19,4,1)=-0.1996, sdata(2,19,4,1)=0.3047E+06,
sdata(1,20,4,1)=-0.1881, sdata(2,20,4,1)=0.2791E+06,
sdata(1,21,4,1)=-0.1768, sdata(2,21,4,1)=0.2524E+06,
sdata(1,22,4,1)=-0.1658, sdata(2,22,4,1)=0.2272E+06,
sdata(1,23,4,1)=-0.1550, sdata(2,23,4,1)=0.2064E+06,
sdata(1,24,4,1)=-0.1445, sdata(2,24,4,1)=0.1877E+06,
sdata(1,25,4,1)=-0.1342, sdata(2,25,4,1)=0.1699E+06,
sdata(1,26,4,1)=-0.1243, sdata(2,26,4,1)=0.1526E+06,
sdata(1,27,4,1)=-0.1145, sdata(2,27,4,1)=0.1358E+06,
sdata(1,28,4,1)=-0.1050, sdata(2,28,4,1)=0.1195E+06,
sdata(1,29,4,1)=-0.0958, sdata(2,29,4,1)=0.1037E+06,
sdata(1,30,4,1)=-0.0868, sdata(2,30,4,1)=0.8930E+05,
sdata(1,31,4,1)=-0.0780, sdata(2,31,4,1)=0.7750E+05,
sdata(1,32,4,1)=-0.0694, sdata(2,32,4,1)=0.6740E+05,
sdata(1,33,4,1)=-0.0612, sdata(2,33,4,1)=0.5880E+05,
sdata(1,34,4,1)=-0.0530, sdata(2,34,4,1)=0.5050E+05,
sdata(1,35,4,1)=-0.0451, sdata(2,35,4,1)=0.4170E+05,
sdata(1,36,4,1)=-0.0374, sdata(2,36,4,1)=0.3320E+05,
sdata(1,37,4,1)=-0.0298, sdata(2,37,4,1)=0.2590E+05,
sdata(1,38,4,1)=-0.0224, sdata(2,38,4,1)=0.1930E+05,
sdata(1,39,4,1)=-0.0153, sdata(2,39,4,1)=0.1300E+05,
sdata(1,40,4,1)=-0.0082, sdata(2,40,4,1)=0.7400E+04,
sdata(1,41,4,1)=-0.0014, sdata(2,41,4,1)=0.3200E+04,
sdata(1,42,4,1)= 0.0035, sdata(2,42,4,1)=0.0000E+00,
/

```

3.5.7 b2.neutrals.parameters

See descriptions in Coupled Eirene chapter 5 and Appendix A.5.

3.5.8 b2.atomic_physics_rescale.parameters

Contains a series of multipliers to the atomic physics rates provided by b2ar in b2frates. These are:

RESCALE_SA real*8 array, length NS, contains the scaling factor per species to be applied to the ionisation rates

RESCALE_RA real*8 array, length NS, contains the scaling factor per species to be applied to the recombination rates

RESCALE_QA real*8 array, length NS, contains the scaling factor per species to be applied to the electron cooling rates

RESCALE_CX real*8 array, length NS, contains the scaling factor per species to be applied to the charge exchange rates

RESCALE_RD real*8 array, length NS, contains the scaling factor per species to be applied to the line radiation rates

RESCALE_BR real*8 array, length NS, contains the scaling factor per species to be applied to the bremsstrahlung radiation rates

See Appendix A.5 for a full description.

3.5.9 b2.user.parameters

See descriptions in Coupled Eirene chapter 5 and Appendix A.5.

3.5.10 b2.XXX.parameters

See descriptions in Appendix A.5.

3.5.11 While the code is running

While the code is running and writing into `run.log`, a number of the scripts described in section 2.5 can be run.

`cpu` shows how much CPU time each iteration has taken and is a quick way of monitoring the progress of a run.

```
18:22> cpu
# /scratch/dpc/solps5.0_test_xpb/src/Braams/b2/runs/JET_B2_comparison/50401/57_coarse/2e19_8MW
1 2.129999935626984
2 1.940000295639038
3 1.730000019073486
4 1.719999313354492
5 1.710000038146973
6 1.720000267028809
7 1.690000534057617
8 1.710000038146973
9 1.679999351501465
10 1.700000762939453
11 1.699998855590820
12 1.710000991821289
13 1.699998855590820
18:22>
```

All of the residuals of a pure deuterium run can be plotted with `resall.D`, figure 3.3.

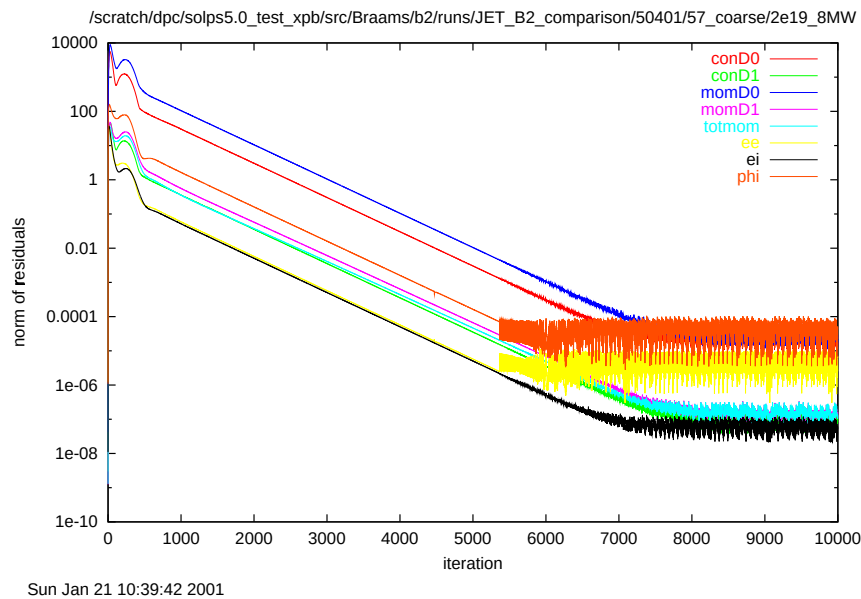
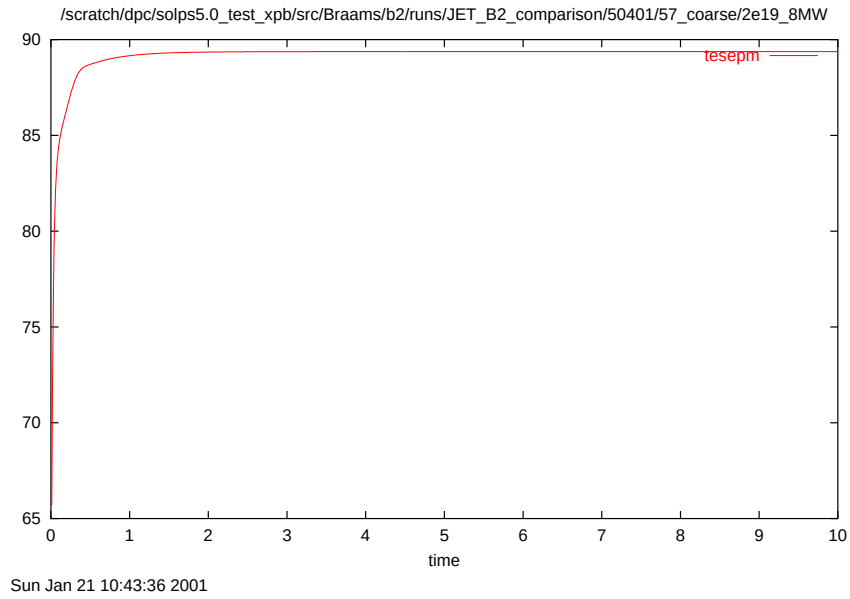
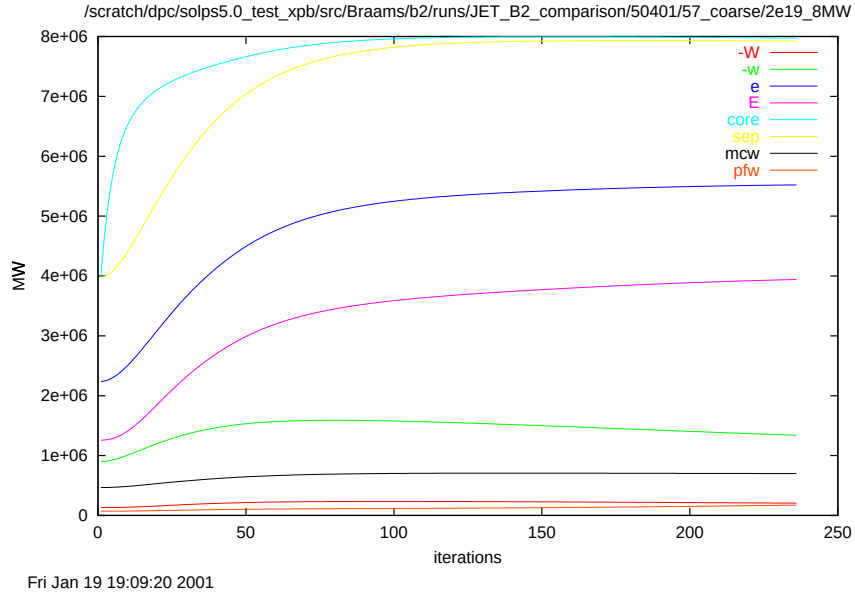


Figure 3.3: Residuals.

The power crossing various surfaces can be seen with `energy_analysis` or `energy_analysis.py`, figure 3.4. The detailed definition of these surfaces can be found in Appendix H or the commented header to the `init_region` subroutine in `b2agfs.F`.

`2dt_teseqm` can be used to plot the separatrix temperature as a function of time, figure 3.5. See table E.1 (page 392) for a list of quantities that can be plotted in this way.



3.6 b2pl, the plotting program

The following commands apply:

pit , pitch	Pitch
bx , bpol	b_{x} (T) (poloidal)
by , brad	b_{y} (T)
bz , btor	b_{z} (T) (toroidal)
bb	bb (T)
b	B (T) (components)

ffbz	Toroidal magnetic flux function (Wb/m)
psi	Poloidal magnetic flux function (Wb)
na	Ion density (m^{-3})
nh	Hydrogenic atom density (m^{-3})
ni	Total density (m^{-3})
ua	Parallel velocity (u_a) ($m.s^{-1}$)
ue	Parallel electron velocity (u_e) ($m.s^{-1}$)
vv	Radial velocity (vv) ($m.s^{-1}$)
ww	Toroidal velocity (ww) ($m.s^{-1}$)
te	Electron temperature (eV)
ti	Ion temperature (eV)
pr	Static pressure (Pa)
pz	Plasma pressure (Pa)
ne	Electron density (m^{-3})
ne2	ne2 (m^{-3})
po	Electric potential (V)
dfhe	$\frac{d}{dt}dfhe$ (W)
dfhi	$\frac{d}{dt}dfhi$ (W)
dhet, dfhet	$\frac{d}{dt}dfhet$ (W)
dhit, dfhit	$\frac{d}{dt}dfhit$ (W)
dfhj	$\frac{d}{dt}dfhj$ (W)
dfch	$\frac{d}{dt}dfch$ (A)
dfna	$\frac{d}{dt}dfna$ (#/s)
dfmo	$\frac{d}{dt}dfmo$ (N.s)
dfhm	$\frac{d}{dt}dfhm$ (W)
dfhp	$\frac{d}{dt}dfhp$ (W)
zeff	Z_{eff}
up	Poloidal velocity (up) ($m.s^{-1}$)
fnax	Poloidal particle flux (s^{-1})
fnay	Radial particle flux (s^{-1})
fnex	Poloidal electron flux (s^{-1})
fney	Radial electron flux (s^{-1})
fmox	Poloidal flux of parallel momentum (N)
fmoy	Radial flux of parallel momentum (N)
fhex	Poloidal electron energy flux (W)
fhey	Radial electron energy flux (W)
fhetx	Poloidal electron energy flux [fhetx] (W)
fhety	Radial electron energy flux [fhety] (W)
fhix	Poloidal ion energy flux (W)
fhiy	Radial ion energy flux (W)
fhitx	Poloidal ion energy flux [fhitx] (W)
fhity	Radial ion energy flux [fhity] (W)
fhm x	Poloidal parallel kinetic energy flux (W)
fhm y	Radial parallel kinetic energy flux (W)
fhp x	Poloidal potential energy flux (W)
fhp y	Radial potential energy flux (W)
fhj x	Poloidal electrostatic energy flux (W)
fhj y	Radial electrostatic energy flux (W)
fht x	Poloidal total energy flux (W)
fht y	Radial total energy flux (W)
fle	Electron heat flow flux limit factor
fli	Ion heat flow flux limit factor
nflfnax	Poloidal neutral particle flux limit factor
nflfnay	Radial neutral particle flux limit factor
nflfhix	Poloidal neutral ion energy flux limit factor
nflfhiy	Radial neutral ion energy flux limit factor
flvisc	Viscosity flux limit factor
fluce	Electron heat diffusivity Luciani flux limit factor
fluci	Ion heat diffusivity Luciani flux limit factor
flucv	Viscosity Luciani flux limit factor
flucs	Electric conductivity Luciani flux limit factor
fluca	Thermo-electric Luciani flux limit factor
flke	Electron heat diffusivity classical flux limit factor
flki	Ion heat diffusivity classical flux limit factor
flal	Thermo-electric classical flux limit factor
flalc	Thermo-electric coefficient limit factor
cvsx	cvsa.cl.x

cvsy	cvsa-cl-y
ceqp	ceqp
ebld	Energy balance (–divergence of fluxes) (W)
ebls	Energy balance (sources) (W)
ebal	Energy balance (W)
pbal	Particle balance (particles)
hx	hx (m)
hy	hy (m)
hyl	hyl (m) (hy corrected)
sx	sx (m^2)
sxperp	sxperp (m^2)
sy	sy (m^2)
sz	sz (m^2)
vol	vol (m^3)
qz0	qz0
qz1	qz1
qc	qc
mach	Mach number
mach4	Mach number (SOLPS4.0)
ehx	ehx
ehy	ehy
dab2	Atomic density (m^{-3})
dmb2	Molecular density (m^{-3})
ahal	Atomic H–\Ga\R emissivity (photons. $\text{m}^{-3}.\text{s}^{-1}$)
mhal	Molecular H–\Ga\R emissivity (photons. $\text{m}^{-3}.\text{s}^{-1}$)
srcml	Molecule dissociation rate ($\#.\text{s}^{-1}$)
edissml	Power to atoms due to molecule dissociation (W)
neutrad	Power loss due to atom radiation (W)
molrad	Power loss due to molecule radiation (W)
ionrad	Power loss due to molecular ion radiation (W)
rqa	Electron cooling rate (W) [rqahe]
b2ra	Line radiation rate (W)
b2br	Bremsstrahlung radiation rate (W)
rsai	Ionisation ion energy losses (W) [rsahi]
rsan	Ionisation particle losses (s^{-1}) [rsana]
rsam	Ionisation parallel momentum losses (N) [rsamo]
rran	Recombination particle losses (s^{-1}) [rrana]
rrai	Recombination ion energy losses (W) [rrahi]
rram	Recombination parallel momentum losses (N) [rramo]
rcxn	Particle neutral losses (s^{-1}) [rcxna]
rcxi	Ion energy neutral losses (W) [rcxhi]
rcxm	Ion parallel momentum neutral losses (N) [rcxmo]
fetx	Total poloidal internal energy flux (W)
fety	Total radial internal energy flux (W)
tab2	Neutral atom temperature (eV)
tmb2	Neutral molecule temperature (eV)
rfla	Radial atomic flux ($\text{s}^{-1}.\text{m}^{-2}$)
rflm	Radial molecular flux ($\text{s}^{-1}.\text{m}^{-2}$)
refa	Radial atomic energy flux ($\text{W}.\text{m}^{-2}$)
refm	Radial molecular energy flux ($\text{W}.\text{m}^{-2}$)
pfla	Poloidal atomic flux ($\text{s}^{-1}.\text{m}^{-2}$)
pflm	Poloidal molecular flux ($\text{s}^{-1}.\text{m}^{-2}$)
pefa	Poloidal atomic energy flux ($\text{W}.\text{m}^{-2}$)
pefm	Poloidal molecular energy flux ($\text{W}.\text{m}^{-2}$)
mch2	Normalised parallel velocity
ro	Mass density ($\text{kg}.\text{m}^{-3}$)
b2espcr	Nuclear sequence index
b2eater	Eirene atomic species index
mi	Ion mass (kg)
zi,za	Ion charge (e)
za2	Squared ion charge (e^2)
zn	Nuclear charge (e)
fmx	Poloidal mass flux (kg/s)
fmy	Radial mass flux (kg/s)
p*	Dynamic plasma pressure (Pa)
conn	Connection length (m)
wpe	\Gx\R-{\pe} (rad/s)

wce	$\backslash Gx \backslash R_{-}\{ce\}$ (rad/s)
wci	$\backslash Gx \backslash R_{-}\{ci\}$ (rad/s)
vthe	$v_{-}\{th,e\}$ (m/s)
vthi	$v_{-}\{th,i\}$ (m/s)
valfven	$v_{-}\{Alfven\}$ (m/s)
cr	CR (m)
cz	CZ (m)
na0	na at previous time step (m^{-3})
ne0	ne at previous time step (m^{-3})
ni0	ni at previous time step (m^{-3})
te0	te at previous time step (eV)
ti0	ti at previous time step (eV)
ua0	ua at previous time step ($m.s^{-1}$)
po0	po at previous time step (V)
smth	Thermal force (N)
smpr	Thermal force (proportional to exhp) (N)
smpt	Thermal force $'//'$ (proportional to $T_{-}\{i\}$ and $T_{-}\{e\}$ gradients) (N)
smfr	Friction force (N)
smcf	Centrifugal force (N)
pot	Potential energy (eV) (cum. ionisation energy)
poti	Ionisation energy (eV)
pot*	Ionisation energy (eV) [shifted down one]
smp	Self-pressure force
nreg	Region
xreg	X-directed flux region
yreg	Y-directed flux region
sna	Particle source ($\#.s^{-1}$)
sne	Electron particle source ($e.s^{-1}$)
sno	Parallel momentum source (N)
smq	smq (momentum sources from atomic physics) (N)
she	Electron heat source (W)
shi	Ion heat source (W)
sch	Charge source (A)
smaf	smaf (extended friction term) (N)
smag	smag (pressure gradient term) (N)
smav	smav (additional viscosity term) (N)
rco	RESCO
rmo	RESMO
rmt	RESMT
ree	RESEE
rei	RESEI
rpo	RESPO
cpa	CORPA
cua	CORUA
cut	CORUT
cte	CORTE
cti	CORTI
cpo	CORPO
ixlt	ixlt
ixle	ixle
ixeq	ixeq
ixne	ixne
ixge	ixge
ixgt	ixgt
iylt	iylt
iyne	iyne
iyeq	iyeq
iyne	iyne
iyge	iyge
iygt	iygt
islt	islt
isle	isle
iseq	iseq
isne	isne
isge	isge
isgt	isgt
lnlam	$\ln \backslash GK \backslash R$

q	q
s	Shear
alph	$\backslash Ga \backslash R$ (ratio or pressure gradient to critical ballooning value)
d	D ($m^{\{2\}}.s^{\{-1\}}$)
dp	$D_{\{pa\}}$ ($m^{\{2\}}.s^{\{-1\}}$)
kye	$\backslash Gv \backslash R_{\{e\}}$ ($m^{\{2\}}.s^{\{-1\}}$)
kyi	$\backslash Gv \backslash R_{\{i\}}$ ($m^{\{2\}}.s^{\{-1\}}$)
kyi0	$\backslash Gv \backslash R_{\{i\}}^{\{0\}}$ ($m^{\{2\}}.s^{\{-1\}}$)
vlax	Poloidal pinch velocity ($m.s^{\{-1\}}$)
vlay	Radial pinch velocity ($m.s^{\{-1\}}$)
visc	Viscosity ($m.kg^{\{-1\}}.s^{\{-1\}}$)
bvis	Braams viscosity piece ($m.kg^{\{-1\}}.s^{\{-1\}}$)
sig	Anomalous conductivity ($mho.m^{\{-1\}}$)
alf	Thermo-electric coefficient ($mho.m^{\{-1\}}$)
fchp	Parallel current (A)
fchx	Poloidal current (A)
fchy	Radial current (A)
ud	pol comp of the drift velocity [uadia.x] ($m.s^{\{-1\}}$)
vd	rad comp of the drift velocity [uadia.y] ($m.s^{\{-1\}}$)
vadia.x	pol comp of the divergent part of the ion diamagnetic velocity [vadia.x] ($m.s^{\{-1\}}$)
vadia.y	rad comp of the divergent part of the ion diamagnetic velocity [vadia.y] ($m.s^{\{-1\}}$)
wadia.x	pol comp of the total ion diamagnetic velocity [wadia.x] ($m.s^{\{-1\}}$)
wadia.y	rad comp of the total ion diamagnetic velocity [wadia.y] ($m.s^{\{-1\}}$)
vexb.x	pol comp of the ion $E \backslash mU \backslash dB$ velocity [vaecrb.x] ($m.s^{\{-1\}}$)
vexb.y	rad comp of the ion $E \backslash mU \backslash dB$ velocity [vaecrb.y] ($m.s^{\{-1\}}$)
vedia.x	pol comp of the divergent part of the electron diamagnetic velocity [vedia.x] ($m.s^{\{-1\}}$)
vedia.y	rad comp of the divergent part of the electron diamagnetic velocity [vedia.y] ($m.s^{\{-1\}}$)
veexb.x	pol comp of the electron $E \backslash mU \backslash dB$ velocity [veecrb.x] ($m.s^{\{-1\}}$)
veexb.y	rad comp of the electron $E \backslash mU \backslash dB$ velocity [veecrb.y] ($m.s^{\{-1\}}$)
cd	fchdia.x
dd	fchdia.y
niso	Isolating region
bc	Boundary conditions
rc	Recycling conditions
ix	ix (normalised)
iy	iy (normalised)
iix	ix
iiy	iy
lix	leftix
liy	leftiy
rix	rightix
riy	rightiy
bix	bottomix
biy	bottomiy
tix	topix
tiy	topiy
rlcl	real cells
gdcl	guard cells
dib2	Test ion density ($m^{\{-3\}}$)
tib2	Test ion temperature (eV)
brna	Recycling particle source ($\#.s^{\{-1\}}$)
brne	Recycling electron source ($e.s^{\{-1\}}$)
brmo	Recycling parallel momentum source (N)
brhe	Recycling electron heat source (W)
brhi	Recycling ion heat source (W)
brch	Recycling charge source (A)
bcna	Boundary particle source ($\#.s^{\{-1\}}$)
bcne	Boundary electron source ($e.s^{\{-1\}}$)
bcmo	Boundary parallel momentum source (N)
bche	Boundary electron heat source (W)
bchi	Boundary ion heat source (W)
bcch	Boundary charge source (A)
bmna	Additional particle source ($\#.s^{\{-1\}}$)

bmne	Additional electron source ($e \cdot s^{-1}$)
bmmo	Additional parallel momentum source (N)
bmhe	Additional electron heat source (W)
bmhi	Additional ion heat source (W)
bmch	Additional charge source (A)
exna	External particle source ($\# \cdot s^{-1}$)
exne	External electron source ($e \cdot s^{-1}$)
exmo	External parallel momentum source (N)
exhe	External electron heat source (W)
exhi	External ion heat source (W)
exch	External charge source (A)
dvue	b2sihs_divue (phm0)
dvua	b2sihs_divua (phm1)
exbe	b2sihs_exbe (phm5)
exba	b2sihs_exba (phm6)
visa	b2sihs_visa (phm2)
joul	b2sihs_joule (phm3)
fraa	b2sihs_fraa (phm4)
str	b2sihs_str (style)
ke	Parallel kinetic energy (J)
facdrift	facdrift
fac_exb	fac.ExB
fac_vis	fac_vis
xymap	xymap
fnax32	Convective poloidal particle flux (s^{-1})
fnay32	Convective radial particle flux (s^{-1})
fnax52	Conductive poloidal particle flux (s^{-1})
fnay52	Conductive radial particle flux (s^{-1})
fnaxnodr	Poloidal particle flux without drifts (s^{-1})
fnaynodr	Radial particle flux without drifts (s^{-1})
fnix32	Convective poloidal total ion flux (s^{-1})
fniy32	Convective radial total ion flux (s^{-1})
fnix52	Conductive poloidal total ion flux (s^{-1})
fniy52	Conductive radial total ion flux (s^{-1})
fnex32	Convective poloidal electron flux (s^{-1})
fney32	Convective radial electron flux (s^{-1})
fnex52	Conductive poloidal electron flux (s^{-1})
fney52	Conductive radial electron flux (s^{-1})
fchx32	Convective poloidal current (A)
fchy32	Convective radial current (A)
fchx52	Conductive poloidal current (A)
fchy52	Conductive radial current (A)
fchxvspa	Pol comp of fchvispar (A)
fchyvspa	Rad comp of fchvispar (A)
fchxvspe	Pol comp of fchvisper (A)
fchyvspe	Rad comp of fchvisper (A)
fchxn	Pol comp of fchin (A)
fchyn	Rad comp of fchin (A)
fnaxmdf	Modified poloidal particle flux (s^{-1})
fnaymdf	Modified radial particle flux (s^{-1})
fhexmdf	Modified poloidal electron heat flux (W)
fheymdf	Modified radial electron heat flux (W)
fhixmdf	Modified poloidal ion heat flux (W)
fhiymdf	Modified radial ion heat flux (W)
fhexps	Pfirsch-Schlueter poloidal electron heat flux (W)
fheyps	Pfirsch-Schlueter radial electron heat flux (W)
fhixps	Pfirsch-Schlueter poloidal ion heat flux (W)
fhiyps	Pfirsch-Schlueter radial ion heat flux (W)
fnaxps	Pfirsch-Schlueter poloidal particle flux (s^{-1})
fnayps	Pfirsch-Schlueter radial particle flux (s^{-1})
fnaxfcor	Poloidal particle flux with Coriolis term (s^{-1})
fnayfcor	Radial particle flux with Coriolis term (s^{-1})
fnaxhe	Poloidal particle flux for heat equations (s^{-1})
fnayhe	Radial particle flux for heat equations (s^{-1})
fchxvisq	Poloidal current due to heat flux viscosity tensor (A)
fchyvisq	Radial current due to heat flux viscosity tensor (A)
fchxinrt	Poloidal current due to inertia and gyroviscosity (A)

fchyinrt	Radial current due to inertia and gyroviscosity (A)
dnneo	Neoclassical particle diffusivity $D_{\{n\}}$ ($m^2.s^{-1}$)
chneo	Neoclassical heat conductivity $\backslash Gv \backslash R$ ($m^2.s^{-1}$)
vnneo	Neoclassical particle pinch $v_{\{n\}}$ ($m.s^{-1}$)
vwneo	Neoclassical heat pinch vw ($m.s^{-1}$)
lnn	Density gradient length (m)
ltt	Temperature gradient length (m)
nuab	Two species collision frequency (s^{-1})
nua	Total collision frequency (s^{-1})
nustar	Normalized collisionality
vth	Thermal velocity ($m.s^{-1}$)
rho1	Full-B Larmor radius (m)
deltal	Ratio of Larmor radius to minimum gradient length
wb	Width of the banana orbit (m)
deltaw	Ratio of banana width to minimum gradient length
deltawn	Ratio of banana width to density gradient length
deltawt	Ratio of banana width to temperature gradient length
omega	Gyro-frequency ($rad.s^{-1}$)
deltab	Ratio of collision frequency to gyro-frequency
deltapar	Ratio of radial to parallel dynamics
chhichii	Chang-Hinton $\backslash Gv \backslash R_{\{i\}}$ (m^2/s)
crx	Corner radial coordinate (m)
cry	Corner vertical coordinate (m)
zn_ext	External species nuclear charge (e)
za_ext	External species charge (e)
za2_ext	Squared external species charge (e^2)
na_ext	External species density (m^{-3})
mi_ext	External species mass (kg)
ua_ext	External species parallel velocity ($m.s^{-1}$)
ta_ext	External species temperature (eV)
fax_ext	External species poloidal flux ($\#.s^{-1}$)
fay_ext	External species radial flux ($\#.s^{-1}$)
ne_ext	External species associated electron density (m^{-3})
mdims	print out matrix stack information
sumx	sum over x
sumy	sum over y
f.x	line plot versus x
f.y	line plot versus y
intx	integrate over x
inty	integrate over y
f.lx	line plot versus log x
f.ly	line plot versus log y
cont	plot contours
surf	plot patch plot
line	!
psx	per sx
psxperp	per sxperp
psy	per sy
abs	abs()
ddx	d/dx
ddy	d/dy
sumz	sum over ns
pvol	per vol
pne	per ne
m+	matrix addition
m-	matrix subtraction
m*	matrix multiplication
m/	matrix division
rm+	matrix + real
rm-	matrix - real
rm*	matrix * real
rm/	matrix / real
rm**	matrix ** real
drop	drop top of matrix stack
dup	duplicate top of matrix stack
swap	swap top two elements of matrix stack
over	m2 m1 $\longrightarrow$ m2 m1 m2

list	list contents of matrix stack
xsel	select column
ysel	select row
zsel	select species
mmin	min(m1,m2)
mmax	max(m1,m2)
mlt	m<r
mle	m<=r
meq	m=r
mne	m!=r
mgt	m>r
mge	m>=r
rmmi,rmmi	min(r1,m1)
rmma,rmma	max(r1,m1)
rtom	real —> matrix stack
rtoe	real —> Eirene stack
log	log10()
ln	ln()
sqrt	sqrt()
exp	exp()
zrng	select species range
chs	change sign
bala	balance ???
traj	trajectories
stoe	matrix —> Eirene array
stow	matrix —> wall array
xtow	x dimension of matrix —> wall array
ytow	y dimension of matrix —> wall array
rtow	real —> wall stack
rtot	real —> target stack
eirc	add the Eirene atomic quantities to the B2 values
nese	set ne from matrix stack
tese	set te from matrix stack
dase	set dab2 from matrix stack
uuse	set uutraaj from matrix stack
vvse	set vvtraaj from matrix stack
arox	set arrow field x-component from matrix
aroy	set arrow field y-component from matrix
pos	matrix with 1 if element > 0, else 0
target	add target elements plot to surface plot
trgoff	turns off target elements plot
idrop	drop top of integer stack
idup	duplicate top of integer stack
iswap	swap top two elements of integer stack
iover	i2 i1 —> i2 i1 i2
rdrop	drop top of real stack
rdup	duplicate top of real stack
rswap	swap top two elements of real stack
rover	r2 r1 —> r2 r1 r2
xmin	Minimum x for plotting in the computational domain
xmax	Maximum x for plotting in the computational domain
ymin	Minimum y for plotting in the computational domain
ymax	Maximum y for plotting in the computational domain
rmin	Minimum r for plotting in the physical domain
rmax	Maximum r for plotting in the physical domain
zmin	Minimum z for plotting in the physical domain
zmax	Maximum z for plotting in the physical domain
fmin	Minimum function value for plotting
fmax	Maximum function value for plotting
omin	Minimum ordinate for 2-d plots
omax	Maximum ordinate for 2-d plots
xref	Reference position in x
wmin	wmin in pltwall
wmax	wmax in pltwall
t_pt	total flight time of particles in caltraj
rshi	relative R shift to be applied to chords
zshi	relative Z shift to be applied to chords

r0	used to specify the R position of an ellipse in chord.F
z0	used to specify the Z position of an ellipse in chord.F
a0	used to specify the R radius of an ellipse in chord.F
b0	used to specify the Z radius of an ellipse in chord.F
tt0	used to specify the value within the ellipse in chord.F
trgxmin	Minimum x for target plotting in the computational domain
trgxmax	Maximum x for target plotting in the computational domain
trgymin	Minimum y for target plotting in the computational domain
trgymin	Maximum y for target plotting in the computational domain
trgrmin	Minimum r for target plotting in the physical domain
trgrmax	Maximum r for target plotting in the physical domain
trgzmin	Minimum z for target plotting in the physical domain
trgzmax	Maximum z for target plotting in the physical domain
trgfmin	Minimum function value for target plotting
trgfmax	Maximum function value for target plotting
trgomin	Minimum ordinate for 2-d target plots
trgomax	Maximum ordinate for 2-d target plots
smin	Minimum ordinate for wall plots
smax	Maximum ordinate for wall plots
emin	Minimum function value for Eirene data plots
emax	Maximum function value for Eirene data plots
neo.ic	Neoclassical model number
extl	extra label
cltb	colour table
chor	chord file
chpl	plot chords
cmd	command file
cmdcdf	read te, ti, ne, na, dab2, dmb2 from the b2movies cdf file and loop calling a command
file	for each frame
chbx	boxcar average of chords
chvl	value at positions
flux	call WOSlib flux routine
chmp	plots a contour plot of the chord contribution to the mesh element
cam	camera plot
movi	movie
sonn	something to do with Sonnet (no longer working)
saw	reads external data onto the matrix stack
saws	reads external data about plasma species onto the matrix stack
sawx	reads external data about external species onto the matrix stack
oson	output geometry in Sonnet format
uedgegrd	output geometry in UEDGE format (string + real arg)
expf	add experimental data to plot
outp	prints the value of a specific element
pdev	PLPLOT device name
mptr	output the top of matrix stack to a file
mtab	output a table from the top of matrix stack to a file
amju,amjuel	accesses Eirene type atomic physics data
adas	accesses ADAS atomic physics data
scat	concatenate two strings on top of string stack
sdrop	drop top of string stack
sdup	duplicate top of string stack
sswap	swap top two elements of string stack
sover	str2 str1 —> str2 str1 str2
sprt	output the string stack
omovie	old movie option
movie,nmovie	new movie option
walllbl	set wall bar label
trgblbl	set target bar label
glbl	set graph label
elbl	set Eirene graph label
wlbl	set wall graph label
tlbl	set target graph label
slbl	set species label
eslbl	set Eirene species label
wslbl	set wall species label
tranprof	output transport profile
pdena	Atoms particle density (m^{-3})

pdenm	Molecules particle density (m^{-3})
pdeni	Test ions particle density (m^{-3})
edena	Atoms energy density ($\text{eV} \cdot \text{m}^{-3}$)
edenm	Molecules energy density ($\text{eV} \cdot \text{m}^{-3}$)
edeni	Test ions energy density ($\text{eV} \cdot \text{m}^{-3}$)
massa	Atom mass (kg)
massm	Molecule mass (kg)
massi	Molecular ion mass (kg)
vxa	Atoms X-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vxm	Molecules X-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vxi	Test ions X-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vya	Atoms Y-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vym	Molecules Y-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vyi	Test ions Y-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vza	Atoms Z-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vzm	Molecules Z-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vzi	Test ions Z-directed velocity ($\text{m} \cdot \text{s}^{-1}$)
vxdena	Atoms X-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vxdenm	Molecules X-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vxdeni	Test ions X-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vydena	Atoms Y-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vydenm	Molecules Y-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vydeni	Test ions Y-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vzdena	Atoms Z-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vzdenm	Molecules Z-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
vzdeni	Test ions Z-directed momentum density ($\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
triarea	Triangle area (m^2)
triavol	Triangle volume (m^3)
tr	Triangle centroid R coordinate (m)
tz	Triangle centroid Z coordinate (m)
pux	PUX
puy	PUY
pvx	PVX
pyy	PVY
tindex	Triangle index
edims	print out Eirene stack information
psta	per triangle area
pstv	per triangle volume
sumt	sum over triangles
esurf	plot triangle patch plot
eclvs	customise Eirene contour levels
eclvd	customise Eirene contour level delta
eclvn	customise Eirene contour level range and number
eclvp	customise Eirene contour level by proportion
eabs	abs() on Eirene stack
esumz	sum over ns for Eirene stack data
em+	Eirene stack addition
em-	Eirene stack subtraction
em*	Eirene matrix multiplication
em/	Eirene matrix division
rem+	Eirene stack + real
rem-	Eirene stack - real
rem*	Eirene stack * real
rem/	Eirene stack / real
rem**	Eirene stack ** real
edrop	drop top of Eirene stack
edup	duplicate top of Eirene stack
eswap	swap top two elements of Eirene stack
eover	$e2 \ e1 \longrightarrow e2 \ e1 \ e2$
elist	list contents of Eirene stack
esel	select Eirene element
ezsel	select species in Eirene stack
emmin	min(e1,e2)
emmax	max(e1,e2)
remmi	min(r1,e1)
remma	max(r1,e1)
emlt	$e < r$

emle	$e \leq r$
emeq	$e = r$
emne	$e \neq r$
emgt	$e > r$
emge	$e \geq r$
elog	$\log_{10}(\text{Eirene})$
eln	$\ln(\text{Eirene})$
esqrt	$\sqrt{\text{Eirene}}$
eexp	$\exp(\text{Eirene})$
ezrng	select species range in Eirene stack
echs	change sign in Eirene stack
etos	Eirene array $\rightarrow$ m
epos	Eirene stack with 1 if element > 0 , else 0
evac	fill vacuum region of Eirene stack with a value
earox	set arrow field x-component from Eirene data
earoy	set arrow field y-component from Eirene data
etal	read Eirene tally output file onto the Eirene stack
echor	chord file for Eirene stack data
echbx	boxcar average of chords for Eirene stack data
echvl	Eirene stack values at positions
echmp	plots a contour plot of the chord contribution to the Eirene element
ecam	camera plot of Eirene stack data
eoutp	prints the value of a specific Eirene element
eprt	output the top of Eirene stack to a file
eavg	
wallarea	Wall area (m^2)
ei	Incident energy (eV)
wallflux	Wall particle flux (s^{-1})
wallhflx	Wall heat flux (W/m^2)
r1	Wall start R position (m)
r2	Wall end R position (m)
z1	Wall start Z position (m)
z2	Wall end Z position (m)
walltemp	Wall surface temperature (K)
plathick	Plate thickness (m)
coathick	Coating thickness (m)
cooltemp	Backplate temperature (K)
vaprttemp	Vaporisation temperature (K)
north	North boundary
south	South boundary
west	West boundary
east	East boundary
nseg	Wall segment
iwlt	iwlt
iwle	iwle
iweq	iweq
iwne	iwne
iwgt	iwgt
iwge	iwge
iswlt	iswlt
iswle	iswle
isweq	isweq
iswne	iswne
iswgt	iswgt
iswge	iswge
ibnd	ibnd
walldist	Distance along wall (m)
wallngth	Wall length (m)
sput	Physical sputtering yield
sput2	Physical sputtering energy yield
sputchem	Chemical sputtering yield
thermev	Thermal evaporation fluxes (m^2/s)
sputres	RES sputtering yield
backscat	Backscattering fraction
backscae	Backscattering energy fraction
erosion	Gross erosion
deposit	Gross deposition

ero_ml	Monolayer erosion
depo_ml	Monolayer deposition
time	Plate time (s)
imapx	imapx
imapy	imapy
kappa_a	Bulk Kappa A factor (1/AT+B) (m/W)
kappa_b	Bulk Kappa B factor (1/AT+B) (m.K/W)
kappa_a1	Bulk Kappa A1 polynomial factor (W/m/K)
kappa_a2	Bulk Kappa A2 polynomial factor (W/m/K ²)
kappa_a3	Bulk Kappa A3 polynomial factor (W/m/K ³)
kappa_a4	Bulk Kappa A4 polynomial factor (W/m/K ⁴)
kappa_a5	Bulk Kappa A5 polynomial factor (W/m/K ⁵)
kappapa1	Bulk Kappa_perp A1 polynomial factor (W/m/K)
kappapa2	Bulk Kappa_perp A2 polynomial factor (W/m/K ²)
kappapa3	Bulk Kappa_perp A3 polynomial factor (W/m/K ³)
kappapa4	Bulk Kappa_perp A4 polynomial factor (W/m/K ⁴)
kappapa5	Bulk Kappa_perp A5 polynomial factor (W/m/K ⁵)
tgdns	Bulk mass density (kg/m ³)
cp_b1	Bulk heat capacity B1 factor (J/kg/K)
cp_b2	Bulk heat capacity B2 factor (J/kg/K ²)
cp_b3	Bulk heat capacity B3 factor
cp_b4	Bulk heat capacity B4 factor (J/kg/K ⁴)
ckappa_a	Coating Kappa A factor (1/AT+B) (m/W)
ckappa_b	Coating Kappa B factor (1/AT+B) (m.K/W)
ckappaa1	Coating Kappa A1 polynomial factor (W/m/K)
ckappaa2	Coating Kappa A2 polynomial factor (W/m/K ²)
ckappaa3	Coating Kappa A3 polynomial factor (W/m/K ³)
ckappaa4	Coating Kappa A4 polynomial factor (W/m/K ⁴)
ckappaa5	Coating Kappa A5 polynomial factor (W/m/K ⁵)
ckapppa1	Coating Kappa_perp A1 polynomial factor (W/m/K)
ckapppa2	Coating Kappa_perp A2 polynomial factor (W/m/K ²)
ckapppa3	Coating Kappa_perp A3 polynomial factor (W/m/K ³)
ckapppa4	Coating Kappa_perp A4 polynomial factor (W/m/K ⁴)
ckapppa5	Coating Kappa_perp A5 polynomial factor (W/m/K ⁵)
ctgdns	Coating mass density (kg/m ³)
c_cp_b1	Coating heat capacity B1 factor (J/kg/K)
c_cp_b2	Coating heat capacity B2 factor (J/kg/K ²)
c_cp_b3	Coating heat capacity B3 factor
c_cp_b4	Coating heat capacity B4 factor (J/kg/K ⁴)
erel	Chemical sputtering factor erel
bbdyemis	Blackbody emissivity
vapprs.a	Vapor pressure A factor (K)
vapprs.b	Vapor pressure B factor
vapprs	Vapor pressure (Pa)
qevap	Energy lost by vaporisation (J/m ² /s)
grad	Energy lost by blackbody radiation (J/m ² /s)
stckcoef	Sticking coefficient
sublheat	Heat of sublimation (J/kg)
lkappa	Layer heat conductivity (W/m/K)
lrho	Layer mass density (kg/m ³)
lcp	Layer heat capacity (J/kg/K)
lemis	Layer blackbody emissivity
lthick	Layer thickness (m)
ldepth	Layer depth (m)
sputfrac	Layer fraction
wall	adds wall stack data to contour plot
wall_ank	add Kukushkin wall data to contour plot
logw	log wall elements
linw	linear wall elements
logxw	log x wall elements plot
linxw	linear x wall elements plot
plwl	plot in wall domain
sumw	sum over walls
intw	integrate over walls
intw*	integrate over walls (no account of segments)
pswa	per wall area
wclvs	customise wall contour levels

wclvd	customise wall contour level delta
wclvn	customise wall contour level range and number
wclvp	customise wall contour level by proportion
wwid	wall line width factor
wabs	abs() on wall stack
ddw	d/dw derivative along wall length
wsumz	sum over ns for wall stack data
pwne	wall data per ne
wm+	wall stack addition
wm-	wall stack subtraction
wm*	wall stack multiplication
wm/	wall stack division
rwm+	wall stack + real
rwm-	wall stack - real
rwm*	wall stack * real
rwm/	wall stack / real
rwm**	wall stack ** real
wdrop	drop top of wall stack
wdup	duplicate top of wall stack
wswap	swap top two elements of wall stack
wover	w2 w1 $\rightarrow$ w2 w1 w2
wlist	list contents of wall stack
wsel	select wall element
wzsel	select species in wall stack
wmmin	min(w1,w2)
wmmax	max(w1,w2)
rwmmin	min(r1,w1)
rwmmax	max(r1,w1)
wmlt	w<r
wmle	w<=r
wmeq	w=r
wmne	w!=r
wmgt	w>r
wmge	w>=r
wlog	log10(wall)
wln	ln(wall)
wsqrt	sqrt(wall)
wexp	exp(wall)
wzrng	select species range in wall stack
wchs	change sign in wall stack
wtos	wall array $\rightarrow$ m
wtot	wall array $\rightarrow$ target
wpos	wall stack with 1 if element > 0, else 0
wsaw	reads external data onto the wall stack
wallf	add experimental wall data to plot
outpw	prints the value of a specific wall element
wprt	output the top of wall stack to a file
wtab	output a table from the top of wall stack to a file
wlld	ANK wall loading
woff	turn off wall plots
logtrg	log target elements
lintrg	linear target elements
logxtrg	log x target elements plot
linxtrg	linear x target elements plot
tclvs	customize target contour levels
tclvd	customize target contour level delta
tclvn	customize target contour level range and number
tclvp	customize target contour level by proportion
trg.w	line plot versus target wall
trg.z	line plot versus target depth
sumtrgw	sum over targets along walls
sumtrgz	sum over targets along z
inttrgw	integrate over targets along walls
inttrgw*	integrate over targets along walls (no account of segments)
inttrgz	integrate over targets along z
trgabs	absolute value of target data
trgddw	target d/dw derivative along wall length

trgddp	target d/dp derivative along perpendicular direction
trgddz	target d/dz derivative along target depth
rtrgm+	target stack + real
rtrgm-	target stack - real
rtrgm*	target stack * real
rtrgm/	target stack / real
rtrgm**	target stack ** real
trgtemp	target temperature (K)
trgdrop	drop top of target stack
trgwsel	select wall element in target stack
trgzsel	select depth element in target stack
rtrgmmin	min(r1, trg)
rtrgmmax	max(r1, trg)
trglt	trg < r
trgle	trg <= r
trgeq	trg = r
trgne	trg != r
trggt	trg > r
trgge	trg >= r
trglog	log10(target)
trgln	ln(target)
trgsqrt	sqrt(target)
trgexp	exp(target)
trgchs	change sign in target stack
trgtow	target array —> wall
trgpos	target stack with 1 if element > 0, else 0
outptrg	prints the value of a specific target element
trgprt	output the target stack to a file
trgtab	output a table from the top of target stack to a file
trgl	ANK target loading
trgsurf	plot target plot
kappa_ab	target heat conductivity computed as 1/(AT+B)
kappa	target heat conductivity computed from polynomial fit
dkappadt	target heat conductivity derivative
kappap	perp. target heat conductivity computed from polynomial fit
kappapdt	perp. target heat conductivity derivative
cp	target specific heat capacity
rho	target mass density
trgwidth	target element width
trgdepth	target element depth position
trgvol	target element volume
trgdup	duplicate top of target stack
trgswap	swap top two elements of target stack
trgover	t2 t1 —> t2 t1 t2
trglist	list contents of target stack
trgmmin	min(t1, t2)
trgmmax	max(t1, t2)
trgr1	target element R1 position (m)
trgr2	target element R2 position (m)
trgz1	target element Z1 position (m)
trgz2	target element Z2 position (m)
trgnorth	North boundary
trgsouth	South boundary
trgwest	West boundary
trgeast	East boundary
trgseg	target segment
trgibnd	trgibnd
trgiz	trgiz
itrslt	itrslt
itrle	itrle
itrgeq	itrgeq
itrne	itrne
itrsgt	itrsgt
itrge	itrge
ztrslt	ztrslt
ztrle	ztrle
ztrgeq	ztrgeq

ztrgne	ztrgne
ztrggt	ztrggt
ztrgge	ztrgge
trgm+	target stack addition
trgm-	target stack subtraction
trgm*	target stack multiplication
trgm/	target stack division
me	electron mass
mp	proton mass
ev	electron charge
qe	electron charge
mu0	magnetic permittivity of the vacuum
eps0	electric permeability of the vacuum
pi	pi
c	speed of light
avogr	Avogadro's number
kbolt	Boltzmann constant
jxi	jxi
jxa	jxa
jsep	jsep
ixref	ixref
iyref	iyref
nx	nx
ny	ny
ns	ns
nspecies	nspecies
natm	natm
nmol	nmol
nion	nion
nxtl	nxtl
nxtr	nxtr
nbnd	nbnd
ndepth	ndepth_nml
ismain	ismain
ismain0	ismain0
ns_ext	ns_ext
f30	write fort.30
f31	write fort.31
glob	transform an extra label into a global header
both	both phys and comp
phys	physical domain only
comp	computational domain only
trigmesh	overlay triangular grid
mesh	overlay mesh
expl	toggle value of expline
page	set number of graphs per page
logf	log f plot
linf	linear f plot
logx	log x plot
linx	linear x plot
clvs	customise contour levels
clvd	customise contour level delta
clvn	customise contour level range and number
clvp	customise contour level by proportion
debu,debug	set debugging level
exps	set symbol for experimental data
eclv	do not fill contour Levels
fclv	do fill contour Levels
font	set font
lacl	label orientation along contour levels
lhor	label orientation horizontal
lbon	labeling on / off
lbof	labeling on / off
lwid	line width factor
zwid	axis line width factor
lfsz	line label font size
grid	grid on

trig	triangular grid on
clvl	contour level lines off
spln	switch on interpolation
arow	change default arrow length
earow	change default arrow length for Eirene stack data
arww	change default arrow width
acol	colour index for arrows
sep	plot separatrix
vesl	plot vessel walls from mesh.extra file
mmid	map to midplane
fpsi	plot versus full psi
mps	plot versus normalised psi
rpsi	plot versus root(normalised psi)
nocp	do not plot contour plot
nice	toggle whether axes will have nice values
bold	toggle whether to bold the axis labels
sepw	change separatrix line width
ewid	change experimental data line width
owid	change outline line width
vwid	change vessel line width
sq	square aspect ratio
xpol	use poloidal distance for f.x
xpar	use parallel distance for f.x
a4p	A4 portrait
a4l	A4 landscape
same	next graph on same axes
filo	fill option
i+	$i_1 + i_2$
i-	$i_1 - i_2$
i*	$i_1 * i_2$
i/	i_1 / i_2
i**	$i_1 ** i_2$
itor	integer $\rightarrow$ real stack
itostr	integer $\rightarrow$ string stack
itom	integer $\rightarrow$ matrix stack
itoe	integer $\rightarrow$ Eirene stack
itow	integer $\rightarrow$ wall stack
itot	integer $\rightarrow$ target stack
iprt	output the integer stack
r+	$r_1 + r_2$
r-	$r_1 - r_2$
r*	$r_1 * r_2$
r/	r_1 / r_2
r**	$r_1 ** r_2$
rtostr	real $\rightarrow$ string stack
rpri	output the real stack
new	new page
lbsz	label size
glsz	globalheader size
ndec	number of decades
aspc	preserve aspect ratio flag
col	colour
bw	black and white
logt	log time for trajectories
lint	linear time for trajectories
zero	plot origin
outl	outline of plasma grid
stdl	"standard" label
mvsk	movie skip
ave	number of things to average over
t_#p	number of trajectories
reso	reset output
swid	part of tcl/tk interface
xvz	part of tcl/tk interface
t_r+	ratio true
t_r-	ratio false
mcol	colour index of mesh

```

scol    colour index of separatrix
vcol    colour index of vessel
ocol    colour index of grid outline
zcol    colour index of zero lines
hcol    colour index of text header
gcol    colour index of graph title
bcol    colour index of bar label title
xcol    colour index of extralabel
dcol    colour index for matrix data line plots
wcol    colour index for wall data line plots
tcol    colour index for target data line plots
rigz    toggle set\_resignore\_regions\_to\_zero
noax    no axes
part    partition colour map
lbsa    toggle label lines
ytra    transform the "y" coordinate for comparison to experiment
ecol    colour index of experimental data
fcoll   colour index for graph frame
nspe,nspec  new spectrometer
spec    old spectrometer
coro    print coronal rates
cora    coronal radiation rate
coni    coronal densities
shco    part of tcl/tk
user    call a user function "b2paux"
writ,write  set writing to file
omp     set jxa (OMP)
imp     set jxi (IMP)
yref    set yref

```

A revised copy of the **b2plot** manual can be found in Appendix I, page 422. Note that some of the quantities have changed name between **SOLPS 4.x** and **SOLPS 5.x** versions, table 3.1, and that whereas *ix* ran from 0 to *nx*+1 it now runs from -1 to *nx*, *iy* which ran from 0 to *ny*+1 now runs from -1 to *ny* and *is* which ran from 1 to *ns* now runs from 0 to *ns*-1. **SOLPS-ITER** uses the same indexing and naming conventions as **SOLPS 5.x**.

SOLPS 4.x	SOLPS 5.x
ni	na
zi	za
up	ua
dx	hx
dy	hy1
feex	fhex
feey	fhey
feix	fhix
feiy	fhiy
fnix	fnax
fniy	fnay

Table 3.1: Changes of quantity names from **SOLPS 4.x b2plot** to **SOLPS 5.x b2pl.exe**.

3.7 b2yg, plotting the geometry

The commands

```
b2run b2yg
```

or

```
gmake -f $$SOLPSTOP/runs/Makefile b2yg.prt
```

or

```
gmake -f $$SOLPSTOP/runs/Makefile b2yg.plt
```

will produce a `b2yg.plt` file containing a series of plots giving some details about the geometry, as well as a `b2yg.prt` ASCII file containing tables of the geometry quantities.

The `b2yg.plt` file is an NCAR GKS file. The presence in the run directory of a file called `ncar.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and more visible lines and an easier-to-read font. The file can be converted into a PostScript file by means of the command:

```
ctrans -d ps.mono b2yg.plt > b2yg.ps
```

The input file `b2yg.dat` looks like:

```
*idout          (int, char*80, free format)
  0, '....'
*nmbpcc
  1
  0.5 0.5
*nmbpdm
  1
  0.5 3.0 -1.25 1.25
```

The meaning of the variables is:

iftp Formatting index for the plot contour lines

idout String specifying the type of data tables requested

nmbpcc Number of computational domain plots requested

cwaxr, cwayr For each computational plot requested, a doublet of numbers indicating the x and y relative positions, in computational space, of the reference location for these plots

nmbpdm Number of physical domain plots requested

cwxmin, cwymax, cwymin, cwymax For each physical domain plot requested, a quadruplet of coordinates (x_{min} , x_{max} , y_{min} , y_{max}) that give the bounds of the domain to be plotted

The data fields that can be requested through **idout** are:

'pgmm' Print min/max values of the geometry data fields

'pgcc' Print geometry data on 5×5 corners

'pgnl' Print singular regions data

'pgmx' Print x -faced data

'pgmy' Print y -faced data

'pgso' Print South boundary data

'pgno' Print North boundary data

'pgwe' Print West boundary data

'pgea' Print East boundary data

'mpcr' Plot the geometry and mesh

'mpff' Plot the flux functions **fpsi** and **ffbz**

'mpbb' Plot the magnetic field

A **'.'** can be used as a wild card when specifying **idout**. Therefore, the string **'....'** will request all data fields. Several fields can be requested through the same **idout** string by listing them in sequence with or without some separating character(s).

3.8 b2yh, plotting the run parameters

The commands

```
b2run b2yh
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2yh.prt
```

will produce a `b2yh.prt` file containing a description of the run parameters: a species list, default boundary conditions parameters, and transport coefficients. These parameters will be extracted from `b2fparam` if it is present (thereby giving the run parameters as they were used by `b2mn`), or from `b2fpardf` which contains the default set of parameters built from `b2ah`.

The input file `b2yh.dat` looks like:

```
*style          (integer, free format)
 1
```

The only number in that file is a style switch that changes the type of output printed out in `b2yh.prt`. Current accepted values are 0 and 1.

3.9 b2yi & b2yp, plotting the plasma state

The commands

```
b2run b2yi
```

or

```
gmake -f $SOLPSTOP/runs/Makefile.plot b2yi.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile.plot b2yi.plt
```

will produce a `b2yi.plt` file containing a series of plots describing the plasma state, as well as a `b2yi.prt` ASCII file containing tables of the plasma quantities. The plasma state will be extracted from `b2fstate` if it is present (thereby giving the plasma state as it was computed by `b2mn`), or from `b2fstati` which contains the initial plasma state built from `b2ai`.

Similarly, the commands

```
b2run b2yp
```

or any of

```
gmake -f $SOLPSTOP/runs/Makefile b2yp.prt
```

```
gmake -f $SOLPSTOP/runs/Makefile b2yp.plt
```

```
gmake -f $SOLPSTOP/runs/Makefile b2specp
```

will produce a `b2yp.plt` file with plots describing the plasma state computed by `b2mn` and stored in the `b2fstate` file, as well as a `b2yp.prt` ASCII file with tables of plasma quantities. `b2yp` also produces a `b2specp` file.

The `b2yi.plt` and `b2yp.plt` files are NCAR GKS files. The presence in the run directory of a file called `ncar.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and more visible lines and an easier-to-read font. The files can be converted into PostScript by means of the commands:

```
ctrans -d ps.mono b2yi.plt > b2yi.ps
ctrans -d ps.mono b2yp.plt > b2yp.ps
```

The input file `b2yi.dat` looks like:

```
*idout          (int, char*80, free format)
0, 'cpne,cpni,cpte,cpti,fne.,fhe.,pf.,pz..'
```

The input file `b2yp.dat` looks like:

```
*idout          (int, char*80, free format)
1, 'cpne,cpte,cpti,pf.,pz..'
*nmbpcc
0
*nmbpdm
1
0.5 3.0 -1.25 1.25
```

The meaning of the variables is:

in both `b2yi.dat` and `b2yp.dat` files:

iftp Formatting index for the plot contour lines

idout String specifying the type of data requested

additionally, in `b2yp.dat`:

nmbpcc Number of computational domain plots requested

cwaxr, cwayr For each computational plot requested, a doublet of numbers indicating the x and y relative positions, in computational space, of the reference location for these plots

nmbpdm Number of physical domain plots requested

cwxmin, cwxmax, cwymmin, cwymax For each physical domain plot requested, a quadruplet of coordinates $(x_{min}, x_{max}, y_{min}, y_{max})$ that give the bounds of the domain to be plotted

The data fields that can be requested through **idout** are:

- 'pzmm'** Print min/max values of the plasma quantities
- 'pfnx'** Print reduced x -face particle fluxes
- 'pfzx'** Print reduced x -face electron and ion fluxes
- 'pfny'** Print reduced y -face particle fluxes
- 'pfzy'** Print reduced y -face electron and ion fluxes
- 'pzso'** Print South boundary data
- 'pzno'** Print North boundary data
- 'pzwe'** Print West boundary data
- 'pzea'** Print East boundary data
- 'cpne'** Plot the electron density
- 'cpni'** Plot the total ion density
- 'cpte'** Plot the electron temperature
- 'cpti'** Plot the ion temperature
- 'cprp'** Plot the total pressure (including neutrals)
- 'cppz'** Plot the plasma pressure
- 'cppo'** Plot the electric potential
- 'cpfn'** Plot the fractional atom density
- 'cppe'** Plot the electron pressure
- 'cpua'** Plot the parallel velocity
- 'cpmc'** Plot the Mach number
- 'cphe'** Plot the electron heat flux
- 'cphi'** Plot the ion heat flux
- 'fnex'** Plot the integrated total particle fluxes as function of x
- 'fhex'** Plot the integrated heat fluxes as function of x
- 'fchx'** Plot the integrated electric current as function of x
- 'fnax'** Plot the integrated ion particle fluxes as function of x
- 'fney'** Plot the integrated total particle fluxes as function of y
- 'fhey'** Plot the integrated heat fluxes as function of y
- 'fchy'** Plot the integrated electric current as function of y
- 'fnay'** Plot the integrated ion particle fluxes as function of y

A ' .' can be used as a wild card when specifying **idout**. Therefore, the string ' ' will request all data fields. Several fields can be requested through the same **idout** string by listing them in sequence with or without some separating character(s).

3.10 b2yr, plotting the atomic physics rates

The commands

```
b2run b2yr
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2yr.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2yr.plt
```

will produce a `b2yr.plt` file containing a series of plots detailing the dependencies of the atomic rates on density and temperature for each isonuclear sequence, as well as a `b2yr.prt` ASCII file containing tables of these rates. If the `fix_recomb` switch has been activated, then the electron cooling rate is plotted with and without the contribution due to the ionization potential of the recombining ion, and additional graphs are included for the negative of that rate if it becomes a heating (instead of cooling) rate.

The `b2yr.plt` file is an NCAR GKS file. The presence in the run directory of a file called `ncar.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and more visible lines and an easier-to-read font. The file can be converted into a PostScript file by means of the command:

```
ctrans -d ps.mono b2yr.plt > b2yr.ps
```

The input file for `b2yr` is `b2yr.dat`, and an example is provided below:

```
*nijdis      (integer, free format)
  4  4
*idis        (nidis*integer, free format)
  8 16 24 32
*jdis        (njdis*integer, free format)
  2  6 10 14
```

The first set of integer is the number of density and temperature slices, respectively, that will be plotted, and the next two sets of integers are the indices within the density and temperatures tables in `b2frates` that are being plotted.

The user has the option when running the main code to rescale the rates given by `b2ar` through the use of the `b2.atomic_physics_rescale.parameters` file. See [3.5.8](#) for details.

3.11 b2ym, producing movie output

The commands

```
b2run b2ym
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2ym.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2ym.plt
```

will produce a `b2ym.plt` file containing a series of movie frames, as well as a `b2ym.prt` ASCII file containing some additional output. The data for the movie is extracted from the `b2fmovie` file.

The `b2ym.plt` file is an NCAR GKS file. The presence in the run directory of a file called `ncar.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and more isible lines and an easier-to-read font. The file can be converted into a PostScript file by means of the command:

```
ctrans -d ps.mono b2ym.plt > b2ym.ps
```

The input file `b2ym.dat` looks like:

```
*idout          (int, char*80, free format)
  0, 'cpne,cpte,cpti'
```

The meaning of the variables is:

iftp Formatting index for the plot contour lines

idout String specifying the type of data requested

The data fields that can be requested through **idout** are:

'**cpne**' Make contour plots of the electron density change

'**cpna**' Make contour plots of the ion density changes

'**cpni**' Make contour plots of the total atom density change

'**cpte**' Make contour plots of the electron temperature change

'**cpti**' Make contour plots of the ion temperature change

'**cppo**' Make contour plots of the electric potential change

'**cpua**' Make contour plots of the ion parallel velocity changes

A '`.`' can be used as a wild card when specifying **idout**. Therefore, the string '`.....`' will request all data fields. Several fields can be requested through the same **idout** string by listing them in sequence with or without some separating character(s).

3.12 b2yn, display the progress of inner B2.5 iterations

The commands

```
b2run b2yn
```

or

```
gmake -f $SOLPSTOP/runs/Makefile.plot b2yn.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile.plot b2yn.plt
```

will produce a `b2yn.plt` file containing contour plots of the residuals and corrections from the B2.5 equations, as well as a `b2yn.prt` ASCII file containing some additional output. The data from the B2.5 inner iterations is extracted from the `b2ftrack` file.

The `b2yn.plt` file is an NCAR GKS file. The presence in the run directory of a file called `ncar.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and more isible lines and an easier-to-read font. The file can be converted into a PostScript file by means of the command:

```
ctrans -d ps.mono b2yn.plt > b2yn.ps
```

The input file `b2yn.dat` looks like:

```
*idout          (int, char*80, free format)
  0, 'cpne,cpte,cpti'
```

The meaning of the variables is:

iftp Formatting index for the plot contour lines

idout String specifying the type of data requested

The data fields that can be requested through **idout** are:

'**cpne**' Make contour plots of electron density residuals and correction

'**cpni**' Make contour plots of total ion density residuals and correction

'**cpte**' Make contour plots of electron temperature residuals and correction

'**cpti**' Make contour plots of ion temperature residuals and correction

'**cppo**' Make contour plots of electric potential residuals and correction

'**cpna**' Make contour plots of ion density residuals and correction

'**cpua**' Make contour plots of ion parallel momentum residuals and correction

'**cput**' Make contour plots of total parallel momentum residuals and correction

'**cppr**' Make contour plots of total pressure correction

A ' .' can be used as a wild card when specifying **idout**. Therefore, the string ' . . . ' will request all data fields. Several fields can be requested through the same **idout** string by listing them in sequence with or without some separating character(s).

3.13 b2yq, display the evolution data

The commands

```
b2run b2yq
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2yq.prt
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2yq.plt
```

will produce a `b2yq.plt` file containing a quick display of the B2.5 evolution data, as well as a `b2yq.prt` ASCII file containing some additional output. The data from B2.5 is extracted from the `b2ftrace` file.

The input file `b2yq.dat` looks like:

```
1 2
1 3
1 6
1 7
1 8
1 9
```

Each line of the `b2yq.dat` file contains a doublet of integers:

indx Index of the data field in the `b2ftrace` file to use as abscissa

indy Index of the data field in the `b2ftrace` file to use as ordinate

The number of quantities actually written to the `b2ftrace` file during a B2.5 run depends on the setting of the `b2mwqt_style` switch in `b2mn.dat`. They are, according to `b2mwqt_style` value:

style=1 default value

- 1 plasma simulation time
- 2 iteration counter
- 3..ns+2 norms of ion continuity residuals
- ns+3..2*ns+2 norms of ion momentum residuals
- 2*ns+3 norm of total momentum residual
- 2*ns+4 norm of electron energy equation
- 2*ns+5 norm of ion energy equation
- 2*ns+6 norm of electric potential equation
- 2*ns+7..3*ns+6 norms of ion continuity corrections
- 3*ns+7..4*ns+6 norms of ion momentum corrections
- 4*ns+7 norm of total momentum correction
- 4*ns+8 norm of electron temperature correction
- 4*ns+9 norm of ion temperature correction
- 4*ns+10 norm of electric potential equation

style=2 all of the above, plus data for 36 grid regions:

- 4*ns+11..40*ns+10 total ion particle content per region

40*ns+11..40*ns+46 total electron energy content per region
40*ns+47..40*ns+82 total ion energy content per region
40*ns+83..40*ns+118 total electric energy content per region

style=3 all of the above, plus 20 integrated cross-region fluxes:

40*ns+119..60*ns+118 *x*-directed particle fluxes
60*ns+119..80*ns+118 *y*-directed particle fluxes
80*ns+119..80*ns+138 *x*-directed electron energy fluxes
80*ns+139..80*ns+158 *y*-directed electron energy fluxes
80*ns+159..80*ns+178 *x*-directed ion energy fluxes
80*ns+179..80*ns+198 *y*-directed ion energy fluxes
80*ns+199..80*ns+218 *x*-directed electric current
80*ns+219..80*ns+238 *y*-directed electric current

style=4 all of the above, plus data for 36 grid regions

80*ns+239..152*ns+238 min/max values of ion densities per region
152*ns+239..152*ns+310 min/max values of electron temperature per region
152*ns+311..152*ns+382 min/max values of ion temperature per region
152*ns+383..152*ns+454 min/max values of electric potential per region

The grid regions are formed as computational rectangles where the first and last region correspond to the domain boundaries ($ix = -1$, $ix = nx$, $iy = -1$ and $iy = ny$) and the intermediate regions each cover one fourth of the extent of the computational domain ($nx/4$ and $ny/4$ respectively). The integrated fluxes are the fluxes connecting those regions, excluding the fluxes within boundary regions.

3.14 b2yt, changing from one grid size and species set to another

The converter program `b2yt` can be used for a variety of purposes, and prospective users are urged to read the *entire* section below carefully before proceeding to use it. As a summary, however, the sequence of steps to be followed for a successful conversion is:

- 1 Create a new case directory: `$SOLPSWORK/<my_new_case>`.
- 2 In the new case directory, create subdirectories `baserun` and `convert` as well as a run subdirectory (`<my_run>`). `baserun` and `convert` are special names and should not be changed.
- 3 Copy from your old case to the new case's `convert` directory all input and output files you wish to have converted. You may copy `baserun` files that you know will not be converted to the new case's `baserun` directory instead.
- 4 Check and correct as necessary the relative ages of the input files, for which you can use the `correct_b2yt_timestamps` script.
- 5 Run the `b2yt` converter.
- 6 Displace the converted case input files to the converted case `baserun` and run directories, for which you can use the `import_files_from_convert` script.

The `b2yt` utility program allows a user to take a `b2fstate` solution file with size $nx \times ny$ and transform it into an $nx1 \times ny1$ `b2fstate` file, provided the numbers of real cells in both x and y directions in the new solution are divisors of the ones in the original mesh. Other files necessary to proceed with the run under the new geometry will also be produced. The two numbers $nx1$ and $ny1$ are provided by the user in the `b2yt.dat` file. The `b2yt.dat` file should contain, for example:

```
*dimens          (nx1, ny1; free format)
  8  8
'solps_version'  'iter'
'eirene_format'  'iter'
```

with any optional switches mentioned afterwards, here to create a solution on an 8×8 grid.

A solution for non-commensurate grids is also available, requiring instead that the topology of the two solutions be the same, and works by 2-D interpolation of the plasma data in the computational domain, by region. This is activated with the `'b2ytdr_non_commensurate'` option set to `'1'` in `b2yt.dat`. This option should also be used when the overall grid size remains the same but the location and/or depth of the cuts has changed, or when mapping a lower single-null (LSN) case to an upper single-null (USN) case or vice versa (or between USN-like and LSN-like limiter geometries). When switching between USN and LSN, the solution will be mirror-reflected to keep the inboard and outboard divertor solutions in the correct geometrical locations.

The conversion of the `Eirene` input file additionally requires for the converter program to know the original version of `Eirene` with which the input file was used, and this is provided by means of the `'eirene_format'` switch. Its possible values (case-insensitive) are:

"old" for input files from **SOLPS4.0** and **SOLPS5.0** runs using "old" `Eirene_96`

"new" for input files from **SOLPS4.0** and **SOLPS5.0** runs using "new" `Eirene_99`

"facelift" for input files from **SOLPS 5.1** runs

"juelich" for input files from Juelich `Eirene` versions (2008 and younger)

"iter" (default) for input files from **SOLPS4.2/4.3** and **SOLPS-ITER** runs

The `input.dat.2` file that the converter will create will follow the **SOLPS-ITER** format.

The conversion of the Eirene output file (`fort.44`) additionally requires for the converter program to know the original version of SOLPS with which the output file was created, and this is provided by means of the 'solps.version' switch. Its possible values (case-insensitive) are:

"4.3" for `fort.44` output files from **SOLPS4.3** runs

"5.0" for `fort.44` output files from **SOLPS5.0** runs

"5.1" for `fort.44` output files from **SOLPS 5.1** runs

"5.2" for `fort.44` output files from **SOLPS5.2** runs

"iter" (default) for input files from **SOLPS-ITER** runs (no conversion necessary)

The `fort.44.2` file that the converter will create will follow the **SOLPS-ITER** format.

One can also use `b2yt` to flip the sign of the poloidal and/or toroidal components of the magnetic field, and the code will make the appropriate adjustments in the input and data files. The switches to obtain this are:

"invert_bpol" "1" Inverts the poloidal field and related quantities

"invert_btor" "1" Inverts the toroidal field and related quantities

This is important because the parallel velocity is co-aligned with the magnetic field in **SOLPS-ITER**, which was not the case in earlier code versions (please consult Section C.1 for a full description of the sign conventions used in **SOLPS-ITER**). For instance, if you are converting from an older **SOLPS4.3** case (where the magnetic field components were always positive by default), you need to use these switches to adapt the plasma solution and the geometry files. This is done automatically for ITER cases when using the `convert_from_solps4.3` script.

`b2yt` may also be used to change the order of species (for example changing from $D + C + He$ to $D + He + C$) or for bundling, de-bundling, or re-bundling species, or even removing species and complete isonuclear sequences. It is however not possible to use this converter to *add* a new isonuclear sequence. When modifying the species data, the user should run `b2ar` with the new species arrangement before running `b2yt` to obtain a temporary file `b2frates`. After `b2yt` has been run successfully, then the user should re-run `b2ah` (using the produced `b2ah.dat.2` file, after renaming it `b2ah.dat`) and `b2ar` before re-launching the run.

If a new geometry is requested, the `b2yt` program should be run after the new geometry file ($nx1 \times ny1$) has been produced and is available in the `convert` or `baserun` directories as `b2fgmtry`. If converting a linear case, it is possible to move the location of the nominal separatrix on the new grid by means of the `b2ytdr_jsep1` switch in `b2yt.dat`. The old geometry file should also remain available in the `convert` directory as `b2fgmtry.old`. If this is a coupled case and the number of triangles for the Eirene grid remains the same, the new `fort.33`, `fort.34` and `fort.35` files corresponding to the new geometry should also be provided. If, however, the number of triangles between the old and new geometries changes, there is currently no conversion possible of the data in the `fort.46` file and it should not be made present in the `convert` directory.

The program also needs the `b2mn.dat` and any other used input files (such as `input.dat`, `b2.*.parameters` or `b2.*.inputfile`) from the original run.

It is important for proper `b2yt` execution that the necessary files have the expected seniority from the Makefile. Therefore, you must ensure that:

1. `b2fstate` is the *youngest* file.
2. `b2frates` (or `../baserun/b2frates`) is *older* than `b2fstate` but *younger* than `b2fgmtry`.
3. `b2fpardf` (or `../baserun/b2fpardf`) is *younger* than `b2ah.dat` and *older* than `b2frates`.
4. `b2fgmtry` must be *younger* than `b2ag.dat` and *older* than `b2fpardf`.

So, ALL of the following must be verified:

`b2ag.dat < b2fgmtry < b2fpardf < b2frates < b2fstate`

```
b2ah.dat < b2fpardf
```

```
b2ar.dat < b2frates
```

If this order is not met, use a series of "touch 'filename' " commands to give the files new dates. The best to check this is to run the command:

```
b2run -n b2yt
```

which will launch a dry-run of the Makefile invoking b2yt. Then, make sure that this dry-run only attempts to run b2yt and no other programs. If it does call, say b2ag and/or b2mn, then the age hierarchy above was not respected. In that case, you can run the script

```
correct_b2yt_timestamps
```

to get the correct ordering of files. Do check again afterwards with the dry-run command to make sure it worked.

Upon successful termination of b2yt, the user will find a b2fstatt file containing the transformed plasma state, as well as a b2mn.dat.2 file which will contain the updated run switches appropriate for the new geometry and species selection. Similarly, if b2.boundary.parameters and b2.neutrals.parameters files were present in the working directory, files b2.boundary.parameters.2 and b2.neutrals.parameters.2, respectively, will contain the updated namelist variables. The program will also create a new b2.wall_save.parameters.2 file containing the new wall namelist settings and a new b2favere.2 file containing the new averaged solution, when appropriate. Other B2.5 input files are treated in the same fashion. The code will also attempt to create a new SOLPS_FLUIDS.2 file appropriate for the new species list.

If an input.dat file is present in the run directory, the code will also attempt to produce a new input.dat.2 file. In order to ensure consistency between the B2.5 and Eirene modified inputs, it is important that the order of the neutrals boundary conditions in b2.neutrals.parameters and the strata in input.dat match! If the B2.5 grid has been changed, and a new Eirene triangular grid is needed, the triangulation will need to be redone. In order to do so, copy the converted input.dat.2 file to ../baserun/triang.eir, modify the copied file to set NLTRIMESH to .true., and then launch the triang script (in the baserun directory, skipping the (U) inp and (b) 2ag steps, i.e. starting at the (E) irene step.

When species are being re-bundled, possible conflicts may occur between their individual settings. In such cases, the code behaviour is to use the setting from the highest charge state within the bundle. If species are being removed, strata, tallies or diagnostics involving these species may either be removed from the input file or revert to their default settings. The user is encouraged to check that those choices are appropriate and modify them as needed.

Once the user has finished converting a case, a script exists, named import_files_from_convert, that will take the converted files from b2yt and place them, after proper renaming, in the current run directory, provided the convert directory where b2yt was run is located at the same level. Some of the files are put in the baserun directory if they would belong there, so it is always best, after the import, to execute correct_baserun_timestamps from the parent directory, followed by a touch b2fstati command in each of the run subdirectories.

Because of the complexity of the conversion process, for which not all possible cases have been tested, users are encouraged to inspect the b2yt output files visually after execution to make sure no mistakes remain. No attempt is made to adjust or modify the text labels pertaining to the various input files, so users should consider adapting those to the new case being built. Report of conversion errors or omissions to the developer team are always welcome.

3.15 b2co, recovering from a crash

In case of a computer crash during a run, it may be possible to recover the results and continue the simulation. The code writes, at regular intervals (default is 3600 CPU seconds, and can be modified with use of the 'b2mndr_cpu' and 'b2mndr_plasmatim' switches, see Appendix A.4 for details and additional options), recovery files entitled `plasmastate.XXXX` and `b2.*_save.parameters.XXXX` in the `b2mn.exe.dir` run subdirectory. The XXXX are numbers indicating the order in which the files were written. The `plasmastate` files contain the same information as the `b2fstati` or `b2fstate` files, but in a more compact format. In order to recover from a crash, the user should then:

1. Check for the presence of `plasmastate.XXXX` files in the `b2mn.exe.dir` run subdirectory
2. If present, check that the most recent file is of the same size as the others (sometimes the crash happens while writing the `plasmastate` file!)
3. You can check the plasma time to which the `plasmastate` file corresponds by means of the `inspect_plasmastate #1` script where #1 is the name of the `plasmastate` file you are considering to use
4. `setenv PLASMASTATE b2mn.exe.dir/plasmastate.XXXX` where XXXX is the number of the desired `plasmastate` file
5. `b2run b2co or gmake -f $SOLPSTOP/runs/Makefile b2co.prt` which will create a `b2fstate` file
6. `mv b2fstate b2fstati`
7. Re-launch the run

If the crash happened early on in the simulation and one wishes to return to the initial state of the simulation to re-launch the run cleanly, one may use the `recover_initial_files` script instead.

This procedure will produce the files necessary to re-start the run from the point at which the `plasmastate` file was written. In cases with feedback schemes active, the feedback values will be grabbed from the relevant `b2.*_save.parameters.XXXX` files and a new set of `b2.*_save.parameters` files will be created. Optional output files (i.e the NetCDF files `b2time.nc`, `b2tallies.nc`, `b2wall.nc`, and `b2movies.nc`, as well as the `*.trc` files in the `tracing` directory), if present, will be truncated to be synchronized with the recovered plasma state file.

It is also possible to use the `b2co` utility *while* SOLPS is running, in order to obtain a `b2fstate` file in the run directory that one may use to plot certain plasma quantities and monitor the current status of the run. However, to do this and not overwrite the current tracing files, `b2co` *must* be invoked as follows:

1. `setenv PLASMASTATE b2mn.exe.dir/plasmastate.XXXX` where XXXX is the number of the desired `plasmastate` file
2. `b2run -b b2co or gmake -f $SOLPSTOP/runs/Makefile b2co.prt B2FSTATE_ONLY='yes'` which will create the provisional `b2fstate` file

3.16 b2uf & b2fu, converting the plasma output file

The commands

```
b2run b2uf
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2uf.prt
```

will convert an 'unformatted' b2fplasma file into a 'formatted' b2fplasmf file. This file contains a full description of the plasma solution and derived quantities.

The reverse operation, converting back from b2fplasmf to b2fplasma, is obtained by means of the commands:

```
b2run b2fu
```

or

```
gmake -f $SOLPSTOP/runs/Makefile b2fu.prt
```

3.17 b2rd, fetching an archived MDSPLUS run

If a run has been archived into the MDSplus tree structure (see 6 for details) it is possible to extract from the tree the information necessary to obtain `b2fstate` and `b2fgmtry` files from which to continue the run. To do so, follow the procedure below:

1. Check for the presence of a `shotnumber.history` file in the run directory
2. If present, the fetched run will be the one with the run number specified last in `shotnumber.history`
3. If absent or incorrect, create or modify the `shotnumber.history` file to add an extra line at the end with the run number of the run you wish to retrieve
4. `b2run b2rd or gmake -f $SOLPSTOP/runs/Makefile.plot b2rd.prt` which will create a `b2fstate` file
5. `mv b2fstate b2fstati`
6. `mv b2fgmtrt b2fgmtry`
7. Re-launch the run

This procedure will re-start the run from the point at which the MDSplus file was written. However, the input files and run settings are not retrieved along with `b2fstate` (these are not stored) so the user must check the input data present in the run directory before re-launching a case. In cases with feedback schemes active, for example, the feedback values will be the ones stored in the appropriate namelists from the `b2.*.parameters` files in the run directory. Concerning the retrieval of the geometry case, some quantities must be recomputed by `b2rd` as they are not saved in MDSplus, so the code assumes a tokamak geometry and that the mesh is a basis mesh.

3.18 b2ymb, post processing for monoblock heat transfer model

This routine first reads the formatted file `b2ymb.dat` and writes the unformatted file `mnbl_in.dat` that is needed by the FreeFem++ (<http://www.freefem.org>) [3] heat transfer model of the monoblock, see the details in the manual in `$SOLPSTOP/doc/SOLPS-ITER_PFCs_heat_transfer_model.pdf`. Then, it reads the files `wl1d.trg[i-o].dat` produced by **b2plot** with the "1000 wl1d" command and writes the unformatted files `W.tg[i-o].dat`, `X.tg[i-o].dat` and `tg[i-o].size.dat` that are needed by the FreeFem++ heat transfer model of the monoblock. In `W.tg[i-o].dat` is stored the value of the heat flux for each tile of the monoblock in Watts. In `X.tg[i-o].dat` is stored the value of the poloidal coordinate of the center of each tile along the monoblock axis (in meters). In `tg[i-o].size.dat` is stored the number of points (i.e. tiles) in which the inner/outer monoblocks have been subdivided according to the assigned tile poloidal length, which is read from the `b2ymb.dat` file. To use the routine:

1. Define in the DG model the plot zone variables for inner and outer targets with labels `trgi` and `trgo` respectively
2. Create the file `b2ymb.dat` in the run directory (see below for an example)
3. echo "1000 wl1d" | `b2plot` will produce the `wl1d.trg[i-o].dat` files
4. Copy the `wl1d.trg[i-o].dat` files in the run directory
5. `b2plot b2ymb`

Example of the `b2ymb.dat` file:

```
&GEOMETRY

HEIGHT      =      28.0e-3,      ! tile height [m]
T_LENGTH    =      28.0e-3,      ! tile toroidal length [m]
CU_TH       =      1.0e-3,      ! Cu interlayer thickness [m]
CUCRZR_TH   =      1.5e-3,      ! CuCrZr tube thickness [m]
DIAMETER     =      12.0e-3,      ! CuCrZr tube internal diameter [m]
W_TH        =      6.0e-3,      ! W armour thickness (from plasma facing surface to Cu interlayer) [m]
P_LENGTH    =      1.2e-2,      ! tile poloidal length [m]
T_GAP       =              0.5e-3,      ! Toroidal gap between monoblocks [m]
BETA        =              1.0,      ! poloidal chamfer [degrees]
ALPHA       =              0.5,      ! Vertical target tilting [degrees]
THETA       =              4.2,      ! Toroidal strike angle of magnetic flux lines [degrees]
REVERSE     =      0,      ! Integer, if >0 reverse the heat flux profile from top to bottom
/

&MESH

NN          =      50,      ! # of mesh points on longest edge
/

&COOLANT

T_COOLANT   =      70.0,      !Coolant inlet temperature [C]
M_FLOW      =      0.734,      ! Coolant mass flow rate      [kg/s]
CP_COOLANT  =      4.22e3,      ! Coolant specific heat capacity [J/kg/K]
/

&OUTPUT

NPOINTS     =              2,      ! Number of points for output surface temperature
LPOINTS     =              0.,14.0e-3,      !Location of output surface temperature
NPLOTS      =              10,      !Plot the 2D profile each NPLOTS cuts
/
```

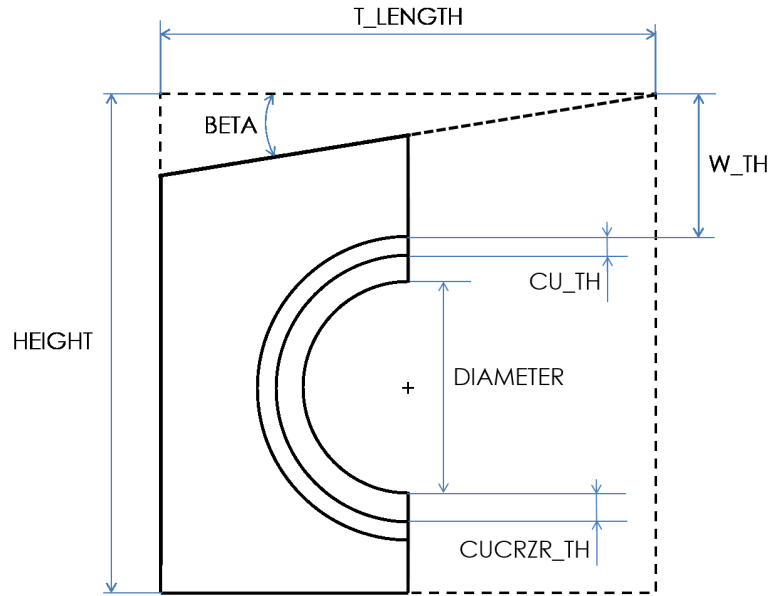



Figure 3.6: Monoblock tile cross section

3.19 b2yrp, post processing for reflector plates heat transfer model

This routine first reads the formatted file `b2yrp.dat` and writes the unformatted file `rps_in.dat` that is needed by the FreeFem++ (<http://www.freefem.org>) [3] heat transfer model of the reflector plates, see the details in the manual in `$SOLPSTOP/doc/SOLPS-ITER_PFCs_heat_transfer_model.pdf`. Then, it reads the files `wl1d_[i-o]rp.dat` produced by **b2plot** with the "1000 wl1d" command and writes the unformatted files `W_[i-o]rp.dat`, `X_[i-o]rp.dat` and `[i-o]rp_size.dat` that are needed by the FreeFem++ heat transfer model. In `W_[i-o]rp.dat` is stored the value of the heat flux for each tile of the RPs in Watts. In `X_[i-o]rp.dat` is stored the value of the poloidal coordinate of the center of each tile along the RPs axis (in meters). In `[i-o]rp_size.dat` is stored the number of points (i.e. tiles) in which the inner/outer RPs have been subdivided according to the assigned tile poloidal length, which is read from the `b2yrp.dat` file. To use the routine:

1. Define in the DG model the plot zone variables for inner and outer reflector plates with labels `irp` and `orp` respectively
2. Create the file `b2yrp.dat` in the run directory (see below for an example)
3. `echo "1000 wl1d" | b2plot` will produce the `wl1d_[i-o]rp.dat` files
4. Copy the `wl1d_[i-o]rp.dat` files in the run directory
5. `b2plot b2yrp`

Example of the `b2yrp.dat` file:

```
&GEOMETRY
L1      =      25.0e-3,      ! [m]
L2      =      17.0e-3,      ! [m]
L3      =      0.25e-3,      ! [m]
H1      =      4.0e-3,      ! [m]
H2      =      5.0e-3,      ! [m]
H3      =      7.0e-3,      ! [m]
H4      =      2.0e-3,      ! [m]
H5      =      8.0e-3,      ! [m]
H_HOLE  =      2.5e-3,      ! [m]
```

```

RH      =      1.5e-3,      ! [m]
P_LENGTH =      1.2e-2,      ! tile poloidal length [m]
T_GAP    =      1.0e-3,      ! Toroidal gap between hypervapotron cahnnels [m]
ALPHA    =      0.5,        ! Reflector plate tilting [degrees]
BETA     =      1.0,        ! poloidal chamfer [degrees]
THETA    =      4.2,        ! Toroidal strike angle of magnetic flux lines [degrees]
REVERSE  =      0,         ! Integer, if >0 reverse the heat flux profile from top to bottom
/

&MESH
NN      =      20,         ! \# of mesh points on H4
/

&COOLANT
T_COOLANT =      128.595,    !Coolant inlet temperature [C]
M_FLOW    =      0.734,      ! Coolant mass flow rate      [kg/s]
CP_COOLANT =      4.22e3,    ! Coolant specific heat capacity [J/kg/K]
/

&OUTPUT
NPOINTS   =      2,         ! Number of points for output surface temperature
LPOINTS   =      0.,14.0e-3, !Location of output surface temperature
NPLOTS    =      10,        !Plot the 2D profile each NPLOTS cuts
/

```

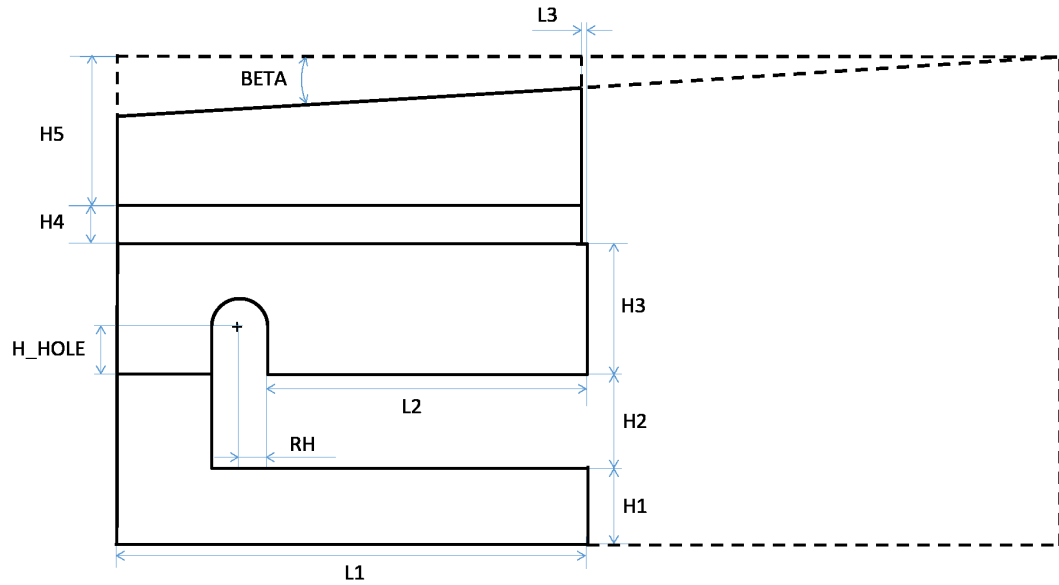


Figure 3.7: Reflector plate tile cross section

3.20 b2ydm, post processing of dome wall loading

This routine first reads the formatted file `b2ydm.dat`. Then, it reads the file `wl1d_dome.dat` produced by **b2plot** with the `"1000 wl1d"` command and writes the unformatted file `dome_out.dat` in which is stored the coolant temperature at the outlet of the dome hypervapotron cooling system. This temperature can be used as inlet temperature for the outer reflector plate heat transfer model. To use the routine:

1. Define in the DG model the plot zone variable for the dome
2. Create the file `b2ydm.dat` in the run directory (see below for an example)
3. `echo "1000 wl1d" | b2plot` will produce the `wl1d_dome.dat` file
4. Copy the `wl1d_dome.dat` file in the run directory
5. `b2plot b2ydm`

Example of the `b2ydm.dat` file:

```
&GEOMETRY
HEIGHT      =      28.0e-3,    ! tile height [m]
T_LENGTH    =      50.0e-3,    ! tile toroidal length [m]
P_LENGTH    =      1.2e-2,     ! tile poloidal length [m]
T_GAP       =      1.0e-3     ! Toroidal gap between hypervapotron cahnnels [m]
ALPHA       =      0.5,       ! Reflector plate tilting [degrees]
BETA        =      1.0,       ! poloidal chamfer [degrees]
THETA       =      4.2,       ! Toroidal strike angle of magnetic flux lines [degrees]
/

&COOLANT
T_COOLANT   =      70.0,      ! Coolant inlet temperature [C]
M_FLOW      =      0.734,     ! Coolant mass flow rate [kg/s]
CP_COOLANT  =      4.22e3,    ! Coolant specific heat capacity [J/kg/K]
/
```

3.21 b2_ual_write, writing last state to IDS

The program can be run with one of the following

```
gmake -f $SOLPSTOP/runs/Makefile b2_ual_write.prt
```

or

```
b2run b2_ual_write
```

This routine saves the final output state of a run into an IDS (Interface Data Structure). Currently, it will populate the `edge_profiles`, `edge_transport`, `edge_sources`, `radiation`, `dataset_description` and `summary data structures`.

The program reads the `b2mndr_id*` switch group, defined in `b2mn.dat`:

- `b2mndr_user` defines the user name and it is set by default to be the same as the `$USER` environment variable.
- `b2mndr_database` defines the IMAS IDS database name and it is set by default to be the same as the `$DEVICE` environment variable. If `DEVICE` has been left undefined, the default database name is `'solps-iter'`.
- `b2mndr_device` superseded by `b2mndr_database`
- `b2mndr_shot_number` defines the shot number
- `b2mndr_run_number` defines the run number

These switches are used as identification parameters for the IDS. Afterwards, a `b2mn` run is initialized but *not* started, so that the final output state data is read. Then the data is written to an IDS, with the `b2mndr_id*` switchgroup used as identification descriptors.

It is possible to replace (or supplement) the IDS switch settings coming from the `b2mn.dat` file by using the command line `-o` option, with the following syntax:

```
b2run -o "-s 123456 -r 1" b2_ual_write
```

which will recognize, within the quotes following the `-o` additional options opener, the switches:

- `-s, --shot` The shot number
- `-r, --run` The run number
- `-u, --user` The user name (default is `$USER`)
- `-d, --database, --device` The IMAS database name (default is `$DEVICE`, if set, or `"solps-iter"` otherwise)
- `-v, --version` The IMAS version to be used (default is 3)

Concerning the assignment of shot and run numbers, the following rules are in effect. The shot number is a six digit number between 1 and 214748. ITER shot numbers shall start with a 1, the second digit represents the majority fuelling species (0:H, 1:He, 2:D, 3:D-T, 4:T, 5:others, the sequence reflecting the order in which fuel gases are expected to be used in the ITER Research Plan), and the last four digits identify the simulation. Because multiple codes are all writing to the same database and we want to avoid clashes, different "slices" of numbers have been assigned for various codes. We will use numbers 1x2000 to 1x2999 for transferring existing simulations from the **SOLPS 4.3** database to IMAS, and 1x3000 to 1x3499 for new **SOLPS-ITER** cases. Run numbers should start at 1, and be incremented every time a new version of the same simulation (say after a re-run because a bug in the code was fixed) is being stored in the archive. When changing important run parameters such as gas puffing rates, turning on drifts, etc..., a new shot number should be used. Until a more automatic way of assigning numbers (and a better overall scheme) can be implemented, please send an email to the IO code responsible officer (Xavier.Bonnin@iter.org), when you wish to obtain a new shot number, as the IO will keep track of those already assigned. **IMPORTANT: When publishing SOLPS-ITER results with IO co-authors involved, archiving into the public IMAS database of the simulations on which the published work is based is mandatory.**

After the IDS files are created, one should also run the `create_db_entry` script, with the syntax:

```
create_db_entry [-u USER] [-d DATABASE] -s SHOT -r RUN
```

which will create `ids_*.yaml` and `ids_*.watcher` files for that shot and run number, to be used in the IMAS scenario database. These files should be edited by hand according to the instructions within them and the IMAS documentation.

3.22 b2xx, the others

Chapter 4

Running the standalone version of Eirene

Eirene is documented at <http://www.eirene.de/>. The user is strongly encouraged to check out that very complete manual. Reference [10] also provides a useful description.

```
sei
gmake depend
gmake listobj
gmake tags
gmake depend
gmake
```

See also the next chapter and section J.1 for setting the array sizes and useful file links for Eirene before compilation.

4.1 Specific SOLPS-ITER blocks in the Eirene input file

In addition to the very complete description of the Eirene input file in the Eirene online manual, the last part of the Eirene input file (blocks 14 and 15) is specific to SOLPS-ITER coupling. We therefore document its content here.

4.1.1 Block 14. Data for interfacing routine INFUSR

The data in this block is meant to allow communication between the B2.5 and Eirene halves of SOLPS-ITER .

The format of this input block for coupling to SOLPS-ITER is as follows:

****14. Data for interfacing routine "infusr"*

```
LSYMET  LBALAN  LCOARSE  NLTRIMESH
NFLA    NCUTB   NCUTL    IMF      NTRFRM   NFULL    IBRAD    IBPOL    IBTOR
```

read NPLSI species lines

```
DO 20, IPL=1,NPLSI
  I   IFLB  FCTE  BMASS  LKINDP
20 CONTINUE
```

Recycling target definitions

```
NDXA    NDYA
NTARGI
```

```

NTGPRT(IT), IT=1,NTARGI
DO 30, IT=1,NTARGI
  DO 33, IPRT=1,NTGPRT(IT)
    I NDT      NINCT  NIXY   NTIN   NTEN   NIFLG   NPTC   NPTCM
      NSPZI   NSPZE   NEMOD
  33 CONTINUE
30 CONTINUE

```

Short cycle parameters

```

CHGP   CHGEE   CHGEI   CHGMOM
NAINB  NCOPIB  NCOPEB
DO 40, IAIN=1,NAINB
  I  NAINS  NAINT
  TXTPLS(IAIN,12)
  TXTPSP(IAIN,12)  TXTPUN(IAIN,12)
40 CONTINUE
NAOTB
DO 50, IAOT=1,NAOTB
  I  NAOTS  NAOTT
50 CONTINUE

```

Meaning of the input variables for interfacing routine INFUSR

We only include here the variables or aspects thereof not already described in the `Eirene` manual, section 2.14.

LCOARSE Coarse mesh switch ???

NLTRIMESH When `.TRUE.`, `Eirene` is only run to obtain the contours necessary to build the triangular grid, which are then written in `fort.78`. This is used by the `triang` script toolchain. No particle trajectories are followed.

NCUTB For **SOLPS-ITER** couplings, should be 0

NCUTL For **SOLPS-ITER** couplings, should be 1

IMF `fort.30` format switch, should be 0

NTRFRM `fort.33` format switch, should be 0

LKINDP Index of the `Eirene` atom this species is linked to

NPTC In addition to the description in the `Eirene` manual, note that only the value from the first segment of the stratum (`IPRT=1`) is used.

NPTCM This corresponds to the `NMINPTS` flag in input block 7, and is needed only if `INDSRC=6`, i.e., if the source is specified by data from block 14 alone. As for `NPTC`, only the value from the first stratum segment is used.

NEMOD This corresponds to the `NEMODS` flag in input block 7, and is needed only if `INDSRC=6`, i.e., if the source is specified by data from block 14 alone. As for `NPTC`, only the value from the first stratum segment is used.

NCOPIB Not in use

NCOPEB Maximum number of additional tallies per plasma background species

4.1.2 Block 15. Data for interfacing routine GEOUSR

This input block is used to adjust the edges of the additional walls in the `Eirene` description of the vacuum vessel walls to match exactly the location of the grid corners from the plasma solver in `B2.5`. This block may begin with an optional text line containing either `GENERAL` or `BIASED_GARCHING`, which changes the format below. The

SOLPS-ITER default is to not include this text line. By default, the grid corners at the targets will be adjusted to the additional surfaces specified, and there should be one such additional surface per grid corner. Each additional surface can only serve for one grid corner adjustment, no more.

***15. Data for interfacing routine "geousr"

If there is no preceding text line, the reading format is:

```
NADMOD  NASMOD
DO I=1, NADMOD
  NRS    IPUNKT
  IF (IPUNKT.LT.0) XPOLPOS  YPOLPOS
ENDDO
DO I=1, NASMOD
  NAS    IPUNKT  NSSIR  NSSIP
ENDDO
```

If the text line is GENERAL, the reading format is:

```
NADMOD  NASMOD  NORMOD
DO I=1, NADMOD
  NRS    IPUNKT  XCOOR  YCOOR  ZCOOR
ENDDO
DO I=1, NASMOD
  NAS    IPUNKT  NSSIR  NSSIP
ENDDO
DO I=1, NORMOD
  IDIR  IR  IP
ENDDO
```

If the text line is BIASED_GARCHING, the reading format is:

```
DO I=1, max (NPPLG/3, 1) * 4
  IPUNKT  NRS
  IF (IPUNKT.LT.0) XPOLPOS  YPOLPOS
ENDDO
```

Meaning of the input variables for interfacing routine GEOUSR

NADMOD Number of additional surfaces to be modified (should be 2 per divertor target, and 2 for a limiter geometry case)

NASMOD Number of standard surfaces to be modified. Should be 0 for standard **SOLPS-ITER** couplings.

NORMOD Number of cell faces whose normal is to inverted. Should be 0 for standard **SOLPS-ITER** couplings.

NRS Index of the additional surface being modified

IPUNKT ABS (IPUNKT) is the index of the vertex of that additional surface being modified

XCOOR New X-coordinate of the point being modified (in cm). Not necessary in standard **SOLPS-ITER** couplings.

YCOOR New Y-coordinate of the point being modified (in cm). Not necessary in standard **SOLPS-ITER** couplings.

ZCOOR New Z-coordinate of the point being modified (in cm). Not necessary in standard **SOLPS-ITER** couplings.

XPOLPOS X-index (poloidal) of the grid cell to which the additional surface vertex must be matched

YPOLPOS Y-index (radial) of the grid cell to which the additional surface vertex must be matched

NAS Index of the standard surface to be modified

NSSIR Radial position of the standard surface to be modified

NSSIP Poloidal position of the standard surface to be modified

IDIR Direction of the face whose normal is being inverted (1 = radial, 2=poloidal)

IR Radial index of the cell where the normal is being inverted

IP Poloidal index of the cell where the normal is being inverted

4.1.3 Information for MPI

The standard way to choose the MPI parallelization strategy in *Eirene* is *via* the `NPRL` variable in block 1 of the input file. However, one can also use an optional block at the very end of the *Eirene* input file which will overwrite the `NPRL` setting. Its format is:

```
*** INFORMATION_FOR_MPI
```

```
AUTOMATIC_DISTRIBUTION
```

There are currently five possible choices supported. They are:

BALANCED A balanced strategy in which the workload is attempted to be divided evenly among processors. If the initial setup for the **BALANCED** strategy fails, the code then attempts the **APCAS** strategy. Also obtainable by setting `NPRL = 3`.

APCAS meaning *All Processors Compute All Strata*, where the work of each stratum is divided among all processors. Usually involves more communication between processors. Also obtainable by setting `NPRL = 2`.

ORIGINAL The original strategy implemented in versions of *Eirene* prior to **SOLPS-ITER** version 3.0.5. Also called “proportional allocation” in the *Eirene* manual. In that strategy, each processor is handed either part of a large stratum or one or several full smaller strata. This strategy works well if the number of processors is larger than the number of strata, but is not recommended if the number of strata is larger than the number of processors. Also obtainable by setting `NPRL = 1`.

EMBARRASS meaning the “embarrassingly parrallel” scheme by which each processor is assigned the same number of particle trajectories. This is the default in non-**SOLPS-ITER** versions of *EIRENE*. Also obtainable if setting `NPRL = 0` or `NPRL > 3`.

AUTOMATIC tells the code to use the current default, whatever it may be.

The format to read the strategy name is free-form. The input file is scanned for the words describing the chosen strategy after the title line `*** INFORMATION_FOR_MPI` is found. The *first* match to an (uppercase) strategy name will be the strategy followed by the code. If the title line is missing or no match is found to any of the strategy names, the default strategy is used. See the *Eirene_balanced_MPI* document on the **SOLPS-ITER** web site *Manuals and Documentation* folder for more details on the performance and algorithm corresponding to each strategy.

The default strategy depends on the version of the MPI library with which *Eirene* was compiled. If MPI version 3 or greater was used, then the default strategy is the **BALANCED** one. Otherwise, the **ORIGINAL** strategy is chosen. If *Eirene* is used outside the **SOLPS-ITER** environment, the default strategy will be the **EMBARRASSINGLY PARALLEL** one.

Chapter 5

Running the coupled version of B2.5-Eirene

In the run directory, you should execute the command

```
setup_baserun_eirene_links
```

to establish the links Eirene needs to the relevant atomic data files, as well as some links needed for diagnostic scripts. Using the "-r" option with this scripts creates relative links. Default behaviour is to create absolute links.

5.1 Input files

Below are given the input files used for the D+C+He benchmark case that can be found on the SVN server. The location of these is https://solps-mdsplus.aug.ipp.mpg.de/repos/SOLPS/trunk/solps_examples/solps5.1/bench_mark/AUG_16151_D+C+He/. Another version of these files can be found in the *Examples* folder in the **SOLPS-ITER** SharePoint web site (<https://portal.iter.org/deploy/ITER/>).

5.1.1 b2ag.dat

```
*dimens          (nx, ny; free format)
 96 36  96 36
*param           (param(0:99); free format)
-1.0, 0.0, 0.75, 1.0, -0.010, 0.0, 0.040,
 5.0, 1.0, 0.02, 0.05, 0.05, 0.05,
 0.06, 1.0,
100*0.0
'b2agfs_geometry'    '16151.4000.AUGD.EQI.00._96x36.geo'
'b2agfs_min_pitch'   '0.01'
```

5.1.2 b2ai.dat

```
*dimens          (ns; free format)
 12
*label           (lblps, character*60; free format)
'Flat initial state for D + C + He, all charge states'
*specs           za      zn      am      za2
'0'      0.0      1.0      2.0      0.0
'1'      1.0      1.0      2.0      1.0
'2'      0.0      6.0     12.011     0.0
'3'      1.0      6.0     12.011     1.0
'4'      2.0      6.0     12.011     4.0
'5'      3.0      6.0     12.011     9.0
```

```

'6'      4.0      6.0      12.011  16.0
'7'      5.0      6.0      12.011  25.0
'8'      6.0      6.0      12.011  36.0
'9'      0.0      2.0      4.003   0.0
'10'     1.0      2.0      4.003   1.0
'11'     2.0      2.0      4.003   4.0
*naini          (na0(0:ns-1); free format)
  1.0e12  1.0e20  7*1.0e16  3*6.0e16  100*1.0e0
*ttini          (tt0; free format)
  1.0e2

```

5.1.3 b2ar.dat

```

*tlahi          (tlo, thi; real, free format)
  1.0e-1  1.0e4
*nlohi          (nlo, nhi; real, free format)
  1.0e18  1.0e22
*numnuc          (nnuc; integer, free format)
  3
*nucspec          (nz, izlo, izhi; integers, one triplet per line)
  1  0  1
  6  0  6
  2  0  2
*flag            (either 'adpak' or 'strahl', free format)
  'strahl'
*tailep          (real, free format)
  0.08

```

5.1.4 b2ah.dat

```

*dimens          (ns; free format)
  12
*label            (lblps, character*60; free format)
  'Species D + C +He'
*b2cmpa          basic parameters
*specs           za      zn      am      za2
'0'      0.0      1.0      2.0      0.0
'1'      1.0      1.0      2.0      1.0
'2'      0.0      6.0      12.011  0.0
'3'      1.0      6.0      12.011  1.0
'4'      2.0      6.0      12.011  4.0
'5'      3.0      6.0      12.011  9.0
'6'      4.0      6.0      12.011  16.0
'7'      5.0      6.0      12.011  25.0
'8'      6.0      6.0      12.011  36.0
'9'      0.0      2.0      4.003   0.0
'10'     1.0      2.0      4.003   1.0
'11'     2.0      2.0      4.003   4.0
*b2cmpb          boundary conditions
*cbregs
'south'    3      0.25      0.75      2.00
'north'    1      2.0
'west'     1      2.0
'east'     1      2.0
*region
0  'south, private flux'
*cbsna      (0)      (1)      (2)      (3)      (4)      (5)      (6)      (7)
'0'      0      0      -1.0e-02  0      0      0      0      0
'1'      0      0      -1.0e-03  0      0      0      0      0
'2'      0      0      0      0      0      0      0      0
'3'      0      0      -1.0e-03  0      0      0      0      0
'4'      0      0      -1.0e-03  0      0      0      0      0
'5'      0      0      -1.0e-03  0      0      0      0      0

```

'6'	0	0	-1.0e-03	0	0	0	0	0
'7'	0	0	-1.0e-03	0	0	0	0	0
'8'	0	0	-1.0e-03	0	0	0	0	0
'9'	0	0	-1.0e-02	0	0	0	0	0
'10'	0	0	-1.0e-03	0	0	0	0	0
'11'	0	0	-1.0e-03	0	0	0	0	0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	-1.0	0
'1'	0	0	0	0	0	0	-1.0	0
'2'	0	0	0	-1.0e+10	0	0	-1.0	0
'3'	0	0	0	0	0	0	-1.0	0
'4'	0	0	0	0	0	0	-1.0	0
'5'	0	0	0	0	0	0	-1.0	0
'6'	0	0	0	0	0	0	-1.0	0
'7'	0	0	0	0	0	0	-1.0	0
'8'	0	0	0	0	0	0	-1.0	0
'9'	0	0	0	-1.0e+10	0	0	-1.0	0
'10'	0	0	0	0	0	0	-1.0	0
'11'	0	0	0	0	0	0	-1.0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'1'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'2'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'3'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'4'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'5'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'6'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'7'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'8'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'9'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'10'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'11'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	-1.0e-04	0	0	0	0	-2.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	1.0	0.3	0	0	0	0	0	0
'11'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*region								
1 'south, core plasma'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)

'0'	0	-1.0e+10	0	0	0	0	0	0
'1'	4.00e+29	-1.0e+10	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	-1.0e+10	0	0	0	0	0	0
'10'	0	-1.0e+10	0	0	0	0	0	0
'11'	1.00e+26	-1.0e+10	0	0	0	0	0	0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	0	0
'1'	0	0	0	-1.0e+10	0	0	0	0
'2'	0	0	0	-1.0e+10	0	0	0	0
'3'	0	0	0	-1.0e+10	0	0	0	0
'4'	0	0	0	-1.0e+10	0	0	0	0
'5'	0	0	0	-1.0e+10	0	0	0	0
'6'	0	0	0	-1.0e+10	0	0	0	0
'7'	0	0	0	-1.0e+10	0	0	0	0
'8'	0	0	0	-1.0e+10	0	0	0	0
'9'	0	0	0	-1.0e+10	0	0	0	0
'10'	0	0	0	-1.0e+10	0	0	0	0
'11'	0	0	0	-1.0e+10	0	0	0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	5.15e+03	0	0	0	0	0	0	-2.5
'1'	5.15e+03	0	0	0	0	0	0	-2.5
'2'	5.15e+03	0	0	0	0	0	0	-2.5
'3'	5.15e+03	0	0	0	0	0	0	-2.5
'4'	5.15e+03	0	0	0	0	0	0	-2.5
'5'	5.15e+03	0	0	0	0	0	0	-2.5
'6'	5.15e+03	0	0	0	0	0	0	-2.5
'7'	5.15e+03	0	0	0	0	0	0	-2.5
'8'	5.15e+03	0	0	0	0	0	0	-2.5
'9'	5.15e+03	0	0	0	0	0	0	-2.5
'10'	5.15e+03	0	0	0	0	0	0	-2.5
'11'	5.15e+03	0	0	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	5.15e+03	0	0	0	0	0	0	-2.5
*cbSCH	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0

'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*region								
2 'south, private flux'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	-1.0e-02	0	0	0	0	0
'1'	0	0	-1.0e-03	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	-1.0e-03	0	0	0	0	0
'4'	0	0	-1.0e-03	0	0	0	0	0
'5'	0	0	-1.0e-03	0	0	0	0	0
'6'	0	0	-1.0e-03	0	0	0	0	0
'7'	0	0	-1.0e-03	0	0	0	0	0
'8'	0	0	-1.0e-03	0	0	0	0	0
'9'	0	0	-1.0e-02	0	0	0	0	0
'10'	0	0	-1.0e-03	0	0	0	0	0
'11'	0	0	-1.0e-03	0	0	0	0	0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	-1.0	0
'1'	0	0	0	0	0	0	-1.0	0
'2'	0	0	0	-1.0e+10	0	0	-1.0	0
'3'	0	0	0	0	0	0	-1.0	0
'4'	0	0	0	0	0	0	-1.0	0
'5'	0	0	0	0	0	0	-1.0	0
'6'	0	0	0	0	0	0	-1.0	0
'7'	0	0	0	0	0	0	-1.0	0
'8'	0	0	0	0	0	0	-1.0	0
'9'	0	0	0	-1.0e+10	0	0	-1.0	0
'10'	0	0	0	0	0	0	-1.0	0
'11'	0	0	0	0	0	0	-1.0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'1'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'2'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'3'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'4'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'5'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'6'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'7'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'8'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'9'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'10'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'11'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	-1.0e-04	0	0	0	0	-2.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	1.0	0.3	0	0	0	0	0	0
'11'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0

'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0

*region
3 'north, SOL/DIV outer wall'

*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	-1.0e-02	0	0	0	0	0
'1'	0	0	-1.0e-03	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	-1.0e-03	0	0	0	0	0
'4'	0	0	-1.0e-03	0	0	0	0	0
'5'	0	0	-1.0e-03	0	0	0	0	0
'6'	0	0	-1.0e-03	0	0	0	0	0
'7'	0	0	-1.0e-03	0	0	0	0	0
'8'	0	0	-1.0e-03	0	0	0	0	0
'9'	0	0	-1.0e-02	0	0	0	0	0
'10'	0	0	-1.0e-03	0	0	0	0	0
'11'	0	0	-1.0e-03	0	0	0	0	0

*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	-1.0e+10	0	0	-1.0	0
'1'	0	0	0	0	0	0	-1.0	0
'2'	0	0	0	-1.0e+10	0	0	-1.0	0
'3'	0	0	0	0	0	0	-1.0	0
'4'	0	0	0	0	0	0	-1.0	0
'5'	0	0	0	0	0	0	-1.0	0
'6'	0	0	0	0	0	0	-1.0	0
'7'	0	0	0	0	0	0	-1.0	0
'8'	0	0	0	0	0	0	-1.0	0
'9'	0	0	0	-1.0e+10	0	0	-1.0	0
'10'	0	0	0	0	0	0	-1.0	0
'11'	0	0	0	0	0	0	-1.0	0

*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'1'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'2'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'3'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'4'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'5'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'6'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'7'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'8'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'9'	1.00e-18	-100.0	-1.0e-01	0	0	0	0	-2.5
'10'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5
'11'	1.00e-18	-100.0	-1.0e-02	0	0	0	0	-2.5

*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	-1.0e-04	0	0	0	0	-2.5

*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0

*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	1.0	0.3	0	0	0	0	0	0
'11'	1.0	0.3	0	0	0	0	0	0

*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*region								
4 'west, target plate'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	-1.0
'1'	0	0	0	0	0	0	0	-1.0
'2'	0	0	0	0	0	0	0	-1.0
'3'	0	0	0	0	0	0	0	-1.0
'4'	0	0	0	0	0	0	0	-1.0
'5'	0	0	0	0	0	0	0	-1.0
'6'	0	0	0	0	0	0	0	-1.0
'7'	0	0	0	0	0	0	0	-1.0
'8'	0	0	0	0	0	0	0	-1.0
'9'	0	0	0	0	0	0	0	-1.0
'10'	0	0	0	0	0	0	0	-1.0
'11'	0	0	0	0	0	0	0	-1.0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	-1.0e+10	0	0	0
'1'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'2'	0	0	0	0	-1.0e+10	0	0	0
'3'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'4'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'5'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'6'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'7'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'8'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'9'	0	0	0	0	-1.0e+10	0	0	0
'10'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'11'	0	0	1.0e+10	0	-1.0e+10	0	0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'2'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'3'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'4'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'5'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'6'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'7'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'8'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'9'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'10'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'11'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	1.0	0.5
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0

'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	1.0	0.3	0	0	0	0	0	0
'11'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*region								
5 'east, target plate'								
*cbsna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	-1.0
'1'	0	0	0	0	0	0	0	-1.0
'2'	0	0	0	0	0	0	0	-1.0
'3'	0	0	0	0	0	0	0	-1.0
'4'	0	0	0	0	0	0	0	-1.0
'5'	0	0	0	0	0	0	0	-1.0
'6'	0	0	0	0	0	0	0	-1.0
'7'	0	0	0	0	0	0	0	-1.0
'8'	0	0	0	0	0	0	0	-1.0
'9'	0	0	0	0	0	0	0	-1.0
'10'	0	0	0	0	0	0	0	-1.0
'11'	0	0	0	0	0	0	0	-1.0
*cbsmo	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	-1.0e+10	0	0	0
'1'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'2'	0	0	0	0	-1.0e+10	0	0	0
'3'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'4'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'5'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'6'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'7'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'8'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'9'	0	0	0	0	-1.0e+10	0	0	0
'10'	0	0	1.0e+10	0	-1.0e+10	0	0	0
'11'	0	0	1.0e+10	0	-1.0e+10	0	0	0
*cbshi	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'2'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'3'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'4'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'5'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'6'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'7'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'8'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'9'	1.00e-18	-100.0	0	0	0	0	0	-1.5
'10'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
'11'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbshe	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.00e-18	-100.0	0	-0.1	0	0	0	-1.5
*cbsch	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	1.0	0.5
*cbrec	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)

'0'	0	0	0	0	0	0	0	0
'1'	1.0	0.3	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	1.0	0.3	0	0	0	0	0	0
'11'	1.0	0.3	0	0	0	0	0	0
*cbmsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cbmsb	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*b2cmpt	transport coefficients							
*cfd0	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	1.0e+02	0	1.0	5.0e-20	1.5e-18	0	0	0
'1'	0.5	0	0	0	0	0	0	0
'2'	1.0e+02	0	1.0	2.0e-20	5.0e-20	0	0	0
'3'	0.5	0	0	0	0	0	0	0
'4'	0.5	0	0	0	0	0	0	0
'5'	0.5	0	0	0	0	0	0	0
'6'	0.5	0	0	0	0	0	0	0
'7'	0.5	0	0	0	0	0	0	0
'8'	0.5	0	0	0	0	0	0	0
'9'	1.0e+02	0	1.0	5.0e-20	1.5e-18	0	0	0
'10'	0.5	0	0	0	0	0	0	0
'11'	0.5	0	0	0	0	0	0	0
*cfdna	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	1.0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	1.0	0	0	0	0	0
'4'	0	0	1.0	0	0	0	0	0
'5'	0	0	1.0	0	0	0	0	0
'6'	0	0	1.0	0	0	0	0	0
'7'	0	0	1.0	0	0	0	0	0
'8'	0	0	1.0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	1.0	0	0	0	0	0
'11'	0	0	1.0	0	0	0	0	0
*cfdpa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	1.0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	1.0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
*cfvla	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	0	0	0	0	0	0
'1'	0	0	0	0	0	0	0	0
'2'	0	0	0	0	0	0	0	0
'3'	0	0	0	0	0	0	0	0
'4'	0	0	0	0	0	0	0	0
'5'	0	0	0	0	0	0	0	0
'6'	0	0	0	0	0	0	0	0
'7'	0	0	0	0	0	0	0	0
'8'	0	0	0	0	0	0	0	0
'9'	0	0	0	0	0	0	0	0
'10'	0	0	0	0	0	0	0	0
'11'	0	0	0	0	0	0	0	0
*cfvsa	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.0	0	0	0	0	0
'1'	0	0	1.0	0	0	0	0	0
'2'	0	0	1.0	0	0	0	0	0
'3'	0	0	1.0	0	0	0	0	0
'4'	0	0	1.0	0	0	0	0	0
'5'	0	0	1.0	0	0	0	0	0
'6'	0	0	1.0	0	0	0	0	0
'7'	0	0	1.0	0	0	0	0	0
'8'	0	0	1.0	0	0	0	0	0
'9'	0	0	1.0	0	0	0	0	0
'10'	0	0	1.0	0	0	0	0	0
'11'	0	0	1.0	0	0	0	0	0
*cfhci	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'0'	0	0	1.4	0	0	0	0	0
'1'	0	0	1.4	0	0	0	0	0
'2'	0	0	1.4	0	0	0	0	0
'3'	0	0	1.4	0	0	0	0	0
'4'	0	0	1.4	0	0	0	0	0
'5'	0	0	1.4	0	0	0	0	0
'6'	0	0	1.4	0	0	0	0	0
'7'	0	0	1.4	0	0	0	0	0
'8'	0	0	1.4	0	0	0	0	0
'9'	0	0	1.4	0	0	0	0	0
'10'	0	0	1.4	0	0	0	0	0
'11'	0	0	1.4	0	0	0	0	0
*cfhce	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0.7	0	0	0	0	0	0	0
*cfsig	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	1.0e-07	0	0	0	0	0	0	0
*cfalf	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
'-1'	0	0	0	0	0	0	0	0
*cflim	cflme	cflmi	cflmv	cflal	cflab	...		
'-1'	0.15	0.15	0.5	0.5	0.5	0	0	0
*endphy								

5.1.5 b2mn.dat

```

*label          (lblmn: character*60)
'D+C+He 15190 3.5s L-mode SNU'
*b2cmpa  basic parameters
*b2cmpb  boundary conditions
*b2cmpt  transport coefficients
*cflim   cflme   cflmi   cflmv   cflal   cflab   ...
'-1'    0.3    10.0    0.5    0.5    0.5    0      0      0
*endphy
'b2mndr_ntim'    '100'
'b2mndr_dtim'    '1.0e-4'
'b2mndt_nstg0'   '1'
'b2mndt_nstg1'   '1'
'b2mndt_nstg2'   '10'
'b2mndt_rxf'     '0.5'

```

```

'b2mndr_mvinc'      '1'
'b2mndr_mvnum'      '0'
'b2mndr_b2time'     '1'
'b2mndr_cpu'        '82800.0'
'b2mndr_savecpu'    '0.0'
'tallies_netcdf'     '1'
'b2ux5p_style'      '2'
'b2ux5p_nltrsol'    '2'
'b2stbc_sna0ep'     '1.0e-10'
'b2trcl_lluciani'   '1'
'b2trcl_lvis21'     '0'
'b2trcl_lthf21'     '0'
'b2news_facdrift_target' '0'
'b2news_facdrift_start' '0'
'b2news_facdrift_inc'  '1.00'
'b2news_facdrift_dec'  '0.5'
'b2news_potit'       '100'
'b2news_potok'       '1e-2'
'b2sihs_phm3'        '1.0'
'b2stbr_sput_frc'    '0.0100'
'b2stbr_sput_phys'   '1.00'
'b2stbr_sput_chem_model' '1'
'b2stbr_sput_dst'    '2'
'b2stbc_boundary_namelist' '1'
'b2srdt_numerics_namelist' '1'
'b2tqna_transport_namelist' '1'
'b2tqna_new_df0'     '0'
'b2mndr_eirene'      '1'
'eirene_repeat_first_call' '1'
'b2mndr_rescale_neutrals' '1e-0'
'b2mndr_rescale_neutrals_sources' '1e-10'
'b2mwti_jxi'        39
'b2mwti_jxa'        54
'b2news_recalculate_contributions' '0'
'eirene_savef31'     '0'
'eirene_dpc_fix'     '1'

```

To activate a coupled run, the master switch is 'b2mndr_eirene' which should be set to '1'. In addition, the densities of the fluid neutral species need to be scaled down, along with the sources that affect them. This is achieved, respectively, by use of the switches 'b2mndr_rescale_neutrals' and 'b2mndr_rescale_neutral_sources'. Typically, it is recommended to use a value of the order of 10^{-10} for scaling down the fluid neutral sources. For the fluid neutral densities, one can scale them down only on the first run, by the same amount as the sources, and then turn off the rescaling, as they are no longer being replenished.

In order to attempt to improve convergence of coupled **SOLPS-ITER** runs, currently three averaging schemes are available to reduce the impact of the Monte Carlo noise intrinsic to the **Eirene** sources. The user may want to activate one of these schemes, although there is no firm recommendation as to which will perform best. The switches for those are:

eirene_neutr_avg The neutral sources are averaged over the last up to *neutr_avg* **Eirene** calls. After *neutr_avg* calls, the history is cleared and the average arrays are rebuilt progressively again (ratcheting average scheme).

eirene_underrelax The neutral sources are multiplied by an under-relaxation factor of $1/eirene\_underrelax$ and added to the sources from the previous timestep (under-relaxation scheme).

b2stbr_eir_src_nhist The neutral sources are averaged over the last *nhist* **Eirene** calls (moving average scheme).

In addition to those, please see the recommended averaging scheme described in Section 5.4.

5.1.6 b2.boundary.parameters

```

&boundary
NBC=6,

```

```

BCCHAR=      'S',      'S',      'S',      'N',      'W',      'E',
BCPOS=       -1,       -1,       -1,       36,       -1,       96,
BCSTART=      24,        0,       72,        0,       -1,       -1,
BCEND=       71,       23,       95,       95,       36,       36,
BCENE=        8,        9,        9,        9,        3,        3,
BCENI=        8,        9,        9,        9,        3,        3,
BCPOT=        0,        0,        0,        0,        3,        3,
ENEPAR=  0.80e6,  0.01,  0.01,  0.01,  0.90,  0.90,  ENEPAR(2,2)=1e2, 1e2, 1e2,
ENIPAR=  0.80e6,  0.01,  0.01,  0.01,  2.50,  2.50,  ENIPAR(2,2)=1e2, 1e2, 1e2,
POTPAR=    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,  POTPAR(1,2)=6*0,
BCCON(0,1)   =      8,      1,      8,      8,      8,      8,      8,      8,      8,      8,      8,      1,
  CONPAR(0,1,1) = 0e19,  2e19,  0e19,  0e19,  0e19,  0e19,  0e19,  0e19,  0e19,  0e19,  0e19,  1e16,
BCCON(0,2)   =     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,
  CONPAR(0,2,1) = -1e-2, -1e-3, -1e-2, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-2, -1e-3, -1e-3,
BCCON(0,3)   =     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,
  CONPAR(0,3,1) = -1e-2, -1e-3, -1e-2, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-2, -1e-3, -1e-3,
BCCON(0,4)   =     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,     10,
  CONPAR(0,4,1) = -1e-2, -1e-3, -1e-2, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-3, -1e-2, -1e-3, -1e-3,
BCCON(0,5)   =      5,      3,      5,      3,      3,      3,      3,      3,      3,      5,      3,      3,
  CONPAR(0,5,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
BCCON(0,6)   =      5,      3,      5,      3,      3,      3,      3,      3,      3,      5,      3,      3,
  CONPAR(0,6,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
BCMOM(0,1)   =      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,
  MOMPARG(0,1,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
  mompar(0,1,2)=12*1e5,
BCMOM(0,2)   =      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,
  MOMPARG(0,2,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
  mompar(0,2,2)=12*1e5,
BCMOM(0,3)   =      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,
  MOMPARG(0,3,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
  mompar(0,3,2)=12*1e5,
BCMOM(0,4)   =      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,      4,
  MOMPARG(0,4,1) = 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
  mompar(0,4,2)=12*1e5,
BCMOM(0,5)   =      1,      3,      1,      3,      3,      3,      3,      3,      3,      1,      3,      3,
  MOMPARG(0,5,1) = 0.00,  1.00,  0.00,  1.00,  1.00,  1.00,  1.00,  1.00,  1.00,  0.00,  1.00,  1.00,  1.00,
  mompar(0,5,2)=12*0,
BCMOM(0,6)   =      1,      3,      1,      3,      3,      3,      3,      3,      3,      1,      3,      3,
  MOMPARG(0,6,1) = 0.00,  1.00,  0.00,  1.00,  1.00,  1.00,  1.00,  1.00,  1.00,  0.00,  1.00,  1.00,  1.00,
  mompar(0,6,2)=12*0,
GAMMAI=1.666666667, GAMMAE=0.5,
LBNDUSR=f,
/

```

5.1.7 b2.numerics.parameters

```

&numerics
dtco(0,1)= 12*1.0, dtco(0,2)=12*1.0, dtco(0,3)=12*1.0, dtco(0,4)=12*1.0,
dtmo(0,1)= 12*1.0, dtmo(0,2)=12*1.0, dtmo(0,3)=12*1.0, dtmo(0,4)=12*1.0,
dtee(1)= 1.0, 1.0, 1.0, 1.0,
dtei(1)= 1.0, 1.0, 1.0, 1.0,
/

```

5.1.8 b2.transport.parameters

```

&transport
flag_dna=1, parm_dna=12*0.4,
flag_dpa=0, parm_dpa=12*0,
flag_vla=0, parm_vla=12*0,
flag_vsa=0, parm_vsa=12*0,
flag_hci=1, parm_hci=12*1.6,
flag_hce=1, parm_hce=1.6,
flag_sig=0, parm_sig=0,

```

```
flag_alf=0, parm_alf=0,
/
```

5.1.9 b2.neutrals.parameters

```
&NEUTRAIS
NSTRAI=8,
CRCSTRA= 'W','E','S','S','N','V','V','V','T',
RCPOS=    -1, 96, -1, -1, 36,  0,  0,  0,  0,
RCSTART=   0,  0,  0, 72,  0,  0,  0,  0,  0,
RCEND=    35, 35, 23, 95, 95,  0,  0,  0,  0,
SPECIES_START= 0, 0, 0, 0, 0, 0, 2, 9, 0,
SPECIES_END=  11,11,11,11,11, 1, 8,11,11,
recyc(0,1)=2*1.00,7*0.00,3*1.00,
recyc(0,2)=2*1.00,7*0.00,3*1.00,
recyc(0,3)=2*1.00,7*0.00,3*1.00,
recyc(0,4)=2*1.00,7*0.00,3*1.00,
recyc(0,5)=2*1.00,7*0.00,3*1.00,
recyc(0,6)=2*1.00,7*0.00,3*1.00,
recyc(0,7)=2*1.00,7*0.00,3*1.00,
recyc(0,8)=2*1.00,7*0.00,3*1.00,
recyc(0,9)=2*1.00,7*0.00,3*1.00,
userfluxparm(1,1)=9*0,
userfluxparm(1,2)=9*0,
volrecinc=1.1,
chemical_sputter_yield(1)=200*0.01,
chemical_sputter_yield(201)=6*0.0,
chemical_sputter_yield(202)=0.01,
chemical_sputter_yield(203)=0.01,
chemical_sputter_yield(204)=0.00,
chemical_sputter_yield(205)=0.00,
chemical_sputter_yield(206)=0.00,
fchar_chemical=1,
igass_chemical=2,
itsput_chemical=1,
issput_chemical=1206,
EIRENE_MOD=1,
eirene_step_dt=1e-0,
eirene_step_cpu=120.0,
l_neutrad=1,
lmolscl(1)=1,
mlcmp(1,1)=2,
targsp(1:5,1)=5*2,
phys_sput(0:11,1:5)=60*1.00,
chem_sput(0:1,1:5)=10*0.010,
track_index(0:11)=2*0,7*1,3*0,
track_chem_sput(1)=.TRUE.,.FALSE.,
/
```

In this file the user defines many parameters that are used for communication between the B2.5 and Eirene parts of the code. Please see Appendix A.5 for full details. In particular, it is essential that the neutrals boundary conditions declared here match, in number and in order, the strata declared in the Eirene input file.

Some differences between the **SOLPS-ITER** and **SOLPS5.0** behaviours should be noted here.

First, the number of strata NSTRAI need no longer include the time-dependent "T" stratum.

Second, the array sizes of chemical_sputter_yield, is used, now need to match the actual number of wall and non-standard surfaces NLIM and NSTS as they are found in the \$Eirene input file. Non-zero values should only be assigned to actual walls (so the 200 in the above example corresponds to the real number of walls declared in block 3B of the Eirene input.dat file used for the same case) and to the non-standard surfaces (block 3A) corresponding to the targets. Note that the surface numbering considers block 3B first, and then block 3A, so non-standard surfaces will have indices varying between NLIM+1 and NLIM+NSTS. Moreover, Eirene adds a last non-standard (absorbing) surface that is used as a "time boundary". If the Eirene input file is being produced by use of the Uinp utility and/or

the `triang` script, then a `b2.neutrals.parameters.2` file meeting these criteria will be created, provided a `b2.neutrals.parameters` file containing any other data not deductible from the DG model is present.

Third, in **SOLPS-ITER**, the North and South boundaries of the B2.5 grid are no longer recycling/sputtering surfaces but rather only switching surfaces to the triangular grid regions.

5.2 Output files

When running the coupled version of **SOLPS-ITER**, you will obtain additional output files, named `fort.44` and `fort.46`, which contain data computed from `Eirene`. The contents of the `fort.44` file is a series of data blocks, each starting with a title line, after a header providing the grid size and species list, as well as a format identifier. The relevant dimensions are (please also refer to the `Eirene` manual for further details and to the code for the actual way the arrays are written out):

NDX Number of poloidal cells in the `Eirene` computational grid

NDY Number of radial cells in the `Eirene` computational grid

NATM Number of atom species declared in the `Eirene` input file

NMOL Number of molecule species declared in the `Eirene` input file

NION Number of test ion species declared in the `Eirene` input file

NPLS Number of plasma background species declared in the `Eirene` input file

NSTRA Number of source strata declared in the `Eirene` input file (Stratum 0 is the sum over strata)

NLIM Number of “additional surfaces” (physical walls) declared in block 3B of the `Eirene` input file

NSTS Number of “non-standard surfaces” (plasma boundaries) declared in block 3A of the `Eirene` input file

NTRII Number of triangles forming the `Eirene` computational grid

NCL Number of segments for resolved tallies on non-standard surfaces

The data blocks are:

dab2 Atom density (m^{-3}), dimension (NDX,NDY,NATM)

tab2 Atom temperature (eV), dimension (NDX,NDY,NATM)

dmb2 Molecule density (m^{-3}), dimension (NDX,NDY,NMOL)

tmb2 Molecule temperature (eV), dimension (NDX,NDY,NMOL)

dib2 Test ion density (m^{-3}), dimension (NDX,NDY,NION)

tib2 Test ion temperature (eV), dimension (NDX,NDY,NION)

rfluxa Radial flux density of atoms ($m^{-2}.s^{-1}$), dimension (NDX,NDY,NATM)

rfluxm Radial flux density of molecules ($m^{-2}.s^{-1}$), dimension (NDX,NDY,NMOL)

pfluxa Poloidal flux density of atoms ($m^{-2}.s^{-1}$), dimension (NDX,NDY,NATM)

pfluxm Poloidal flux density of molecules ($m^{-2}.s^{-1}$), dimension (NDX,NDY,NMOL)

refluxa Radial energy flux density carried by atoms ($W.m^{-2}$), dimension (NDX,NDY,NATM)

refluxm Radial energy flux density carried by molecules ($W.m^{-2}$), dimension (NDX,NDY,NMOL)

pefluxa Poloidal energy flux density carried by atoms ($W.m^{-2}$), dimension (NDX,NDY,NATM)

pefluxm Poloidal energy flux density carried by molecules ($W.m^{-2}$), dimension (NDX,NDY,NMOL)

emiss H_α emissivity due to atoms ($photons.m^{-3}.s^{-1}$), dimension (NDX,NDY)

emissmol H_α emissivity due to molecules and molecular ions ($photons.m^{-3}.s^{-1}$), dimension (NDX,NDY)

srcml Molecule particle source (A), dimension (NDX,NDY,NMOL)

edissml Energy spent for dissociating hydrogenic molecules (W), dimension (NDX,NDY,NMOL)

wldnek Heat transferred by neutrals (W), total over strata, dimension (NLIM+NSTS)

wldnep Potential energy released by neutrals (W), total over strata, dimension (NLIM+NSTS)

wldna Flux of atoms impinging on surface (A), total over strata, dimension (NLIM+NSTS,NATM)

ewlda Average energy of impinging atoms on surface (eV), total over strata, dimension (NLIM+NSTS,NATM)

wldnm Flux of molecules impinging on surface (A), total over strata, dimension (NLIM+NSTS,NMOL)

ewldm Average energy of impinging molecules on surface (eV), total over strata, dimension (NLIM+NSTS,NMOL)

p1,p2 Endpoints of surface (X and Y coordinates, in m), total over strata, dimension (NLIM)

wldra Flux of reflected atoms from surface (A), total over strata, dimension (NLIM+NSTS,NATM)

wldrm Flux of reflected molecules from surface (A), total over strata, dimension (NLIM+NSTS,NMOL)

For each stratum $i = 1, nstra$ The above data, resolved per stratum:

- wldnek(i)** Heat transferred by neutrals (W), dimension (NLIM+NSTS)
- wldnep(i)** Potential energy released by neutrals (W), dimension (NLIM+NSTS)
- wldna(i)** Flux of atoms impinging on surface (A), dimension (NLIM+NSTS,NATM)
- ewlda(i)** Average energy of impinging atoms on surface (eV), dimension (NLIM+NSTS,NATM)
- wldnm(i)** Flux of molecules impinging on surface (A), dimension (NLIM+NSTS,NMOL)
- ewldm(i)** Average energy of impinging molecules on surface (eV), dimension (NLIM+NSTS,NMOL)
- wldra(i)** Flux of reflected atoms from surface (A), dimension (NLIM+NSTS,NATM)
- wldrm(i)** Flux of reflected molecules from surface (A), dimension (NLIM+NSTS,NMOL)

wldpp Flux of plasma ions impinging on surface (A), total over strata, dimension (NLIM+NSTS,NPLS)

wldpa Net flux of atoms emitted from surface (A), total over strata, dimension (NLIM+NSTS,NATM)

wldpm Net flux of molecules emitted from surface (A), total over strata, dimension (NLIM+NSTS,NMOL)

wldpeb Power carried by particles emitted from surface (W), total over strata, dimension (NLIM+NSTS)

wldspt Flux of sputtered wall material (A), total over strata, dimension (NLIM+NSTS)

wldspta Flux of sputtered wall material per atom (A), total over strata, dimension (NLIM+NSTS,NATM)

wldsptm Flux of sputtered wall material per molecule (A), total over strata, dimension (NLIM+NSTS,NMOL)

For each stratum $i = 1, nstra$ The above data, resolved per stratum:

- wldpp(i)** Flux of plasma ions impinging on surface (A), dimension (NLIM+NSTS,NPLS)
- wldpa(i)** Net flux of atoms emitted from surface (A), dimension (NLIM+NSTS,NATM)
- wldpm(i)** Net flux of molecules emitted from surface (A), dimension (NLIM+NSTS,NMOL)
- wldpeb(i)** Power carried by particles emitted from surface (W), dimension (NLIM+NSTS)
- wldspt(i)** Flux of sputtered wall material (A), dimension (NLIM+NSTS)
- wldspta(m)** Flux of sputtered wall material per atom (A), dimension (NLIM+NSTS,NATM)

wldsptm(m) Flux of sputtered wall material per molecule (A), dimension (NLIM+NSTS,NMOL)

isrftype ILLIIN surface type variable in Eirene, dimension (NLIM+NSTS)

wlarea Surface area (m^2), dimension (NLIM+NSTS)

wlabsrp(A) Absorption rate for atoms, dimension (NATM, NLIM+NSTS)

wlabsrp(M) Absorption rate for molecules, dimension (NMOL, NLIM+NSTS)

wlabsrp(I) Absorption rate for test ions, dimension (NION, NLIM+NSTS)

wlabsrp(P) Absorption rate for plasma ions, dimension (NPLS, NLIM+NSTS)

wlpump(A) Pumped flux per atom (A), dimension (NATM, NLIM+NSTS)

wlpump(M) Pumped flux per molecule (A), dimension (NMOL, NLIM+NSTS)

wlpump(I) Pumped flux per test ion (A), dimension (NION, NLIM+NSTS)

wlpump(P) Pumped flux per plasma ion (A), dimension (NPLS, NLIM+NSTS)

eneutrad Radiation rate due to atoms (W), dimension (NDX,NDY,NATM)

emolrad Radiation rate due to molecules (W), dimension (NDX,NDY,NMOL)

eionrad Radiation rate due to test ions (W), dimension (NDX,NDY,NION)

eirdiag_nds Indices for segments on resolved non-standard surfaces, dimension ($5 \times \text{NSTS} + 1$)

sarea_res Surface area of surface segment (m^2), dimension (NCL)

wldna_res Flux of atoms impinging on surface segment (A), dimension (NATM, NCL)

wldnm_res Flux of molecules impinging on surface segment (A), dimension (NMOL, NCL)

ewlda_res Average energy of impinging atoms on surface segment (eV), dimension (NATM, NCL)

ewldm_res Average energy of impinging molecules on surface segment (eV), dimension (NMOL, NCL)

ewldea_res Energy flux carried by emitted atoms from surface segment (W), dimension (NATM, NCL)

ewldem_res Energy flux carried by emitted molecules from surface segment (W), dimension (NMOL, NCL)

ewldrp_res Total energy flux carried by emitted particles from surface segment (W), dimension (NCL)

ewldmr_res Flux of emitted molecules from recycling atoms (A), dimension (NMOL, NCL)

wldspt_res Flux of sputtered wall material (A), dimension (NCL)

wldspta_res Flux of sputtered wall material per atom (A), dimension (NCL, NATM)

wldsptm_res Flux of sputtered wall material per molecule (A), dimension (NCL, NMOL)

wlpump_res(A) Pumped flux per atom (A), dimension (NCL, NATM)

wlpump_res(M) Pumped flux per molecule (A), dimension (NCL, NMOL)

wlpump_res(I) Pumped flux per test ion (A), dimension (NCL, NION)

wlpump_res(P) Pumped flux per plasma ion (A), dimension (NCL, NPLS)

pdena_int Integral number of atoms over the entire Eirene computational grid, dimension (NATM, 0 : NSTRA)

pdenm_int Integral number of molecules over the entire Eirene computational grid, dimension (NMOL, 0 : NSTRA)

pdeni_int Integral number of test ions over the entire Eirene computational grid, dimension (NION, 0 : NSTRA)

pdena_int.b2 Integral number of atoms over the B2 . 5 computational grid, dimension (NATM, 0 : NSTRA)

pdenm_int.b2 Integral number of molecules over the B2 . 5 computational grid, dimension (NMOL, 0 : NSTRA)

pdeni_int.b2 Integral number of test ions over the B2.5 computational grid, dimension (NION, 0:NSTRA)

edena_int Integral energy carried by atoms over the entire Eirene computational grid (J), dimension (NATM, 0:NSTRA)

edenm_int Integral energy carried by molecules over the entire Eirene computational grid (J), dimension (NMOL, 0:NSTRA)

edeni_int Integral energy carried by test ions over the entire Eirene computational grid (J), dimension (NION, 0:NSTRA)

edena_int.b2 Integral energy carried by atoms over the B2.5 computational grid (J), dimension (NATM, 0:NSTRA)

edenm_int.b2 Integral energy carried by molecules over the B2.5 computational grid (J), dimension (NMOL, 0:NSTRA)

edeni_int.b2 Integral energy carried by test ions over the B2.5 computational grid (J), dimension (NION, 0:NSTRA)

Similarly, the `fort.46` file contains the following data blocks, this time all defined on the Eirene triangle grid, which are:

PDENA Atom particle density (in cm^{-3}), dimension (NTRII,NATM)

PDENM Molecule particle density (in cm^{-3}), dimension (NTRII,NMOL)

PDENI Test ion particle density (in cm^{-3}), dimension (NTRII,NION)

EDENA Energy density carried by atoms (in $eV.cm^{-3}$), dimension (NTRII,NATM)

EDENM Energy density carried by molecules (in $eV.cm^{-3}$), dimension (NTRII,NMOL)

EDENI Energy density carried by test ions (in $eV.cm^{-3}$), dimension (NTRII,NION)

VXDENA X-directed momentum density carried by atoms (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NATM)

VXDENM X-directed momentum density carried by molecules (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NMOL)

VXDENI X-directed momentum density carried by test ions (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NION)

VYDENA Y-directed momentum density carried by atoms (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NATM)

VYDENM Y-directed momentum density carried by molecules (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NMOL)

VYDENI Y-directed momentum density carried by test ions (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NION)

VZDENA Z-directed momentum density carried by atoms (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NATM)

VZDENM Z-directed momentum density carried by molecules (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NMOL)

VZDENI Z-directed momentum density carried by test ions (in $g.cm^{-2}.s^{-1}$), dimension (NTRII,NION)

VOL Triangle volumes (in cm^3), dimension (NTRII)

PUX X-component of the poloidal unit vector at the triangle center

PUY Y-component of the poloidal unit vector at the triangle center

PVX X-component of the radial unit vector at the triangle center

PVY Y-component of the radial unit vector at the triangle center

Please note that, for the momentum densities, the coordinate system used by Eirene is Cartesian, i.e. independent of the local magnetic field, and that this data is given in CGS units. For cases without a certain species type, for example a pure Helium case with no molecules, the relevant data fields have a species dimension of 1 as they are mapped internally in Eirene to a dummy pointer array.

5.3 Changing dt

With **SOLPS 4.x** one was often limited to a timestep of $1 \times 10^{-6} \text{ s} - 1 \times 10^{-5} \text{ s}$. With **SOLPS 5.x** this restriction seems to have been lifted, at least for some parameter regimes. In figure 5.1 we show the time evolution of the upstream separatrix temperature for time-steps of $1 \times 10^{-6} \text{ s}$, $1 \times 10^{-5} \text{ s}$, $1 \times 10^{-4} \text{ s}$, $1 \times 10^{-3} \text{ s}$ and $1 \times 10^{-2} \text{ s}$. All of the runs seem to be behaving similarly, though at the highest time-step there is a time shift.

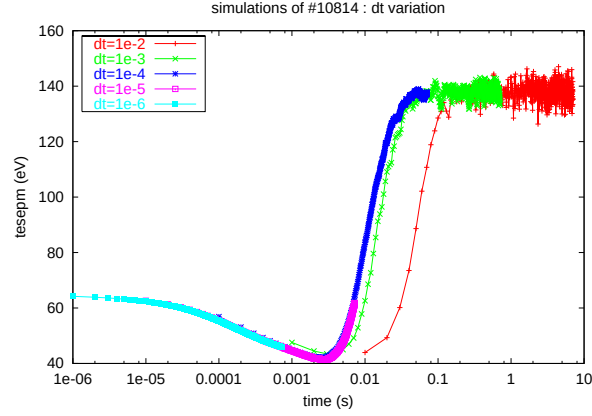


Figure 5.1: Upstream (outer midplane) electron temperature for a range of time-steps.

In figure 5.2 we see the same quantity plotted for a range of time-steps but with an additional core speed-up factor coming from `b2.numerics.parameters`.

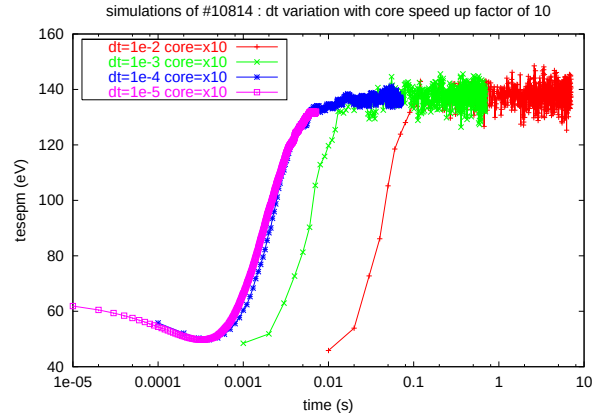


Figure 5.2: Upstream (outer midplane) electron temperature for a range of time-steps with a core speed-up factor.

Figure 5.3 combines the data from figures 5.1 and 5.2.

Figures 5.4, 5.5 and 5.6 show the outer target separatrix temperatures corresponding to figures 5.1, 5.2 and 5.3.

Figures 5.7, 5.8 and 5.9 show continuation runs.

A similar study was undertaken using **SOLPS-ITER** with fluid neutrals, where a subset of the PFR region radial particle boundary conditions were set as total particle flux to simulate a gas puff. In this case it was found that a time step on the order of 10^{-7} was required to obtain converged behavior. In particular, large time steps resulted in significant deviation in the total number of particles injected after an elapsed time, meaning the effective puff rate varied with the value of Δ_t . Figures 5.10 and 5.11 show the evolution of the midplane temperature and the total number of electron after the puff B.C. was applied for several values of Δ_t .

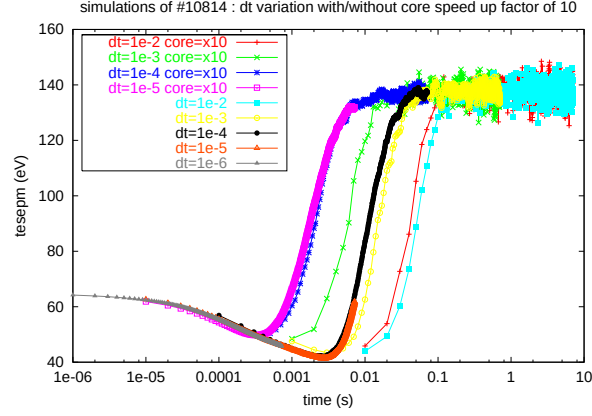


Figure 5.3: Upstream (outer midplane) electron temperature for a range of time-steps with and without a core speed-up factor.

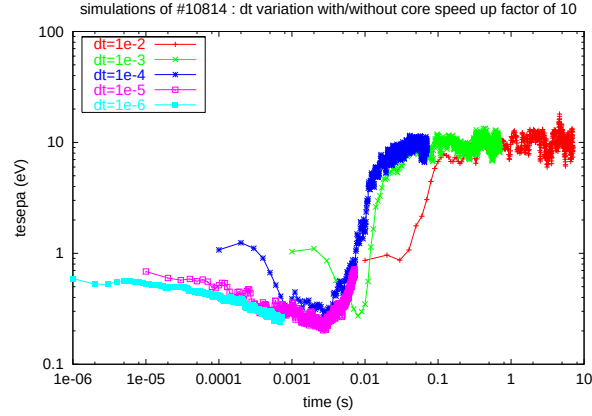


Figure 5.4: Downstream (outer target) electron temperature for a range of time-steps.

5.4 Averaging scheme

When running a coupled **SOLPS-ITER** case to obtain a steady-state solution, it is recommended to use the source averaging scheme detailed in `$SOLPSTOP/doc/SOLPS-ITER.Eirene_averaging.pdf` and references therein. This averaging scheme is meant to yield significant run speed-up and lower the statistical error introduced by the Eirene Monte Carlo noise, as well as provide means to estimate the errors on the solution, following the formalism detailed in [11, 12]. Error estimates are obtained by means of the `b2ye` program (See Section 5.5 below). It should be noted that this averaging scheme does NOT affect the physical run behaviour as such and therefore does not have any stabilizing effect on the run.

The averaging scheme works in three stages:

1. A transient phase during which the code is run normally, until all the short transients die out and the solution approaches its steady state.
2. An averaging phase, turned on by means of the `'b2mndt_av'` switch in the `b2mn.dat` input file, usually with a smaller number of Eirene trajectories, during which batch averaging will be done to obtain an averaged steady-state solution. This stage can be prepared by means of the `run_av` script and uses the `input_avg.eir` Eirene input file prepared by `Uinp` if available. When continuing a run in this phase, the user needs to provide both the instantaneous and averaged solutions from the previous run, i.e. `do mv b2faver b2faveri in`

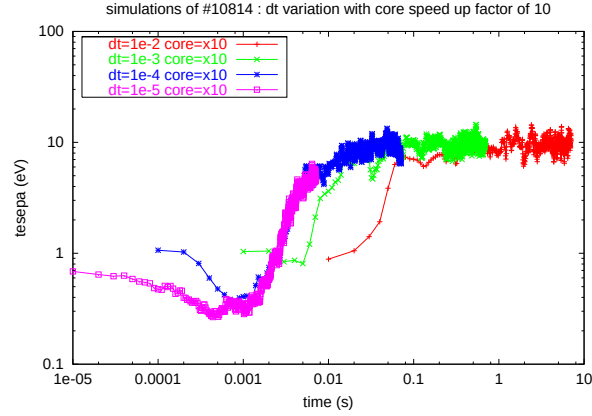


Figure 5.5: Downstream (outer target) electron temperature for a range of time-steps with a core speed-up factor.

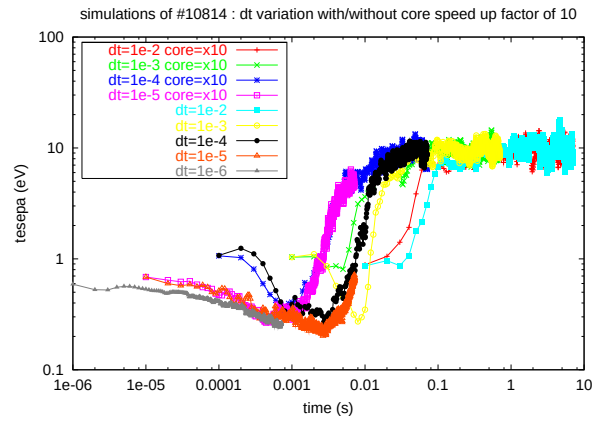


Figure 5.6: Downstream (outer target) electron temperature for a range of time-steps with and without a core speed-up factor.

addition to mv b2fstate b2fstati.

3. A finalization phase where the averaged steady-state solution is used for a small number of timesteps, and a larger number of Eirene trajectories, to reduce statistical noise. This is done by running the `res_av` script and uses the `input_fin.eir` Eirene input file prepared by Uinp if available.

5.5 b2ye, statistical error assessment

The program `b2ye` is meant for computing statistical error assessment according to [13] of a coupled B2.5-Eirene run where the averaging scheme described in Section 5.4 above has been used. It is called by means of the command:

```
b2run b2ye
```

or

```
gmake -f $$SOLPSTOP/runs/Makefile b2ye.prt
```

It uses the data in the `b2time.nc` and `b2favere` files and calculates statistical errors based on averaged quantities collected during the averaging phase of a B2.5-Eirene calculation. For each averaged quantity stored in `b2time.nc` it evaluates:

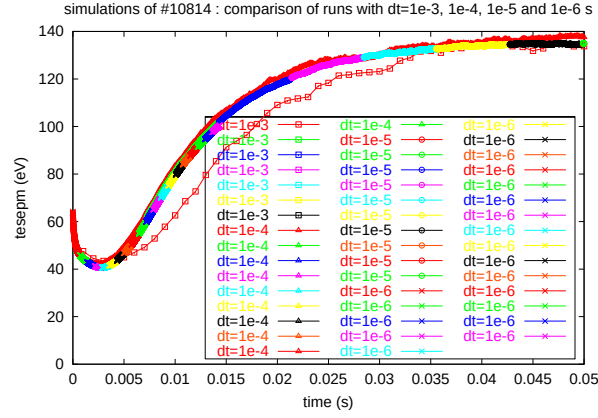


Figure 5.7: Upstream (outer midplane) electron temperature.

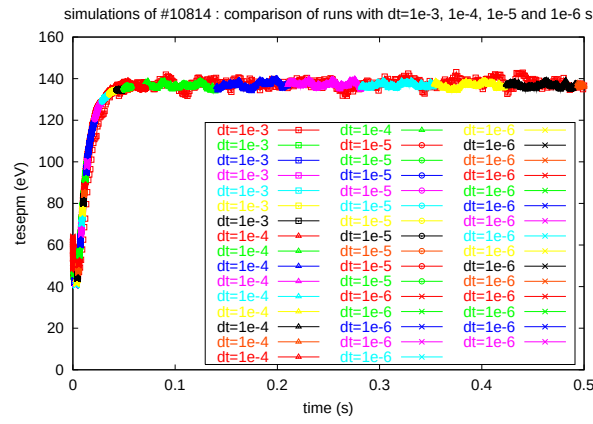


Figure 5.8: Upstream (outer midplane) electron temperature.

- The sample mean.
- The correlation time according to the *batching* method.
- The correlation time according to the *autocorrelation* method.
- The statistical error σ_s according to the maximum of the two correlation times.

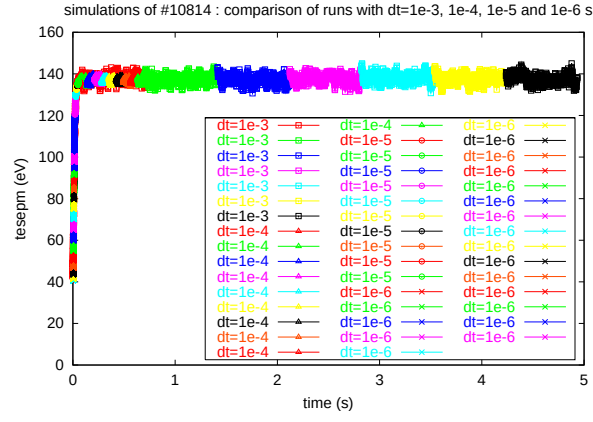


Figure 5.9: Upstream (outer midplane) electron temperature.

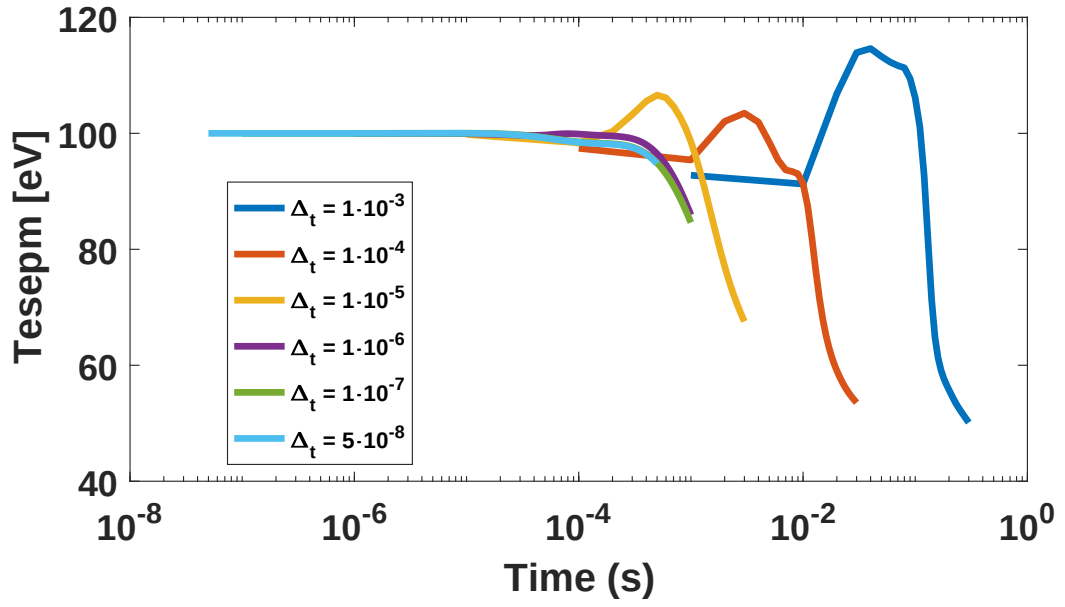


Figure 5.10: Upstream electron temperature for several timesteps after puff b.c. applied.

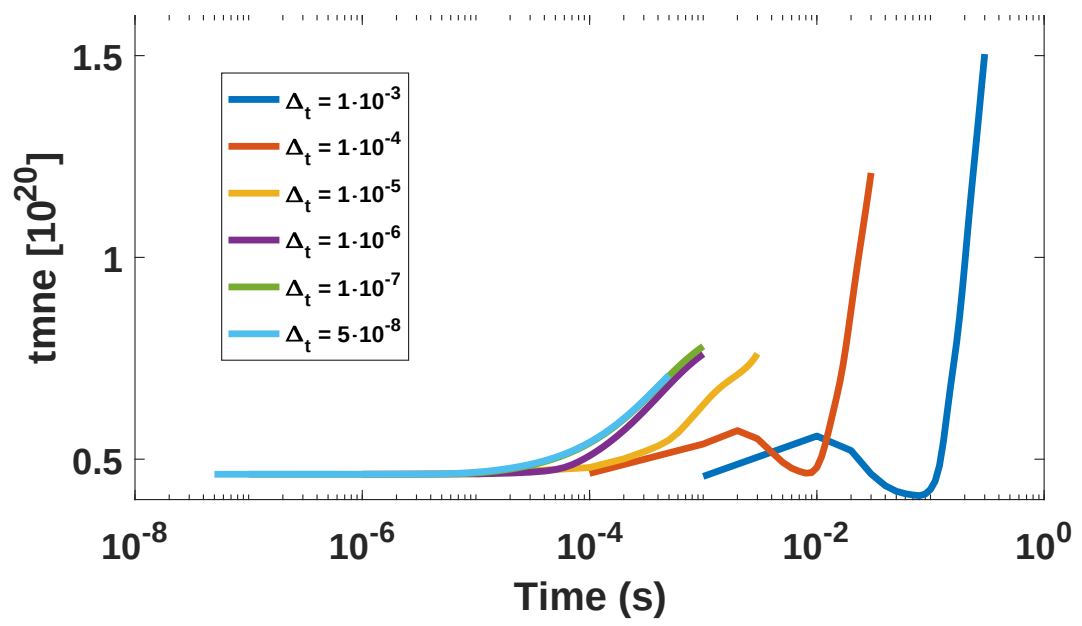


Figure 5.11: Upstream electron temperature for several timesteps after puff b.c. applied.

Chapter 6

Using MDSPLUS

A routine has been added, `b2md`, which allows one to save the output from **SOLPS 5.x** and **SOLPS-ITER** to an MDSplus server in Garching. (For more information about MDSplus see the Web site at www.mdsplus.org.)

The reverse operation, namely recovering an archived run from the MDSplus server, is performed by `b2rd`.

In order to use this facility, you need to ask David Coster at IPP-Garching (david.coster@ipp.mpg.de) to add you to the list of authorized MDSplus writers, supplying the username which will be used, as well as the machine/domain.

In order to compile the code with MDSplus, the `MDSPLUS_DIR` environment needs to be set, as well as the compiler linking instruction `LD_MDSPLUS` — consult your systems people about this, or the MDSplus documentation. Both of these variables should be set in your `/${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}(.local)` file. Then, a full recompilation with

```
sbb ; gmake clean ; gmake depend ; gmake
```

is safest. Before doing this, the `MDSPLUS_DIR` environment needs to be set — consult your systems people about this, or the MDSplus documentation.

[At IPP on the TOK Linux cluster, add the following to `.login`:

```
setenv MDSPLUS_DIR /afs/ipp-garching.mpg.de/home/d/dpc/@sys/mdsplus
source $MDSPLUS_DIR/setup.csh
```

`b2md` requires as input `b2md.dat` and this is searched for in the present working directory, then the `baserun` directory, in order. See Appendix A.5 for a content description.

An example `b2md.dat` is:

```
&b2md_namelist
exp='JET',
shot=45462,
time=59.381,
comment='Parameter scan based (mostly) on g045462.59381.txt.x4.96x24.mod',
timedep=.true.,
snapshot=.true.,
tallies=.true.,
movies=.false.,
&end
```

To save a new run to the MDSplus server, execute

```
save_mds
```

which should create `shotnumber.history` which contains the "shotnumber" for the run on the server (not to be confused with the experimental shot number). The output from the operation will be saved in `b2md.log` and this should be checked for errors.

A run can be re-saved to the same shotnumber with

```
resave_mds
```

which uses the "shotnumber" in `shotnumber.history`.

The `shotnumber.history` file can later be read by `b2rd` which will then fetch from the MDSplus tree the archived shot corresponding to the last number stored within.

6.1 SOLPS tree structure

The `solps` tree consists of the main tree, `SOLPS`, and six sub-trees: `IDENT`, `SNAPSHOT`, `TALLIES`, `TIMEDEP`, `MOVIES` and `DIAG`.

The signals in, for example, `TIMEDEP` can be accessed as `SOLPS::TOP.TIMEDEP.DP:IMPSEP` or alternatively `TIMEDEP::TOP.DP:IMPSEP`.

6.1.1 SOLPS

```
\SOLPS::TOP.DIAG
\SOLPS::TOP.IDENT
\SOLPS::TOP.MOVIES
\SOLPS::TOP.SNAPSHOT
\SOLPS::TOP.TALLIES
\SOLPS::TOP.TIMEDEP
```

6.1.2 IDENT

```
\IDENT::TOP "Top level for information identifying a run"

\IDENT::TOP.IMPSEP
\IDENT::TOP.IMPSEP:CR "Inner midplane separatrix R"
\IDENT::TOP.IMPSEP:CZ "Inner midplane separatrix Z"
\IDENT::TOP.IMPSEP:D "Inner midplane separatrix diffusion coefficient"
\IDENT::TOP.IMPSEP:KYE "Inner midplane separatrix electron diffusivity"
\IDENT::TOP.IMPSEP:KYI "Inner midplane separatrix ion diffusivity for ns=1"
\IDENT::TOP.IMPSEP:KYI0 "Inner midplane separatrix ion/atom diffusivity"
\IDENT::TOP.IMPSEP:NE "Inner midplane separatrix ne"
\IDENT::TOP.IMPSEP:TE "Inner midplane separatrix te"
\IDENT::TOP.IMPSEP:TI "Inner midplane separatrix ti"

\IDENT::TOP.OMPSEP
\IDENT::TOP.OMPSEP:CR "Outer midplane separatrix R"
\IDENT::TOP.OMPSEP:CZ "Outer midplane separatrix Z"
\IDENT::TOP.OMPSEP:D "Outer midplane separatrix diffusion coefficient"
\IDENT::TOP.OMPSEP:KYE "Outer midplane separatrix electron diffusivity"
\IDENT::TOP.OMPSEP:KYI "Outer midplane separatrix ion diffusivity for ns=1"
```

```

\IDENT::TOP.OMPSEP:KYIO "Outer midplane separatrix ion/atom diffusivity"
\IDENT::TOP.OMPSEP:NE "Outer midplane separatrix ne"
\IDENT::TOP.OMPSEP:TE "Outer midplane separatrix te"
\IDENT::TOP.OMPSEP:TI "Outer midplane separatrix ti"

\IDENT::TOP.SEP
\IDENT::TOP.SEP:FHE "core-sol electron energy flux"
\IDENT::TOP.SEP:FHI "core-sol ion energy flux"
\IDENT::TOP.SEP:FNE "core-sol electron flux"
\IDENT::TOP.SEP:FNI "core-sol ion flux"
\IDENT::TOP.B2FGMTRYID "md5sum hash of b2fgmtry"
\IDENT::TOP.B2FPLASMAID "md5sum hash of b2fplasma"
\IDENT::TOP.B2FRATESID "md5sum hash of b2frates"
\IDENT::TOP.B2FSTATEID "md5sum hash of b2fstate"
\IDENT::TOP.B2FSTATIID "md5sum hash of b2fstati"
\IDENT::TOP.COMMENTS "NOT_SET"
\IDENT::TOP.DATE "Date when the run was saved"
\IDENT::TOP.DIRECTORY "Directory in which the run was stored"
\IDENT::TOP.EXP "Experiment name, derived from the b2md_namelist in b2md.dat"
\IDENT::TOP.GEOMETRY "Label in b2fgmtry (derived from b2ag.dat)"
\IDENT::TOP.HOST "Hostname from which the run was saved"
\IDENT::TOP.MAIN "Label in b2fstate derived from b2mn.dat"
\IDENT::TOP.NS "Number of species"
\IDENT::TOP.NX "Grid size in poloidal direction"
\IDENT::TOP.NY "Grid size in radial direction"
\IDENT::TOP.OBJECTCODE "SOLPS $OBJECTCODE"
\IDENT::TOP.PHYSICS "Label in b2fparam derived from b2ah.dat"
\IDENT::TOP.SHOT "Experiment shot number (int) derived from the b2md_namelist in b2md.dat"
\IDENT::TOP.SOLPSTOP "SOLPS $OBJECTCODE"
\IDENT::TOP.SOLPSVERSION "Version number from SOLPS"
\IDENT::TOP.SPECIES "List of plasma species"
\IDENT::TOP.TIME "Experiment time point (double) derived from the b2md_namelist in b2md.dat"
\IDENT::TOP.USER "Username"
\IDENT::TOP.VERSION "Version number of the SOLPS MDS+ data output routine"
\IDENT::TOP.VERSION:B2FMOVIE "Version of b2fmovie"
\IDENT::TOP.VERSION:B2FPLASMA "Version of b2fplasma"
\IDENT::TOP.VERSION:B2FRATES "Version of b2frates"
\IDENT::TOP.VERSION:B2FSTATE "Version of b2fstate"
\IDENT::TOP.VERSION:B2FSTATI "Version of b2ftati"
\IDENT::TOP.VERSION:B2FTRACK "Version of b2ftrack"
\IDENT::TOP.VERSION:GIT "SOLPS GIT Repository Revision"
\IDENT::TOP.VERSION:PLASMASTATE "Version of plasmastate"
\IDENT::TOP.VERSION:SVN "SOLPS SVN Repository Revision"
\IDENT::TOP.WALL_SPECIES "List of wall species"

```

6.1.3 SNAPSHOT

```

\SNAPSHOT::TOP "Detailed results from the end of the run"

\SNAPSHOT::TOP.DIMENSIONS "Dimensions of the grid, number of species, position of separatrix etc."
\SNAPSHOT::TOP.DIMENSIONS:IMP "Index of the inner midplane"
\SNAPSHOT::TOP.DIMENSIONS:NATM "Number of atomic species"
\SNAPSHOT::TOP.DIMENSIONS:NION "Number of test ion species"
\SNAPSHOT::TOP.DIMENSIONS:NMOL "Number of molecular species"
\SNAPSHOT::TOP.DIMENSIONS:NS "Number of fluid species"
\SNAPSHOT::TOP.DIMENSIONS:NSW "Number of wall species"
\SNAPSHOT::TOP.DIMENSIONS:NWALL "Number of wall elements"
\SNAPSHOT::TOP.DIMENSIONS:NX "Number of cells in the parallel or poloidal direction"
\SNAPSHOT::TOP.DIMENSIONS:NY "Number of cells in the radial direction"
\SNAPSHOT::TOP.DIMENSIONS:OMP "Position of the outer midplane"
\SNAPSHOT::TOP.DIMENSIONS:SEP "Position of the separatrix"
\SNAPSHOT::TOP.DIMENSIONS:TARG1 "Position of target 1 (usually inner)"
\SNAPSHOT::TOP.DIMENSIONS:TARG2 "Position of target 2"

```

```

\SNAPSHOT::TOP.DIMENSIONS:TARG3 "Position of target 3"
\SNAPSHOT::TOP.DIMENSIONS:TARG4 "Position of target 4 (usually outer)"

\SNAPSHOT::TOP.GRID "Details of the grid"
\SNAPSHOT::TOP.GRID:AM "Atom/ion mass"
\SNAPSHOT::TOP.GRID:BOTTOMIX "bottom ix neighbourhood array"
\SNAPSHOT::TOP.GRID:BOTTOMIY "bottom iy neighbourhood array"
\SNAPSHOT::TOP.GRID:CFLAGS "cell flags array"
\SNAPSHOT::TOP.GRID:CR "R-coordinate of the cell centre"
\SNAPSHOT::TOP.GRID:CR_X "R-coordinate of the cell x-face"
\SNAPSHOT::TOP.GRID:CR_Y "R-coordinate of the cell y-face"
\SNAPSHOT::TOP.GRID:CZ "Z-coordinate of the cell centre"
\SNAPSHOT::TOP.GRID:CZ_X "Z-coordinate of the cell x-face"
\SNAPSHOT::TOP.GRID:CZ_Y "Z-coordinate of the cell y-face"
\SNAPSHOT::TOP.GRID:DSPAR "Parallel distance"
\SNAPSHOT::TOP.GRID:DSPOL "Poloidal distance"
\SNAPSHOT::TOP.GRID:DSRAD "Radial distance wrt separatrix"
\SNAPSHOT::TOP.GRID:IMAPX "IX-position of wall elements"
\SNAPSHOT::TOP.GRID:IMAPY "IY-position of wall elements"
\SNAPSHOT::TOP.GRID:LEFTIX "left ix neighbourhood array"
\SNAPSHOT::TOP.GRID:LEFTIY "left iy neighbourhood array"
\SNAPSHOT::TOP.GRID:POT "Potential Energy"
\SNAPSHOT::TOP.GRID:POTI "Ionization Potential"
\SNAPSHOT::TOP.GRID:R "R-coordinates of cell vertices"
\SNAPSHOT::TOP.GRID:REGFLX "x-directed flux region indices"
\SNAPSHOT::TOP.GRID:REGFLY "y-directed flux region indices"
\SNAPSHOT::TOP.GRID:REGVOL "volume region indices"
\SNAPSHOT::TOP.GRID:RIGHTIX "right ix neighbourhood array"
\SNAPSHOT::TOP.GRID:RIGHTIY "right iy neighbourhood array"
\SNAPSHOT::TOP.GRID:TOPIX "top ix neighbourhood array"
\SNAPSHOT::TOP.GRID:TOPIY "top iy neighbourhood array"
\SNAPSHOT::TOP.GRID:VESSEL "Vessel structure"
\SNAPSHOT::TOP.GRID:XYMAP "Wall index array"
\SNAPSHOT::TOP.GRID:Z "Z-coordinates of cell vertices"
\SNAPSHOT::TOP.GRID:ZA "Atom charge"
\SNAPSHOT::TOP.GRID:ZAMAX "Atom charge (max over bundle)"
\SNAPSHOT::TOP.GRID:ZAMIN "Atom charge (min over bundle)"
\SNAPSHOT::TOP.GRID:ZN "Nuclear charge"
\SNAPSHOT::TOP:ALF "Thermo-electric coefficient"
\SNAPSHOT::TOP:B "B field"
\SNAPSHOT::TOP:BRHE "Electron heat source from recycling"
\SNAPSHOT::TOP:BRHI "Ion heat source from recycling"
\SNAPSHOT::TOP:BRMO "Momentum source from recycling"
\SNAPSHOT::TOP:BRNA "Particle source from recycling"
\SNAPSHOT::TOP:CEQP "Electron-ion thermal equipartition factor"
\SNAPSHOT::TOP:D "D"
\SNAPSHOT::TOP:DAB2 "Atomic density"
\SNAPSHOT::TOP:DIB2 "Test ion density"
\SNAPSHOT::TOP:DMB2 "Molecular density"
\SNAPSHOT::TOP:DP "Dpa"
\SNAPSHOT::TOP:EDISSML "Power to atoms due to molecule dissociation (W)"
\SNAPSHOT::TOP:EIONRAD "Power loss due to (test) ion radiation (W)"
\SNAPSHOT::TOP:EMISS "Atomic H-alpha emissivity (photons/m3/s)"
\SNAPSHOT::TOP:EMISSMOL "Molecular H-alpha emissivity (photons/m3/s)"
\SNAPSHOT::TOP:EMOLRAD "Power loss due to molecule radiation (W)"
\SNAPSHOT::TOP:ENEUTRAD "Power loss due to atom radiation (W)"
\SNAPSHOT::TOP:FCHDIAX "Poloidal diamagnetic current"
\SNAPSHOT::TOP:FCHDIAY "Radial diamagnetic current"
\SNAPSHOT::TOP:FCHINERTX "Poloidal ion inertial current"
\SNAPSHOT::TOP:FCHINERTY "Radial ion inertial current"
\SNAPSHOT::TOP:FCHINX "Poloidal ion-neutral current"
\SNAPSHOT::TOP:FCHINY "Radial ion-neutral current"
\SNAPSHOT::TOP:FCHVISPARX "Poloidal parallel viscosity current"
\SNAPSHOT::TOP:FCHVISPARY "Radial parallel viscosity current"
\SNAPSHOT::TOP:FCHVISPERX "Poloidal perpendicular viscosity current"
\SNAPSHOT::TOP:FCHVISPERY "Radial perpendicular viscosity current"

```

```

\SNAPSHOT::TOP:FCHVISQX "Poloidal heat viscosity current"
\SNAPSHOT::TOP:FCHVISQY "Radial heat viscosity current"
\SNAPSHOT::TOP:FCHX "Poloidal current"
\SNAPSHOT::TOP:FCHX_32 "Convective poloidal current"
\SNAPSHOT::TOP:FCHX_52 "Conductive poloidal current"
\SNAPSHOT::TOP:FCHY "Radial current"
\SNAPSHOT::TOP:FCHY_32 "Convective radial current"
\SNAPSHOT::TOP:FCHY_52 "Conductive radial current"
\SNAPSHOT::TOP:FCH_P "Parallel current"
\SNAPSHOT::TOP:FHEPSCHX "Poloidal electron Pfirsch-Schlueter energy flux"
\SNAPSHOT::TOP:FHEPSCHY "Radial electron Pfirsch-Schlueter energy flux"
\SNAPSHOT::TOP:FHEX "Poloidal electron energy flux"
\SNAPSHOT::TOP:FHEX_COND "Poloidal electron energy flux, conductive part"
\SNAPSHOT::TOP:FHEX_CONV "Poloidal electron energy flux, convective part"
\SNAPSHOT::TOP:FHEX_EIR "Poloidal electron energy flux (passed to Eirene)"
\SNAPSHOT::TOP:FHEX_MDF "Poloidal electron energy flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FHEY "Radial electron energy flux"
\SNAPSHOT::TOP:FHEY_COND "Radial electron energy flux, conductive part"
\SNAPSHOT::TOP:FHEY_CONV "Radial electron energy flux, convective part"
\SNAPSHOT::TOP:FHEY_EIR "Radial electron energy flux (passed to Eirene)"
\SNAPSHOT::TOP:FHEY_MDF "Radial electron energy flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FHIPSCHX "Poloidal ion Pfirsch-Schlueter energy flux"
\SNAPSHOT::TOP:FHIPSCHY "Radial ion Pfirsch-Schlueter energy flux"
\SNAPSHOT::TOP:FHIX "Poloidal ion energy flux"
\SNAPSHOT::TOP:FHIX_COND "Poloidal ion energy flux, conductive part"
\SNAPSHOT::TOP:FHIX_CONV "Poloidal ion energy flux, convective part"
\SNAPSHOT::TOP:FHIX_EIR "Poloidal ion energy flux (passed to Eirene)"
\SNAPSHOT::TOP:FHIX_MDF "Poloidal ion energy flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FHIY "Radial ion energy flux"
\SNAPSHOT::TOP:FHIY_COND "Radial ion energy flux, conductive part"
\SNAPSHOT::TOP:FHIY_CONV "Radial ion energy flux, convective part"
\SNAPSHOT::TOP:FHIY_EIR "Radial ion energy flux (passed to Eirene)"
\SNAPSHOT::TOP:FHIY_MDF "Radial ion energy flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FHJX "Poloidal electrostatic energy flux"
\SNAPSHOT::TOP:FHJY "Radial electrostatic energy flux"
\SNAPSHOT::TOP:FHMX "Poloidal kinetic energy flux"
\SNAPSHOT::TOP:FHMY "Radial kinetic energy flux"
\SNAPSHOT::TOP:FHPX "Poloidal potential energy flux"
\SNAPSHOT::TOP:FHPY "Radial potential energy flux"
\SNAPSHOT::TOP:FHTX "Poloidal total energy flux"
\SNAPSHOT::TOP:FHTY "Radial total energy flux"
\SNAPSHOT::TOP:FLLIME "Electron thermal flux limiter factor"
\SNAPSHOT::TOP:FLLIMI "Ion thermal flux limiter factor"
\SNAPSHOT::TOP:FLOEX_NOC "Poloidal electron energy flow"
\SNAPSHOT::TOP:FLOEY_NOC "Radial electron energy flow"
\SNAPSHOT::TOP:FLOIX_NOC "Poloidal ion energy flow"
\SNAPSHOT::TOP:FLOIY_NOC "Radial ion energy flow"
\SNAPSHOT::TOP:FMOX "Poloidal momentum flux"
\SNAPSHOT::TOP:FMOY "Radial momentum flux"
\SNAPSHOT::TOP:FNAPSCHX "Poloidal Pfirsch-Schlueter particle flux"
\SNAPSHOT::TOP:FNAPSCHY "Radial Pfirsch-Schlueter particle flux"
\SNAPSHOT::TOP:FNAX "Poloidal particle flux"
\SNAPSHOT::TOP:FNAX_32 "Poloidal particle flux (3/2 piece)"
\SNAPSHOT::TOP:FNAX_52 "Poloidal particle flux (5/2 piece)"
\SNAPSHOT::TOP:FNAX_EIR "Poloidal particle flux (passed to Eirene)"
\SNAPSHOT::TOP:FNAX_FCOR "Poloidal particle flux (including HZ toroidal factors)"
\SNAPSHOT::TOP:FNAX_HE "Poloidal particle flux (for electron heat equation)"
\SNAPSHOT::TOP:FNAX_MDF "Poloidal particle flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FNAX_NODRIFT "Poloidal particle flux (not including drift contributions)"
\SNAPSHOT::TOP:FNAY "Radial particle flux"
\SNAPSHOT::TOP:FNAY_32 "Radial particle flux (3/2 piece)"
\SNAPSHOT::TOP:FNAY_52 "Radial particle flux (5/2 piece)"
\SNAPSHOT::TOP:FNAY_EIR "Radial particle flux (passed to Eirene)"
\SNAPSHOT::TOP:FNAY_FCOR "Radial particle flux (including HZ toroidal factors)"
\SNAPSHOT::TOP:FNAY_HE "Radial particle flux (for electron heat equation)"
\SNAPSHOT::TOP:FNAY_MDF "Radial particle flux (with modified divergent diamagnetic flow)"
\SNAPSHOT::TOP:FNAY_NODRIFT "Radial particle flux (not including drift contributions)"

```

```

\SNAPSHOT::TOP:FNEX_32 "Poloidal electron flux (3/2 piece)"
\SNAPSHOT::TOP:FNEX_52 "Poloidal electron flux (5/2 piece)"
\SNAPSHOT::TOP:FNEX_EIR "Poloidal electron flux (passed to Eirene)"
\SNAPSHOT::TOP:FNEY_32 "Radial electron flux (3/2 piece)"
\SNAPSHOT::TOP:FNEY_52 "Radial electron flux (5/2 piece)"
\SNAPSHOT::TOP:FNEY_EIR "Radial electron flux (passed to Eirene)"
\SNAPSHOT::TOP:FNIX_32 "Poloidal total ion flux (3/2 piece)"
\SNAPSHOT::TOP:FNIX_52 "Poloidal total ion flux (5/2 piece)"
\SNAPSHOT::TOP:FNIY_32 "Radial total ion flux (3/2 piece)"
\SNAPSHOT::TOP:FNIY_52 "Radial total ion flux (5/2 piece)"
\SNAPSHOT::TOP:HX "Length of cell"
\SNAPSHOT::TOP:HY "Width of cell"
\SNAPSHOT::TOP:HY1 "Corrected width of cell"
\SNAPSHOT::TOP:KINRGY "Total kinetic energy"
\SNAPSHOT::TOP:KYE "kye"
\SNAPSHOT::TOP:KYI "kyi"
\SNAPSHOT::TOP:KYIO "kyi0"
\SNAPSHOT::TOP:NA "Ion/Neutral density"
\SNAPSHOT::TOP:NA0 "Ion/Neutral density (previous time-step)"
\SNAPSHOT::TOP:NE "Electron density"
\SNAPSHOT::TOP:NEO "Electron density (previous time-step)"
\SNAPSHOT::TOP:PEFA "Poloidal atomic energy flux"
\SNAPSHOT::TOP:PEFM "Poloidal molecular energy flux"
\SNAPSHOT::TOP:PFLA "Poloidal atomic flux"
\SNAPSHOT::TOP:PFLM "Poloidal molecular flux"
\SNAPSHOT::TOP:PO "Electric potential"
\SNAPSHOT::TOP:POO "Electric potential (previous time-step)"
\SNAPSHOT::TOP:PR "Pressure"
\SNAPSHOT::TOP:QZ "qz"
\SNAPSHOT::TOP:RCXHI "CX ion energy neutral losses"
\SNAPSHOT::TOP:RCXMO "CX momentum neutral losses"
\SNAPSHOT::TOP:RCXNA "CX particle neutral losses"
\SNAPSHOT::TOP:REFA "Radial atomic energy flux"
\SNAPSHOT::TOP:REFM "Radial molecular energy flux"
\SNAPSHOT::TOP:RESCO "Continuity equation residual"
\SNAPSHOT::TOP:RESHE "Electron heat equation residual"
\SNAPSHOT::TOP:RESHI "Ion heat equation residual"
\SNAPSHOT::TOP:RESMO "Momentum equation residual"
\SNAPSHOT::TOP:RESMT "Total momentum equation residual"
\SNAPSHOT::TOP:RESPO "Potential equation residual"
\SNAPSHOT::TOP:RFLA "Radial atomic flux"
\SNAPSHOT::TOP:RFLM "Radial molecular flux"
\SNAPSHOT::TOP:RPI "Effective Ionisation Potential (function of ix,iy,is)"
\SNAPSHOT::TOP:RPT "Cumulative Ionisation Potential (function of ix,iy,is)"
\SNAPSHOT::TOP:RQAHE "Electron cooling rate"
\SNAPSHOT::TOP:RQBRM "Bremsstrahlung radiation rate"
\SNAPSHOT::TOP:RQRAD "Line radiation rate"
\SNAPSHOT::TOP:RRAHI "Recombination ion energy losses"
\SNAPSHOT::TOP:RRAMO "Recombination momentum losses"
\SNAPSHOT::TOP:RRANA "Recombination particle losses"
\SNAPSHOT::TOP:RSAHI "Ionization ion energy losses"
\SNAPSHOT::TOP:RSAMO "Ionization momentum losses"
\SNAPSHOT::TOP:RSANA "Ionization particle losses"
\SNAPSHOT::TOP:RZ2 "Ion charge squared (function of ix,iy,is)"
\SNAPSHOT::TOP:RZA "Ion charge (function of ix,iy,is)"
\SNAPSHOT::TOP:SIG "Anomalous conductivity"
\SNAPSHOT::TOP:SMAF "smaf (extended friction term)"
\SNAPSHOT::TOP:SMAG "smag (pressure gradient term)"
\SNAPSHOT::TOP:SMAV "smav (additional viscosity term)"
\SNAPSHOT::TOP:SMFR "Friction force"
\SNAPSHOT::TOP:SMO "Parallel momentum source"
\SNAPSHOT::TOP:SMPR "Thermal force (proportional to exhp)"
\SNAPSHOT::TOP:SMPT "Thermal force (proportional to Ti and Te gradients)"
\SNAPSHOT::TOP:SMQ "smq (momentum sources from atomic physics)"
\SNAPSHOT::TOP:SMTH "Thermal force"
\SNAPSHOT::TOP:SRCML "Molecule dissociation rate (#/s)"
\SNAPSHOT::TOP:SX "sx (poloidal contact area)"

```

```

\SNAPSHOT::TOP:XPERP "sxperp (poloidal area perpendicular to flux tube)"
\SNAPSHOT::TOP:SY "sy"
\SNAPSHOT::TOP:TAB2 "Neutral atom temperature"
\SNAPSHOT::TOP:TE "Electron temperature"
\SNAPSHOT::TOP:TE0 "Electron temperature (previous time-step)"
\SNAPSHOT::TOP:TEXTAN "Atom species"
\SNAPSHOT::TOP:TEXTCOMP "Components"
\SNAPSHOT::TOP:TEXTDRIFT "Directions of drift components"
\SNAPSHOT::TOP:TEXTIN "Test ion species"
\SNAPSHOT::TOP:TEXTLIN "Linearization components"
\SNAPSHOT::TOP:TEXTMN "Molecule species"
\SNAPSHOT::TOP:TEXTPL "Plasma species"
\SNAPSHOT::TOP:TEXTWL "Wall species"
\SNAPSHOT::TOP:TI "Ion temperature"
\SNAPSHOT::TOP:TIO "Ion temperature (previous time-step)"
\SNAPSHOT::TOP:TIB2 "Test ion temperature"
\SNAPSHOT::TOP:TMB2 "Neutral molecule temperature"
\SNAPSHOT::TOP:UA "Ion/neutral parallel velocity"
\SNAPSHOT::TOP:UA0 "Ion/neutral parallel velocity (previous time-step)"
\SNAPSHOT::TOP:UADIA "Drift velocities in diamagnetic and radial directions"
\SNAPSHOT::TOP:VADIA "Divergent diamagnetic drift velocities in diamagnetic and radial directions"
\SNAPSHOT::TOP:VAECRB "ExB drift velocities in diamagnetic and radial directions"
\SNAPSHOT::TOP:VEDIA "Divergent diamagnetic drift electron velocities in diamagnetic and radial direc
\SNAPSHOT::TOP:VEECRB "Electron ExB drift velocities in diamagnetic and radial directions"
\SNAPSHOT::TOP:VISC "Viscosity"
\SNAPSHOT::TOP:VLAX "Poloidal pinch velocity"
\SNAPSHOT::TOP:VLAY "Radial pinch velocity"
\SNAPSHOT::TOP:VOL "Volume"
\SNAPSHOT::TOP:WADIA "Diamagnetic drift velocities in diamagnetic and radial directions"
\SNAPSHOT::TOP:WALL_TEMP "Wall temperature"

```

6.1.4 TALLIES

```

\tallies::TOP "Various tallies (volume or surface integrals) as a function of time"
\tallies::TOP:B2BREMREG "Bremsstrahlung energy loss per region"
\tallies::TOP:B2DIVUA "B2DIVUA"
\tallies::TOP:B2DIVUE "B2DIVUE"
\tallies::TOP:B2EXBA "B2EXBA"
\tallies::TOP:B2EXBE "B2EXBE"
\tallies::TOP:B2FRAA "B2FRAA"
\tallies::TOP:B2JOULE "B2JOULE"
\tallies::TOP:B2QIE "B2QIE"
\tallies::TOP:B2RADREG "Line radiation energy loss per region"
\tallies::TOP:B2SEXTSCHREG "External charge source"
\tallies::TOP:B2SEXTSHEREG "External electron energy source"
\tallies::TOP:B2SEXTSHIREG "External ion energy source"
\tallies::TOP:B2SEXTSMOREG "External parallel momentum source"
\tallies::TOP:B2SEXTSNAREG "External particle source"
\tallies::TOP:B2SEXTSNEREG "External electron particle source"
\tallies::TOP:B2SHE "B2SHE"
\tallies::TOP:B2SHE0 "B2SHE0"
\tallies::TOP:B2SHI "B2SHI"
\tallies::TOP:B2SHI0 "B2SHI0"
\tallies::TOP:B2STBCSCHREG "B2STBCSCHREG"
\tallies::TOP:B2STBCSHEREG "B2STBCSHEREG"
\tallies::TOP:B2STBCSHIREG "B2STBCSHIREG"
\tallies::TOP:B2STBCSMOREG "B2STBCSMOREG"
\tallies::TOP:B2STBCSNAREG "B2STBCSNAREG"
\tallies::TOP:B2STBCSNEREG "Boundary electron particle source"
\tallies::TOP:B2STBMSCHREG "Additional charge source"
\tallies::TOP:B2STBMSHEREG "Additional electron energy source"
\tallies::TOP:B2STBMShIREG "Additional ion energy source"
\tallies::TOP:B2STBMSMOREG "Additional parallel momentum source"
\tallies::TOP:B2STBMSNAREG "Additional particle source"

```



```

\TALLIES::TOP:B2STBMSNEREG "Additional electron particle source"
\TALLIES::TOP:B2STBRSCHREG "B2STBRSCHREG"
\TALLIES::TOP:B2STBRSHEREG "B2STBRSHEREG"
\TALLIES::TOP:B2STBRSHIREG "B2STBRSHIREG"
\TALLIES::TOP:B2STBRSMOREG "B2STBRSMOREG"
\TALLIES::TOP:B2STBRSNAREG "B2STBRSNAREG"
\TALLIES::TOP:B2STBRSNEREG "Recycling electron particle source"
\TALLIES::TOP:B2STR "B2STR"
\TALLIES::TOP:B2VISA "B2VISA"
\TALLIES::TOP:B2WRONG1 "B2WRONG1"
\TALLIES::TOP:B2WRONG2 "B2WRONG2"
\TALLIES::TOP:B2WRONG3 "B2WRONG3"
\TALLIES::TOP:FCHXREG "FCHXREG"
\TALLIES::TOP:FCHYREG "FCHYREG"
\TALLIES::TOP:FHEXREG "FHEXREG"
\TALLIES::TOP:FHEYREG "FHEYREG"
\TALLIES::TOP:FHIXREG "FHIXREG"
\TALLIES::TOP:FHIYREG "FHIYREG"
\TALLIES::TOP:FHJXREG "FHJXREG"
\TALLIES::TOP:FHJYREG "FHJYREG"
\TALLIES::TOP:FHMXREG "FHMXXREG"
\TALLIES::TOP:FHMYREG "FHMYREG"
\TALLIES::TOP:FHPXREG "FHPXREG"
\TALLIES::TOP:FHPYREG "FHPYREG"
\TALLIES::TOP:FHTXREG "fhtxreg"
\TALLIES::TOP:FHTYREG "fhtyreg"
\TALLIES::TOP:FNAXREG "FNAXREG"
\TALLIES::TOP:FNAYREG "FNAYREG"
\TALLIES::TOP:NAREG "NAREG"
\TALLIES::TOP:NE2REG "NE2REG"
\TALLIES::TOP:NEREG "NEREG"
\TALLIES::TOP:NIREG "NIREG"
\TALLIES::TOP:NS "NS"
\TALLIES::TOP:POREG "POREG"
\TALLIES::TOP:QCONVEXREG "qconvexreg"
\TALLIES::TOP:QCONVEYREG "qconveyreg"
\TALLIES::TOP:QCONVIXREG "qconvixreg"
\TALLIES::TOP:QCONVIYREG "qconviyreg"
\TALLIES::TOP:RCXHIREG "RCXHIREG"
\TALLIES::TOP:RCXMOREG "RCXMOREG"
\TALLIES::TOP:RCXNAREG "RCXNAREG"
\TALLIES::TOP:RESCOREG "RESCOREG"
\TALLIES::TOP:RESCO_REG "RESCO_REG"
\TALLIES::TOP:RESHEREG "RESHEREG"
\TALLIES::TOP:RESHE_REG "RESHE_REG"
\TALLIES::TOP:RESHIREG "RESHIREG"
\TALLIES::TOP:RESHI_REG "RESHI_REG"
\TALLIES::TOP:RESMOREG "RESMOREG"
\TALLIES::TOP:RESMO_REG "RESMO_REG"
\TALLIES::TOP:RESMTREG "RESMTREG"
\TALLIES::TOP:RESMT_REG "RESMT_REG"
\TALLIES::TOP:RESPOREG "RESPOREG"
\TALLIES::TOP:RESPO_REG "RESPO_REG"
\TALLIES::TOP:RQAHHEREG "RQAHHEREG"
\TALLIES::TOP:RQBRMREG "Bremsstrahlung radiation rate per region"
\TALLIES::TOP:RQADREG "Line radiation rate per region"
\TALLIES::TOP:RRAHIREG "RRAHIREG"
\TALLIES::TOP:RRAMOREG "RRAMOREG"
\TALLIES::TOP:RRANAREG "RRANAREG"
\TALLIES::TOP:RSAHIREG "RSAHIREG"
\TALLIES::TOP:RSAMOREG "RSAMOREG"
\TALLIES::TOP:RSANAREG "RSANAREG"
\TALLIES::TOP:SPECIES "Species"
\TALLIES::TOP:TEREG "TEREG"
\TALLIES::TOP:TIME "TIME"
\TALLIES::TOP:TIREG "TIREG"
\TALLIES::TOP:VOLREG "VOLREG"

```

```

\tALLIES::TOP:VREG "VREG"
\tALLIES::TOP:XREG "XREG"
\tALLIES::TOP:YREG "YREG"

```

6.1.5 TIMEDEP

```

\tIMEDEP::TOP "Various time dependent quantities"

\tIMEDEP::TOP.DN "diffusion coefficient"
\tIMEDEP::TOP.DN:IMPSEP
\tIMEDEP::TOP.DN:OMPSEP "diffusion coefficient, outer midplane"

\tIMEDEP::TOP.DP "pressure diffusion coefficient"
\tIMEDEP::TOP.DP:IMPSEP
\tIMEDEP::TOP.DP:OMPSEP "pressure diffusion coefficient, outer midplane"

\tIMEDEP::TOP.FCH
\tIMEDEP::TOP.FCH:DIVINT "integrated radial current, divertor region"
\tIMEDEP::TOP.FCH:MCWINT "integrated radial current, main chamber"
\tIMEDEP::TOP.FCH:TARG1INT "integrated poloidal current, inboard divertor"
\tIMEDEP::TOP.FCH:TARG2INT
\tIMEDEP::TOP.FCH:TARG3INT
\tIMEDEP::TOP.FCH:TARG4INT "integrated poloidal current, outboard divertor"

\tIMEDEP::TOP.FEE
\tIMEDEP::TOP.FEE:DIVINT "integrated radial electron energy flux, divertor region"
\tIMEDEP::TOP.FEE:MCWINT "integrated radial electron energy flux, main chamber"
\tIMEDEP::TOP.FEE:TARG1INT "integrated poloidal electron energy flux, inboard divertor"
\tIMEDEP::TOP.FEE:TARG2INT
\tIMEDEP::TOP.FEE:TARG3INT
\tIMEDEP::TOP.FEE:TARG4INT "integrated poloidal electron energy flux, outboard divertor"

\tIMEDEP::TOP.FEI
\tIMEDEP::TOP.FEI:DIVINT "integrated radial ion energy flux, divertor region"
\tIMEDEP::TOP.FEI:MCWINT "integrated radial ion energy flux, main chamber"
\tIMEDEP::TOP.FEI:TARG1INT "integrated poloidal ion energy flux, inboard divertor"
\tIMEDEP::TOP.FEI:TARG2INT
\tIMEDEP::TOP.FEI:TARG3INT
\tIMEDEP::TOP.FEI:TARG4INT "integrated poloidal ion energy flux, outboard divertor"

\tIMEDEP::TOP.FET
\tIMEDEP::TOP.FET:DIVINT
\tIMEDEP::TOP.FET:MCWINT
\tIMEDEP::TOP.FET:TARG1INT
\tIMEDEP::TOP.FET:TARG2INT
\tIMEDEP::TOP.FET:TARG3INT
\tIMEDEP::TOP.FET:TARG4INT

\tIMEDEP::TOP.FNI
\tIMEDEP::TOP.FNI:DIVINT "integrated radial particle flux, divertor region"
\tIMEDEP::TOP.FNI:MCWINT "integrated radial particle flux, main chamber"
\tIMEDEP::TOP.FNI:TARG1INT "integrated poloidal particle flux, inboard divertor"
\tIMEDEP::TOP.FNI:TARG2INT
\tIMEDEP::TOP.FNI:TARG3INT
\tIMEDEP::TOP.FNI:TARG4INT "integrated poloidal particle flux, outboard divertor"

\tIMEDEP::TOP.IMP
\tIMEDEP::TOP.IMP:AN "atom density, inboard midplane"
\tIMEDEP::TOP.IMP:DN "diffusion coefficient, inboard midplane"
\tIMEDEP::TOP.IMP:DP "pressure diffusion coefficient, inboard midplane"
\tIMEDEP::TOP.IMP:DS "S-S_sep (inner midplane)"
\tIMEDEP::TOP.IMP:KE

```

```

\TIMEDEP::TOP.IMP:KI "ion thermal diffusivity, inboard midplane"
\TIMEDEP::TOP.IMP:LH "Conductive flux limit factor for neutrals, inner midplane"
\TIMEDEP::TOP.IMP:LN "Convective flux limit factor for neutrals, inner midplane"
\TIMEDEP::TOP.IMP:MN "molecule density, inboard midplane"
\TIMEDEP::TOP.IMP:NE "electron density, inboard midplane"
\TIMEDEP::TOP.IMP:PO "potential, inboard midplane"
\TIMEDEP::TOP.IMP:TE "electron temperature, inboard midplane"
\TIMEDEP::TOP.IMP:TI "ion temperature, inboard midplane"
\TIMEDEP::TOP.IMP:VS "viscosity coefficient, inboard midplane'"
\TIMEDEP::TOP.IMP:VX "poloidal pinch velocity, inboard midplane'"
\TIMEDEP::TOP.IMP:VY "radial pinch velocity, inboard midplane"

\TIMEDEP::TOP.INPUT
\TIMEDEP::TOP.INPUT:DS "S-S_sep (input profile location)"

\TIMEDEP::TOP.KE "electron thermal diffusivity"
\TIMEDEP::TOP.KE:IMPSEP
\TIMEDEP::TOP.KE:OMPSEP "electron thermal diffusivity, outer midplane"

\TIMEDEP::TOP.KI "ion thermal diffusivity"
\TIMEDEP::TOP.KI:IMPSEP
\TIMEDEP::TOP.KI:OMPSEP "ion thermal diffusivity, outer midplane"

\TIMEDEP::TOP.NE
\TIMEDEP::TOP.NE:IMPSEP
\TIMEDEP::TOP.NE:OMPSEP "separatrix electron density, outer midplane"
\TIMEDEP::TOP.NE:PWDENMAX
\TIMEDEP::TOP.NE:TARG1MAX "maximum electron density, inboard divertor"
\TIMEDEP::TOP.NE:TARG1SEP "separatrix electron density, inboard divertor"
\TIMEDEP::TOP.NE:TARG2MAX
\TIMEDEP::TOP.NE:TARG2SEP
\TIMEDEP::TOP.NE:TARG3MAX
\TIMEDEP::TOP.NE:TARG3SEP
\TIMEDEP::TOP.NE:TARG4MAX "maximum electron density, outboard divertor"
\TIMEDEP::TOP.NE:TARG4SEP "separatrix electron density, outboard divertor"

\TIMEDEP::TOP.OMP
\TIMEDEP::TOP.OMP:AN "atom density, outboard midplane"
\TIMEDEP::TOP.OMP:DN "diffusion coefficient, outboard midplane"
\TIMEDEP::TOP.OMP:DP "pressure diffusion coefficient, outboard midplane"
\TIMEDEP::TOP.OMP:DS "S-S_sep (outer midplane)"
\TIMEDEP::TOP.OMP:KE
\TIMEDEP::TOP.OMP:KI "ion thermal diffusivity, outboard midplane"
\TIMEDEP::TOP.OMP:LH "Conductive flux limit factor for neutrals, outer midplane"
\TIMEDEP::TOP.OMP:LN "Convective flux limit factor for neutrals, outer midplane"
\TIMEDEP::TOP.OMP:MN "molecule density, outboard midplane"
\TIMEDEP::TOP.OMP:NE "electron density, outboard midplane"
\TIMEDEP::TOP.OMP:PO "potential, outboard midplane"
\TIMEDEP::TOP.OMP:TE "electron temperature, outboard midplane"
\TIMEDEP::TOP.OMP:TI "ion temperature, outboard midplane"
\TIMEDEP::TOP.OMP:VS "viscosity coefficient, outboard midplane'"
\TIMEDEP::TOP.OMP:VX "poloidal pinch velocity, outboard midplane'"
\TIMEDEP::TOP.OMP:VY "radial pinch velocity, outboard midplane"

\TIMEDEP::TOP.PO
\TIMEDEP::TOP.PO:IMPSEP
\TIMEDEP::TOP.PO:OMPSEP "separatrix potential, outer midplane"
\TIMEDEP::TOP.PO:PWDENMAX
\TIMEDEP::TOP.PO:TARG1MAX "maximum potential, inboard divertor"
\TIMEDEP::TOP.PO:TARG1SEP "separatrix potential, inboard divertor"
\TIMEDEP::TOP.PO:TARG2MAX
\TIMEDEP::TOP.PO:TARG2SEP
\TIMEDEP::TOP.PO:TARG3MAX
\TIMEDEP::TOP.PO:TARG3SEP
\TIMEDEP::TOP.PO:TARG4MAX "maximum potential, outboard divertor "

```

```

\TIMEDEP::TOP.PO:TARG4SEP "separatrix potential, outboard divertor max"

\TIMEDEP::TOP.TARGET1
\TIMEDEP::TOP.TARGET1:AN "atom density, inboard divertor"
\TIMEDEP::TOP.TARGET1:DA "Area element, target1 (usually inner target)"
\TIMEDEP::TOP.TARGET1:DS "S-S_sep, target1 (usually inner target)"
\TIMEDEP::TOP.TARGET1:FC "poloidal current, inboard divertor "
\TIMEDEP::TOP.TARGET1:FE "poloidal electron energy flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:FI "poloidal ion energy flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:FL "poloidal electron flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:FN "poloidal spec one flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:FO "poloidal ion flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:FT "poloidal total energy flux, inboard divertor"
\TIMEDEP::TOP.TARGET1:MN "molecule density, inboard divertor"
\TIMEDEP::TOP.TARGET1:NE "electron density, inboard divertor"
\TIMEDEP::TOP.TARGET1:PO "potential, inboard divertor"
\TIMEDEP::TOP.TARGET1:TE "electron temperature, inboard divertor"
\TIMEDEP::TOP.TARGET1:TI "ion temperature, inboard divertor"
\TIMEDEP::TOP.TARGET1:TP "Target surface temperature, inboard divertor"

\TIMEDEP::TOP.TARGET2
\TIMEDEP::TOP.TARGET2:AN
\TIMEDEP::TOP.TARGET2:DA
\TIMEDEP::TOP.TARGET2:DS
\TIMEDEP::TOP.TARGET2:FC
\TIMEDEP::TOP.TARGET2:FE
\TIMEDEP::TOP.TARGET2:FI
\TIMEDEP::TOP.TARGET2:FL
\TIMEDEP::TOP.TARGET2:FN
\TIMEDEP::TOP.TARGET2:FO
\TIMEDEP::TOP.TARGET2:FT
\TIMEDEP::TOP.TARGET2:MN
\TIMEDEP::TOP.TARGET2:NE
\TIMEDEP::TOP.TARGET2:PO
\TIMEDEP::TOP.TARGET2:TE
\TIMEDEP::TOP.TARGET2:TI
\TIMEDEP::TOP.TARGET2:TP "Target surface temperature"

\TIMEDEP::TOP.TARGET3
\TIMEDEP::TOP.TARGET3:AN
\TIMEDEP::TOP.TARGET3:DA
\TIMEDEP::TOP.TARGET3:DS
\TIMEDEP::TOP.TARGET3:FC
\TIMEDEP::TOP.TARGET3:FE
\TIMEDEP::TOP.TARGET3:FI
\TIMEDEP::TOP.TARGET3:FL
\TIMEDEP::TOP.TARGET3:FN
\TIMEDEP::TOP.TARGET3:FO
\TIMEDEP::TOP.TARGET3:FT
\TIMEDEP::TOP.TARGET3:MN
\TIMEDEP::TOP.TARGET3:NE
\TIMEDEP::TOP.TARGET3:PO
\TIMEDEP::TOP.TARGET3:TE
\TIMEDEP::TOP.TARGET3:TI
\TIMEDEP::TOP.TARGET3:TP "Target surface temperature"

\TIMEDEP::TOP.TARGET4
\TIMEDEP::TOP.TARGET4:AN "atom density, outboard divertor"
\TIMEDEP::TOP.TARGET4:DA "Area element, target4 (usually outer target)"
\TIMEDEP::TOP.TARGET4:DS "S-S_sep, target4 (usually outer target)"
\TIMEDEP::TOP.TARGET4:FC "poloidal current, outboard divertor "
\TIMEDEP::TOP.TARGET4:FE "poloidal electron energy flux, outboard divertor"
\TIMEDEP::TOP.TARGET4:FI "poloidal ion energy flux, outboard divertor"
\TIMEDEP::TOP.TARGET4:FL "poloidal electron flux, outboard divertor"
\TIMEDEP::TOP.TARGET4:FN "poloidal spec one flux, outboard divertor"
\TIMEDEP::TOP.TARGET4:FO "poloidal ion flux, outboard divertor"
\TIMEDEP::TOP.TARGET4:FT "poloidal total energy flux, outboard divertor"

```

```

\TIMEDEP::TOP.TARGET4:MN "molecule density, outboard divertor"
\TIMEDEP::TOP.TARGET4:NE "electron density, outboard divertor"
\TIMEDEP::TOP.TARGET4:PO "potential, outboard divertor"
\TIMEDEP::TOP.TARGET4:TE "electron temperature, outboard divertor"
\TIMEDEP::TOP.TARGET4:TI "ion temperature, outboard divertor"
\TIMEDEP::TOP.TARGET4:TP "Target surface temperature, outboard divertor"

\TIMEDEP::TOP.TE
\TIMEDEP::TOP.TE:IMPSEP
\TIMEDEP::TOP.TE:OMPSEP "separatrix electron temperature, outer midplane"
\TIMEDEP::TOP.TE:PWDENMAX
\TIMEDEP::TOP.TE:TARG1MAX "maximum electron temperature, inboard divertor"
\TIMEDEP::TOP.TE:TARG1SEP "separatrix electron temperature, inboard divertor"
\TIMEDEP::TOP.TE:TARG2MAX
\TIMEDEP::TOP.TE:TARG2SEP
\TIMEDEP::TOP.TE:TARG3MAX
\TIMEDEP::TOP.TE:TARG3SEP
\TIMEDEP::TOP.TE:TARG4MAX "maximum electron temperature, outboard divertor"
\TIMEDEP::TOP.TE:TARG4SEP "separatrix electron temperature, outboard divertor"

\TIMEDEP::TOP.TI
\TIMEDEP::TOP.TI:IMPSEP
\TIMEDEP::TOP.TI:OMPSEP "separatrix ion temperature, outer midplane"
\TIMEDEP::TOP.TI:PWDENMAX
\TIMEDEP::TOP.TI:TARG1MAX "maximum ion temperature, inboard divertor"
\TIMEDEP::TOP.TI:TARG1SEP "separatrix ion temperature, inboard divertor"
\TIMEDEP::TOP.TI:TARG2MAX
\TIMEDEP::TOP.TI:TARG2SEP
\TIMEDEP::TOP.TI:TARG3MAX
\TIMEDEP::TOP.TI:TARG3SEP
\TIMEDEP::TOP.TI:TARG4MAX "maximum ion temperature, outboard divertor "
\TIMEDEP::TOP.TI:TARG4SEP "separatrix ion temperature, outboard divertor"

\TIMEDEP::TOP.TP
\TIMEDEP::TOP.TP:TARG1MAX "Maximum surface temperature along target, inboard divertor"
\TIMEDEP::TOP.TP:TARG1SEP "Target surface temperature at strike point, inboard divertor"
\TIMEDEP::TOP.TP:TARG2MAX "Maximum surface temperature along target"
\TIMEDEP::TOP.TP:TARG2SEP "Target surface temperature at strike point"
\TIMEDEP::TOP.TP:TARG3MAX "Maximum surface temperature along target"
\TIMEDEP::TOP.TP:TARG3SEP "Target surface temperature at strike point"
\TIMEDEP::TOP.TP:TARG4MAX "Maximum surface temperature along target, outboard divertor"
\TIMEDEP::TOP.TP:TARG4SEP "Target surface temperature at strike point, outboard divertor"

\TIMEDEP::TOP.VS "viscosity coefficient"
\TIMEDEP::TOP.VS:IMPSEP
\TIMEDEP::TOP.VS:OMPSEP "viscosity coefficient, outer midplane"

\TIMEDEP::TOP.VX "poloidal pinch velocity"
\TIMEDEP::TOP.VX:IMPSEP
\TIMEDEP::TOP.VX:OMPSEP "poloidal pinch velocity, outer midplane"

\TIMEDEP::TOP.VY "radial pinch velocity"
\TIMEDEP::TOP.VY:IMPSEP
\TIMEDEP::TOP.VY:OMPSEP "radial pinch velocity, outer midplane"
\TIMEDEP::TOP:HALCORE
\TIMEDEP::TOP:HALDIV
\TIMEDEP::TOP:HALSOL
\TIMEDEP::TOP:TIME "time"

```

6.1.6 MOVIES

```
\MOVIES::TOP "Saved slices of quantities for movies"
\MOVIES::TOP:DAB2 "Atom density"
\MOVIES::TOP:DMB2 "Molecule density"
\MOVIES::TOP:NA "Ion density"
\MOVIES::TOP:NE "Electron density"
\MOVIES::TOP:RQAHESUM "Electron cooling rate (summed over all species)"
\MOVIES::TOP:RQRADSUM "Radiation rate"
\MOVIES::TOP:TE "Electron temperature"
\MOVIES::TOP:TI "Ion temperature"
\MOVIES::TOP:TIME "Time base"
```

6.1.7 DIAG

```
\DIAG::TOP "Top level for simulated experimental diagnostic signals"
```

```
\DIAG::TOP.AUG "AUG simulated experimental diagnostic signals'"
```

```
\DIAG::TOP.AUG.BLS_II_DPC
\DIAG::TOP.AUG.BLS_II_DPC:AHAL_MHAL
\DIAG::TOP.AUG.BLS_II_DPC:CIII_4650
\DIAG::TOP.AUG.BLS_II_DPC:CII_6581
\DIAG::TOP.AUG.BLS_II_DPC:H0_4340
\DIAG::TOP.AUG.BLS_II_DPC:H0_4861
\DIAG::TOP.AUG.BLS_II_DPC:H0_6561
\DIAG::TOP.AUG.BLS_II_DPC:HALPHA
\DIAG::TOP.AUG.BLS_II_DPC:HBETA
\DIAG::TOP.AUG.BLS_II_DPC:HGAMMA
```

```
\DIAG::TOP.AUG.LAST10
\DIAG::TOP.AUG.LAST10:AN3DA
\DIAG::TOP.AUG.LAST10:AN3DI
\DIAG::TOP.AUG.LAST10:AN3DL
\DIAG::TOP.AUG.LAST10:AN3DR
\DIAG::TOP.AUG.LAST10:DN3DA
\DIAG::TOP.AUG.LAST10:DN3DI
\DIAG::TOP.AUG.LAST10:DP3DA
\DIAG::TOP.AUG.LAST10:DP3DI
\DIAG::TOP.AUG.LAST10:FC3DL
\DIAG::TOP.AUG.LAST10:FC3DR
\DIAG::TOP.AUG.LAST10:FE3DL
\DIAG::TOP.AUG.LAST10:FE3DR
\DIAG::TOP.AUG.LAST10:FI3DL
\DIAG::TOP.AUG.LAST10:FI3DR
\DIAG::TOP.AUG.LAST10:FL3DL
\DIAG::TOP.AUG.LAST10:FL3DR
\DIAG::TOP.AUG.LAST10:FN3DL
\DIAG::TOP.AUG.LAST10:FN3DR
\DIAG::TOP.AUG.LAST10:FO3DL
\DIAG::TOP.AUG.LAST10:FO3DR
\DIAG::TOP.AUG.LAST10:FT3DL
\DIAG::TOP.AUG.LAST10:FT3DR
\DIAG::TOP.AUG.LAST10:KE3DA
\DIAG::TOP.AUG.LAST10:KE3DI
\DIAG::TOP.AUG.LAST10:KI3DA
\DIAG::TOP.AUG.LAST10:KI3DI
\DIAG::TOP.AUG.LAST10:MN3DA
\DIAG::TOP.AUG.LAST10:MN3DI
\DIAG::TOP.AUG.LAST10:MN3DL
\DIAG::TOP.AUG.LAST10:MN3DR
\DIAG::TOP.AUG.LAST10:NE3DA
```

```

\DIAG::TOP.AUG.LAST10:NE3DI
\DIAG::TOP.AUG.LAST10:NE3DL
\DIAG::TOP.AUG.LAST10:NE3DR
\DIAG::TOP.AUG.LAST10:PO3DA
\DIAG::TOP.AUG.LAST10:PO3DI
\DIAG::TOP.AUG.LAST10:PO3DL
\DIAG::TOP.AUG.LAST10:PO3DR
\DIAG::TOP.AUG.LAST10:TE3DA
\DIAG::TOP.AUG.LAST10:TE3DI
\DIAG::TOP.AUG.LAST10:TE3DL
\DIAG::TOP.AUG.LAST10:TE3DR
\DIAG::TOP.AUG.LAST10:TI3DA
\DIAG::TOP.AUG.LAST10:TI3DI
\DIAG::TOP.AUG.LAST10:TI3DL
\DIAG::TOP.AUG.LAST10:TI3DR
\DIAG::TOP.AUG.LAST10:TP3DL
\DIAG::TOP.AUG.LAST10:TP3DR
\DIAG::TOP.AUG.LAST10:VS3DA
\DIAG::TOP.AUG.LAST10:VS3DI
\DIAG::TOP.AUG.LAST10:VX3DA
\DIAG::TOP.AUG.LAST10:VX3DI
\DIAG::TOP.AUG.LAST10:VY3DA
\DIAG::TOP.AUG.LAST10:VY3DI

\DIAG::TOP.AUG.Z_0_2
\DIAG::TOP.AUG.Z_0_2:AHAL_MHAL
\DIAG::TOP.AUG.Z_0_2:CIII_4650
\DIAG::TOP.AUG.Z_0_2:CII_6581
\DIAG::TOP.AUG.Z_0_2:H0_4340
\DIAG::TOP.AUG.Z_0_2:H0_4861
\DIAG::TOP.AUG.Z_0_2:H0_6561
\DIAG::TOP.AUG.Z_0_2:HALPHA
\DIAG::TOP.AUG.Z_0_2:HBETA
\DIAG::TOP.AUG.Z_0_2:HGAMMA
\DIAG::TOP.AUG:DIA
\DIAG::TOP.AUG:DIA:AHAL_MHAL
\DIAG::TOP.AUG:DIA:CIII_4650
\DIAG::TOP.AUG:DIA:CII_6581
\DIAG::TOP.AUG:DIA:H0_4340
\DIAG::TOP.AUG:DIA:H0_4861
\DIAG::TOP.AUG:DIA:H0_6561
\DIAG::TOP.AUG:DIA:HALPHA
\DIAG::TOP.AUG:DIA:HBETA
\DIAG::TOP.AUG:DIA:HGAMMA

```

6.2 Accessing data from the MDSPLUS server

6.2.1 Using standard experimental data viewing programs

The MDSplus server name is `solps-mdsplus.aug.ipp.mpg.de:8001`.

The tree name is `solps`.

An example shotnumber is 4293.

Example signal names are

`top.timedep.ne:ompsep`,

`top.timedep.te:ompsep` (Outer midplane electron density and temperature), and

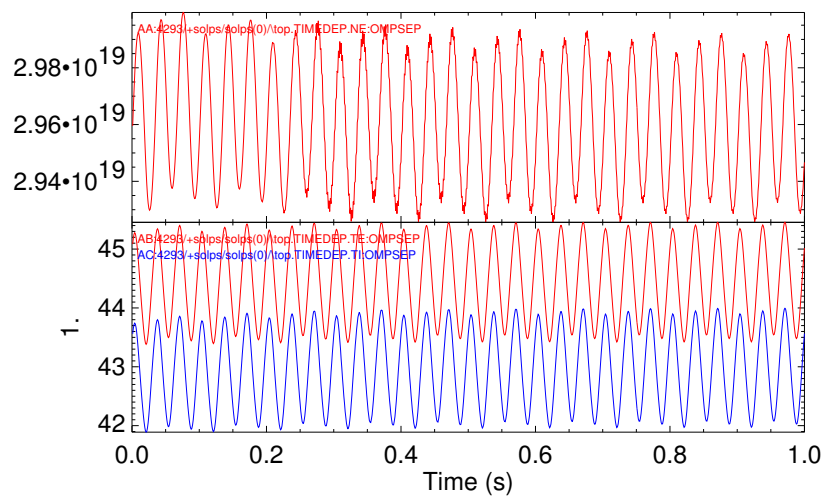
`top.timedep.ti:ompsep` (Outer midplane ion temperature).

cvview

- start cvview
- Shot = 4293
- In the table:
 - shot = 0
 - shot = +solps
 - diag = solps
 - name =
top.timedep.ne:ompsep

Adding

top.timedep.ne:ompsep and
top.timedep.ne:ompsep, the graph in figure 6.1 is produced.



cvview (gdc) v3.17 - User: dpc - Mon Mar 17 16:35:21 2003 No standard set file loaded : 4293

Figure 6.1: Using CVIEW to access SOLPS MDSplus data.

jetdsp

- start JETDSP
- File|Read Signal
- MDS+ option
- MDS+ server: = solps-mdsplus.aug.ipp.mpg.de:8001
- Shot number: = 4293
- Tree: = solps
- Signal: =
top.timedep.ne:ompsep

- Read button

You can also use

- start JETDSP
- File|Read Signal
- MDS+ option
- MDS+ server: = solps-mdsplus.aug.ipp.mpg.de:8001
- Shot number: = 4293
- Tree: = timedep
- Signal: =
top.ne:ompsep
- Read button

Adding te:ompsep and ti:ompsep, the graph in figure 6.2 is produced.

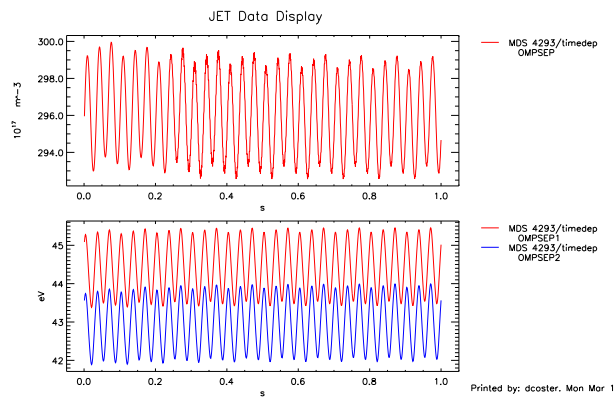


Figure 6.2: Using JETDSP to access SOLPS MDSplus data.

reviewplus

6.2.2 Using IDL

```
mdsconnect,'solps-mdsplus.aug.ipp.mpg.de:8001'
mdsopen,'solps',251
te_coarse=mdsvalue('\top.snapshot:te')
cr_coarse=mdsvalue('\top.snapshot.grid:cr')
mdsopen,'solps',288
te_fine=mdsvalue('\top.snapshot:te')
cr_fine=mdsvalue('\top.snapshot.grid:cr')
mdsclose
mdsdisconnect
plot,cr_coarse[:,8],te_coarse[:,8],linestyle=1,psym=-1,yrange=[0,50]
oplot,cr_fine[:,10],te_fine[:,10],linestyle=2,psym=-2
```

6.2.3 Using MATLAB

```
mdsopen('solps-mdsplus.aug.ipp.mpg.de:8001::solps',nshot);
onx= mdsvalue('\SOLPS::TOP.SNAPSHOT.DIMENSIONS:NX');
ny = mdsvalue('\SOLPS::TOP.SNAPSHOT.DIMENSIONS:NY'); %number of radial cells
ne = mdsvalue('\SOLPS::TOP.SNAPSHOT:NE');
te = mdsvalue('\SOLPS::TOP.SNAPSHOT:TE');
ti = mdsvalue('\SOLPS::TOP.SNAPSHOT:TI');
sep= mdsvalue('\SOLPS::TOP.TIMEDEP.OMP:DS');
qe = 1.602e-19; %Elementary charge in C
amu= 1.660542e-27; %Atomic mass unit in kg

%Calculate the ion sound speed at outer target
cs2o = qe .* (ti(onx-1,:) + te(onx-1,:)) ./ (2*amu);

%Outer target
neo = ne(onx-1,:);
osja = neo .* qe .* sqrt(cs2o) / 1.0e4;

cr = mdsvalue('\SOLPS::TOP.SNAPSHOT.GRID:CR'); %R-coordinate of cell centre
ocr= cr(onx,:); %R coordinates of outer target cells
cz = mdsvalue('\SOLPS::TOP.SNAPSHOT.GRID:CZ'); %Z-coordinate of cell centre
ocz= cz(onx,:); %Z coordinate of outer target cells
mdsclose('MDSIP');

[map2mid,oms]=jhmap2mid(26730,ocr,ocz,0.7);
rrsep= map2mid.Rmid-map2mid.Rlcfs;

plot(rrsep,osja,'k');
```

6.2.4 Using Python

You will need the MDSplus interface for python.

```
import pmds
from pylab import *

pmds.mdsconnect ('solps-mdsplus.aug.ipp.mpg.de:8001')
pmds.mdsopen('solps',12380)
r=pmds.mdsvalue('\top.snapshot.grid:r')
z=pmds.mdsvalue('\top.snapshot.grid:z')
pmds.mdsdisconnect ()

print(r)
print(z)
```

For additional information about using python, see

https://wiki.eu-euforia.eu/index.php/Fusion_dataset_with_Python_and_VisIt
(this activity occurred within the EUFORIA project, <http://www.euforia-project.eu/>).

Typing

```
python
import compare_shots
```

with the following as `compare_shots.py`:

```
'''
Date : 05/12/2008
Authors : Matthieu Haeefe based on IDL scripts provided by David Coster

This script demonstrates the capabilities of Python/NumPy/matplotlib tools to show fusion data.
Basically it reads datasets from solps server through MDS+ and either display or write a
    resulting
picture on disk thanks to matplotlib.

It provides the function compare_shots which reads ID datasets from a list of shots on a server
through MDS+ and either displays or writes a resulting picture on disk thanks to matplotlib.
The picture consists in a 9X9 plot on which data from all the specified shots are displayed. If
filename is omitted, a X window pop up, if filename is specified, no window pop up, but a picture
is generated on disk with name filename.
'''

import pmds
import numpy

# matplotlib.pyplot is a collection of command style functions that make matplotlib work like
    matlab
import matplotlib.pyplot as plt

# matplotlib.ticker contains classes to support completely configurable tick locating and
    formatting.
import matplotlib.ticker as ticker

def compare_shots(shot, filename=None, server='solps-mdsplus.aug.ipp.mpg.de:8001'):
    '''
    This function reads ID datasets from shot <shot> on solps server <server> through MDS+ and either
    displays or writes a resulting picture on disk thanks to matplotlib.
    * shot : python list of integer which are the shots. Mandatory argument
    * filename : python string which contains the name of the generated file. Optional argument.
    * server : MDS+ server to connect. Optional argument, default value=solps-mdsplus.aug.ipp.mpg.de
        :8001
    '''

    params = {'axes.labelsize': 10,
              'axes.titlesize': 10,
              'xtick.labelsize': 4,
              'ytick.labelsize': 4,
              'lines.markersize': 3,
              'text.usetex': True
             }

    if filename == None:
        # No filename supplied, displaying on screen
        params['backend'] = 'QtAgg'
    else:
        # Filename provided, then it's a png
        params['backend'] = 'png'

    plt.rcParams.update(params)

    # compare_shots is called with an array of shots. Let's get its length
    ncase = len(shot)

    # Let's create a few identical arrays whose dimensions and type are in arrayspec
    arrayspec=(ncase,numpy.int32)

    # Creating arrays expanding array specification
    nx = numpy.zeros(*arrayspec)
    ny = numpy.zeros(*arrayspec)
```

```

ns = numpy.zeros(*arrayspec)
imp = numpy.zeros(*arrayspec)
omp = numpy.zeros(*arrayspec)
sep = numpy.zeros(*arrayspec)

pmds.mdsconnect(server)
for i in range(ncase):
    pmds.mdsopen('solps', shot[i])
    nx[i]=pmds.mdsvalue('\top.snapshot.dimensions:nx')
    ny[i]=pmds.mdsvalue('\top.snapshot.dimensions:ny')
    print 'opening shot '+ str(shot[i]) + ' => nx, ny='+str(nx[i]) + ' ' + str(ny[i])
    # Someday maybe, all this will work in Python 3. One shall then write
    # print(''opening shot '+ str(shot[i]) + ' => nx, ny='+str(nx[i]) + ' ' + str(ny[
        i]))

    ns[i]=pmds.mdsvalue('\top.snapshot.dimensions:ns')
    imp[i]=pmds.mdsvalue('\top.snapshot.dimensions:imp')
    omp[i]=pmds.mdsvalue('\top.snapshot.dimensions:omp')
    sep[i]=pmds.mdsvalue('\top.snapshot.dimensions:sep')
    pmds.mdsfclose('solps', shot[i])

# Another array specification
arrayspec=((ncase,ny[0],nx[0]),numpy.float64)

# Let's create these float arrays
dsrad = numpy.zeros(*arrayspec)
dspol = numpy.zeros(*arrayspec)
dspar = numpy.zeros(*arrayspec)
te     = numpy.zeros(*arrayspec)
ti     = numpy.zeros(*arrayspec)
nel    = numpy.zeros(*arrayspec)

for i in range(ncase):
    pmds.mdsopen('solps', shot[i])
    dsrad[i,:,:]=pmds.mdsvalue('\top.snapshot.grid:dsrad')
    dspol[i,:,:]=pmds.mdsvalue('\top.snapshot.grid:dspol')
    dspar[i,:,:]=pmds.mdsvalue('\top.snapshot.grid:dspar')
    te[i,:,:]=pmds.mdsvalue('\top.snapshot:te')
    ti[i,:,:]=pmds.mdsvalue('\top.snapshot:ti')
    nel[i,:,:]=pmds.mdsvalue('\top.snapshot:ne')
    pmds.mdsfclose('solps', shot[i])

pmds.mdsdisconnect()

#build the figure (Let's go Picasso)
# Layout is a list composed of
# [number of sub-plot, legend of the left axis, data, location in the tokamak (Inner
    target, Outer midplane or Outer target), subplot upper title, subplot lower title ]
layout=[[1,'Te',te, 0, 'Inner target', ''],
        [2,'',te,omp[0], 'Outer midplane', ''],
        [3,'',te,nx[0]-1, 'Outer target', ''],
        [4,'Ti',ti, 0, '', ''],
        [5,'',ti,omp[0], '', ''],
        [6,'',ti,nx[0]-1, '', ''],
        [7,'ne',nel, 0, '', 'distance wrt sep'],
        [8,'',nel,omp[0], '', 'distance wrt sep'],
        [9,'',nel,nx[0]-1, '', 'distance wrt sep']]

# Matplotlib plot definitions : point shape, linked or not, color
line_style = ['o-b', '^-g', 's-r', '+-c', 'x-m', 'D-y', 'l-k', 'h-']

```

```

# sub-plot number j[0] in a 3x3 subplot. last number ranges from 1 to 3x3, line-major
print 'generating figures'
for j in layout:
    plotgridshape=(3,3,j[0])
    cur_fig = plt.subplot(*plotgridshape)

    # Finds with the ticker module nice ticks locations, up to 5 locators
    cur_fig.xaxis.set_major_locator(ticker.MaxNLocator(5))

    cur_fig.set_xlabel(j[5])
    cur_fig.set_ylabel(j[1])
    cur_fig.set_title(j[4])

    # In each subplot, plot all the cases, each with its line style
    for i in range(ncase):
        cur_fig.plot(dsrad[i,:,j[3]],j[2][i,:,j[3]], line_style[i])

# Some more space between sub-plots
plt.subplots_adjust(wspace=0.15, hspace=0.2)

if filename == None:
    plt.show()
else:
    plt.savefig(filename)

compare_shots([11437,11438,11447,11448,11449,11450])

```

produced figure 6.3.

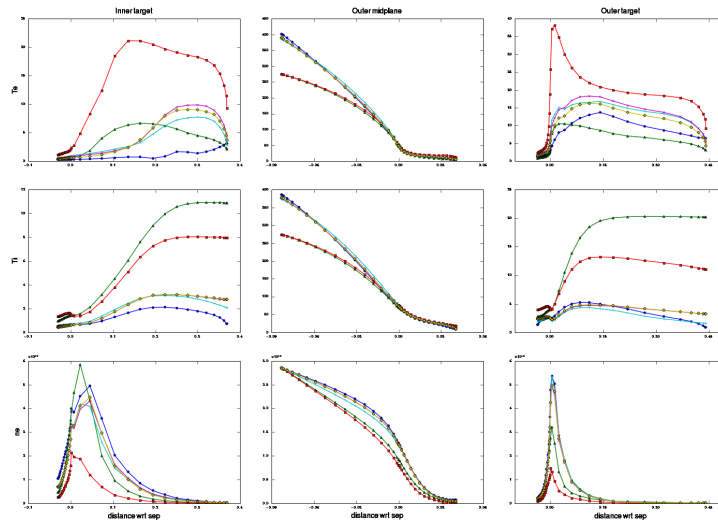


Figure 6.3: Comparison of profiles for a range of shots.

In another, more complicated, example, python was used to create the frames of a movie on the ITM Gateway machine.

```

set c=0
foreach i ( `cat shots` )
    set tbase=`echo $c | awk '{print $1*1000*1e-5}'`
    mkdir $i

```

```

    echo Submitting $i with offset $tbase
    ( cd $i ; sed -e "s:#1#:${i}:g" -e "s:#2#:${tbase}:g" < ../QSUB.shot | bsub )
    @ c++
    sleep 5
end

```

with QSUB.shot containing

```

#!/bin/tcsh

#BSUB -e PYTHON.e%J.%I
#BSUB -o PYTHON.o%J.%I
#BSUB -J PYTHON
#BSUB -W 36:00
#BSUB -q priority

setenv PATH /afs/efda-itm.eu/imp3/user/coster/public/python/bin:$PATH
setenv PYTHONPATH /afs/efda-itm.eu/imp3/user/coster/python:/afs/efda-itm.eu/imp3/user/coster/
    public/python/lib/python2.5/site-packages:/afs/efda-itm.eu/imp3/user/coster/public/python:
    $PYTHONPATH

echo 1200 | awk '{srand();print int($1*srand())}' | xargs sleep
echo "import movie_patch ; movie_patch.movie_patch(shot=#1#,data='rqahesum',fmin=0.0,fmax
    =812649219.41426456,tbase=#2#,rmin=1.155,rmax=1.725,zmin=-1.25,zmax=-0.65)" | python

```

and shots containing

```

18559
18577
18593
18615
18637
18662
18685
18711
18731
18744
18757
18772
18785
18795
18808
18822
18835
18847
18879
18897
18917
18951
18978
19005
19034
19059
19089
19113
19139
19157
19190
19211
19232
19258
19280
19300
19331
19352
19379
19405

```

19432
19456
19485
19523
19540
19575
19605
19628
19658
19690
19710
19732
19770
19793
19827
19856
19896
19928
19950
19971
19994
20025
20051
20076
20268
20269
20301
20323
20349
20379
20405
20445
20473
20516
20537
20559
20588
20613
20638
20663
20687
20710
20732
20763
20780
20806
20824
20845
20869
20890
20911
21116
21117
21118
21119
21120
21121
21160
21161
21182
21203
21224
21245
21266
21287
21308
21329

21350
21371
21392
21414
21435
21459
21478
21499
21520
21541
21562
21583
21604
21625
21646
21669
21840
21841
21842
21843
21845
21847
21848
21850
21868
21886
21904
21929
21958
21995
22025
22052
22079
22106
22125
22160
22188
22215
22240
22268
22308
22336
22408
22452
22486
22539
22568
22611
22649
22685
22710
22745
22767
22803
22831
22872
22902
22935
22971
23024
23058
23103
23141
23179
23209
23239
23269


```

23299
23333
23363
23395
23436
23465
23498
23533
23563
23586
23624
23664
23738
23793
23839
23887
23946
23982
24049
24050
24078
24106
24134
24169
24204

```

This launches one job per SOLPS shot-number. Then the frames were renumbered by

```

mkdir frames
cd frames/
set c=0
foreach i ( `cat ../shots` )
  set p=`echo $c | awk '{printf "%03i\n",$1}'`
  echo $p
  @ c++
  foreach j ( `cd ../$i ; echo tmp_?????.png` )
    ln -s ../$i/$j `echo $j | sed "s:tmp_0:tmp_${p}:"`
  end
end
end

```

The individual frames were then combined into a movie with

```

mencoder 'mf://frames/tmp_??????.png' -mf type=png:fps=30 -ovc lavc -lavcopts \
  vcodec=wmv2 -oac copy -o fb_nesepm=5.0e19.ELM_2cm_X10.rqahesum.mpg

```

The resultant movie can be found at

http://solps-mdsplus.aug.ipp.mpg.de/SOLPS/fb_nesepm=5.0e19.ELM_2cm_X10.rqahesum.mpg

6.3 Querying the database

All cases are also entered into a MySQL (<http://www.mysql.com/>) database.

A Web interface to the database is available at <http://solps-mdsplus.aug.ipp.mpg.de/solps/> via Zope (<http://www.zope.org/>).

The database can also be queried via MySQL commands passed via MDSplus. An IDL example of querying the database to find the number of shots supplied by each user follows:

```

mdsconnect,'solps-mdsplus.aug.ipp.mpg.de:8001'
ans=mdsvalue("sql('select user,count(*) from id group by user','solps')")
mdsdisconnect

```

```
help,ans
print,transpose(ans)
```

Another, more complicated, IDL example is:

```
mdsconnect,'solps-mdsplus.aug.ipp.mpg.de:8001'
ans=mdsvalue("sql('select solpsindex,shot,time,ompsep_te,ompsep_ti,user,exp
      from id where ompsep_te>110.0 and ompsep_te<120.0','solps')")
mdsdisconnect
help,ans
print,format='(i6,1x,i6,1x,f6.3,1x,f6.2,1x,1x,f6.2,1x,a10,1x,a)',transpose(ans)
plot,ans(*,3),ans(*,4),psym=4,xtitle='Te (eV)',ytitle='Ti (eV)'
```

And a similar example in python:

```
python sql_solps.py
```

with sql_solps.py containing

```
import pmds
import numpy

pmds.mdsconnect('solps-mdsplus.aug.ipp.mpg.de:8001')
shape=pmds.mdsvalue("shape(_s=sql(\`select solpsindex,shot,time,ompsep_ne,ompsep_te,user,exp from
      id where ompsep_te>10.0 and ompsep_te<1000.0\`,`solps`)")")
tmp=pmds.mdsvalue("_s")
ans=numpy.array(tmp).reshape(shape[:-1])
solpsindex=numpy.array(ans[0,:].tolist()).astype('i8')
shot=numpy.array(ans[1,:].tolist()).astype('i8')
time=numpy.array(ans[2,:].tolist()).astype('f8')
ompsep_ne=numpy.array(ans[3,:].tolist()).astype('f8')
ompsep_te=numpy.array(ans[4,:].tolist()).astype('f8')

import matplotlib.cm as cm
import matplotlib.pyplot as plt

plt.loglog(ompsep_ne,ompsep_te,'o')
plt.savefig('sql_scatter.png',dpi=200)
plt.clf()

plt.hexbin(ompsep_ne,ompsep_te,xscale='log',yscale='log',cmap=cm.OrRd)
plt.title("SOLPS")
plt.xlabel('ompsep_ne')
plt.ylabel('ompsep_te')
cb = plt.colorbar()
cb.set_label('N')
plt.savefig('sql_hexbin_lin.png',dpi=200)
plt.clf()

plt.hexbin(ompsep_ne,ompsep_te,bins='log',xscale='log',yscale='log',cmap=cm.OrRd)
plt.title("SOLPS")
plt.xlabel('ompsep_ne')
plt.ylabel('ompsep_te')
cb = plt.colorbar()
cb.set_label('log10(N)')
plt.savefig('sql_hexbin_log.png',dpi=200)
plt.clf()
```

The resultant plots are given in figures 6.4, 6.5 and 6.6.

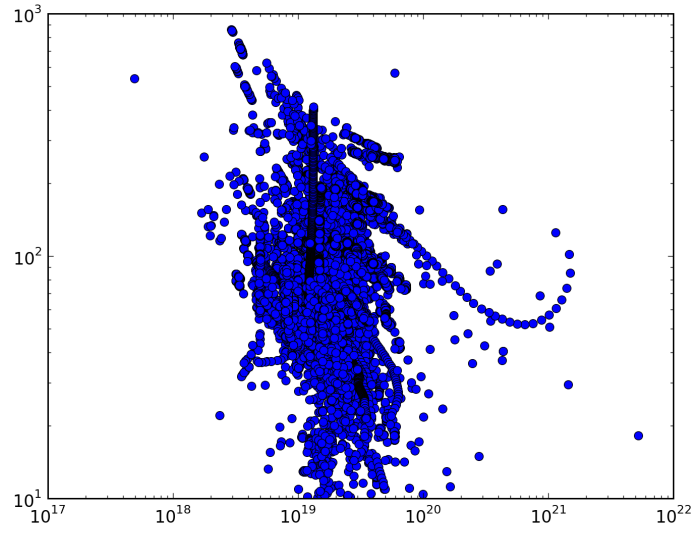


Figure 6.4: Comparison of profiles for a range of shots.

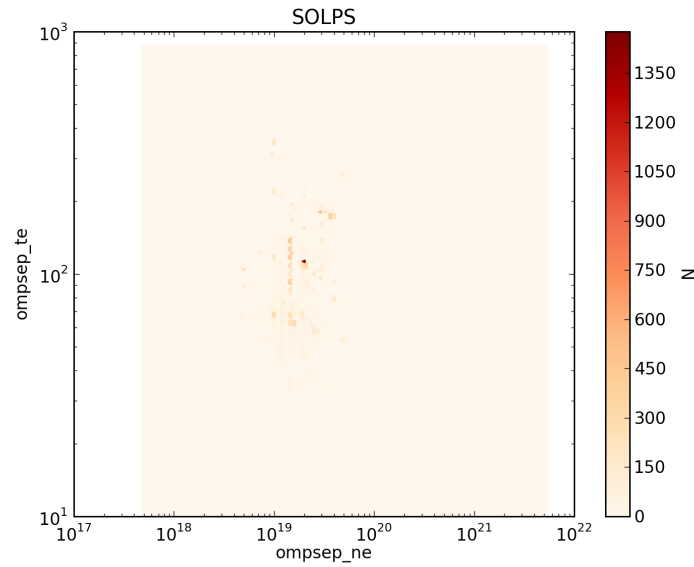


Figure 6.5: Comparison of profiles for a range of shots.

6.4 Coupling to other codes via MDSplus

Coupling of SOLPS to other codes has in the past been done mostly by exchanging files. The preferred route for the future is to use the SOLPS MDSplus tree.

As an example of how to do this, a short example FORTRAN code which queries the SOLPS MDSplus server and then writes geometry information in the standard Sonnet format is presented below. The code uses Wolfgang Suttrop's libitdb library (<http://www.ipp.mpg.de/Wolfgang.Suttrop/mdsplus/libitdb/>).

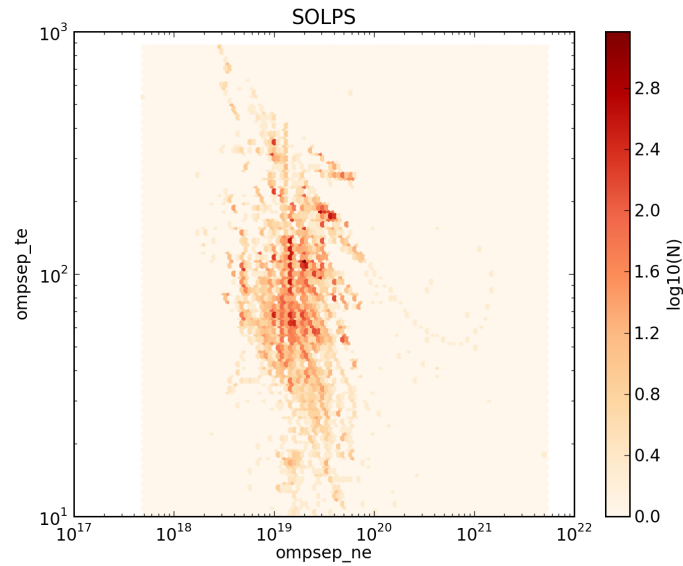


Figure 6.6: Comparison of profiles for a range of shots.

```

program write_files
implicit none
integer nxdmax, nydmax
parameter (nxdmax=1024, nydmax=1024)
integer ishot, status, iargc, lng,
1 sizes(4), ndim, ndim1, ndim2, ndim3,
2 icell, ix, iy, nx, ny
character arg*256, string*256
real*8 cr(-1:nxdmax, -1:nydmax), cz(-1:nxdmax, -1:nydmax),
1 crx(-1:nxdmax, -1:nydmax, 0:3), cry(-1:nxdmax, -1:nydmax, 0:3),
2 bb(-1:nxdmax, -1:nydmax, 0:3),
3 dum1(1), dum2(1), dum3(1), b0r0

if(iargc().ne.1) then
  write(*,*) 'One argument expected, not ',iargc()
  stop
endif
call getarg(1,arg)
read(arg,*) ishot

call itdb_open(status,"solps-mdsplus.aug.ipp.mpg.de:8001",
1 "solps", ishot)
call itdb_err_msg('itdb_open',status)

lng=len(string)
call itdb_gettext(status,"\TOP.IDENT:USER",string,lng)
call itdb_err_msg('itdb_gettext',status)
write(*,*) string(1:lng)

ndim=2
ndim1=0                      ! nxdmax+2

```

```

sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2
ndim3=0                ! 1+3
sizes(3)=1+3

ndim=2
ndim1=0                ! nxdmax+2
sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2
ndim3=0                ! 1+3
CALL itdb_getsig(status,' \TOP.SNAPSHOT.GRID:CR',
*      ndim,sizes,cr,ndim1,dum1,ndim2,dum2,ndim3,dum3)
call itdb_err_msg(' itdb_getsig',status)
write(*,*) 'CR: ',ndim,sizes(1:ndim),ndim1,ndim2,ndim3
ndim=2
ndim1=0                ! nxdmax+2
sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2
ndim3=0                ! 1+3
CALL itdb_getsig(status,' \TOP.SNAPSHOT.GRID:CZ',
*      ndim,sizes,cz,ndim1,dum1,ndim2,dum2,ndim3,dum3)
call itdb_err_msg(' itdb_getsig',status)
write(*,*) 'CZ: ',ndim,sizes(1:ndim),ndim1,ndim2,ndim3
ndim=3
ndim1=0                ! nxdmax+2
sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2
ndim3=0                ! 1+3
sizes(3)=1+3
CALL itdb_getsig(status,' \TOP.SNAPSHOT.GRID:R',
*      ndim,sizes,crx,ndim1,dum1,ndim2,dum2,ndim3,dum3)
call itdb_err_msg(' itdb_getsig',status)
write(*,*) 'R: ',ndim,sizes(1:ndim),ndim1,ndim2,ndim3
ndim=3
ndim1=0                ! nxdmax+2
sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2
ndim3=0                ! 1+3
sizes(3)=1+3
CALL itdb_getsig(status,' \TOP.SNAPSHOT.GRID:Z',
*      ndim,sizes,cry,ndim1,dum1,ndim2,dum2,ndim3,dum3)
call itdb_err_msg(' itdb_getsig',status)
write(*,*) 'Z: ',ndim,sizes(1:ndim),ndim1,ndim2,ndim3
ndim=3
ndim1=0                ! nxdmax+2
sizes(1)=nxdmax+2
ndim2=0                ! nydmax+2
sizes(2)=nydmax+2

```

```

ndim3=0                      ! 1+3
sizes(3)=1+3
CALL itdb_getsig(status,'\TOP.SNAPSHOT:B',
*      ndim,sizes,bb,ndim1,dum1,ndim2,dum2,ndim3,dum3)
call itdb_err_msg('itdb_getsig',status)
write(*,*) 'B: ',ndim,sizes(1:ndim),ndim1,ndim2,ndim3

call itdb_close(status)
call itdb_err_msg('itdb_close',status)

nx=sizes(1)-2
ny=sizes(2)-2

open(99,file=trim(arg)//'.sonnet')
icell=0
write(99,*)
write(99,'(3x,a/)') 'Element output:'
b0r0=cr(-1,-1)*bb(-1,-1,3)

write(99,*) '      R*Btor =',b0r0
write(99,'(/3x,2a)')
1 '=====',
2 '=====',
do iy=-1,ny
do ix=-1,nx
write(99,10001) icell,ix+1,iy+1,
1 crx(ix,iy,2),cry(ix,iy,2),crx(ix,iy,3),cry(ix,iy,3)
10001 format(3x,'Element',i5,' = (' ,i3,',',i3,') : (' ,
. 1pe17.10,',',1pe17.10,')',
. 6x,' (' ,1pe17.10,',',1pe17.10,')')
write(99,10002)
1 sqrt(bb(ix,iy,0)**2+bb(ix,iy,1)**2)/bb(ix,iy,3),
2 cr(ix,iy),cz(ix,iy)
10002 format(3x,'Field ratio = ',1pe17.10,13x,
. ' (' ,1pe17.10,',',1pe17.10,')')
write(99,10003)
1 crx(ix,iy,0),cry(ix,iy,0),crx(ix,iy,1),cry(ix,iy,1)
10003 format(
. t30,' (' ,1pe17.10,',',1pe17.10,')',
. 6x,' (' ,1pe17.10,',',1pe17.10,')')
write(99,10005)
10005 format(3x,
1 '-----',
2 '-----')
icell=icell+1
enddo
enddo
close(99)

end

subroutine itdb_err_msg(name,status)
implicit none

```

```

character*(*) name
integer status

if (status.ne.0) then
    write(*,*) 'Routine ',trim(name),' returned error code ',status
endif
return
end

```

6.5 Some information about the server

The IPP MDSplus server is a dual processor 2.8 GHz AMD Opteron machine with 4 GB RAM running SUSE LINUX 10.1 (X86-64). The data is stored in MR-AFS.

As of Wed Jun 1 21:30:57 CEST 2005 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 12497 cases consisting of 387.158 GB in 261099 files.

As of Wed Jul 26 19:12:53 CEST 2006 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 15999 cases consisting of 995.904 GB in 334534 files.

As of Sat Mar 8 00:46:36 CET 2008 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 22190 cases consisting of 2824.17 GB in 464547 files.

As of Tue Oct 7 19:45:57 CEST 2008 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 25623 cases consisting of 3646.37 GB in 536644 files.

As of Tue Jan 13 10:05:51 MET 2009 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 26448 cases consisting of 3802.68 GB in 553970 files.

As of Thu Jun 8 11:22:51 CEST 2017 the SOLPS MDSplus server "solps-mdsplus.aug.ipp.mpg.de:8001" contained 95977 cases consisting of 19859.1 GB in 1929218 files.

Chapter 7

Running DivGeo

You will need to choose your device, *e.g.*:

```
setenv DEVICE iter
```

Available choices are:

aditya ADITYA (Gandhinagar, India)

asdex ASDEX (Garching, Germany), now HL-2A (Chengdu, China)

cfns Compact Fusion Neutron Source (China)

cmod Alcator C-Mod (Boston, USA)

demo DEMO (many incarnations)

d3d DIII-D (San Diego, USA)

dream (Japan)

east EAST (Hefei, China)

globus Globus-2M (St. Petersburg, Russia)

hl2a HL-2A (Chengdu, China)

hl2m HL-2M (Chengdu, China)

iter ITER (Cadarache, France) (default)

iter.dg link to `iter`

iterm ITER (modified)

jet JET (Culham, UK)

jt60sa JT60-SA (Naka, Japan)

jt60u JT60-U (Naka, Japan)

kstar KSTAR (Daejeon, Korea)

mast MAST (Culham, UK)

nhtx NHTX (Hefei, China)

sst SST (Gandhinagar, India)

t15m T15-M (Moscow, Russia)

tcv TCV (Lausanne, Switzerland)

ts Tore Supra (Cadarache, France)

upgrade ASDEX–Upgrade (Garching, Germany)

upgrade.dg link to upgrade

The default setting for `DEVICE` can be changed for each local implementation *via* use of the `setup.csh.*` files listed in section 2.1.

If `DivGeo` is not yet compiled, you will need to do

```
cd $DG
gmake tags
gmake listobj
gmake depend
gmake
cd equtrnr
gmake depend
gmake
cd ../convert
gmake
```

The environment variable `$DG` contains the path `${SOLPSTOP}/modules/DivGeo`. To start DG, the command is:

```
./dg &
```

or

```
./dg case.dg &
```

where `case.dg` is an existing case.

The reader is referred to the DG manual in Appendix K for further information of running DG itself.

After saving (File|Save) and outputting (File|Output) you will then need to

```
./lns case
```

where `case` is the name of the just saved case (without the `.dg` extension). This links the DG output files to files named `dg.*` in the `Carre run` directory.

Then you will probably want to generate the grid — this is described in the next chapter. After the grid has been created, you can import it back into DG (File|Import|Mesh) to check it.

Chapter 8

Running Carre

If Carre is not yet compiled, you will need to do

```
ssc
gmake tags
gmake listobj
gmake depend
gmake
```

With the installation instructions from Section 1.1, users get a local clone of the Carre GIT submodule, and **this is the only supported option!** The files in `src` will be version-controlled and updated whenever the user will pull in new code version. If, however, one wishes to make local changes to a file and protect those from GIT updates, the procedure for that purpose is to copy the file you want to change to `src.local` and then edit that file. Once you have copied and edited the file(s) you need to change in `src.local`, you should recompile with:

```
stop
gmake listobj
gmake tags
gmake depend
gmake (carre)
```

so that the local file is used. The last `gmake` command may of course be replaced by `gmake carre(_debug)`, `gmake solps(_debug)`, `gmake solps_mpi(_debug)`, `gmake all(_debug)` or `gmake all_mpi(_debug)` as needed.

If this local change should no longer be needed, then it should be removed from `src.local` and the following performed:

```
stop
gmake clean(_carre)
gmake listobj
gmake tags
gmake depend
gmake (carre)
```

If you want to avoid a long recompilation, you need to remove from the `builds/*/` directories, the object and module files corresponding to your locally changed file(s), and then ask for a normal recompilation with a single `gmake` command.

To run Carre, assuming that you are working on a case from DG, you would then need to do

```
carre -
```

and then either type `return` to accept the default, or a `y` if asked to accept Carre's choices.

The presence in the run directory of a file called `near.cfg`, with the first character of the first line non-blank, will cause the program to use 'publication-ready' settings for the graphs, namely thicker and visible lines and easier-to-read font.

The program will propose a table (see next chapter for an example) containing, for each region of the grid, the relevant meshing parameters. The first table (titled `SEPARATRIX SEGMENTS`) includes the variables for the poloidal meshing along the separatrix. For a single-null case, segment 1 goes from the inner divertor to the X-point, segment 2 from the outer divertor to the X-point, and segment 3 around the core in the clockwise direction. For more complicated geometries, see the Carre original paper [14]. The variables in the table mean:

nptseg Number of poloidal mesh points along the segment

lg Length of the segment (in m)

deltp1 Distance from the 1st to the 2nd mesh point (The first mesh point is defined to be the X-point) (in m)

deltpn Distance from the penultimate to the last mesh point (The last mesh point is defined to be the strike point) (in m)

dpmin Minimum distance achieved between two neighbouring points (in m)

dpmax Maximum distance achieved between two neighbouring points (in m)

Of the above, the user is only allowed to change `nptseg`, `deltp1` and `deltpn`. If the code, when using the default parameters (which are imported from the DG run), initially fails, re-run trying a different set of parameters. The code in some cases gives some indication of which region and which surface was problematic. For the most common error scenarios, some sanity checks are performed at this stage and the code will not allow the user to continue unless all of these are satisfied. In particular, make sure that for each individual segment `lg`, `deltp1`, `deltpn`, `dpmin` and `dpmax` are all of the same sign. It is recommended to have the sizes of the mesh steps on all sides of the X-points to be comparable, and to increase resolution gradually towards the strike points. The values of `dpmin` and `dpmax` can be used to check the monotonicity of the mesh step size. For reasons related to numerical convergence of the plasma solvers, the grid spacing between neighbouring cells should not change too quickly and must be prevented from collapsing to zero. To this effect, Carre imposes a minimum grid spacing given by the `pasmin` value. However, the spacing of the points as initially provided by DG is unaware of that limit, and thus it is often the case that the user must adapt either the grid spacings or `pasmin` (or both) before being allowed to continue.

The second table (titled `RADIAL DISTRIBUTIONS`) includes the variables for the radial meshing away from the separatrix. For a single-null case, region 1 is the private flux region (PFR), and region 2 the scrape-off layer (SOL). For more complicated geometries, see the Carre original paper [14].

repart Radial repartition of points is done according to distance if `repart=1` (recommended) or according to the magnetic flux function `psi` if `repart=2`. Units for the quantities below are then either m or Wb .

npr Number of radial mesh points in the region

width Width of the region

deltr1 Distance from the 1st to the 2nd mesh point (The first mesh point is defined to be on the separatrix)

deltrn Distance from the penultimate to the last mesh point

drmin Minimum distance achieved between two neighbouring points

drmax Maximum distance achieved between two neighbouring points

The user is only allowed to change `repart`, `npr`, `deltr1` and `deltrn`. Again, make sure that for each individual segment `width`, `deltr1`, `deltrn`, `drmin` and `drmax` are all of the same sign. Regions on opposite sides of the separatrix will usually have opposite signs for their respective `width` values. It is recommended to have the sizes of the mesh steps on both sides of the separatrix be comparable, and to decrease resolution gradually towards the

boundaries. For disconnected double-null cases, the region between the two separatrices must be meshed with special care. The values of `drmin` and `drmax` can be used to check the monotonicity of the mesh step size.

The third table (titled `CENTRAL REGION`) includes the variables for the radial meshing in the core. For a single-null case, this is region 3. The `repart` switch from above applies here as well. For more complicated geometries, see the Carre original paper [14].

pntrat Radial penetration from the separatrix into the core to the innermost meshed flux surface. The O-point is located at the value `pntrat max`.

npr Number of radial mesh points in the core

width Width of the region

deltrl Distance from the 1st to the 2nd mesh point (The first mesh point is defined to be on the separatrix)

deltrn Distance from the penultimate to the last mesh point

drmin Minimum distance achieved between two neighbouring points

drmax Maximum distance achieved between two neighbouring points

The user is only allowed to change `pntrat`, `npr`, `deltrl` and `deltrn`. Again, the code forces the user to ensure that `width`, `deltrl`, `deltrn`, `drmin` and `drmax` are all of the same sign. It is recommended to keep the sizes of the mesh steps on both sides of the separatrix to be comparable (compare with the `RADIAL DISTRIBUTIONS` table), and to decrease resolution gradually towards the inner core boundary and outer regions. The values of `drmin` and `drmax` can be used to check the monotonicity of the mesh step size. For a B2 or B2.5 run, it is necessary that `npr` in the PFR and core regions be the same. This check will be performed by the `b2ag` grid preprocessor and an error message returned if the condition is not satisfied. Similar more complicated checks will be done for double-null geometries.

The fourth table (titled `GUARD LENGTH FOR EACH DIVERTOR PLATE`) gives the poloidal length (in *m*) `tgarde` over which the grid is to be adjusted from strictly orthogonal to coincident with the target geometry. This is a highly device-dependent quantity, but a value of the order of the length of the relevant separatrix segment is recommended. In single-null geometries, `tgarde (1)` refers to the inner target, and `tgarde (2)` to the outer target.

The last table (titled `RELAXATION PARAMETERS USED TO CONSTRUCT THE MESH`) gives numerical parameters used to smooth out the mesh. Carre uses a relaxation scheme to optimize the mesh so as to make it as orthogonal as possible within the given constraints of the field geometry and the shape of the target plates. The parameters governing this scheme are:

nrelax The maximum number of iterations allowed

relax The relaxation factor used

pasmin The minimal spacing required between neighbouring surfaces in the poloidal direction

rlcept The maximum desired deviation from orthogonality (averaged over a flux surface)

If the code fails to position the points poloidally in a way that satisfies the optimization criteria, an error message will be printed showing the deviation from orthogonality obtained and options likely to improve it. The user may ignore the message and proceed, but the mesh provided may not then be of sufficient quality.

Lastly, there is a "hidden" language variable `sellan` (default value 'english') that can be changed to its original 'français'.

Once all the settings have been chosen, it is often convenient to save them, so as to not have to re-enter them on the next Carre meshing attempt. This is achieved by means of the '(S)ave' option within the `carre` script, provided that the user does not start the next meshing attempt with the '(P)repare' option, but rather jumps directly to '(G)rid'.

After running Carre, it is recommended to have a look at the mesh and decide if one wants to modify it before proceeding much further (see previous section). The default settings may not be appropriate for all cases. It is particularly important to check the mesh near the separatrix, the X-point(s) and the target plates.

In some cases, the `carre` - script will fail. In that case, one can still run Carre in "standalone" mode by invoking:

```
ssc  
builds/${HOST_NAME}.${COMPILER}(.debug)/carre
```

Chapter 9

Running agg

agg is an interface to Carre for generating the grid for a specified ASDEX Upgrade shot at a particular time point. There is also a similar program, aggj for JET. agg was written by Heimo Bürbaumer.

Running the code at IPP produces:

```
15:36> ./aggs
AGG: Automatic Grid Generator
For fully automatic use type: aggs -s shotnumber -t time
and optionally: -nx pol.grid -ny rad.grid -depth distance
with shotnumber as integer, time in s (2.7), poloidal grid (96,...)
rad.grid (32, ... ) and distance betw. X-Point and inner boundary in cm
For half-automatic use type: aggs -i
Requesting shot 10095(1) of DPC:EQI 10095(1)
Requesting shot 8650 of AUGD:YGC 8650
kkEQPFM: ierr=0
kkEQpfx: ierr=0
kkrzBrzt: ierr=0
1.74294448, 1.52229261, 2.15042973, 1.68617392, 1.11173415
0.130604938, -0.858187616, 0.449555516, 1.26027465, -0.581599176
ngr,ngz=65, 129
10095, 3.20000005, 1 dpc EQI 96x24
Starting
newpag OK
cpsets ok
cprect ok
cpcldr ok
Pre-selected points are identified.
o-point: 1.7426E+00 1.3054E-01
x-point: 1.5218E+00 -8.5777E-01

Do you accept the selection (y/n)?

Entering sptris: npx= 1

ipx,ix,jx,ptx,pty = 1 26 28 1.5217840062009 -0.85776841963238
psi(ix,jx),CCW = -7.98106000000000D-04 8.99434000000000D-05
4.31895000000000D-04 -1.71310000000000D-03
fctpx = -1.5440357642937D-04

idir,nbdfav,isep = 1 0 0
il,i2,j1,j2,iref,jref = 28 28 27 28 28 27
psi(il,j1),psi(i2,j2),fctpx(ipx)= -2.46859000000000D-03 -5.03957000000000D-03
-1.5440357642937D-04
il,i2,j1,j2,iref,jref = 28 28 28 29 28 28
psi(il,j1),psi(i2,j2),fctpx(ipx)= -5.03957000000000D-03 -3.98707000000000D-03
-1.5440357642937D-04
```

```

i1,i2,j1,j2,iref,jref = 28 28 29 30 28 29
psi(i1,j1),psi(i2,j2),fctpx(ipx)= -3.9870700000000D-03 8.0841800000000D-04
-1.5440357642937D-04
Cell crossing: k,isep,ipx,separx,separ y = 2 1 1 1.5600000000000
-0.82721824912211
insert: indstr,nbdef,ipx 1 0 1

idir,nbdfav,isep = 2 1 1
i1,i2,j1,j2,iref,jref = 28 27 30 30 27 30
psi(i1,j1),psi(i2,j2),fctpx(ipx)= 8.0841800000000D-04 4.4249300000000D-03
-1.5440357642937D-04
i1,i2,j1,j2,iref,jref = 27 26 30 30 26 30
psi(i1,j1),psi(i2,j2),fctpx(ipx)= 4.4249300000000D-03 9.2840200000000D-04
-1.5440357642937D-04
i1,i2,j1,j2,iref,jref = 26 25 30 30 25 30
psi(i1,j1),psi(i2,j2),fctpx(ipx)= 9.2840200000000D-04 -8.7230800000000D-03
-1.5440357642937D-04
Cell crossing: k,isep,ipx,separx,separ y = 2 2 1 1.4966342819380
-0.822500000000000
insert: indstr,nbdef,ipx 2 1 1

idir,nbdfav,isep = 3 2 2
i1,i2,j1,j2,iref,jref = 25 25 30 29 24 29
psi(i1,j1),psi(i2,j2),fctpx(ipx)= -8.7230800000000D-03 -9.5526000000000D-03
-1.5440357642937D-04
i1,i2,j1,j2,iref,jref = 25 25 29 28 24 28
psi(i1,j1),psi(i2,j2),fctpx(ipx)= -9.5526000000000D-03 -7.2130600000000D-03
-1.5440357642937D-04
i1,i2,j1,j2,iref,jref = 25 25 28 27 24 27
psi(i1,j1),psi(i2,j2),fctpx(ipx)= -7.2130600000000D-03 -1.7010600000000D-03
-1.5440357642937D-04

idir,nbdfav,isep = 4 2 2
i1,i2,j1,j2,iref,jref = 25 26 27 27 25 26
psi(i1,j1),psi(i2,j2),fctpx(ipx)= -1.7010600000000D-03 3.5539300000000D-03
-1.5440357642937D-04
Cell crossing: k,isep,ipx,separx,separ y = 2 3 1 1.4788296443394
-0.893000000000000
insert: indstr,nbdef,ipx 1 2 1
i1,i2,j1,j2,iref,jref = 26 27 27 27 26 26
psi(i1,j1),psi(i2,j2),fctpx(ipx)= 3.5539300000000D-03 3.4159400000000D-03
-1.5440357642937D-04
i1,i2,j1,j2,iref,jref = 27 28 27 27 27 26
psi(i1,j1),psi(i2,j2),fctpx(ipx)= 3.4159400000000D-03 -2.4685900000000D-03
-1.5440357642937D-04
Cell crossing: k,isep,ipx,separx,separ y = 2 4 1 1.5482020156738
-0.893000000000000
insert: indstr,nbdef,ipx 2 2 1

After idir loop: isep,ipx,nbdef,nbdfav = 4 1 2 2

after x-point loop: isep,ipx,nbdef,nbdfav = 4 2 2 2
ixp_hlp : 1 1
xsttmp : 1.2536294134836 1.6897170950977
ysttmp : -1.0691543089581 -1.0565671252574
nptot
226 220 18 14
0 0 0 0
indplq
1 2 1 2
0 0 0 0
Entering trgarng: nbdef= 2
inndef: 1 2
indxpt: 1 1
xst: 1.254E+00 1.690E+00
yst: -1.069E+00 -1.057E+00
Single intersections : 2

```

```

jj : 1 2
ll : 1 1
Complete x-points: 1
kk : 1
Before re-arrangement. k = 2
jj : 1 2
After re-arrangement: 2
indef: 1 2
xst: 1.254E+00 1.690E+00
yst: -1.069E+00-1.057E+00
Leaving trgarng: 2
Entering orddef: npx,nbdef= 1 2
indef: 1 2
xst: 1.25E+00 1.69E+00
yst: -1.07E+00 -1.06E+00
ptx: 1.52E+00
pty: -8.58E-01
After ordering in y
indef: 2 1
i ipx xy pty aire : wx : wy
1 1 -8.58E-01 1.31E-01-4.44E-02 1.52E+00 1.69E+00 1.25E+00 1.52E+00-8.58E-01-1.06E+00-1.07E+00-8.58E-01
After ordering in x
indef: 2 1
Leaving sptris...

```

```

=== carre *.10.0 - before frtier
ptx: 1.5218E+00 1.7426E+00 0.0000E+00 0.0000E+00
pty: -8.5777E-01 1.3054E-01 0.0000E+00 0.0000E+00
nptot(4,nxpoints)
204 1 17 13 0 0 0 0 0 0 0 0 0 0 0 0
Strike points (presumably)
x: 1.5218E+00 1.5218E+00 1.2536E+00 1.6897E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
y: -8.5777E-01 -8.5777E-01 -1.0692E+00 -1.0566E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

```

SEPARATRICES

#	nptseg(i)	lg(i)	deltpl(i)	deltpn(i)	dpmin	dpmax
1	25	2.6059E-01	5.2083E-03	5.2083E-03	5.2083E-03	1.4051E-02
2	25	3.4168E-01	5.2083E-03	5.2083E-03	5.2083E-03	1.9340E-02
3	49	4.1421E+00	5.2083E-03	5.2083E-03	5.2083E-03	1.2942E-01

RADIAL DISTRIBUTIONS

Distribution in psi: repart= 2

region	npr(i)	width	deltrl(i)	deltrn(i)	drmin	drmax
1	13	-1.3291E-01	-1.0000E-03	-2.0833E-03	-1.7192E-02	-1.0000E-03
2	13	5.0702E-02	1.0000E-03	2.0833E-03	1.0000E-03	5.9821E-03

CENTRAL REGION: region i= 3 pntrat max.=1.01267691

pntrat	npr(i)	width	deltrl(i)	deltrn(i)	drmin	drmax
0.3000	13	2.3836E-01	1.0000E-03	2.0833E-03	1.0000E-03	3.1572E-02

GUARD LENGTH FOR EACH DIVERTOR PLATE

tgarde(1)	tgarde(2)
0.00000	0.00000

RELAXATION PARAMETERS USED TO CONSTRUCT THE MESH

```
=====
nrelax      relax      pasmin      rlcept
  200        0.200      5.000E-03    1.000E-06
```

Do you wish to accept these values (y/n)?

```
ir= 2
ir= 3
ir= 4
ir= 5
ir= 6
ir= 7
ir= 8
ir= 9
ir= 10
ir= 11
ir= 12
ir= 13
ir= 2
ir= 3
ir= 4
ir= 5
ir= 6
ir= 7
ir= 8
ir= 9
ir= 10
ir= 11
ir= 12
ir= 13
ir= 2
ir= 3
ir= 4
ir= 5
ir= 6
ir= 7
ir= 8
ir= 9
ir= 10
ir= 11
ir= 12
ir= 13
```

Warning: Distance Separatix-outer grid line is small: 2.40 cm !

But mesh-generation is valid !

Distance Separatix-outer grid line in private flux region is: 6.37 cm !

Name of the file containing the carre grid

Select the output format

1: standard mailtri format

2: format B2.5

3: format SONNET-DIVIMP

4: format DG-SONNET-B2-EIRENE

==> Entering b2agfz. nreg = 3

Input: b0r0 = -4.1984658200000

topology: sn-down

15:37>

Appendix A

Internal B2.5 documentation

A.1 b2cdca.F

```
*-----
*.specification

      subroutine b2cdca ()
      implicit none

*-----
*.documentation
*
* 1. purpose
*
*   B2CDCA provides some general documentation of the B2 code.
*   No computation is performed in this routine.
*
*-----
*.text
*
* 0. Introduction
*
*   The B2 code /1,2/ solves a system of fluid equations that models
*   a multi-species plasma in a two-dimensional geometry. Its main
*   application is simulation of the edge plasma in tokamaks.
*
*   B2 is composed of several independent programs that communicate
*   through external files. The main sequence involves the programs
*   b2ag (prepares a geometry and magnetic field), b2ah (prepares
*   a table of default physics parameters), b2ai (prepares an
*   initial plasma state), b2mn (principal calculation), and b2yp
*   (postprocessing).
*   Additional programs may prepare tables of rate coefficients for
*   atomic processes, tables of plasma-wall interaction processes,
*   external source terms, and other physics, or may perform further
*   postprocessing.
*
*   ...
*
* 1. Mathematical model
*
*   ...
*
* 2. Geometry
*
*   B2 employs a two-dimensional, curvilinear, topologically
```

```

*      rectangular mesh. The coordinates of the mesh are labelled 'x'
*      and 'y'. There are nx+2 cells along the x-coordinate and ny+2
*      cells along the y-coordinate. The cells of the mesh are indexed
*      by (ix,iy) in (-1:nx,-1:ny); the interior cells have index in
*      (0:nx-1,0:ny-1). The mesh may contain cuts, which typically
*      originate from an x-point in the real*8 geometry.
*
*      The primary data describing the mesh are the coordinates of the
*      four vertices of all the cells of the mesh. These data are stored
*      in the arrays crx and cry. The metric parameters (vol, hx, hy, qz,
*      qc and gs; described elsewhere) are derived from crx and cry.
*
*      The primary data describing the magnetic field are the flux
*      functions fpsi and ffbz at the cell vertices. The magnetic field
*      components at cell centers are derived from fpsi and ffbz.
*
*      Singular regions in the geometry are specified by the data in the
*      common block /b2cmgs/.
*
*      ...
*
* 3. Numerical method
*
*      ...
*
* 4. References
*
*      (Refs. /1/ and /2/ are out of date; !! to be replaced.)
*      /1/ B. J. Braams, Computational Studies in Tokamak Equilibrium
*      and Transport, thesis, State University at Utrecht, June 1986.
*      /2/ B. J. Braams, A Multi-Fluid Code for Simulation of the Edge
*      Plasma in Tokamaks, NET Report Nr 68, January 1987.
*      /3/ R. A. Hulse, Numerical Studies of Impurities in Fusion
*      Plasmas, Nuclear Technol./Fusion, vol.3 (1983) pp.259-272.
*      /4/ R. A. Hulse, !! (detailed description?).
*      /5/ K. Behringer, Description of the Impurity Transport Code
*      STRAHL, JET Report R(87)08, 1987.
*      /6/ K. Behringer, !! (journal article?).
*      /7/ Jon C. Weisheit, Recombination in Dense Plasmas, J. Phys. B
*      (Atom. Molec. Phys.) 8 (1975) 2556.
*      /8/ R. K. Janev, D. E. Post, W. D. Langer, K. Evans, D. B.
*      Heifetz and J. C. Weisheit, Survey of Atomic Processes in Edge
*      Plasmas, J. Nuclear Mater. 121 (1984) 10-16.
*
*-A1. Notes
*
*      ..Adherence to ANSI Fortran-77.
*      We respect the standard with the following exceptions.
*      - 'implicit none' is used to force type declarations for all
*        constants, variables and functions. We also declare, as external
*        or intrinsic, all procedures.
*      - 'do, enddo' syntax is used; a pre-processor based on Emacs
*        (courtesy of Charles Karney) may convert to labelled do loops.
*      - CPP 'include' statements are used for common blocks and for some
*        parameter statements.
*      - symbolic names may have up to eight characters.
*      These extensions to ANSI Fortran-77 are accepted in the identical
*      syntax on all popular workstations as well as on the Cray.
*      The code can be distributed using do-enddo syntax or using
*      labelled do loops, and with or without use of include statements.
*
*      ..Scientific libraries.
*      The BLAS (basic linear algebra subroutines) and LAPACK (linear
*      algebra package) are employed; these are included in the standard
*      scientific libraries on many computer systems, and are also in
*      NAG and SLATEC; moreover, they are available by electronic mail
*      or ftp from netlib@ornl.gov. Some simple additional routines

```

```

*      from the NAG library may be employed, for constants of the machine
*      arithmetic and for time and date.
*      An implementation of these scientific libraries is required in
*      which real*8 variables occupy eight bytes.
*
*      ..Graphics libraries.
*      As distributed, B2 relies on NCAR and GKS for graphics output.
*      The graphics postprocessing codes for B2 are completely separate
*      from the main calculation.
*
*      ..Concerning 'unformatted' files.
*      The programs that make up B2 communicate through external files,
*      which could be unformatted if all programs run on the same
*      computer system, and otherwise would be formatted. In the latter
*      case the format is very simple, and not designed for convenience
*      of a human reader. We refer to these files as 'unformatted'.
*      Reading from and writing to an unformatted file is done through
*      the routines cfruch and cfwuch for character variables, cfruin
*      and cfwuin for integer variables, and cfrure and cfwure for real
*      variables.
*      The unformatted files could easily be replaced by files written
*      using 'NetCDF' or 'HDF' routines if either one of those packages
*      is desired as a standard interface.
*
*      ..Concerning internal documentation.
*      The style of the internal documentation follows that of the NAG
*      library routine documents.
*      It has been the intent to specify the meaning of any non-trivial
*      variable in the outermost routine in which it is declared, and
*      also in the routine in which it is first computed.
*
*      ..Special patterns in comments.
*      The pattern !! is used to draw attention to segments of code
*      that might have to be changed. The pattern *$$$ in columns 1-4 is
*      used for code that is commented out with the text retained as a
*      reminder.
*
*      ..Contacting the author.
*      Use e-mail to braams@cims.nyu.edu, phone to 212-998-3276,
*      FAX to 212-995-4121, or regular mail to
*      B. J. Braams
*      Courant Institute of Mathematical Sciences
*      New York University
*      251 Mercer Street
*      New York, NY 10012
*
*-----
*.end b2cdca

      end subroutine b2cdca

```

```

*.specification

      subroutine b2cdcr ()
      implicit none

*-----
*.documentation
*
*  1. purpose
*
*      B2CDCR contains a brief statement of the purpose of many of the
*      subroutines of the B2 code. No computation is performed in this
*      routine.
*-----
*.text
*
*  0. Overview and naming conventions.
*
*      B2A*, B2B* - Auxiliary programs and their subroutines.
*      B2AG* - Preparation of a geometry.
*      B2AH* - Preparation of default physics parameters.
*      B2AI* - Preparation of an initial state.
*      B2AR* - Preparation of atomic rate coefficients.
*      B2AW* - Reserved for preparation of wall data.
*      B2C* - Documentation and 'include' segments.
*      B2CD* - Documentation.
*      B2CM* - 'include' segments: parameters and common blocks.
*      B2D* - Reserved for DEGAS interface.
*      B2E* - Reserved for EIRENE interface.
*      B2F*-B2L* - Not used.
*      B2M* - Core routines for main program: driver and timestepping.
*      B2MN_INIT - Main program initialisation.
*      B2MN_STEP - Main program call.
*      B2MN_FIN - Main program finalization.
*      B2MNDR_0 - Main driver routine initialisation.
*      B2MNDR_1 - Main driver routine.
*      B2MNDR_2 - Main driver routine finalisation.
*      B2MNDS - Initialisation.
*      B2MNDT - Time stepping.
*      B2MW* - Monitoring of the calculation.
*      B2MX* - Auxiliaries for b2m*.
*      B2N* - Driver routines for inner iteration.
*      B2NEWS - Controls inner iteration.
*      B2NP* - Iteration for individual equations.
*      B2NX* - Auxiliary routines for b2n*.
*      B2O*-B2Q* - Reserved for alternative numerical core.
*      B2R* - Read from an external file.
*      B2RF* - Read formatted.
*      B2RU* - Read unformatted.
*      B2S* - Evaluate source term coefficients.
*      B2SI* - Evaluate internal source terms.
*      B2SP* - Evaluate log-log linearised rate coefficients.
*      B2SQ* - Evaluate rate coefficients.
*      B2SR* - Evaluate or modify total source term coefficients.
*      B2ST* - Evaluate partial source term coefficients.
*      B2T* - Evaluate transport coefficients and fluxes.
*      B2TF* - Evaluate fluxes.
*      B2TL* - Apply flux limits to transport coefficients.
*      B2TQ* - Evaluate differential transport coefficients.
*      B2TR* - Evaluate cell-integrated transport coefficients.
*      B2TX* - Auxiliary routines for b2t*.
*      B2U* - Inner routines for the inner iteration.

```

```

*      B2UP* - Update some quantity.
*      B2UR* - Evaluate a residual.
*      B2US* - Solve a correction equation.
*      B2UX* - Auxiliaries related to b2u*.
*      B2V* - Not used.
*      B2W* - Write to an external file.
*      B2WF* - Write formatted.
*      B2WG* - Write graphical.
*      B2WS* - Write special purpose.
*      B2WU* - Write unformatted.
*      B2X* - Basic auxiliary routines.
*      B2XG* - Auxiliaries for geometry.
*      B2XP* - Auxiliaries for plasma state.
*      B2XV* - Test validity of some quantities.
*      B2XX* - Miscellaneous auxiliaries.
*      B2Y* - Postprocessing.
*      B2YG* - Display the geometry and magnetic field.
*      B2YH* - Display the table of physics parameters.
*      B2YI* - Display the plasma state.
*      B2YM* - Display the evolution of the plasma state.
*      B2YN* - Display the progress of the inner iterations.
*      B2YP* - Comprehensive postprocessing of a calculation.
*      B2YQ* - Quick display of the evolution of a calculation.
*      B2YR* - Display the table of atomic rate coefficients.
*      B2YS* - Display the externally imposed source terms.
*      B2Z* - Reserved for application-dependent postprocessing.
*
*
*      1. Purpose of individual B2 routines.
*
*      B2AG (main program) prepares a geometry and magnetic field for
*      B2. This program calls the driver routine b2agdr for the actual
*      computation.
*
*      B2AGDR computes and writes to an external file the specification
*      of a B2 geometry and magnetic field.
*
*      B2AGBB computes the magnetic field from the flux functions.
*
*      B2AGBS computes a correction to the pbsx array to ensure the
*      constancy of magnetic flux along flux tubes.
*
*      B2AGCF computes the locations of the cell vertices, and computes
*      the flux functions at those locations.
*
*      B2AGFS obtains a description of the flux surface structure.
*
*      B2AGGS marks a singular region in the geometry.
*
*      B2AGMT computes the metric coefficients given the locations of
*      cell vertices.
*
*      B2AGMX projects the metric coefficients from a fine mesh to a
*      coarsening of that mesh.
*
*      B2AGSP evaluates the coordinates and the flux functions at one
*      single point on the mesh. This routine is application dependent,
*      and defines the specific geometry and magnetic field for a B2
*      calculation.
*
*      B2AGX0 reads the mesh dimensions from a formatted file.
*
*      B2AH (main program) prepares default physics parameters for B2.
*      This program calls the driver routine b2ahdr for the actual
*      computation.
*
*      B2AHDR prepares and writes to an external file a table of default

```

```

* physics parameters for B2.
*
* B2AHX0 reads the number of species from a formatted file.
*
* B2AI (main program) prepares an initial plasma state for B2.
* This program calls the driver routine b2aidr for the actual
* computation.
*
* B2AIDR prepares and writes to an external file an initial plasma
* state for B2.
*
* B2AIX0 reads the mesh dimensions and the number of species from a
* formatted file.
*
* B2AR (main program) prepares a table of atomic rate coefficients
* for B2. This program calls the driver routine b2ardr for the
* actual computation.
*
* B2ARDR computes and writes to an external file a table of atomic
* rate coefficients for B2. The coefficients are obtained using
* either the ADPAK or the STRAHL set of routines.
*
* B2ARIN reads from a formatted file the list of nuclear species
* and atomic states for which the rate coefficients are to be
* computed.
*
* B2CDCA provides some general documentation of the B2 code.
* No computation is performed in this routine.
*
* B2CDCR contains a brief statement of the purpose of many of the
* subroutines of the B2 code. No computation is performed in this
* routine.
*
* B2CDCV contains a description of some of the variables used in
* the B2 code. No computation is performed in this routine.
*
* B2MN is the main program for the B2 code. The code solves a
* system of fluid equations that models a multi-species plasma
* in a two-dimensional geometry. This program calls the driver
* routines b2mndr_[0-2] for the actual computation.
*
* B2MNDR is the main driver routine for the B2 code.
*
* B2MNDS performs initialisation functions for a B2 calculation.
*
* B2MNDT performs one implicit timestep for the B2 system of
* equations.
*
* B2MWMV writes to an unformatted file a selection of data
* describing the state of the plasma.
*
* B2MWIT writes to an unformatted file detailed monitoring output
* for the inner iterations of the B2 solution process.
*
* B2MWQ0 writes a header for the output from b2mwqt.
*
* B2MWQT writes brief data concerning the numerical evolution.
*
* B2MXAC computes the (maximum-)norms of the corrections.
*
* B2MXAR computes scaled (L1-)norms of the residuals.
*
* B2MXNP extrapolates a positive quantity to the next time.
*
* B2MXNU extrapolates an indefinite quantity to the next time.
*
* B2MXZR sets the residuals to zero along the boundaries of the

```

```

*      mesh and on the special regions.
*
*      B2NEWS performs one inner iteration for the B2 equations.
*      This routine defines the core structure of the solution
*      algorithm.
*
*      B2NPCO performs one iteration of a relaxation process for the
*      solution of the continuity equation for one atomic species.
*
*      B2NPHT performs one iteration of a relaxation process for the
*      solution of the equations for internal electron and atom energy.
*
*      B2NPMO performs one iteration of a relaxation process for the
*      solution of the B2 equations for parallel velocity.
*
*      B2NPPO performs one iteration of a relaxation process for the
*      solution of the charge conservation equation, which governs the
*      electric potential.
*
*      B2NXCM computes the adjustment to the coefficients pcca that
*      comes from the combined parallel momentum equation solver.
*
*      B2NXDP computes the correction to the electric potential following
*      an update to the electron density.
*
*      B2NXDU computes the artificial velocity shift that is
*      required to make the total numerical friction force cancel.
*
*      B2NXDV computes the correction to the parallel momentum source
*      that is required to make the total friction force cancel.
*
*      B2NXFC computes the transport terms in the parallel momentum
*      correction equation.
*
*      B2NXFM computes the flow terms in the parallel momentum equation
*      (not used anymore).
*
*      B2NXFX computes an expression that occurs in the electron-atom
*      thermal force.
*
*      B2NXSM corrects the particle transport coefficients as required
*      for the Spalding-Patankar pressure correction procedure.
*
*      B2NXST accumulates the parallel momentum source linearisation
*      over the distinct atomic species.
*
*      B2PRNTI prints out the residuals in integer format.
*
*      B2RFCN reads from a formatted file some parameters in the
*      numerics common block.
*
*      B2RFCP reads from a formatted file some parameters in the
*      physics common blocks.
*
*      B2RFLB reads from a formatted file a label for one run of one
*      of the programs that make up B2.
*
*      B2RUCP reads from an unformatted file the parameters in the
*      physics common blocks.
*
*      B2RUGM reads from an unformatted file the geometry and magnetic
*      field.
*
*      B2RUPS reads the state of the plasma from an unformatted file.
*
*      B2RURC reads a table of atomic rate data from an unformatted
*      file.

```



```

*
* B2RUSR reads a table of source term data from an unformatted
* file.
*
* B2SIAN computes contribution to the parallel momentum source due
* to the anomalous current
*
* B2SIIV computes a linearised representation of the additional
* viscosity.
*
* B2SIFR computes a linearised representation of the friction
* force between one species and all others.
*
* B2SIFRTF computes a linearised representation of the thermal
* and friction forces.
*
* B2SIGP computes the pressure gradient term on the right hand
* of the parallel momentum balance equation for one species.
*
* B2SIHS computes the source coefficients for a set of terms in the
* heat equations. These terms specify the contributions of the
* divergence of the flow field, of viscous heating, and of the
* electron-atom and atom-atom friction.
*
* B2SPCX computes on all cells of the mesh a log-log representation
* of the rate coefficients for charge exchange.
*
* B2SPEL computes on all cells of the mesh a log-log representation
* of the rate coefficients for ionisation, recombination, and
* electron heat loss.
*
* B2SQCX computes on all cells of the mesh the rate coefficients
* for charge exchange with one neutral hydrogen species.
*
* B2SQEL computes on all cells of the mesh the rate coefficients
* for ionisation, recombination, and electron heat loss.
*
* B2SRAL provides linearised expressions for the sources of
* density, parallel momentum, electron heat, atom heat, and charge.
*
* B2SRDT modifies the source coefficients to account for the
* implicit time discretisation.
*
* B2SRSM smoothes expressions the sources strengths of
* density, parallel momentum, electron heat, atom heat, and charge.
*
* B2SRST modifies the source coefficients to provide numerical
* stabilisation.
*
* B2STBC provides linearised expressions for the sources of
* density, parallel momentum, electron heat, atom heat, and charge,
* due to the boundary conditions.
*
* B2STBM provides linearised expressions for the sources of
* density, parallel momentum, and heat, due to miscellaneous
* application specific processes.
*
* B2STBR provides linearised expressions for the sources of
* density, parallel momentum, and heat, due to special boundary
* conditions and the neutral gas recycling model.
*
* B2STCX provides linearised expressions for the sources of atom
* density and atom parallel momentum, due to charge exchange.
*
* B2STEL provides linearised expressions for the sources of
* density, parallel momentum, electron heat and atom heat, due
* to the electron-atom collision processes of ionisation,

```

```

*      recombination, and radiation.
*
*      B2TBFL computes the total flux through the boundaries
*
*      B2TDIA computes the X and Y components of the divergent part of
*      the diamagnetic current
*
*      B2TEPSCH computes the X and Y components of the Pfirsch-Schlueter
*      heat flux
*
*      B2TFCH computes the electric current between neighboring cells,
*      and the coefficients that specify its linearisation.
*
*      B2TFED computes the electron drift velocities.
*
*      B2TFHE computes the electron heat flux between neighboring cells,
*      and the coefficients that specify its linearisation.
*
*      B2TFHI computes the all atom heat flux between neighboring cells,
*      and the coefficients that specify its linearisation.
*
*      B2TFNB computes the particle flux of one atomic species between
*      neighboring cells, and the coefficients that specify its
*      linearisation.
*
*      B2TFRN computes the particle and energy fluxes between
*      neighbouring cells which are to be provided to Eirene
*
*      B2TIPSCH computes the Pfirsch-Schlueter all atom heat flux between
*      neighboring cells, and the coefficients that specify its linearization.
*
*      B2TLHE applies a flux limit to the electron heat and charge
*      transport coefficients.
*
*      B2TLHI applies a flux limit to the all atom heat transport
*      coefficients.
*
*      B2TLH0 computes the conductive neutral heat transport flux limit.
*
*      B2TLMV applies a flux limit to the viscosity.
*
*      B2TNPSCH computes the Pfirsch-Schlueter particle fluxes of one atomic
*      species between neighbouring cells
*
*      B2TQCA computes the classical contributions to the atom
*      transport coefficients.
*
*      B2TQCE computes the classical contributions to the electron
*      transport coefficients.
*
*      B2TQNA computes the anomalous contributions to the transport
*      coefficients.
*
*      B2TRAL computes the transport coefficients, including geometric
*      factors such as  $vol/hx^{**2}$  and  $vol/hy^{**2}$ .
*
*      B2TRCL computes the classical transport coefficients, including
*      geometric factors such as  $vol/hx^{**2}$  and  $vol/hy^{**2}$ .
*
*      B2TREQ computes the cell-integrated energy equipartition
*      multiplier.
*
*      B2TRNO computes the anomalous transport coefficients, including
*      geometric factors such as  $vol/hx^{**2}$  and  $vol/hy^{**2}$ .
*
*      B2TVSPA computes the X and Y components of the parallel viscosity
*      current

```

```

*
* B2TVSPR computes the X and Y components of the perpendicular viscosity
* current
*
* B2TVSQ computes the X and Y components of the heat viscosity
* current
*
* B2TXCV computes viscous conductances between neighbouring
* staggered cells, given the viscosities (not used anymore).
*
* B2TXCX computes conductances between neighbouring cells in the
* x-direction, given the conductivities.
*
* B2TXCY computes conductances between neighbouring cells in the
* y-direction, given the conductivities.
*
* B2TXFX computes conductances between neighbouring cells in the
* x-direction, given the flux-limited conductivities.
*
* B2TXFY computes conductances between neighbouring cells in the
* y-direction, given the flux-limited conductivities.
*
* B2TXSX computes the integrals over cell x-faces of a function
* that is provided on the cells.
*
* B2TXSY computes the integrals over cell y-faces of a function
* that is provided on the cells.
*
* B2UNAVSPR7A evaluates the coefficients of matrix of the contribution
* from divergence of the viscosity perpendicular current by analytical
* manipulation with respect to main ion density using a 7-point stencil.
*
* B2UPCO corrects the state of the plasma towards the solution of
* continuity equation for one atomic species.
*
* B2UPHT adjusts the state of the plasma towards the solution
* of the heat equations.
*
* B2UPPO corrects the electric potential po.
*
* B2URMO computes the residual of the parallel momentum balance
* equation for one atomic species.
*
* B2URSC computes the residual of a scalar conservation equation.
*
* B2URSD computes the residual of a scalar conservation equation.
*
* B2USCO computes a correction towards the solution of the
* continuity equation for one atomic species.
*
* B2USHT computes a correction to the fields te and ti.
*
* B2USMO Computes a correction towards the solution of the
* parallel momentum balance equation for one atomic species.
*
* B2USPO computes a correction towards the solution of the
* potential equation.
*
* B2UX5P solves a five-point system of equations. This routine is
* an interface to an iterative or a direct solver; the choice
* depends upon an internal parameter.
*
* B2WAOS prints out the results of a B2 calculation in old-fashioned
* style
*
* B2WDAT is an output routine that writes out B2 arrays into ASCII
* data files.

```

```

*
* B2WFCP writes to a formatted file the parameters in the physics
* common blocks.
*
* B2WFGM writes to a formatted file a summary of the B2 geometry
* and magnetic field.
*
* B2WFPI writes a summary of the plasma state to a formatted file.
* This routine is an interface to b2wfpj.
*
* B2WF PJ writes a summary of the plasma state to a formatted file.
* This routine does not make use of any geometric information.
*
* B2WFPK writes supplementary information about the plasma state to
* a formatted file.
*
* B2WFPS writes a summary of the plasma state to a formatted file.
*
* B2WFR0 writes to a formatted file a text containing credits for
* the ADPAK atomic physics rate coefficients.
*
* B2WFR1 writes to a formatted file a text containing credits for
* the STRAHL atomic physics rate coefficients.
*
* B2WFRC writes to a formatted file a table of atomic rate
* coefficients that has been prepared for B2.
*
* B2WGCR provides graphical output of the residuals and the
* corrections.
*
* B2WGDT provides graphical output of the change in the state.
*
* B2WGGM produces graphical displays of the B2 geometry and
* magnetic field.
*
* B2WGPI provides graphical output of the plasma state on a uniform
* rectangular mesh. This routine is an interface to b2wgpj.
*
* B2WG PJ provides graphical output of the plasma state on a uniform
* cartesian mesh.
*
* B2WGPS provides graphical output of the plasma state.
* This routine is an interface to b2wgst.
*
* B2WGST provides graphical output of the plasma state.
*
* B2WGRC displays a table of atomic rate coefficients that has been
* prepared for B2.
*
* B2WGX0 displays the B2 mesh in a cartesian representation.
*
* B2WGX1 displays the B2 mesh on the physical plane.
*
* B2WGX2 produces contour plots on a sequence of cartesian domains.
*
* B2WGX3 produces contour plots on the physical domain.
*
* B2WGX4 produces log-contour plots on a sequence of cartesian
* domains.
*
* B2WGX5 produces log-contour plots on the physical domain.
*
* B2WGXB plots an outline of the B2 mesh on the physical domain.
*
* B2WGXP produces contour plots on a cartesian domain.
*
* B2WGXQ produces log-contour plots on a cartesian domain.

```

```

*
* B2WGYF performs the core NCAR operations to produce a contour
* plot.
*
* B2WGYQ performs the core NCAR operations to produce a log-contour
* plot.
*
* B2WSAV writes the state of the plasma to an unformatted file,
* in the style required for input into the 'old B2' code.
* The routine operates as an interface to b2wsax.
*
* B2WSAX writes the state of the plasma to an unformatted file,
* in the style required for input into the 'old B2' code.
*
* B2WSCP provides special purpose printed output of data in the
* physics common blocks.
*
* B2WSPS provides special purpose printed output concerning the
* plasma state. This routine is an interface to b2wsst.
*
* B2WSST provides special purpose printed output concerning the
* plasma state.
*
* B2WUCP writes to an unformatted file the parameters in the
* physics common blocks.
*
* B2WUGM writes to an unformatted file the geometry and magnetic
* field.
*
* B2WUPS writes the state of the plasma to an unformatted file.
*
* B2WURC writes a table of atomic rate coefficients to an
* unformatted file.
*
* B2XBZB calculates  $B_z/B^{*2} - B_z/\langle B^{*2} \rangle$ .
*
* B2XGBS computes the product (magnetic field pitch)*(surface area)
* at cell faces.
*
* B2XGXY computes cartesian coordinates suitable for graphical and
* printed output.
*
* B2XPFE computes the electron particle flux, fne:
*   fne(,,) = (sum is :: rza(,,is)*fna(,,is))-(1/qe)*fch(,,) .
*
* B2XPFI computes the total atom particle flux, fni:
*   fni(,,) = (sum is :: fna(,,is)) .
*
* B2XPIC computes the poloidal projection of the ion parallel
* current on cell centers.
*
* B2XPIF computes the ion parallel current on cell faces.
*
* B2XPMC computes the mach number of the parallel flow.
*
* B2XPNE computes the electron density, ne:
*   ne(,) = (sum is :: rza(,,is)*na(,,is)) .
*
* B2XPNI computes the total atom density, ni0:
*   ni(,,0) = (sum is :: na(,,is)) .
* and the total ion density, ni1:
*   ni(,,1) = (sum is :: na(,,is) if is_neutral(is).eqv..false.) .
*
* B2XPNM computes the atom density weighted by  $rz^2/am$ , ne2m:
*   ne2m(,) = (sum is :: (rz2(,,is)/am(is))*na(,,is)) .
*
* B2XPNR computes the atom density weighted by  $1/\sqrt{am}$ , nirm:

```

```

*      nirm(,) = (sum is :: na(,,is)/sqrt(am(is))) .
*
*      B2XPPB computes the partial pressure, pb = nb*(rzb*te+ti).
*
*      B2XPPE computes the electron pressure, pe = ne*te.
*
*      B2XPPR computes the total pressure, pr:
*      pr(,) = ne(,)*te(,) + (sum is :: na(,,is)*ti(,)) .
*
*      B2XPPZ computes the plasma pressure, pz:
*      pz(,) = ne(,)*te(,) + (sum is :: na(,,is)*ti(,)), not including neutrals).
*
*      B2XPRO computes the mass density, ro:
*      ro(,) = (sum is :: am(is)*mp*na(,,is)) .
*
*      B2XPRZ computes the ion mass density, rz:
*      rz(,) = (sum is :: am(is)*mp*na(,,is), including only charged species).
*
*      B2XPVE computes the parallel electron velocity on cell centers.
*
*      B2XPXC interpolates a function from x-faces to cell centers.
*
*      B2XVCP tests the validity of parameters in the physics common
*      blocks.
*
*      B2XVFF verifies the zero entries in the edge locations of some
*      quantity defined on faces of the B2 mesh. This routine is an
*      interface to b2xvfx and b2xvfy.
*
*      B2XVFFV verifies the zero entries in the edge locations of some
*      quantity that is defined in the manner of a viscosity on the
*      cells and corners of the B2 mesh (not used anymore).
*
*      B2XVFX verifies the zero entries in the edge locations of some
*      quantity defined on x-faces of the B2 mesh.
*
*      B2XVFX verifies the zero entries in the edge locations of some
*      quantity defined on y-faces of the B2 mesh.
*
*      B2XVGM tests the validity of the geometric parameters.
*
*      B2XVPS tests the validity of the state of the plasma.
*
*      B2XVSG verifies the sign of some vector quantity.
*
*      B2XXGS sets a mesh function to a constant value on the singular
*      regions of the B2 geometry.
*
*      B2XXID returns an identification of the current version of B2.
*
*      B2XXMM computes the locations of the minimum and maximum entries
*      of a quantity defined on the B2 mesh.
*
*      B2XZEF computes zeff:
*      zeff(,) = (sum is :: rz2(,,is)*na(,,is))/ne .
*
*      B2XZMF computes zmff:
*      zmff(,) = (sum over is /= ismain :: rz2(,,is)*na(,,is))/
*                  na(,,ismain)*rz2(,,ismain)
*
*      B2YE (main program) is a postprocessor for B2. It calculates
*      statistical errors based on averaged quantities collected during
*      the averaging phase of a B2 calculation.
*
*      B2YG (main program) is a postprocessor for B2. It reads and
*      displays the geometry and magnetic field. This program calls the
*      driver routine b2ygdr for the actual computation.

```

```

*
* B2YGDR is a postprocessor for B2. It reads and displays the
* geometry and magnetic field.
*
* B2YH (main program) is a postprocessor for B2. It reads and
* displays the table of physics parameters. This program calls the
* driver routine b2yhdr for the actual computation.
*
* B2YHDR is a postprocessor for B2. It reads and displays the table
* of physics parameters.
*
* B2YI (main program) is a postprocessor for B2. It reads and
* displays the plasma state. This program calls the driver routine
* b2yidr for the actual computation.
*
* B2YIDR is a postprocessor for B2. It reads and displays the
* plasma state. This routine does not make use of any metric
* information.
*
* B2YM (main program) is a postprocessor for B2. It is meant to
* produce movie output. This program calls the driver routine b2ymdr
* for the actual computation.
*
* B2YMDR is a postprocessor for B2. It is meant to produce movie
* output.
*
* B2YN (main program) is a postprocessor for B2. It displays the
* progress of the inner iterations. This program calls the driver
* routine b2yndr for the actual computation.
*
* B2YNDR is a postprocessor for B2. It displays the progress of
* the inner iterations.
*
* B2YP (main program) is a postprocessor for B2. It displays the
* plasma state. This program calls the driver routine b2ypdr for
* the actual computation.
*
* B2YPDR is a postprocessor for B2. It displays the plasma state,
* making use also of the geometric information.
*
* B2YQ (main program) is a postprocessor for B2. It provides a
* quick display of the evolution data. This program calls the driver
* routine b2yqdr for the actual computation.
*
* B2YQDR is a postprocessor for B2. It provides a quick display of
* the evolution data.
*
* B2YR (main program) is a postprocessor for B2. It reads and
* displays a table of atomic rate coefficients. This program calls
* the driver routine b2yrdr for the actual computation.
*
* B2YRDR is a postprocessor for B2. It reads and displays a table
* of atomic rate coefficients.
*
* 2. Purpose of some additional routines employed with B2.
*
* ..2a program management and error handling
*
* PRGINI performs any tasks required at the start of program
* execution.
*
* PRGEND performs any tasks required at the end of program
* execution.
*
* SUBSYS is a group of routines (entry points in SUBSYS) that
* perform certain functions related to workspace management and

```

```

* error handling. The entry points are subini, subend, and subxnd.
* SUBINI performs any tasks required at the start of subprogram
* execution.
* SUBEND performs any tasks required at the end of subprogram
* execution.
* SUBXND is employed to build down the stack when the program
* is terminated because of an error condition.
*
* IPMAIN is a group of routines (entry points in ipmain) that
* operate on a collection of names and values representing
* "internal parameters" in a code. The entry points are ipsetc,
* ipgetc, and ipfini.
* IPSETC saves the value of an internal parameter.
* IPGETC returns the value of an internal parameter.
* IPFINI writes an error message for any parameter that has been
* set but not read.
*
* IPGETI is an interface to ipgetc; it retrieves an integer value
* ival from the table of internal parameters, or returns ival
* unchanged if the name is not found or if the corresponding entry
* is a blank string. The internal parameter may have been set by an
* earlier call to ipseti or to ipsetc.
*
* IPGETR is an interface to ipgetc; it retrieves a real*8 value
* rval from the table of internal parameters, or returns rval
* unchanged if the name is not found or if the corresponding entry
* is a blank string. The internal parameter may have been set by an
* earlier call to ipsetr or to ipsetc.
*
* IPSETI is an interface to ipsetc; it saves an integer value in
* the table of internal parameters.
*
* IPSETR is an interface to ipsetc; it saves a real*8 value in the
* table of internal parameters.
*
* LWIMAI is a group of routines (entry points in lwimai) that
* maintain a base address for free space in the workspace area
* stackw%wi. The entry points are lwiset, lwiget, and lwimax.
* LWISET frees workspace by resetting the base address.
* LWIGET reserves workspace by incrementing the base address.
* LWIMAX returns the maximum workspace use.
*
* LWMAIN is a group of routines (entry points in lwmain) that
* maintain a base address for free space in the workspace area
* stackw%w. The entry points are lwset, lwget, and lwmax.
* LWSET frees workspace by resetting the base address.
* LWGET reserves workspace by incrementing the base address.
* LWMAX returns the maximum workspace use.
*
* XERGET gets the unit number for error messages.
*
* XERSET sets the unit number for error messages.
*
* XERRAB writes an error message and the call chain, and terminates
* execution of the program.
*
* XERTST tests a condition; if violated, calls xerrab to terminate
* execution.
*
* ..2b File management and unformatted input/output
*
* CFENDF tests whether an unformatted file is positioned at
* end-of-file.
*
* CFOPEN opens an external file. The file may be formatted or
* unformatted.
*!! This routine may contain system dependencies.

```



```

*
*   CFRUCH reads a character array from an unformatted file.
*
*   CFRUIN reads an integer array from an unformatted file.
*
*   CFRURE reads a real*8 array from an unformatted file.
*
*   CFWUCH writes a character array to an unformatted file.
*
*   CFWUIN writes an integer array to an unformatted file.
*
*   CFWURE writes a real*8 array to an unformatted file.
*
* ..2c Low-level auxiliary routines
*
*   BLAS and BLAS-like routines (basic linear algebra subprograms):
*   IFILL (BLAS-like) fills an integer vector with a constant value.
*   SMIN (BLAS-like) computes the minimum value in a vector.
*   SMAX (BLAS-like) computes the maximum value in a vector.
*   SAMAX (BLAS-like) computes the maximum absolute value in
*   a vector.
*   B2SSUM (BLAS-like) sums the elements of a vector.
*   B2SASUM (BLAS) sums the absolute values of elements of a vector.
*   SFILL (BLAS-like) fills a vector with a constant value.
*   SBIAS (BLAS-like) adds a constant value to a vector.
*   B2SSCAL (BLAS) scales a vector by a constant.
*   B2SCOPY (BLAS) copies a vector to another.
*   B2SAXPY (BLAS) adds a constant-times-vector to a vector.
*
*   ESSEQ determines whether two real*8 numbers differ by no more than
*   a small multiple of their machine precision.
*
*   EVAL returns its real*8 argument through the function name. It may
*   be used to force a compiler to store an intermediate operand.
*
*   RATIO evaluates very carefully the real*8 ratio of two integers.
*
*   SCAL10 is meant to compute a suitable scale for graphical output.
*
*   CHCASE converts a character variable to either all uppercase or
*   all lowercase.
*
*   STRCOPY copies the first 10 characters of string str1 into str0.
*
*   STREQ1 checks for equality between two strings, ignoring case
*   shift and trailing blanks.
*
*   STRMAS checks for a certain matching between two strings.
*
* ..2d Auxiliary routines for discretised PDEs
*
*   ILU5F computes the ILU factorisation of a 2d 5-point matrix.
*
*   ILU5G computes the ILU factorisation of a 2d 5-point matrix and
*   applies the factors for one iteration of an update equation.
*
*   ILU5S solves an equation of the form  $L*U*x=r$ , where L and U are
*   the factors in an ILU factorisation of a 2d 5-point matrix.
*
*   ILUTER performs a number of iterations of the Stone strongly
*   implicit procedure for a system of equations of five-point form.
*
*   SIP5F computes the SIP-ILU factorisation of a 2d 5-point matrix.
*
*   SIP5G computes the SIP-ILU factorisation of a 2d 5-point matrix,
*   and applies the factors for one iteration of an update equation.
*

```

```

*      SLV5PT solves a five-point system of equations by a direct
*      method.
*
*      STABEQ adjusts the diagonal of a five-point matrix to satisfy
*      a property related to diagonal dominance.
*
* ..2e System-dependent routines
*
*      DAYTIM returns the date and time in a character*17 format.
*!!   This routine may contain system dependencies.
*
*      JOBNAM returns an identification of the present job.
*!!   This routine may contain system dependencies.
*
*      MACHEPS returns the machine precision.
*!!   This routine may contain system dependencies.
*
*      MACHSFR returns the safe numerical range (a tiny positive machine
*      value e0 for which -e0, 1/e0, -1/e0 are all computable).
*!!   This routine may contain system dependencies.
*
*      PRVRT sets all elements of a real*8 array to a value that is
*      expected to cause an error when used in an arithmetic operation.
*      This is used to protect against certain programming errors.
*!!   This routine may contain system dependencies.
*
*      PRVRTI sets all elements of an integer array to a value that is
*      expected to cause an error when used in an arithmetic operation.
*      This is used to protect against certain programming errors.
*!!   This routine may contain system dependencies.
*
*      SYSINI performs any system dependent calls that may be required
*      at the start of program execution.
*!!   This routine may contain system dependencies.
*
*      SYSEND performs any system dependent calls that may be required
*      at the finish of program execution.
*!!   This routine may contain system dependencies.
*
*      USRNAM returns the name or other appropriate identification of
*      the submitter of the present job.
*!!   This routine may contain system dependencies.
*
* ..2f Basic graphics routines
*
*      GNOPEN opens a graphics file for use with GKS.
*
*      GWSREP defines the workstation representation for GKS. It should
*      normally be called just once for each workstation, immediately
*      after opening the workstation.
*
*      PAGDEF sets GKS and NCAR properties to default values. It should
*      normally be called at the start of each new plot.
*
*      PAGINI starts a new plot.
*
*      PAGEND terminates a plot.
*
*      CPMPXY is a substitute for a routine of the same name in the
*      CONPACK sub-package of NCARG graphics. It defines a mapping from
*      rectangular array index coordinates in (1:*,1:*) to the plotting
*      window.
*
*      GNMAP2 (together with the alternate entry GNDFM2) is a service
*      routine for use with the CONPACK routines in the NCARG graphics
*      package. It defines a rectangular map for CONPACK. The map is
*      specified by a call to the entry gndfm2.

```

```

*
*   GNMAP3 (together with the alternate entry GNDFM3) is a service
*   routine for use with the CONPACK routines in the NCARG graphics
*   package. It defines a curvilinear, logically rectangular map for
*   CONPACK. The map is specified by a call to the entry gndfm3.
*
*   GNCOLR is for use by the routine ARSCAM in the AREAS utility of
*   NCAR graphics. It defines the filling of a polygon.
*
*   LBFILL is a substitute for a routine of the same name in the
*   LABELBAR utility of NCAR graphics. It fills a specified area.
*   LBFILX and LBFILQ are alternate entries in LBFILL. LBFILX is
*   called to define the filling style and LBFILQ is called to query
*   the filling style.
*
*   ..2g Atomic physics and surface physics
*
*   RATADP uses the ADPAK code to prepare tables of rate coefficients
*   for ionisation, recombination, electron energy loss, and charge
*   exchange; as function of temperature and density.
*
*   RATSTR uses the STRAHL code to prepare tables of rate coefficients
*   for ionisation, recombination, electron energy loss, and charge
*   exchange; as function of temperature and density.
*
*   RATPRT prints an excerpt from a table of atomic rate
*   coefficients.
*
*   RATTAI computes a smoothed set of rate coefficients to represent
*   the effect of a power-law tail on the distribution function.
*
*   RATWEI uses WEISHEIT data to improve tables of rate coefficients
*   for ionisation, recombination, electron energy loss, and charge
*   exchange; as function of temperature and density.
*   The WEISHEIT dataset is incomplete. The call to ratwei must
*   be preceded by a call to one of the comprehensive routines.
*
*-----
*.end b2cdcr

      end subroutine b2cdcr

```

A.3 b2cdcv.F

```

*-----
*.specification

      subroutine b2cdcv ()
      implicit none

*-----
*.documentation
*
* 1. purpose
*
*   B2CDCV contains a description of some of the variables used in
*   the B2 code. No computation is performed in this routine.
*
*-----
*.text
*
* 0. Introduction
*
*   ...
*
* 1. Principal code variables
*
*   ..variables describing the geometry
*
*   nx, ny - integer.
*   nx and ny specify the number of interior cells along the first
*   and the second coordinate, respectively. The total number of
*   cells is (nx+2)*(ny+2); they are indexed by (-1:nx,-1:ny).
*   It will hold that 0.le.nx and 0.le.ny.
*
*   crx, cry - (-1:nx,-1:ny,0:3) real*8 array.
*   For (ix,iy) in (-1:nx,-1:ny), crx(ix,iy,0:3) specify the
*   horizontal coordinates and cry(ix,iy,0:3) specify the vertical
*   coordinates (in the physical plane) of the four corners of the
*   (ix,iy) cell, mapped as follows:
*   (,,0): lower left corner, (,,1): lower right corner,
*   (,,2): upper left corner, (,,3): upper right corner.
*   One expects the following identities to hold away from the
*   boundaries:
*   crx(ix,iy,0)=crx(ix-1,iy,1)=crx(ix,iy-1,2)=crx(ix-1,iy-1,3),
*   cry(ix,iy,0)=cry(ix-1,iy,1)=cry(ix,iy-1,2)=cry(ix-1,iy-1,3),
*   However, there may be cuts in the geometry, in which case these
*   identities need not hold.
*
*   fpsci - (-1:nx,-1:ny,0:3) real*8 array.
*   For (ix,iy,i) in (-1:nx,-1:ny,0:3), fpsci(ix,iy,i) specifies the
*   potential for the magnetic field components (bx,by) at the
*   (ix,iy,i) position.
*   (In the case of straight symmetry, bx=dpsi/dy and by=-dpsi/dx.
*   In the case of toroidal or cylindrical symmetry,
*   bx=(dpsi/dy)/(2*pi*R) and by=-(dpsi/dx)/(2*pi*R),
*   where R is the distance to the axis of symmetry.)
*
*   ffbz - (-1:nx,-1:ny,0:3) real*8 array.
*   For (ix,iy,i) in (-1:nx,-1:ny,0:3), ffbz(ix,iy,i) specifies the
*   field function for the magnetic field component bz at the
*   (ix,iy,i) position.
*   (In the case of straight symmetry, bz=ffbz. In the case of
*   toroidal or cylindrical symmetry, bz=ffbz/(2*pi*R), where R is
*   the distance to the axis of symmetry.)
*
*   bb - (-1:nx,-1:ny,0:3) real*8 array.

```

```

* For (ix,iy) in (-1:nx,-1:ny), bb(ix,iy,0:3) specifies the
* magnetic field at the center of the (ix,iy) cell.
* bb(ix,iy,0:2) are the (x,y,z)-components and bb(ix,iy,3) is
* the absolute magnetic field strength.
* (bb(,,3)=sqrt(bb(,,0)**2+bb(,,1)**2+bb(,,2)**2.)
* It will hold that 0.lt.bb(,,3).
*
* wbb1 - (-1:nx,-1:ny,0:3) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), wbb1(ix,iy,0:3) specifies the
* magnetic field at the midpoint of the left face of the (ix,iy) cell.
* Meaning of components is similar to that of the bb array.
*
* wbb2 - (-1:nx,-1:ny,0:3) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), wbb2(ix,iy,0:3) specifies the
* magnetic field at the midpoint of the right face of the (ix,iy) cell.
* Meaning of components is similar to that of the bb array.
*
* wbb3 - (-1:nx,-1:ny,0:3) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), wbb3(ix,iy,0:3) specifies the
* magnetic field at the midpoint of the bottom face of the (ix,iy) cell.
* Meaning of components is similar to that of the bb array.
*
* wbbc - (-1:nx+1,-1:ny+1,0:3) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), wbbc(ix,iy,0:3) specifies the
* magnetic field at the left bottom corner of the (ix,iy) cell.
* For ix=nx+1 wbbc is related to the right bottom corner of
* the cell with ix=nx, similarly for iy=ny+1 wbbc is related to
* the left top corner of the cell iy=ny. wbbc(nx+1,ny+1,:) is related
* to the right top corner of the cell (ix=nx,iy=ny)
* Meaning of components are similat to the bb array.
*
* vol - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), vol(ix,iy) specifies the volume of
* the (ix,iy) cell. (In the case of toroidal symmetry this volume
* includes a factor 2*pi*crx, while in the case of cylindrical
* symmetry it includes a factor 2*pi*cry.)
* It will hold that 0.lt.vol(,,).
*
* hx, hy, hz - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), hx(ix,iy), hy(ix,iy), and hz(ix,iy)
* specify the x-diameter, y-diameter, and z-diameter, respectively,
* of the (ix,iy) cell.
* If redef_gmtry=1, hz(:,,2)=sum(ffbz(:,,0:3))/4 is enforced.
* It will hold that 0.lt.hx(,,) and 0.lt.hy(,,) and 0.lt.hz(,,).
*
* qz - (-1:nx,-1:ny,0:1) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), qz(ix,iy,0) specifies sin(t) and
* qz(ix,iy,1) specifies cos(t), where t is the complementary angle
* between the x-direction and the y-direction on the (ix,iy) cell.
* t is in the range (-pi/2 to pi/2) and vanishes for an orthogonal
* geometry. (qz(ix,iy,0) is the inner product of the x and y unit
* vectors.)
* It will hold that -1.lt.qz(,,0).lt.1 and 0.lt.qz(,,1).le.1.
*
* qc - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), qc(ix,iy) specifies cos(t), where
* t is the complementary angle between the x-direction of the
* (ix,iy) cell and its left face.
* t is in the range (-pi/2 to pi/2) and vanishes for an orthogonal
* geometry.
* It will hold that 0.lt.qc(,,).le.1.
*
* gs - (-1:nx,-1:ny,0:2) real*8 array.
* (Let gsx(,,)=gs(,,0), gsy(,,)=gs(,,1) and gsz(,,)=gs(,,2).)
* For (ix,iy) in (0:nx,-1:ny), gsx(ix,iy) specifies the area of the
* face between the (ix,iy) cell and its left neighbor. For ix.eq.-1,
* gsx(ix,-1:ny) will hold 0.

```

```

* For (ix,iy) in (-1:nx,0:ny), gsy(ix,iy) specifies the area of the
* face between the (ix,iy) cell and its bottom neighbor. For iy.eq.-1,
* gsy(-1:nx,iy) will hold 0.
* For (ix,iy) in (-1:nx,-1:ny), gsz(ix,iy) specifies the area of the
* (ix,iy) cell normal to the third (ignorable) coordinate.
*
* hyl - (-1:nx,-1:ny) real*8 array.
* Actual radial distance between neighbouring flux surfaces.
* hyl(:, :) = hy(:, :) * qc(:, :) and
* vol(:, :) = hx(:, :) * hyl(:, :) * hz(:, :)
* It will hold that 0.lt.hyl(,).
*
* hxb, hyl - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), hxb(ix,iy) and hyl(ix,iy) specify the
* x-diameter and the y-diameter, respectively, of the bottom and
* left faces of the (ix,iy) cell.
* It will hold that 0.lt.hxb(,) and 0.lt.hyl(,).
*
* hzb, hzl - (-1:nx,-1:ny) real*8 array.
* hzc - (-1:nx+1,-1:ny+1) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), hzb(ix,iy) and hzl(ix,iy) specify the
* length along the toroidal direction, respectively, of the bottom and
* left faces of the (ix,iy) cell. It follows that:
* hzl(:, :) * hyl(:, :) = gs(:, :, 0)
* hzb(:, :) * hxb(:, :) = gs(:, :, 1)
* For (ix,iy) in (-1:nx,-1:ny), hzc(ix,iy) specifies the length along
* the toroidal direction of the bottom left vertex of cell (ix,iy).
* For (ix,iy) in (nx+1,-1:ny), hzc(ix,iy) specifies the length along
* the toroidal direction of the top left vertex of cell (nx,iy).
* For (ix,iy) in (-1:nx,ny+1), hzc(ix,iy) specifies the length along
* the toroidal direction of the bottom right vertex of cell (ix,ny).
* For (ix,iy) in (nx+1,ny+1), hzc(ix,iy) specifies the length along
* the toroidal direction of the top right vertex of cell (nx,ny).
* If redef_gmtry=1, the following identities are enforced:
* hzl(:, :) * wbb1(:, :, 2) = (ffbz(:, :, 0)+ffbz(:, :, 2))/2,
* hzb(:, :) * wbbv(:, :, 2) = (ffbz(:, :, 0)+ffbz(:, :, 1))/2,
* hzc(:, :) * wbbc(:, :, 2) = ffbz(:, :, :),
* and conservation of magnetic flux for constant iy (for all ix values):
* hyl(:, iy) * qc(:, iy) * hzl(:, iy) * wbb1(:, iy, 0) = hy(:, iy) * qz(:, iy, 1) * hz(:, iy) * bb(:, iy, 0)
* It will hold that 0.lt.hzb(,) and 0.lt.hzl(,) and 0.lt.hzc(,).
*
* isymm - integer.
* isymm specifies the type of symmetry of the geometry: isymm.eq.0
* implies a slab geometry, isymm.eq.1.or.isymm.eq.2 imply rotational
* symmetry about the crx=0 axis, and isymm.eq.3.or.isymm.eq.4
* indicate rotational symmetry about the cry=0 axis.
* Other values are not allowed.
* isymm.eq.1.or.isymm.eq.3 mean a toroidal geometry case, i.e.
* with a toroidal (out-of-plane) magnetic field component.
* isymm.eq.2.or.isymm.eq.4 mean a cylindrical geometry case, i.e.
* no toroidal magnetic field component.
*
* periodic_bc - integer.
* periodic_bc indicates whether the geometry closes upon itself (i.e.
* whether periodic boundary conditions are necessary). The default value
* is 0. periodic_bc.eq.1 implies that periodic BC are being used. This
* case is used for the simulation of diverted stellarator islands or
* limiters.
*
* /b2cmfs/ - common block.
* /b2cmfs/ is reserved for numerical mhd equilibrium data.
* isymm - integer.
* See above for description.
*
* /b2cmgs/ - common block.
* /b2cmgs/ specifies the singular regions in the geometry.
* nlrms - integer constant.

```

```

*      nlrms is an upper bound for nlrreg, and is used in dimension
*      specifications.
*      nlrreg - integer.
*      nlrreg specifies the number of singular regions. It will hold that
*      0.le.nlrreg.le.nlrms.
*      nlxlo, nlxhi, nlylo, nlyhi - (0:nlrms-1) integer array.
*      For 0.le.i.lt.nlrreg, nlxlo(i), nlxhi(i), nlylo(i) and nlyhi(i) are
*      the bounds of the i-th singular region. Mesh locations (ix,iy)
*      in (nlxlo(i):nlxhi(i),nlylo(i):nlyhi(i)) make up the region, which
*      will be included in (-1:nx,-1:ny).
*      For nlrreg.le.i.lt.nlrms, nlxlo(i), nlxhi(i), nlylo(i) and nlyhi(i)
*      are unspecified.
*      nlloc - (0:nlrms-1) integer array.
*      For 0.le.i.lt.4, nlloc(i) specifies the class of quantities for
*      which the i-th region is special, as follows:
*      nlloc(i) = 0: cell centers, indices (-1:nx,-1:ny).
*      nlloc(i) = 1: cell x-faces, indices (0:nx,-1:ny).
*      nlloc(i) = 2: cell y-faces, indices (-1:nx,0:ny).
*      nlloc(i) = 3: cell corners, indices (0:nx,0:ny).
*      For nlrreg.le.i.lt.nlrms, nlloc(i) is unspecified.
*
*      ..auxiliary variables describing the geometry
*
*      pbs - (-1:nx,-1:ny,0:1) real*8 array.
*      (Let pbsx(,)=pbs(, ,0) and pbsy(,)=pbs(, ,1).)
*      For (ix,iy) in (-1:nx,-1:ny), pbsx(ix,iy) specifies the product
*      (bx/bb)*sx and pbsy(ix,iy) specifies (by/bb)*sy, or (magnetic
*      field pitch)*(surface area), namely the parallel contact area,
*      on the (ix,iy) cell left and bottom faces respectively.
*      In the case where the grid covers the full vacuum vessel,
*      pbsy also contains a contribution from the narrowing of the
*      flux tube when impacting the wall structures.
*      See routine b2xgbs for the form of interpolation employed.
*
*      pbshz - (-1:nx,-1:ny,0:1) real*8 array. !srv 09.06.08
*      (Let pbshzx(,)=pbshz(, ,0) and pbshzy(,)=pbshz(, ,1).)
*      For (ix,iy) in (-1:nx,-1:ny), pbshzx(ix,iy) specifies the product
*      hz*(bx/bb)*sx and pbshzy(ix,iy) specifies hz*(by/bb)*sy, or (magnetic
*      field pitch)*(surface area)*hz
*
*      bzb - (-1:nx,-1:ny) real*8 array contains Bz/B**2 - Bz/<B**2> !srv 09.06.08
*      on the cell centre.
*
*      OnedBsq - (-1:nx,-1:ny) real*8 array. !srv
*      For (ix,iy) in (-1:nx,-1:ny), OnedBsq(ix,iy) specifies the
*      1/bb(, ,3)**2 on the cell centre
*
*      ..basic physical parameters
*
*      /b2cmpa/ - common block
*      /b2cmpa/ specifies basic physics parameters.
*      me, mp, ev, qe, eps0, mu0 - real*8 constant.
*      These parameters are included with /b2cmpa/.
*      me = electron mass (kg)
*      mp = proton mass (kg)
*      ev = 1 electron-Volt (Joule)
*      qe = elementary charge (Coulomb)
*      eps0 = vacuum dielectric permittivity (SI)
*      mu0 = vacuum magnetic permeability (SI)
*      nsdecl - integer constant.
*      nsdecl is an upper bound for the number of species (ns), and is
*      used in dimension specifications.
*      zamin, zamax, zn, am - (0:nsdecl-1) real*8 array.
*      For (is) in (0:ns-1), zn(is) specifies the nuclear charge of
*      species (is), am(is) specifies its atomic mass, and zamin(is) and
*      zamax(is) specify its atomic charge range. All are dimensionless.
*      It will hold that 0.le.zamin(is).le.zamax(is), 0.lt.zn(is) and

```

```

*      0.lt.am(is). It is required that neutral species not be bundled
*      with charged species.
*      For ns.le.is.lt.nsdecl, zamin(is), zamax(is), zn(is), am(is)
*      are undefined.
*
*      ..variables describing the plasma state
*
*      ns - integer.
*      ns specifies the number of atomic species in the calculation.
*      The species are indexed by (0:ns-1).
*      It will hold that 1.le.ns.
*
*      na - (-1:nx,-1:ny,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), na(ix,iy,is) specifies
*      the density of atomic species (is) on the (ix,iy) cell.
*      It will hold that 0.lt.na(,,).
*
*      ua - (-1:nx,-1:ny,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), ua(ix,iy,is) specifies
*      the parallel velocity of atomic species (is) on the (ix,iy) cell.
*      The sign of the parallel velocity indicates whether it is flowing
*      in the same direction as the local poloidal magnetic field.
*
*      te - (-1:nx,-1:ny) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), te(ix,iy) specifies the electron
*      temperature on the (ix,iy) cell. It will hold that 0.lt.te(,).
*
*      ti - (-1:nx,-1:ny) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), ti(ix,iy) specifies the all atom
*      temperature on the (ix,iy) cell. It will hold that 0.lt.ti(,).
*
*      po - (-1:nx,-1:ny) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), po(ix,iy) specifies the electric
*      potential on the (ix,iy) cell.
*
*      For all the fluxes below, the sign convention is as follows.
*      Fluxes between poloidal neighbors are positive if directed in the
*      direction of increasing poloidal (ix) index.
*      Fluxes between radial neighbors are positive if directed in the
*      direction of increasing radial (iy) index.
*
*      fna - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      (Let fnax(,,)=fna(,,0,) and fnay(,,)=fna(,,1,).)
*      For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), fnax(ix,iy,is) specifies
*      the flux of atoms of species (is) through the face between the
*      (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, fnax(ix,-1:ny,0:ns-1) will hold 0.
*      For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), fnay(ix,iy,is) specifies
*      the flux of atoms of species (is) through the face between the
*      (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, fnay(-1:nx,iy,0:ns-1) will hold 0.
*
*      fna_mdf - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      (Let fnax_mdf(,,)=fna_mdf(,,0,) and fnay_mdf(,,)=fna_mdf(,,1,).)
*      For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), fnax_mdf(ix,iy,is) specifies
*      the modified flux of atoms of species (is) through the face between
*      the (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, fnax_mdf(ix,-1:ny,0:ns-1) will hold 0.
*      For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), fnay_mdf(ix,iy,is) specifies
*      the modified flux of atoms of species (is) through the face between
*      the (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, fnay_mdf(-1:nx,iy,0:ns-1) will hold 0.
*      fna_mdf is equal to fna in runs without drifts.
*
*      fna_fcor - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      (Let fna_fcorx(,,)=fna_fcor(,,0,) and fna_fcory(,,)=fna_fcor(,,1,).)
*      For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), fna_fcorx(ix,iy,is) specifies

```



```

* the flux of atoms of species (is) for calculating momentum transport
* through the face between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fna_fcorx(ix,-1:ny,0:ns-1) will hold 0.
* For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), fna_fcory(ix,iy,is) specifies
* the flux of atoms of species (is) for calculating momentum transport
* through the face between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fna_fcory(-1:nx,iy,0:ns-1) will hold 0.
* fna_fcor takes into account the hz toroidal length derivative and
* the Coriolis force.
*
* fhe - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fhex(,)=fhe(,,0) and fhey(,)=fhe(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fhex(ix,iy) specifies the electron
* heat flux through the face between the (ix,iy) cell and
* its left neighbor. For ix.eq.-1, fhex(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fhey(ix,iy) specifies the electron
* heat flux through the face between the (ix,iy) cell and
* its bottom neighbor. For iy.eq.-1, fhey(-1:nx,iy) will hold 0.
*
* fhe_mdf - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fhex_mdf(,)=fhe_mdf(,,0) and fhey_mdf(,)=fhe_mdf(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fhex_mdf(ix,iy) specifies the modified
* electron heat flux through the face between the (ix,iy) cell
* and its left neighbor. For ix.eq.-1, fhex_mdf(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fhey_mdf(ix,iy) specifies the modified
* electron heat flux through the face between the (ix,iy) cell and
* its bottom neighbor. For iy.eq.-1, fhey_mdf(-1:nx,iy) will hold 0.
* fhe_mdf is equal to fhe in runs without drifts.
*
* fhi - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fhix(,)=fhi(,,0) and fhiy(,)=fhi(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fhix(ix,iy) specifies the all atom
* heat flux through the face between the (ix,iy) cell
* and its left neighbor. For ix.eq.-1, fhix(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fhiy(ix,iy) specifies the all atom
* heat flux through the face between the (ix,iy) cell and its
* bottom neighbor. For iy.eq.-1, fhiy(-1:nx,iy) will hold 0.
*
* fhi_mdf - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fhix_mdf(,)=fhi_mdf(,,0) and fhiy_mdf(,)=fhi_mdf(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fhix_mdf(ix,iy) specifies the
* modified all atom heat flux through the face between the (ix,iy) cell
* and its left neighbor. For ix.eq.-1, fhix_mdf(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fhiy_mdf(ix,iy) specifies the modified
* all atom heat flux through the face between the (ix,iy) cell and
* its bottom neighbor. For iy.eq.-1, fhiy_mdf(-1:nx,iy) will hold 0.
*
* fch - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchx(,)=fch(,,0) and fchy(,)=fch(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchx(ix,iy) specifies the electric
* current through the face between the (ix,iy) cell and its left
* neighbor. For ix.eq.-1, fchx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchy(ix,iy) specifies the electric
* current through the face between the (ix,iy) cell and its bottom
* neighbor. For iy.eq.-1, fchy(-1:nx,iy) will hold 0.
*
* ..auxiliary variables describing the plasma state
*
* nb - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), nb(ix,iy) specifies the density of
* one selected atomic species on the (ix,iy) cell. If that species
* has index (is) then nb(,).eq.na(,,is).
*
* ub - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), ub(ix,iy) specifies the parallel
* velocity of one selected atomic species on the (ix,iy) cell.
* The sign of the parallel velocity indicates whether it is flowing

```

```

*      in the same direction as the local poloidal magnetic field.
*
*      For all the fluxes below, the sign convention is as follows.
*      Fluxes between poloidal neighbors are positive if directed in the
*      direction of increasing poloidal (ix) index.
*      Fluxes between radial neighbors are positive if directed in the
*      direction of increasing radial (iy) index.
*
*      fnb - (-1:nx,-1:ny,0:1) real*8 array.
*      fnb specifies the flux of one selected atomic species.
*      If that species has index (is) then fnb(,,)=fna(,,is).
*      precisely: (Let fnbx(,)=fnb(,,0) and fnby(,)=fnb(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), fnbx(ix,iy) specifies the flux of
*      the selected atomic species through the face between the
*      (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, fnbx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), fnby(ix,iy) specifies the flux of
*      the selected atomic species through the face between the
*      (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, fnby(-1:nx,iy) will hold 0.
*      fnb_mdf is equal to fnb in runs without drifts.
*
*      fnb_mdf - (-1:nx,-1:ny,0:1) real*8 array.
*      fnb_mdf specifies the flux of one selected atomic species.
*      If that species has index (is) then fnb_mdf(,,)=fna_mdf(,,is).
*      precisely: (Let fnbx_mdf(,)=fnb_mdf(,,0) and fnby_mdf(,)=fnb_mdf(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), fnbx_mdf(ix,iy) specifies the modified
*      flux of the selected atomic species through the face between the
*      (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, fnbx_mdf(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), fnby_mdf(ix,iy) specifies the modified
*      flux of the selected atomic species through the face between the
*      (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, fnby_mdf(-1:nx,iy) will hold 0.
*
*      fnb_fcor - (-1:nx,-1:ny,0:1) real*8 array.
*      fnb_fcor specifies the flux of one selected atomic species
*      for calculating momentum transport.
*      If that species has index (is) then fnb_fcor(,,)=fna_fcor(,,is).
*      precisely: (Let fnbx_fcor(,)=fnb_fcor(,,0) and fnby_fcor(,)=fnb_fcor(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), fnb_fcorx(ix,iy) specifies the flux of
*      the selected atomic species through the face between the
*      (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, fnb_fcorx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), fnb_fcory(ix,iy) specifies the flux of
*      the selected atomic species through the face between the
*      (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, fnb_fcory(-1:nx,iy) will hold 0.
*      fnb_fcor is equal to fnb in runs without drifts.
*
*      ni - (-1:nx,-1:ny,0:1) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), ni(ix,iy,0) specifies the total atom
*      density on the (ix,iy) cell. It will hold that 0.lt.ni(,0).
*      The defining expression for ni is
*      ni(,0) = (sum is : 0.le.is.lt.ns : na(,is)).
*      For (ix,iy) in (-1:nx,-1:ny), ni(ix,iy,1) specifies the total ion
*      density on the (ix,iy) cell. It will hold that 0.lt.ni(,1).
*      The defining expression for ni is
*      ni(,1) = (sum is : 0.le.is.lt.ns : na(,is)) when rza(,is)>0.
*
*      ne - (-1:nx,-1:ny) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), ne(ix,iy) specifies the electron
*      density on the (ix,iy) cell. It will hold that 0.lt.ne(,).
*      The defining expression for ne is
*      ne(,) = (sum is : 0.le.is.lt.ns : rza(,is)*na(,is)).
*
*      ne2 - (-1:nx,-1:ny) real*8 array.

```

```

* For (ix,iy) in (-1:nx,-1:ny), ne2(ix,iy) specifies the atom
* density weighted by rz2 on the (ix,iy) cell. It will hold that
* 0.lt.ne2(,).
* The defining expression for ne2 is
*   ne2(,) = (sum is : 0.le.is.lt.ns : rz2(,,is)*na(,,is)).
*
* pr - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), pr(ix,iy) specifies the total
* pressure on the (ix,iy) cell. The defining expression for pr is
*   pr(,) = ne(,)*te(,) + (sum is :: na(,,is)*ti(,)).
*
* pz - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), pz(ix,iy) specifies the plasma
* pressure on the (ix,iy) cell. The defining expression for pr is
*   pr(,) = ne(,)*te(,) + (sum is :: na(,,is)*ti(,), not including neutrals).
*
* pe - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), pe(ix,iy) specifies the electron
* pressure on the (ix,iy) cell.
* The defining expression for pe is pe(,) = ne(,)*te(,).
*
* ro - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), ro(ix,iy) specifies the mass
* density on the (ix,iy) cell. The defining expression for ro is
*   ro(,) = (sum is : 0.le.is.lt.ns : am(is)*mp**na(,,is)).
*
* rz - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), rz(ix,iy) specifies the ion mass
* density on the (ix,iy) cell. The defining expression for rz is
*   rz(,) = (sum is : 0.le.is.lt.ns .and. .not.is_neutral(is) :
*             am(is)*mp**na(,,is)).
*
* ne2m - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), ne2m(ix,iy) specifies the atom
* density weighted by rz2/am on the (ix,iy) cell. It will hold
* that 0.lt.ne2m(,).
* The defining expression for ne2m is
*   ne2m(,) = (sum is :: (rz2(,,is)/am(is))*na(,,is)).
*
* fne - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fnex(,)=fne(,,0) and fney(,)=fne(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fnex(ix,iy) specifies the flux of
* electrons through the face between the (ix,iy) cell and its
* left neighbor. For ix.eq.-1, fnex(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fney(ix,iy) specifies the flux of
* electrons through the face between the (ix,iy) cell and its
* bottom neighbor. For iy.eq.-1, fney(-1:nx,iy) will hold 0.
* The defining expression for fne is
*   fne(,,) = (sum is :: rza(,,is)*fna(,,is))-(1/qe)*fch(,,)
*
* fni - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fnix(,)=fni(,,0) and fniy(,)=fni(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fnix(ix,iy) specifies the flux of
* atoms (all species) through the face between the (ix,iy) cell
* and its left neighbor. For ix.eq.-1, fnix(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fniy(ix,iy) specifies the flux of
* atoms (all species) through the face between the (ix,iy) cell
* and its bottom neighbor. For iy.eq.-1, fniy(-1:nx,iy) will hold 0.
* The defining expression for fni is
*   fni(,,) = (sum is :: fna(,,is))
*
* fchvispar - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchvisparx(,)=fchvispar(,,0) and fchvispary(,)=fchvispar(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchvisparx(ix,iy) specifies the
* electric current driven by parallel viscosity through the face
* between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fchvisparx(ix,-1:ny) will hold 0.

```

```

* For (ix,iy) in (-1:nx,0:ny), fchvispary(ix,iy) specifies the
* electric current driven by parallel viscosity through the face
* between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fchvispary(-1:nx,iy) will hold 0.
*
* fchvisq - (-1:nx,-1:ny,0:1) real*8 array.
* fchvisq specifies the electric current produced by the components
* of the viscosity tensor which is connected with the heat fluxes.
* (Let fchvisqx(,)=fchvisq(,,0) and fchvisqy(,)=fchvisq(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchvisqx(ix,iy) specifies the current
* through the face between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fchvisqx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchvisqy(ix,iy) specifies the current
* through the face between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fchvisqy(-1:nx,iy) will hold 0.
*
* fchinert - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchinertx(,)=fchinert(,,0) and fchinerty(,)=fchinert(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchinertx(ix,iy) specifies the
* contributions from inertia and gyroviscosity current through
* the face between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fchinertx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchinerty(ix,iy) specifies the
* contributions from inertia and gyroviscosity current through
* the face between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fchinerty(-1:nx,iy) will hold 0.
*
* fchanml - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchanmlx(,)=fchanml(,,0) and fchanmly(,)=fchanml(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchanmlx(ix,iy) specifies the
* anomalous current through the face between the (ix,iy) cell
* and its left neighbor.
* For ix.eq.-1, fchanmlx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchanmly(ix,iy) specifies the
* amomalous current through the face between the (ix,iy) cell
* and its bottom neighbor.
* For iy.eq.-1, fchanmly(-1:nx,iy) will hold 0.
*
* fchdia - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchdiax(,)=fchdia(,,0) and fchdiay(,)=fchdia(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchdiax(ix,iy) specifies the
* modified diamagnetic current through the face between
* the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fchdiax(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchdiay(ix,iy) specifies the
* modified diamagnetic current through the face between
* the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fchdiay(-1:nx,iy) will hold 0.
*
* fchin - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchinx(,)=fchin(,,0) and fchiny(,)=fchin(,,1).)
* For (ix,iy) in (0:nx,-1:ny), fchinx(ix,iy) specifies the
* contribution from ion-neutral friction through the face between
* the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fchinx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), fchiny(ix,iy) specifies the
* contribution from ion-neutral friction through the face between
* the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, fchiny(-1:nx,iy) will hold 0.
*
* fch_p - (-1:nx,-1:ny,0:1) real*8 array.
* For (ix,iy) in (0:nx,-1:ny), fch_p(ix,iy,0) specifies the product of
* the parallel electric current and the poloidal magnetic field component
* sign through the face between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, fch_p(ix,-1:ny,0) will hold 0.
* fch_p(-1:nx,-1:ny,1) = 0
*

```

```

* fchvisper - (-1:nx,-1:ny,0:1) real*8 array.
* (Let fchvisperx(,)=fchvisper(,,0) and fchvispery(,)=fchvisper(,,1).)
* For (ix,iy) in (-1:nx,-1:ny), fchvisperx(ix,iy) = 0.
* For (ix,iy) in (-1:nx, 0:ny), fchvispery(ix,iy) specifies the
* electric current connected with contribution from
* the perpendicular viscosity through the face between the (ix,iy) cell
* and its bottom neighbor.
* For iy.eq.-1, fchvispery(-1:nx,iy) will hold 0.
*
* uadia - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let uadiax(,,)=uadia(,,0,) and uadiay(,)=uadia(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), uadiax(ix,iy,is) specifies
* the total effective drift velocity of species (is) in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the
* poloidal direction.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), uadiay(ix,iy,is) specifies
* the total effective drift velocity of species (is) in cell (ix,iy)
* in the radial direction.
* uadia is computed at cell centers if 'b2tfnb_drift_style' is '0'
* uadia is computed at cell faces otherwise (default)
*
* vadia - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let vadiax(,,)=vadia(,,0,) and vadiay(,)=vadia(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vadiax(ix,iy,is) specifies
* the effective diamagnetic drift velocity of species (is) in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the
* poloidal direction.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vadiay(ix,iy,is) specifies
* the effective diamagnetic drift velocity of species (is) in cell (ix,iy)
* in the radial direction.
* vadia is computed at cell centers if 'b2tfnb_drift_style' is '0'
* vadia is computed at cell faces otherwise (default)
*
* vedia - (-1:nx,-1:ny,0:1) real*8 array.
* (Let vediax(,)=vedia(,,0) and vediay(,)=vedia(,,1).)
* For (ix,iy) in (-1:nx,-1:ny), vediax(ix,iy) specifies
* the effective diamagnetic drift velocity of electrons in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the
* poloidal direction.
* For (ix,iy) in (-1:nx,-1:ny), vediay(ix,iy) specifies
* the effective diamagnetic drift velocity of electrons in cell (ix,iy)
* in the radial direction.
* vedia is computed at cell centers if 'b2tfnb_drift_style' is '0'
* vedia is computed at cell faces otherwise (default)
*
* wadia - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let wadiax(,,)=wadia(,,0,) and wadiay(,)=wadia(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), wadiax(ix,iy,is) specifies
* the total diamagnetic drift velocity of species (is) in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the
* poloidal direction.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), wadiay(ix,iy,is) specifies
* the total diamagnetic drift velocity of species (is) in cell (ix,iy)
* in the radial direction.
* wadia is computed at cell centers if 'b2tfnb_drift_style' is '0'
* wadia is computed at cell faces otherwise (default)
*
* vaecrb - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let vaecrbx(,,)=vaecrb(,,0,) and vaecrby(,)=vaecrb(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vaecrbx(ix,iy,is) specifies
* the ExB drift velocity of species (is) in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the

```

```

* poloidal direction.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vaecrb(ix,iy,is) specifies
* the ExB drift velocity of species (is) in cell (ix,iy)
* in the radial direction.
* vaecrb is computed at cell centers if 'b2tfnb_drift_style' is '0'
* vaecrb is computed at cell faces otherwise (default)
*
* veecrb - (-1:nx,-1:ny,0:1) real*8 array.
* (Let veecrbx(,)=veecrb(,0) and veecrby(,)=veecrb(,1).)
* For (ix,iy) in (-1:nx,-1:ny), veecrbx(ix,iy) specifies
* the ExB drift velocity of electrons in cell (ix,iy)
* in the diamagnetic direction (i.e. within the flux surface but
* perpendicular to the magnetic field), but projected onto the
* poloidal direction.
* For (ix,iy) in (-1:nx,-1:ny), veecrby(ix,iy) specifies
* the ExB drift velocity of electrons in cell (ix,iy)
* in the radial direction.
* veecrb is computed at cell centers if 'b2tfnb_drift_style' is '0'
* veecrb is computed at cell faces otherwise (default)
*
* kinrgy - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), kinrgy(ix,iy,is) specifies
* the kinetic energy of a particle of species (is) in cell (ix,iy).
*
* ..residuals and corrections
*
* resco - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), resco(ix,iy,is) specifies
* the residual of the continuity equation on the (ix,iy) cell for
* species (is).
*
* resmo - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), resmo(ix,iy,is) specifies
* the residual of the parallel momentum equation on the (ix,iy)
* cell for species (is).
*
* resmb - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), resmb(ix,iy) specifies the residual
* of the parallel momentum equation on the (ix,iy) cell for the
* selected species. If that species has index (is) then
* resmb(,).eq.resmo(,,is).
*
* resmt - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), resmt(ix,iy) specifies the residual
* of the total parallel momentum equation on the (ix,iy) cell.
*
* reshe - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), reshe(ix,iy) specifies the residual
* of the electron heat balance equation on the (ix,iy) cell.
*
* reshi - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), reshe(ix,iy) specifies the residual
* of the ion heat balance equation on the (ix,iy) cell.
*
* resht - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), resht(ix,iy) specifies the residual
* of the total heat balance equation on the (ix,iy) cell.
*
* respo - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), respo(ix,iy) specifies the
* residual of the electric potential equation on the (ix,iy) cell.
*
* corpa - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), corpa(ix,iy,is) specifies
* the last computed correction to the pressure of species (is) on
* the (ix,iy) cell. corpa is computed by solving an equation that

```

```

* has resco as its right hand side. It is not necessarily the same
* as the pressure correction over one complete iteration.
*
* corua - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), corua(ix,iy,is) specifies
* the last computed correction to the parallel velocity ua of
* species (is) on the (ix,iy) cell.
* corua is computed by solving an equation that has resmo as its
* right hand side. It is not necessarily the same as the change in
* the parallel velocity over one complete iteration.
*
* corut - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (0:nx,-1:ny), corut(ix,iy) specifies the last
* computed correction to the parallel velocity ua of all species
* simultaneously on the (ix,iy) cell.
* corut is computed by solving an equation that has resmt as its
* right hand side. It is not necessarily the same as the change in
* the parallel velocity over one complete iteration.
*
* corte - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), corte(ix,iy) specifies the last
* computed correction to the electron temperature on the (ix,iy)
* cell. corte is computed by solving an equation that has reshe as
* its right hand side. It is not necessarily the same as the change
* in the electron temperature over one complete iteration.
*
* corti - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), corti(ix,iy) specifies the last
* computed correction to the atom temperature on the (ix,iy) cell.
* corti is computed by solving an equation that has reshi as its
* right hand side. It is not necessarily the same as the change in
* the atom temperature over one complete iteration.
*
* cortt - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), cortt(ix,iy) specifies the last
* computed correction to the combined temperature on the (ix,iy)
* cell. cortt is computed by solving an equation that has resht as
* its right hand side. It is not necessarily the same as the change
* in the electron or atom temperature over one complete iteration.
*
* corpo - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), corpo(ix,iy) specifies the last
* computed correction to the electric potential on the (ix,iy)
* cell. corpo is computed by solving an equation that has respo as
* its right hand side. It is not necessarily the same as the change
* in the electric potential over one complete iteration.
*
* aresco - (0:ns-1) real*8 array.
* For (is) in (0:ns-1), aresco(is) specifies the scaled norm of the
* residual of the continuity equation for species (is).
*
* aresmo - (0:ns-1) real*8 array.
* For (is) in (0:ns-1), aresmo(is) specifies the scaled norm of the
* residual of the parallel momentum equation for species (is).
*
* aresmt - real.
* aresmt specifies the scaled norm of the residual of the total
* parallel momentum equation.
*
* areshe - real.
* areshe specifies the scaled norm of the residual of the electron
* heat balance equation.
*
* areshi - real.
* areshi specifies the scaled norm of the residual of the all atom
* heat balance equation.
*

```

```

*      arespo - real.
*      arespo specifies the scaled norm of the residual of the electric
*      potential equation.
*
*      acorpa - (0:ns-1) real*8 array.
*      For (is) in (0:ns-1), acorpa(is) specifies the norm of the last
*      computed correction for the continuity equation for species (is).
*
*      acorua - (0:ns-1) real*8 array.
*      For (is) in (0:ns-1), acorua(is) specifies the norm of the last
*      computed correction for the momentum balance equation for
*      species (is).
*
*      acorut - real.
*      acorut specifies the norm of the last computed correction for the
*      parallel momentum balance equation for all species simultaneously.
*
*      acorte - real.
*      acorte specifies the norm of the last computed correction for the
*      electron heat balance equation.
*
*      acorti - real.
*      acorti specifies the norm of the last computed correction for the
*      all atom heat balance equation.
*
*      acorpo - real.
*      acorpo specifies the norm of the last computed correction for the
*      electric potential equation.
*
*      ..rate coefficients
*
*      /b2cmrc/ - common block.
*      /b2cmrc/ contains a table of atomic rate coefficients.
*      rtnt - integer constant.
*      rtnt specifies the number of intervals in the discretisation
*      of the temperature range. The table will contain data for
*      (rtnt+1) values of the temperature variable.
*      rtnn - integer constant.
*      rtnn specifies the number of intervals in the discretisation
*      of the density range. The table will contain data for (rtnn+1)
*      values of the density variable.
*      rtns - integer.
*      rtns specifies the total number of atomic species; i.e.
*      (sum inuc : 0.le.inuc.lt.nnuc : izhi(inuc)-izlo(inuc)+1) ,
*      where inuc is a nuclear species index, izlo(inuc) is the lowest
*      and izhi(inuc) is the highest atomic charge considered in B2 for
*      that nuclear species.
*      rtns equals the variable ns used elsewhere in B2.
*      rtdumm - integer.
*      rtdumm is a dummy integer variable, introduced in order to obtain
*      an even-length sequence of integer variables in the common block.
*      rtzmin, rtzmax, rtzn - (0:rtns-1) real*8 array.
*      For 0.le.is.lt.rtns, rtzmin(is) and rtzmax(is) specify the atomic
*      charge range, and rtzn(is) specifies the nuclear charge
*      of atomic species (is).
*      The izhi(inuc)-izlo(inuc)+1 charge states for each nuclear
*      species are taken in natural order, starting at izlo(inuc); the
*      highest state of nuclear species (inuc) is followed by the lowest
*      state of (inuc+1). rtzmin, rtzmax and rtzn are equal to the arrays
*      zamin, zamax and zn used elsewhere in B2.
*      rtt - (0:rtnt) real*8 array.
*      rtt(0:rtnt) specifies the sequence of values of the temperature
*      variable, expressed in ev, for which the rate coefficients are
*      evaluated. rtt(0:rtnt) will be positive and increasing.
*      rtn - (0:rtnn) real*8 array.
*      rtn(0:rtnn) specifies the sequence of values of the density
*      variable, expressed in m**-3, for which the rate coefficients

```



```

* are evaluated. rtn(0:rtnn) will be positive and increasing.
*   rltt - (0:rtnt) real*8 array.
*   rltt(0:rtnt) holds log(rtt(0:rtnt)).
*   rltln - (0:rtnn) real*8 array.
*   rltln(0:rtnn) holds log(rtn(0:rtnn)).
*   rltlsa - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlsa(i,j,k) specifies
*   logu(rtrsa), where rtrsa is the ionisation rate coefficient at
*   electron temperature rtt(i), electron density rtn(j), for the
*   process k->k+1. rtrsa has dimension m**3/s. The function logu is
*   the same as (log) except that logu(0.0) is equal to some huge
*   negative number.
*   rltlra - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlra(i,j,k) specifies
*   logu(rtrra), where rtrra is the recombination rate coefficient at
*   electron temperature rtt(i) and electron density rtn(j), for the
*   process k->k-1. rtrra has dimension m**3/s.
*   rltlqa - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlqa(i,j,k) specifies
*   logu(rtrqa), where rtrqa is the electron heat loss rate
*   coefficient at electron temperature rtt(i) and electron density
*   rtn(j), for processes starting from atomic species k. rtrqa has
*   dimension ev*m**3/s.
*   rltlcx - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlcx(i,j,k) specifies
*   logu(rtrcx), where rtrcx is the charge exchange rate coefficient
*   at neutral hydrogen temperature/amu rtt(i) and electron density
*   rtn(j), for the process k->k-1. rtrcx has dimension m**3/s.
*   rltlza - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlza(i,j,k) specifies
*   the effective charge state at
*   electron temperature rtt(i), electron density rtn(j), for the
*   superstage k. rltlza has dimension e.
*   rltlz2 - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlz2(i,j,k) specifies
*   the effective squared charge at
*   electron temperature rtt(i), electron density rtn(j), for the
*   superstage k. rltlz2 has dimension e**2.
*   rltlpt - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlpt(i,j,k) specifies
*   the cumulative ionisation energy at
*   electron temperature rtt(i), electron density rtn(j),
*   to reach superstage k. rltlpt has dimension eV.
*   rltlpi - (0:rtnt,0:rtnn,0:rtns-1) real*8 array.
*   For (i,j,k) in (0:rtnt,0:rtnn,0:rtns-1), rltlpi(i,j,k) specifies
*   the effective ionisation energy at
*   electron temperature rtt(i), electron density rtn(j),
*   to reach superstage k from superstage k-1. rltlpi has dimension eV.
*
*   nscx - integer.
*   nscx is the number of charge exchanging neutral hydrogen species
*   that are present in the calculation. (So, usually nscx.eq.1).
*   It will hold that 0.le.nscx.lt.10.
*
*   iscx - (0:nscxmax-1) integer array.
*   iscx(0:nscx-1) will hold the species indices for the various kinds
*   of neutral hydrogen that are present in the calculation.
*   For 0.le.k.lt.nscx it will hold that 0.le.iscx(k).lt.ns.
*
*   rlsa - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*   For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlsa(ix,iy,0,is) and
*   rlsa(ix,iy,1,is) are the coefficients in the representation
*   rsa = exp(rlsa(,,0)+rlsa(,,1))*log(te(ix,iy)/ev)
*   that specifies the rate coefficient for ionisation from atomic
*   species (is) to species (is+1). (For is.eq.ns-1 or in case the
*   atomic species (is) and (is+1) belong to different nuclear
*   species this expression will underflow, and its value is then

```

```

*      understood as 0.)
*      rsa has dimension m**3/s.
*
*      rlra - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlra(ix,iy,0,is) and
*      rlra(ix,iy,1,is) are the coefficients in the representation
*      rra = exp(rlra(.,0,)+rlra(.,1,)*log(te(ix,iy)/ev))
*      that specifies the rate coefficient for recombination from atomic
*      species (is) to species (is-1). (For is.eq.0 or in case the
*      atomic species (is) and (is-1) belong to different nuclear
*      species this expression will underflow, and its value is then
*      understood as 0.)
*      rra has dimension m**3/s.
*
*      rlqa - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlqa(ix,iy,0,is) and
*      rlqa(ix,iy,1,is) are the coefficients in the representation
*      rqa = ev*exp(rlqa(.,0,)+rlqa(.,1,)*log(te(ix,iy)/ev))
*      that specifies the rate coefficient for electron heat loss terms
*      that are proportional to the density of the atomic species (is).
*      rqa has dimension J*m**3/s.
*
*      rlcx - (-1:nx,-1:ny,0:1,0:ns-1,0:nscxmax-1) real*8 array.
*      For (ix,iy,is,k) in (-1:nx,-1:ny,0:ns-1,0:nscx-1),
*      rlcx(ix,iy,0,is,k) and rlcx(ix,iy,1,is,k) are the coefficients in
*      the expression
*      rcx = exp(rlcx(.,0,,)+rlcx(.,1,,)*log(th0/(amh0*ev)))
*      that specifies the rate coefficient for charge exchange
*      recombination from state (is) to state (is-1) by collisions with
*      the neutral hydrogen species iscx(k); here, amh0 is the atomic
*      mass of that neutral species. (For is.eq.0 or in case the atomic
*      species (is) and (is-1) belong to different nuclear species this
*      expression will underflow, and its value is then understood as 0.)
*      rcx has dimension m**3/s.
*
*      rlza - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlza(ix,iy,0,is) and
*      rlza(ix,iy,1,is) are the coefficients in the representation
*      rza = rlza(.,0,)+rlza(.,1,)*log(te(ix,iy)/ev)
*      that specifies the effective charge of stage (is).
*      rza has dimension e.
*
*      rlz2 - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlz2(ix,iy,0,is) and
*      rlz2(ix,iy,1,is) are the coefficients in the representation
*      rz2 = rlz2(.,0,)+rlz2(.,1,)*log(te(ix,iy)/ev)
*      that specifies the effective square charge of stage (is).
*      rz2 has dimension e**2.
*
*      rlpt - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlpt(ix,iy,0,is) and
*      rlpt(ix,iy,1,is) are the coefficients in the representation
*      rpt = rlpt(.,0,)+rlpt(.,1,)*log(te(ix,iy)/ev)
*      that specifies the cumulative ionisation potential to stage (is).
*      rpt has dimension eV.
*
*      rlpi - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rlpi(ix,iy,0,is) and
*      rlpi(ix,iy,1,is) are the coefficients in the representation
*      rpi = rlpi(.,0,)+rlpi(.,1,)*log(te(ix,iy)/ev)
*      that specifies the effective ionisation potential to stage (is)
*      from stage (is-1).
*      rpi has dimension eV.
*
*      rsa - (-1:nx,-1:ny,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rsa(ix,iy,is) specifies
*      the rate coefficient for ionisation from atomic species (is) to

```

```

* species (is+1). For is.eq.ns-1 or in case the atomic species (is)
* and (is+1) belong to different nuclear species, this coefficient
* will be 0.
* rsa has dimension m**3/s.
*
* rra - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rra(ix,iy,is) specifies
* the rate coefficient for recombination from atomic species (is)
* to species (is-1). For is.eq.0 or in case the atomic species (is)
* and (is-1) belong to different nuclear species, this coefficient
* will be 0.
* rra has dimension m**3/s.
*
* rqa - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rqa(ix,iy,is) specifies
* the rate coefficient for electron heat loss terms that are
* proportional to the density of the atomic species (is).
* rqa has dimension J*m**3/s.
*
* rcx - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), rcx(ix,iy,is) specifies
* the rate coefficient for charge exchange recombination from
* atomic species (is) to species (is-1) by collisions with the
* selected species of neutral hydrogen. For is.eq.0 or in case the
* atomic species (is) and (is-1) belong to different nuclear
* species, this coefficient will be 0.
* rcx has dimension m**3/s.
*
* ..boundary conditions, parameters
*
* /b2cmpb/ - common block.
* /b2cmpb/ specifies physics parameters for boundary conditions.
* For this purpose the boundary is divided up into regions, and for
* each region a set of coefficients defines the local conditions.
* For detailed description of /b2cmpb/ see the code, especially the
* routine b2stbc.
*
* cbnrmx - integer constant.
* cbnrmx is an upper bound for the number of boundary regions nreg,
* (nreg.eq.cbnrso+cbnrno+cbnrwe+cbnrea) and is used in dimension
* specifications.
*
* cbirso, cbirno, cbirwe, cbirea - integer.
* cbirso(-no,-we,-ea) is the index of the first region on the
* south (north, west, east) boundary.
*
* cbnrso, cbnrno, cbnrwe, cbnrea - integer.
* cbnrso(-no,-we,-ea) is the number of regions on the south
* (north, west, east) boundary.
* It will hold that 1.le.cbnrso, 1.le.cbnrno, 1.le.cbnrwe, and
* 1.le.cbnrea; and 0.eq.cbirso, cbirso+cbnrso.eq.cbirno,
* cbirno+cbnrno.eq.cbirwe, cbirwe+cbnrwe.eq.cbirea.
*
* cbrbrk - (0:cbnrmx-1) real*8 array.
* cbrbrk defines the boundaries between the boundary regions.
*
* cbsna - (0:7,0:nsdecl-1,0:cbnrmx-1) real*8 array.
* For 0.le.is.lt.ns and 0.le.ireg.lt.nreg, the eight coefficients
* cbsna(0:7,is,ireg) specify standard form boundary conditions for
* the continuity equation for species (is) on the (ireg) boundary
* region.
*
* cbsmo - (0:7,0:nsdecl-1,0:cbnrmx-1) real*8 array.
* For 0.le.is.lt.ns and 0.le.ireg.lt.nreg, the eight coefficients
* cbsmo(0:7,is,ireg) specify standard form boundary conditions for
* the parallel momentum balance equation for species (is) on the
* (ireg) boundary region.
*
* cbshi - (0:7,0:nsdecl-1,0:cbnrmx-1) real*8 array.
* For 0.le.is.lt.ns and 0.le.ireg.lt.nreg, the eight coefficients
* cbshi(0:7,is,ireg) specify standard form boundary conditions for
* the contribution to that atom heat equation by species (is) on
* the (ireg) boundary region.
*
* cbshe - (0:7,0:cbnrmx-1) real*8 array.

```

```

*      For 0.le.ireg.lt.nreg, the eight coefficients cbshe(0:7,ireg)
*      specify standard form boundary conditions for the electron heat
*      equation on the (ireg) boundary region.
*      cbsch - (0:7,0:cbnrmx-1) real*8 array.
*      For 0.le.ireg.lt.nreg, the eight coefficients cbsch(0:7,ireg)
*      specify standard form boundary conditions for the electric
*      potential equation on the (ireg) boundary region.
*      cbrec - (0:7,0:nsdecl-1,0:cbnrmx-1) real*8 array.
*      For 0.le.is.lt.ns-1 and 0.le.ireg.lt.nreg, the eight coefficients
*      cbrec(0:7,is,ireg) specify the recycling process for species (is)
*      on the boundary region ireg. The exact interpretation of these
*      coefficients is defined by the routine that computes the source
*      terms due to the recycling process (b2stbr). Routine b2stbr is
*      problem dependent.
*      cbmsa - (0:7,0:cbnrmx-1) real*8 array.
*      cbmsc - (0:7,0:nsdecl-1,0:cbnrmx-1) real*8 array.
*      For 0.le.ireg.lt.nreg, the eight coefficients cbmsa(0:7,ireg)
*      specify miscellaneous processes on the boundary region ireg.
*      For 0.le.is.lt.ns-1 and 0.le.ireg.lt.nreg, the eight coefficients
*      cbmsc(0:7,is,ireg) specify miscellaneous processes for species
*      (is) on the boundary region ireg. The interpretation of these
*      coefficients is defined by the routine that computes the source
*      terms due to miscellaneous processes (b2stbm). Routine b2stbm is
*      application dependent.
*
*      ..source terms from neutral gas model
*
*      /b2cmsr/ - common block.
*      /b2cmsr/ specifies source terms from neutral gas model.
*      (Presently empty--to be used in conjunction with a Monte Carlo
*      neutral gas code.)
*
*      ..source coefficients
*
*      sna - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), sna(ix,iy,0,is) and
*      sna(ix,iy,1,is) are the coefficients in the expression
*      sna(ix,iy,0,is)+sna(ix,iy,1,is)*na(ix,iy,is)
*      that specifies a linear approximation to the assumed particle
*      source of species (is) integrated over the (ix,iy) cell.
*
*      sne - (-1:nx,-1:ny,0:1) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), sne(ix,iy,0) and
*      sne(ix,iy,1) are the coefficients in the expression
*      sne(ix,iy,0)+sne(ix,iy,1)*ne(ix,iy)
*      that specifies a linear approximation to the assumed non-ambipolar
*      electron source of integrated over the (ix,iy) cell.
*
*      smo - (-1:nx,-1:ny,0:3,0:ns-1) real*8 array.
*      For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), smo(ix,iy,0:3,is) are the
*      coefficients in the expression
*      smo(ix,iy,0,is)+
*      smo(ix,iy,1,is)*ua(ix,iy,is)+
*      smo(ix,iy,2,is)*roxa(ix,iy,is)+
*      smo(ix,iy,3,is)*roxa(ix,iy,is)*ua(ix,iy,is)
*      that specifies a linear approximation to the parallel momentum
*      source for species (is) integrated over the (ix,iy) cell; here
*      roxa(ix,iy,is) = am(is)*mp*na(ix,iy,is) .
*
*      she - (-1:nx,-1:ny,0:3) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), she(ix,iy,0:3) are the coefficients
*      in the expression
*      she(ix,iy,0)+
*      she(ix,iy,1)*te(ix,iy)+
*      she(ix,iy,2)*ne(ix,iy)+
*      she(ix,iy,3)*ne(ix,iy)*te(ix,iy)
*      that specifies a linear approximation to the assumed electron heat

```

```

*      source integrated over the (ix,iy) cell.
*
*      shi = (-1:nx,-1:ny,0:3) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), shi(ix,iy,0:3) are the coefficients
*      in the expression
*      shi(ix,iy,0)+
*      shi(ix,iy,1)*ti(ix,iy)+
*      shi(ix,iy,2)*ni(ix,iy)+
*      shi(ix,iy,3)*ni(ix,iy)*ti(ix,iy)
*      that specifies a linear approximation to the assumed all atom heat
*      source integrated over the (ix,iy) cell.
*
*      sch = (-1:nx,-1:ny,0:3) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), sch(ix,iy,0:3) are the coefficients
*      in the expression
*      sch(ix,iy,0)+
*      sch(ix,iy,1)*po(ix,iy)+
*      sch(ix,iy,2)*ne(ix,iy)+
*      sch(ix,iy,3)*ne(ix,iy)*po(ix,iy)
*      that specifies a linear approximation to the assumed electric
*      charge source integrated over the (ix,iy) cell.
*
*      ..partial source coefficients
*
*      sna0 = (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
*      For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), sna0(ix,iy,0,is) and
*      sna0(ix,iy,1,is) are the coefficients in the expression
*      sna0(ix,iy,0,is)+sna0(ix,iy,1,is)*na(ix,iy,is)
*      that specifies a linear approximation to some specific
*      contribution to the particle source of species (is) integrated
*      over the (ix,iy) cell.
*
*      sne0 = (-1:nx,-1:ny,0:1) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), sne0(ix,iy,0) and
*      sne0(ix,iy,1) are the coefficients in the expression
*      sne0(ix,iy,0)+sne0(ix,iy,1)*ne(ix,iy)
*      that specifies a linear approximation to some specific
*      contribution to the particle non-ambipolar electron source
*      integrated over the (ix,iy) cell.
*
*      smo0 = (-1:nx,-1:ny,0:3,0:ns-1) real*8 array.
*      For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), smo0(ix,iy,0:3,is) are the
*      coefficients in the expression
*      smo0(ix,iy,0,is)+
*      smo0(ix,iy,1,is)*ua(ix,iy,is)+
*      smo0(ix,iy,2,is)*roxa(ix,iy,is)+
*      smo0(ix,iy,3,is)*roxa(ix,iy,is)*ua(ix,iy,is)
*      that specifies a linear approximation to some specific
*      contribution to the non-frictional parallel momentum source for
*      species (is) integrated over the (ix,iy) cell. Here,
*      roxa(ix,iy,is) = am(is)*mp*na(ix,iy,is) .
*
*      she0 = (-1:nx,-1:ny,0:3) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), she0(ix,iy,0:3) are the coefficients
*      in the expression
*      she0(ix,iy,0)+
*      she0(ix,iy,1)*te(ix,iy)+
*      she0(ix,iy,2)*ne(ix,iy)+
*      she0(ix,iy,3)*ne(ix,iy)*te(ix,iy)
*      that specifies a linear approximation to some specific
*      contribution to the electron heat source integrated over the
*      (ix,iy) cell.
*
*      shi0 = (-1:nx,-1:ny,0:3) real*8 array.
*      For (ix,iy) in (-1:nx,-1:ny), shi0(ix,iy,0:3) are the coefficients
*      in the expression
*      shi0(ix,iy,0)+

```

```

*      shi0(ix,iy,1)*ti(ix,iy)+
*      shi0(ix,iy,2)*ni(ix,iy)+
*      shi0(ix,iy,3)*ni(ix,iy)*ti(ix,iy)
* that specifies a linear approximation to some specific
* contribution to the all atom heat source integrated over the
* (ix,iy) cell.
*
* sch0 - (-1:nx,-1:ny,0:3) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), sch0(ix,iy,0:3) are the coefficients
* in the expression
*      sch0(ix,iy,0)+
*      sch0(ix,iy,1)*po(ix,iy)+
*      sch0(ix,iy,2)*ne(ix,iy)+
*      sch0(ix,iy,3)*ne(ix,iy)*po(ix,iy)
* that specifies a linear approximation to some specific
* contribution to the electric charge source integrated over the
* (ix,iy) cell.
*
* ..transport coefficients, parameters
*
* /b2cmpt/ - common block.
* (Note: the integer constant nsdecl, which comes with /b2cmpa/,
* is used instead of (ns) in dimension specifications.)
* cdfd0 - (0:7,0:ns-1) real*8 array.
* cdfd0 contains some auxiliary coefficients. For 0.le.is.lt.ns,
* the local expression
*      df0(,,is) = cdfd0(0,is)+cdfd0(1,is)/(ni(,,0)/1.0e20)+
*      cdfd0(2,is)*sqrt(ti(,)/(am(is)*mp))/
*      (cdfd0(3,is)*ni(,,0)+cdfd0(4,is)*ne(,))
* is used in the definition of the anomalous transport coefficients.
* df0 is meant to represent a diffusivity. If cdfd0(2,is).eq.0 then
* the corresponding term in df0 is understood as 0 independent of
* cdfd0(3:4,is).
* cfdna - (0:7,0:ns-1) real*8 array.
* For 0.le.is.lt.ns, cfdna(,is) are the coefficients in the
* local expression
*      difna(,,is) = cfdna(0,is)+cfdna(1,is)/(ne(,)/1.0e20)+
*      cfdna(2,is)*df0(,,is)+
*      cfdna(3,is)*te(,)/ev/abs(bb(,,3))/16
* that specifies the anomalous diffusivity with respect to
* grad.na; the transport equation for species (is) contains a term
* difna(,,is)*'grad.na'.
* If cfdna(7,is) is not zero, the dna0 array is flux-scaled with
* a scaling factor of cfdna(7,is)*(hyl(,)/hyl(ixref,))*2
* where hyl is the radial width of the cell and ixref is a reference
* poloidal position (usually the outer midplane or 'jxa').
* cfdna(4:6,) are not used in the default version of the code.
* cfdpa - (0:7,0:ns-1) real*8 array.
* For 0.le.is.lt.ns, cfdpa(,is) are the coefficients in the
* local expression
*      difpa(,,is) = cfdpa(0,is)+cfdpa(1,is)/(ne(,)/1.0e20)+
*      cfdpa(2,is)*df0(,,is)+
*      cfdpa(3,is)*te(,)/ev/abs(bb(,,3))/16
* that specifies the anomalous diffusivity with respect to
* grad.pa; the transport equation for species (is) contains a term
* (1/(rza(,,is)*te(,)+ti(,)))*difpa(,,is)*'grad.pa'.
* If cfdpa(7,is) is not zero, the dpa0 array is flux-scaled with
* a scaling factor of cfdpa(7,is)*(hyl(,)/hyl(ixref,))*2
* where hyl is the radial width of the cell and ixref is a reference
* poloidal position (usually the outer midplane or 'jxa').
* cfdpa(4:6,) are not used in the default version of the code.
* cfvla - (0:7,0:ns-1) real*8 array.
* For 0.le.is.lt.ns, cfvla(,is) are the coefficients in the
* local expression
*      vlay(,,is) = cfvla(0,is)+cfvla(1,is)/(ne(,)/1.0e20)+
*      cfvla(2,is)*df0(,,is)+
*      cfvla(3,is)*te(,)/ev/abs(bb(,,3))/16

```

```

*      that specifies the anomalous y-velocity of species (is); the
*      transport equation for species (is) contains a term
*      vlay(,is)*na(,is).
*      If cfvla(7,is) is not zero, the vla0 array is flux-scaled with
*      a scaling factor of cfvla(7,is)*(hyl(,)/hyl(ixref,))**2
*      where hyl is the radial width of the cell and ixref is a reference
*      poloidal position (usually the outer midplane or 'jxa').
*      cfvla(4:6,) are not used in the default version of the code.
*      cfvsa - (0:7,0:ns-1) real*8 array.
*      For 0.le.is.lt.ns, cfvsa(is) are the coefficients in the
*      local expression
*      dfvsa(,is) = cfvsa(0,is)+cfvsa(1,is)/(ne(,)/1.0e20)+
*      cfvsa(2,is)*df0(,is)+
*      cfvsa(3,is)*te(,)/ev/abs(bb(,3))/16
*      that specifies the diffusivity corresponding to the anomalous
*      transverse viscosity, so that this viscosity is
*      vsa(,is) = mp*am(is)*na(,is)*dfvsa
*      If cfvsa(7,is) is not zero, the vsa0 array is flux-scaled with
*      a scaling factor of cfvsa(7,is)*(hyl(,)/hyl(ixref,))**2
*      where hyl is the radial width of the cell and ixref is a reference
*      poloidal position (usually the outer midplane or 'jxa').
*      cfvsa(4:6,) are not used in the default version of the code.
*      cfhci - (0:7,0:ns-1) real*8 array.
*      For 0.le.is.lt.ns, cfhci(is) are the coefficients in the
*      local expression
*      dfhca(,is) = cfhci(0,is)+cfhci(1,is)/(ne(,)/1.0e20)+
*      cfhci(2,is)*df0(,is)+
*      cfhci(3,is)*te(,)/ev/abs(bb(,3))/16
*      that specifies the anomalous atom heat conductivity associated
*      with species (is), so that the atom heat conduction coefficient
*      contains a term dfhca(,is)*na(,is).
*      If cfhci(7,is) is not zero, the hci arrays are flux-scaled with
*      a scaling factor of cfhci(7,is)*(hyl(,)/hyl(ixref,))**2
*      where hyl is the radial width of the cell and ixref is a reference
*      poloidal position (usually the outer midplane or 'jxa').
*      cfhci(4:6,) are not used in the default version of the code.
*      cfhce - (0:7) real*8 array.
*      cfhce() are the coefficients in the local expression
*      dfhce(, ) = cfhce(0)+cfhce(1)/(ne(,)/1.0e20)+
*      cfhce(3)*te(,)/ev/abs(bb(,3))/16
*      that specifies the anomalous electron heat conductivity, so that
*      the electron heat conduction coefficient is dfhce(,)*ne(,).
*      If cfhce(7) is not zero, the hce0 array is flux-scaled with
*      a scaling factor of cfhce(7)*(hyl(,)/hyl(ixref,))**2
*      where hyl is the radial width of the cell and ixref is a reference
*      poloidal position (usually the outer midplane or 'jxa').
*      cfhce(2) and (4:6) are not used in the default version of the code.
*      cfsig - (0:7) real*8 array.
*      cfsig() are the coefficients in the local expression
*      dfsig(, ) = cfsig(0)+cfsig(1)/(ne(,)/1.0e20)+
*      cfsig(3)*te(,)/ev/abs(bb(,3))/16+
*      cfsig(4)*exp(-cfsig(5)*(y/L)**2)
*      that specifies the diffusivity corresponding to the anomalous
*      electrical conductivity, which is
*      sig(, ) = qe*ne(,)*dfsig(,).
*      L is the local radial width of the plasma and
*      y is the radial distance from the inner (South) boundary.
*      If cfsig(7) is not zero, the sig0 array is flux-scaled with
*      a scaling factor of cfsig(7)*(hyl(,)/hyl(ixref,))**2
*      where hyl is the radial width of the cell and ixref is a reference
*      poloidal position (usually the outer midplane or 'jxa').
*      cfsig(2) and (6) are not used in the default version of the code.
*      cfalf - (0:7) real*8 array.
*      cfalf() are the coefficients in the local expression
*      dfalf(, ) = cfalf(0)+cfalf(1)/(ne(,)/1.0e20)+
*      cfalf(3)*te(,)/ev/abs(bb(,3))/16
*      that specifies the diffusivity corresponding to the anomalous

```

```

* thermo-electric coefficient, which is
*   alf(,) = ne(,)*sqrt(qe/te(,))*dfalf(,).
* If cfalf(7) is not zero, the alf0 array is flux-scaled with
* a scaling factor of cfalf(7)*(hyl(,)/hyl(ixref,))*2
* where hyl is the radial width of the cell and ixref is a reference
* poloidal position (usually the outer midplane or 'jxa').
* cfalf(2) and (4:6) are not used in the default version of the code.
*   cflim - (0:7) real*8 array.
* cflim contains coefficients for flux limiting. cflim(0)=cflme
* specifies the flux limit factor for poloidal electron heat
* transport. cflim(1)=cflmi specifies the flux limit factor for
* poloidal all atom heat transport. cflim(2)=cflmv specifies the
* flux limit factor for poloidal atom momentum transport.
* cflim(3)=cflal specifies the flux limit factor for the
* thermo-electric coefficient.
* cflim(4)=cflab specifies the flux limit factor for the
* friction force.
* Zero settings turn off these flux limits.
* cflim(5:7) are not used in the default version of the code.
*
* ..transport coefficients, differential
*
* dna - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let dnax(,,)=dna(,,0) and dnay(,,)=dna(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), dnax(ix,iy,is) and
* dnay(ix,iy,is) specify the x-component and the y-component of
* the diffusivity of species (is) on the (ix,iy) cell. The
* corresponding term in the particle flux density is
* -dna*grad.na .
*
* dpa - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let dpax(,,)=dpa(,,0) and dpay(,,)=dpa(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), dpax(ix,iy,is) and
* dpay(ix,iy,is) specify the x-component and the y-component of
* the diffusion coefficient (w.r.t. grad.pa) of species (is) on
* the (ix,iy) cell. The corresponding term in the particle flux
* density is -dpa*grad.pa .
*
* vla - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let vlax(,,)=vla(,,0) and vlay(,,)=vla(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vlax(ix,iy,is) specifies
* the x-component of the anomalous velocity of species (is) on the
* (ix,iy) cell.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vlay(ix,iy,is) specifies
* the y-component of the anomalous velocity of species (is) on the
* (ix,iy) cell.
*
* vsa - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let vsax(,,)=vsa(,,0) and vsay(,,)=vsa(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vsax(ix,iy,is) specifies
* the x-component and vsay(ix,iy,is) the y-component of the
* viscosity of species (is) on the (ix,iy) cell.
*
* hce - (-1:nx,-1:ny,0:1) real*8 array.
* (Let hcex(,)=hce(,,0) and hcey(,)=hce(,,1).)
* For (ix,iy) in (-1:nx,-1:ny), hcex(ix,iy) specifies the
* x-component and hcey(ix,iy) the y-component of the electron
* thermal conductivity on the (ix,iy) cell.
*
* hci - (-1:nx,-1:ny,0:1) real*8 array.
* (Let hcix(,)=hci(,,0) and hciy(,)=hci(,,1).)
* For (ix,iy) in (-1:nx,-1:ny), hcix(ix,iy) specifies the
* x-component and hciy(ix,iy) the y-component of the all atom
* thermal conductivity on the (ix,iy) cell.
*
* sig - (-1:nx,-1:ny,0:1) real*8 array.
* (Let sigx(,)=sig(,,0) and sigy(,)=sig(,,1).)

```



```

* For (ix,iy) in (-1:nx,-1:ny), sigx(ix,iy) specifies the
* x-component and sigy(ix,iy) the y-component of the electrical
* conductivity on the (ix,iy) cell.
*
* alf - (-1:nx,-1:ny,0:1) real*8 array.
* (Let alfx(,)=alf(,0) and alfy(,)=alf(,1).)
* For (ix,iy) in (-1:nx,-1:ny), alfx(ix,iy) specifies the
* x-component and alfy(ix,iy) the y-component of the thermo-
* electric coefficient on the (ix,iy) cell.
*
* dna0 - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), dna0(ix,iy,is) specifies
* the anomalous diffusivity with respect to grad(na) of species
* (is) on the (ix,iy) cell.
*
* dpa0 - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), dpa0(ix,iy,is) specifies
* the anomalous diffusivity with respect to grad(pa) of species
* (is) on the (ix,iy) cell.
*
* vla0 - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let vla0x(,,)=vla0(,,0,) and vla0y(,,)=vla0(,,1).)
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vla0x(ix,iy,is) specifies
* the x-component and vla0y(ix,iy,is) the y-component of the
* anomalous velocity of species (is) on the (ix,iy) cell.
*
* vsa0 - (-1:nx,-1:ny,0:ns-1) real*8 array.
* For (ix,iy,is) in (-1:nx,-1:ny,0:ns-1), vsa0(ix,iy,is) specifies
* the anomalous viscosity of species (is) on the (ix,iy) cell.
*
* hce0 - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), hce0(ix,iy) specifies the anomalous
* electron thermal conductivity on the (ix,iy) cell.
*
* hci0 - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), hci0(ix,iy) specifies the anomalous
* atom thermal conductivity on the (ix,iy) cell.
*
* sig0 - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), sig0(ix,iy) specifies the anomalous
* electrical conductivity on the (ix,iy) cell.
*
* alf0 - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), alf0(ix,iy) specifies the anomalous
* thermo-electric coefficient on the (ix,iy) cell.
*
* ..transport coefficients, discrete
*
* ceqp - (-1:nx,-1:ny) real*8 array.
* For (ix,iy) in (-1:nx,-1:ny), ceqp(ix,iy) specifies the energy
* equipartition multiplier integrated over the (ix,iy) cell.
* (The cell-integrated energy equipartition coefficient is
* ceqp*ne*ni/te*1.5.)
*
* cdna - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let cdnax=cdna(,0,) and cdnay=cdna(,1).)
* For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), cdnax(ix,iy,is) specifies
* the particle diffusivity w.r.t. grad(na) of species (is) between
* the (ix,iy) cell and its left neighbor. This coefficient enters into
* the expression for the particle flux in the form
* -cdnax(ix,iy,is)*(na(ix,iy,is)-na(ix-1,iy,is))
* For ix.eq.-1, cdnax(ix,-1:ny,0:ns-1) will hold 0.
* For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), cdnay(ix,iy,is) specifies
* the particle diffusivity w.r.t. grad(na) of species (is) between
* the (ix,iy) cell and its bottom neighbor. This coefficient enters into
* the expression for the particle flux in the form
* -cdnay(ix,iy,is)*(na(ix,iy,is)-na(bottom(ix,iy),is))

```

```

* For iy.eq.-1, cday(-1:nx,iy,0:ns-1) will hold 0.
*
* cdpax - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let cdpax=cdpax(,,0) and cdpay=cdpax(,,1).)
* For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), cdpax(ix,iy,is) specifies
* the particle diffusivity w.r.t. grad(pa) of species (is) between
* the (ix,iy) cell and its left neighbor. This coefficient enters into
* the expression for the particle flux in the form
* -cdpax(ix,iy,is)*(pa(ix,iy,is)-pa(ix-1,iy,is))
* For ix.eq.-1, cdpax(ix,-1:ny,0:ns-1) will hold 0.
* For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), cdpay(ix,iy,is) specifies
* the particle diffusivity w.r.t. grad(pa) of species (is) between
* the (ix,iy) cell and its bottom neighbor. This coefficient enters into
* the expression for the particle flux in the form
* -cdpay(ix,iy,is)*(pa(ix,iy,is)-pa(bottom(ix,iy),is))
* For iy.eq.-1, cdpay(-1:nx,iy,0:ns-1) will hold 0.
*
* cvlax - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let cvlax=cvla(,,0) and cvlay=cvla(,,1).)
* For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), cvlax(ix,iy,is) specifies
* the surface-integrated anomalous velocity of species (is) between
* the (ix,iy) cell and its left neighbor. This coefficient enters into
* the expression for the particle flux in the form
* cvlax(ix,iy,is)*(na)
* in which (na) involves some interpolation to cell faces.
* For ix.eq.-1, cvlax(ix,-1:ny,0:ns-1) will hold 0.
* For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), cvlay(ix,iy,is) specifies
* the surface-integrated anomalous velocity of species (is) between
* the (ix,iy) cell and its bottom neighbor. This coefficient enters into
* the expression for the particle flux in the form
* cvlay(ix,iy,is)*(na)
* in which (na) involves some interpolation to cell faces.
* For iy.eq.-1, cvlay(-1:nx,iy,0:ns-1) will hold 0.
*
* cvsax - (-1:nx,-1:ny,0:1,0:ns-1) real*8 array.
* (Let cvsax=cvsax(,,0) and cvsay=cvsax(,,1).)
* For (ix,iy,is) in (0:nx,-1:ny,0:ns-1), cvsax(ix,iy,is) specifies
* the viscous conductance of species (is) between the (ix,iy) cell
* and its left neighbor.
* For ix.eq.-1, cvsax(ix,-1:ny,0:ns-1) will hold 0.
* For (ix,iy,is) in (-1:nx,0:ny,0:ns-1), cvsay(ix,iy,is) specifies
* the viscous conductance of species (is) between the (ix,iy) cell
* and its bottom neighbor.
* For iy.eq.-1, cvsay(-1:nx,iy,0:ns-1) will hold 0.
*
* chcx - (-1:nx,-1:ny,0:1) real*8 array.
* (Let chcx=chcx(,,0) and chcy=chcx(,,1).)
* For (ix,iy) in (0:nx,-1:ny), chcx(ix,iy) specifies the electron
* thermal conductance between the (ix,iy) cell and its left neighbor.
* This coefficient enters into the expression for the electron
* heat flux in the form
* -chcx(ix,iy)*(te(ix,iy)-te(left(ix,iy))) .
* For ix.eq.-1, chcx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), chcy(ix,iy) specifies the electron
* thermal conductance between the (ix,iy) cell and its bottom neighbor.
* This coefficient enters into the expression for the electron
* heat flux in the form
* -chcy(ix,iy)*(te(ix,iy)-te(bottom(ix,iy))) .
* For iy.eq.-1, chcy(-1:nx,iy) will hold 0.
*
* chve - (-1:nx,-1:ny,0:1) real*8 array.
* (Let chvex=chve(,,0) and chvey=chve(,,1).)
* For (ix,iy) in (0:nx,-1:ny), chvex(ix,iy) specifies the electron
* thermal strange velocity between the (ix,iy) cell and its left
* neighbor. This coefficient enters into the expression for the
* electron heat flux in the form chvex*ne*te , interpolated to
* cell faces in a suitable manner.

```

```

* For ix.eq.-1, chvex(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), chvey(ix,iy) specifies the electron
* thermal strange velocity between the (ix,iy) cell and its bottom
* neighbor. This coefficient enters into the expression for the
* electron heat flux in the form chvey*ne*te , interpolated to
* cell faces in a suitable manner.
* For iy.eq.-1, chvey(-1:nx,iy) will hold 0.
*
* chci - (-1:nx,-1:ny,0:1) real*8 array.
* (Let chci=chci(,,0) and chciy=chci(,,1).)
* For (ix,iy) in (0:nx,-1:ny), chciy(ix,iy) specifies the all atom
* thermal conductance between the (ix,iy) cell and its left neighbor.
* This coefficient enters into the expression for the atom heat
* flux in the form
* -chciy(ix,iy)*(ti(ix,iy)-ti(left(ix,iy))) .
* For ix.eq.-1, chciy(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), chciy(ix,iy) specifies the all atom
* thermal conductance between the (ix,iy) cell and its bottom neighbor.
* This coefficient enters into the expression for the atom heat
* flux in the form
* -chciy(ix,iy)*(ti(ix,iy)-ti(bottom(ix,iy))) .
* For iy.eq.-1, chciy(-1:nx,iy) will hold 0.
*
* chvi - (-1:nx,-1:ny,0:1) real*8 array.
* (Let chvix=chvi(,,0) and chviy=chvi(,,1).)
* For (ix,iy) in (0:nx,-1:ny), chvix(ix,iy) specifies the all atom
* thermal strange velocity between the (ix,iy) cell and its left
* neighbor. This coefficient enters into the expression for the
* atom heat flux in the form chvix*ni*ti , interpolated to cell
* faces in a suitable manner.
* For ix.eq.-1, chvix(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), chviy(ix,iy) specifies the all atom
* thermal strange velocity between the (ix,iy) cell and its bottom
* neighbor. This coefficient enters into the expression for the
* atom heat flux in the form chviy*ni*ti , interpolated to
* cell faces in a suitable manner.
* For iy.eq.-1, chviy(-1:nx,iy) will hold 0.
*
* csig - (-1:nx,-1:ny,0:1) real*8 array.
* (Let csigx=csig(,,0) and csigy=csig(,,1).)
* For (ix,iy) in (0:nx,-1:ny), csigx(ix,iy) specifies the electrical
* conductance between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, csigx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), csigy(ix,iy) specifies the electrical
* conductance between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, csigx(-1:nx,iy) will hold 0.
*
* calfx - (-1:nx,-1:ny,0:1) real*8 array.
* (Let calfx=calfx(,,0) and calfy=calfx(,,1).)
* For (ix,iy) in (0:nx,-1:ny), calfx(ix,iy) specifies the thermo-
* electric coefficient between the (ix,iy) cell and its left neighbor.
* For ix.eq.-1, calfx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), calfy(ix,iy) specifies the thermo-
* electric coefficient between the (ix,iy) cell and its bottom neighbor.
* For iy.eq.-1, calfy(-1:nx,iy) will hold 0.
*
* csiginx - (-1:nx,-1:ny,0:1,0:ns-1,0:ns-1) real*8 array. !xpb
* (Let csiginx=csigin(,,0,,) and csiginy=csigin(,,1,,).)
* For (ix,iy,is,is0) in (0:nx,-1:ny,0:ns-1,0:ns-1),
* csiginx(ix,iy,is,is0) specifies the ion-neutral
* conductivity between the (ix,iy) cell and its left neighbor, where
* the ion species is species is and the neutral species is is0.
* For ix.eq.-1, csiginx(ix,-1:ny) will hold 0.
* For (ix,iy) in (-1:nx,0:ny), csiginy(ix,iy) specifies the ion-neutral
* conductivity between the (ix,iy) cell and its bottom neighbor, where
* the ion species is species is and the neutral species is is0.
* For iy.eq.-1, csiginx(-1:nx,iy) will hold 0.

```

```

*      When is is a neutral species or is0 is an ionised species,
*      csigin(,,is,is0) will be zero.
*
*      csig_cl - (-1:nx,-1:ny,0:1) real*8 array. !xpb
*      (Let csig_clx=csig_cl(,,0) and csig_cly=csig_cl(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), csig_clx(ix,iy) specifies the classical
*      conductivity between the (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, csig_clx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), csig_cly(ix,iy) specifies the classical
*      conductivity between the (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, csig_cly(-1:nx,iy) will hold 0.
*
*      calf_cl - (-1:nx,-1:ny,0:1) real*8 array.
*      (Let calf_clx=calf_cl(,,0) and calf_cly=calf_cl(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), calf_clx(ix,iy) specifies the
*      classical part of the thermo-electric coefficient
*      between the (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, calf_clx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), calf_cly(ix,iy) specifies the
*      classical part of the thermo-electric coefficient
*      between the (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, calf_cly(-1:nx,iy) will hold 0.
*
*      csig_an - (-1:nx,-1:ny,0:1) real*8 array. !xpb
*      (Let csig_anx=csig_an(,,0) and csig_any=csig_an(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), csig_anx(ix,iy) specifies the anomalous
*      conductivity between the (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, csig_anx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), csig_any(ix,iy) specifies the anomalous
*      conductivity between the (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, csig_anx(-1:nx,iy) will hold 0.
*
*      calf_an - (-1:nx,-1:ny,0:1) real*8 array.
*      (Let calf_anx=calf_an(,,0) and calf_any=calf_an(,,1).)
*      For (ix,iy) in (0:nx,-1:ny), calf_anx(ix,iy) specifies the
*      anomalous part of the thermo-electric coefficient
*      between the (ix,iy) cell and its left neighbor.
*      For ix.eq.-1, calf_anx(ix,-1:ny) will hold 0.
*      For (ix,iy) in (-1:nx,0:ny), calf_any(ix,iy) specifies the
*      anomalous part of the thermo-electric coefficient
*      between the (ix,iy) cell and its bottom neighbor.
*      For iy.eq.-1, calf_any(-1:nx,iy) will hold 0.
*
*      ..parameters for core numerical iteration
*
*      ntim - integer.
*      ntim specifies the number of timesteps to be performed.
*
*      itim - integer.
*      itim specifies the timestep counter. It is initialised to 0 and
*      incremented by 1 following each successful call to b2mndt.
*
*      dtim - real.
*      dtim specifies the timestep for the present call to b2mndt.
*      It will hold that 0.lt.dtim.
*
*      itcnt - integer.
*      itcnt counts the number of inner iterations. It is initialised to
*      0 and incremented by 1 following each call to b2news.
*
*      nstg - (0:2) integer array.
*      nstg(0:2) specifies the maximum number of iterations for each of
*      the three main loops in the subroutine b2mndt.
*
*      rxf - real.
*      rxf specifies the main under-relaxation parameter. Computed
*      corrections to the unknowns are scaled by a factor rxf before

```

```

*      being applied. It will hold that 0.le.rxf.le.1.
*
*      rxg - real.
*      rxg specifies a special under-relaxation parameter. A term of the
*      form "abs(residual)/rxg" is added to the diagonal of the matrix
*      of the correction equation, with the effect of limiting the
*      computed correction. rxg is dimensionless and of order unity;
*      smaller values of rxg imply stronger damping.
*
*      ..variables for input-output
*
*      nrid - integer.
*      nrid specifies a unit number for formatted input.
*
*      nget - integer.
*      nget specifies a unit number for unformatted input.
*
*      nprt - integer.
*      nprt specifies a unit number for formatted output.
*
*      nput - integer.
*      nput specifies a unit number for unformatted output.
*
*      idout - character*(*).
*      idout specifies the desired selection of output segments for
*      printed and graphical output. Associated with each segment is a
*      four-character word, which is identified on the printed output.
*      (The same word may be associated with several segments.)
*      A particular segment will be produced only if its associated word
*      has a match, as defined by the routine strmas, in idout.
*
*      /b2cmwg/ - common block.
*      /b2cmwg/ contains some parameters to control graphical output.
*      nmbpmx - integer constant.
*      nmbpmx is used in dimension declarations instead of nmbpcc or
*      nmbpdm.
*      nmbpcc - integer.
*      nmbpcc specifies the number of cartesian domains on which any
*      quantity may be plotted.
*      nmbpdm - integer.
*      nmbpdm specifies the number of physical domains on which any
*      quantity may be plotted.
*      cwaxr, cwayr - (0:nmbpmx-1) real*8 array.
*      For each iplot in (0:nmbpcc-1), cwaxr(iplot) and cwayr(iplot)
*      specify coordinates for a cartesian domain plot. The coordinates
*      are obtained by integrating the grid metric along two reference
*      lines. The reference line for the x-coordinate is at position
*      nint(cwaxr*nx), and the reference line for the y-coordinate is
*      at nint(cwayr*ny). It will hold that 0.le.cwaxr.le.1 and
*      0.le.cwayr.le.1.
*      cwxmin, cwxmax, cwymmin, cwymax - (0:nmbpmx-1) real*8 array.
*      For each iplot in (0:nmbpdm-1), the quadruple of cwxmin(iplot)
*      through cymax(iplot) specifies a domain in the physical plane
*      for contour plots.
*
*      ..workspace
*
*      /stackw/ - common block.
*      /stackw/ contains two stacks, one for real*8 and one for integer
*      workspace. They are used to provide local dynamic arrays within
*      the rules of Fortran 77.
*      nstckw - integer constant.
*      nstckw specifies the size of the real*8 workspace array w. If
*      nstckw is too small, the program will abort via a call to
*      xertst/xerrab.
*      nstcwi - integer constant.
*      nstcwi specifies the size of the integer workspace array wi. If

```

```

*      nstcwi is too small, the program will abort via a call to
*      xertst/xerrab.
*      w - (0:nstckw-1) real*8 array.
*      w is used as a stack for local dynamic real*8 arrays. Base
*      addresses for arrays in this stack are to be maintained by
*      appropriate calls to the subroutines subini, subend, and lwget.
*      wi - (0:nstcwi-1) integer array.
*      wi is used as a stack for local dynamic integer arrays. Base
*      addresses for arrays in this stack are to be maintained by
*      appropriate calls to the subroutines subini, subend, and lwiget.
*
*      ..some other variables
*
*      lblgm - character*120.
*      lblgm identifies the instance of the program employed to prepare
*      the geometry.
*
*      lblcp - character*120.
*      lblcp identifies the instance of the program employed to prepare
*      the table of physics parameters.
*
*      lblps - character*120.
*      lblps identifies the instance of the program employed to prepare
*      the initial state.
*
*      lblrc - character*120.
*      lblrc identifies the instance of the program employed to prepare
*      the table of atomic rate coefficients.
*
*      lblsr - character*120.
*      lblsr identifies the instance of the program employed to prepare
*      the neutral gas source data.
*
*      lblmn - character*120.
*      lblmn identifies the instance of the program employed for the main
*      calculation.
*
*      ncall - integer.
*      ncall counts the number of calls made to the present routine.
*
*      2. Notes
*
*      Variables local to a subroutine are not expected to be preserved
*      between calls unless they are declared in a 'save' statement.
*      Variables in common are not expected to be preserved between
*      calls to the outermost routine in which the common block appears.
*      (See ANSI X3.9-1978 Fortran 77, Section 17.3.6d.) The code may
*      be compiled and loaded under an option that initialises all
*      quantities not mentioned in a data statement to an invalid
*      numerical value.
*
*-----
*.end b2cdcv

      end subroutine b2cdcv

```

A.4 b2cdci.F

```
*-----
* Generated by b2cdci.py from b2input.xml. DO NOT EDIT THIS FILE!
*.specification

      subroutine b2cdci ()
      implicit none

*-----
*.documentation
*
* 1. purpose
*
*   B2CDCI provides information about internal parameters of the B2 code.
*   No computation is performed in this routine.
*
*-----
*.text
*
* 0. Overview and naming conventions.
*
*   Here we list the switches in alphabetical order. By convention, a
*   switch name starts with the name of the routine in which it is used.
*   Unless noted otherwise, the desired switches are to be declared at the
*   end of the b2mn.dat input file.
*   Integer switches are given integer default values, etc...
*   Some switches are used by several routines and may thus also appear in
*   routines other than the one of which they carry the name.
*   The basic philosophy behind the assignment of the default values is
*   that the default values do not change the expected standard behaviour
*   of the solution and solution method. In most cases, the code checks
*   that a switch modified by the user is within some expected range, and
*   stops otherwise. Setting a switch to its default value will not stop
*   the code.
*
*   The switches are divided into 6 broad categories : Geometry, Physics,
*   Run, Output, Numerics and Atomic Physics. The following section will
*   describe each category separately. Within categories, an effort is
*   made to list the most relevant and/or common switches first, and
*   rarely used or esoteric ones last.
*
*   Geometry switches refer to the plasma geometry. These usually will
*   need to be set up only once for a family of runs. The user should be
*   familiar with most of these.
*
*   Run switches refer to parameters for the code run. These are likely to
*   change for every run and the user should know how to use them. There
*   is some overlap between Physics and Run switches, and such cases will
*   be explained below individually.
*
*   Physics switches refer to the physics terms in the code. These are
*   likely to be different within a family of runs. The user should be
*   familiar with the ones relevant to the case being studied.
*
*   Output switches refer to the level and type of output. A few switches
*   are standard, but most need only be activated for specific debugging
*   purposes.
*
*   Numerics switches refer to the numerical treatment of the equations.
*   Most of these switches should only be modified with the most extreme
*   caution. The user need be concerned with only very few of these.
*
*   Atomic Physics includes switches used in the atomic physics pre-
*   processing module B2AR.
```

Switch name:	Default value:	Category:
'ank_tracing'	'0'	Output
'b2agdr_nxisol'	'-2'	Geometry
'b2agdr_nxiso2'	'-2'	Geometry
'b2agdr_nyisol'	'-2'	Geometry
'b2agdr_nyiso2'	'-2'	Geometry
'b2agdr_redef_gmtry'	'1'	Geometry
'b2agdr_redef_qc'	'0'	Geometry
'b2agfs_Bt_adjust'	'0'	Geometry
'b2agfs_Bt_rescale'	'1.0'	Geometry
'b2agfs_Bt_reversal'	'0'	Geometry
'b2agfs_bottomcut'	See description (integer)	Geometry
'b2agfs_bottomcut2'	See description (integer)	Geometry
'b2agfs_geom_match_dist'	'1.0e-6'	Geometry
'b2agfs_geometry'	'upgrade.geometry'	Geometry
'b2agfs_leftcut'	See description (integer)	Geometry
'b2agfs_leftcut2'	See description (integer)	Geometry
'b2agfs_min_pitch'	'1.0'	Geometry
'b2agfs_nncut'	See description (integer)	Geometry
'b2agfs_periodic_bc'	'0'	Geometry
'b2agfs_pit_rescale'	'1.0'	Geometry
'b2agfs_rightcut'	See description (integer)	Geometry
'b2agfs_rightcut2'	See description (integer)	Geometry
'b2agfs_topcut'	See description (integer)	Geometry
'b2agfs_topcut2'	See description (integer)	Geometry
'b2agfs_xoffset'	'0.0'	Geometry
'b2agfs_xrescale'	'1.0'	Geometry
'b2agfs_yoffset'	'0.0'	Geometry
'b2agfs_yrescale'	'1.0'	Geometry
'b2agmt_ld_width'	'1.0'	Geometry
'b2agmt_qc_style'	'1'	Geometry
'b2agmx_pbs_from_basis_mesh'	'1'	Geometry
'b2agsi_isymm'	'1'	Geometry
'b2aidr_read_b2fstate'	'0'	Run
'b2ardr_fix_cx'	'1'	Atomic Physics
'b2ardr_fix_recomb'	'0'	Atomic Physics
'b2ardr_no_smoothering'	'0'	Atomic Physics
'b2ardr_no_weisheit'	'0'	Atomic Physics
'b2ardr_rtnn'	'32'	Atomic Physics
'b2ardr_rtnt'	'40'	Atomic Physics
'b2ardr_zmax_recomb'	'1'	Atomic Physics
'b2mndr_astra'	'0'	Run
'b2mndr_atomic_physics_rescale'	'0'	Physics
'b2mndr_av_read'	'0'	Output
'b2mndr_b2time'	'1'	Output
'b2mndr_cdf_default'	'0'	Output
'b2mndr_cdfmovietim'	'0.0'	Output
'b2mndr_coronal_model'	'0'	Physics
'b2mndr_cpu'	'0.0'	Run
'b2mndr_database'	\$(DEVICE)	Run
'b2mndr_delta_max'	'0.0'	Run
'b2mndr_delta_min'	'0.0'	Run
'b2mndr_density_rescale'	'1.0'	Run
'b2mndr_device'	\$(DEVICE)	Run
'b2mndr_dpc_mod_rates_ne_hot_frac'	'0.0'	Atomic Physics
'b2mndr_dpc_mod_rates_te_hot'	'0.0'	Atomic Physics
'b2mndr_dt_change_dec'	'1.0'	Run
'b2mndr_dt_change_inc'	'1.0'	Run
'b2mndr_dt_max'	'1.0e+01'	Run
'b2mndr_dt_min'	'1.0e-30'	Run
'b2mndr_dtim'	'1.0'	Run
'b2mndr_eirene'	'0'	Run
'b2mndr_elapsed'	'0.0'	Run
'b2mndr_etim'	'0.0'	Run

*	'b2mndr_hz'	'1.0'	Physics
*	'b2mndr_idout0'	'pgnl;pgmm;pzmm'	Output
*	'b2mndr_idout1'	'pzmm'	Output
*	'b2mndr_inverse_ua'	'0'	Output
*	'b2mndr_isfb'	ismain	Numerics
*	'b2mndr_ismain'	'1'	Run
*	'b2mndr_ixfb'	See description	Numerics
*	'b2mndr_iyfb'	ny/2	Numerics
*	'b2mndr_min_aresco'	'0.0'	Numerics
*	'b2mndr_min_areshe'	'0.0'	Numerics
*	'b2mndr_min_areshi'	'0.0'	Numerics
*	'b2mndr_mvinc'	'1'	Output
*	'b2mndr_mvnum'	'0'	Output
*	'b2mndr_na_eps'	'1.0e19'	Output
*	'b2mndr_na_min'	'1.0e4'	Numerics
*	'b2mndr_na_new'	'1.0e14'	Numerics
*	'b2mndr_ne_wanted'	'0.0'	Numerics
*	'b2mndr_ne_wanted_time'	'0.0'	Numerics
*	'b2mndr_ntim'	'1'	Run
*	'b2mndr_ntim_save'	'0'	Output
*	'b2mndr_old_style'	'0'	Output
*	'b2mndr_plasinc'	'1'	Output
*	'b2mndr_plasmatim'	'0.0'	Output
*	'b2mndr_plasnum'	'0'	Output
*	'b2mndr_po_eps'	'1.0e+1'	Output
*	'b2mndr_rescale_neutrals'	'1.0'	Run
*	'b2mndr_rescale_neutrals_sources'	'1.0'	Run
*	'b2mndr_run_number'	'1000'	Run
*	'b2mndr_savecpu'	'3600.0'	Run
*	'b2mndr_shot_number'	'0'	Run
*	'b2mndr_stim'	'0.0'	Run
*	'b2mndr_tally'	'1'	Output
*	'b2mndr_te_eps'	'1.0e+1'	Output
*	'b2mndr_ti_eps'	'1.0e+1'	Output
*	'b2mndr_trantim'	'0.0'	Output
*	'b2mndr_ua_eps'	'1.0e+4'	Output
*	'b2mndr_user'	\$(USER)	Run
*	'b2mndr_zn_rescale'	'0'	Run
*	'b2mndt_av'	'0'	Output
*	'b2mndt_av_batch_all'	'0'	Output
*	'b2mndt_av_continue'	'1'	Output
*	'b2mndt_av_ntim_batch'	'500'	Output
*	'b2mndt_av_ntim_run'	'1000'	Output
*	'b2mndt_density_control'	'0'	Run
*	'b2mndt_moitlv'	'-1'	Output
*	'b2mndt_moqtlv'	'3'	Output
*	'b2mndt_nstg0'	'1'	Run
*	'b2mndt_nstg1'	'1'	Run
*	'b2mndt_nstg2'	'1'	Run
*	'b2mndt_nstg_aresco'	'0.0'	Numerics
*	'b2mndt_nstg_areshe'	'0.0'	Numerics
*	'b2mndt_nstg_areshi'	'0.0'	Numerics
*	'b2mndt_ntim_step_out'	'1'	Numerics
*	'b2mndt_rxf'	'0.5'	Numerics
*	'b2mndt_style'	'1'	Numerics
*	'b2mndt_use_b2srst'	'1'	Numerics
*	'b2mwgt_style'	'1'	Output
*	'b2mwti_2dwrite'	'0'	Output
*	'b2mwti_ismain0'	'0'	Output
*	'b2mwti_jsep'	See description (integer)	Geometry
*	'b2mwti_jxa'	See description (integer)	Geometry
*	'b2mwti_jxi'	See description (integer)	Geometry
*	'b2mwti_target_offset'	'1'	Output
*	'b2news_BoRiS'	'0.0'	Physics
*	'b2news_ExB'	'0.0'	Physics
*	'b2news_area_fix'	'3'	Numerics
*	'b2news_do_2nd_b2npco_call'	'0'	Numerics

*	'b2news_facExB_dec'	'0.0'	Run
*	'b2news_facExB_inc'	'1.0'	Run
*	'b2news_facExB_start'	'0.0'	Run
*	'b2news_facExB_target'	'0.0'	Run
*	'b2news_fac_ExB_tanh_a'	'0.0'	Numerics
*	'b2news_fac_ExB_tanh_b'	'0.0'	Numerics
*	'b2news_fac_ref'	See description (integer)	Numerics
*	'b2news_fac_vis_tanh_a'	'0.0'	Numerics
*	'b2news_fac_vis_tanh_b'	'0.0'	Numerics
*	'b2news_facdrift_dec'	'0.0'	Run
*	'b2news_facdrift_inc'	'1.0'	Run
*	'b2news_facdrift_start'	'0.0'	Run
*	'b2news_facdrift_tanh_a'	'0.0'	Numerics
*	'b2news_facdrift_tanh_b'	'0.0'	Numerics
*	'b2news_facdrift_target'	'0.0'	Run
*	'b2news_facvis_dec'	'0.0'	Run
*	'b2news_facvis_inc'	'1.0'	Run
*	'b2news_facvis_start'	'0.0'	Run
*	'b2news_facvis_target'	'0.0'	Run
*	'b2news_guard_flows'	'2'	Numerics
*	'b2news_iy_nocoreExB'	'-2'	Numerics
*	'b2news_l_enebound'	'0'	Numerics
*	'b2news_l_fbound'	'0'	Numerics
*	'b2news_l_tmbound'	'0'	Numerics
*	'b2news_ncallout'	'-1'	Output
*	'b2news_no_b2sral_call'	'1'	Numerics
*	'b2news_no_solve'	'0'	Run
*	'b2news_nsmax'	ns	Numerics
*	'b2news_nsmin'	'0'	Numerics
*	'b2news_poteq'	'1'	Numerics
*	'b2news_potit'	'50'	Numerics
*	'b2news_potitmin'	'0'	Numerics
*	'b2news_potok'	'1.0e-2'	Numerics
*	'b2news_ramp_slow'	'0'	Numerics
*	'b2news_re_eval_prtls_fluxes'	'0'	Numerics
*	'b2news_recalculate_contributions'	'1'	Numerics
*	'b2news_xfm0'	'1.0'	Numerics
*	'b2news_xfm1'	'1.0'	Numerics
*	'b2news_xfm2'	'1.0'	Numerics
*	'b2news_xfm3'	'1.0'	Numerics
*	'b2npco_pcm0'	'1.0'	Numerics
*	'b2npco_pcm1'	'1.0'	Numerics
*	'b2npco_rxcg'	'1.0'	Numerics
*	'b2nph9_style'	'1'	Numerics
*	'b2npht_pcm0'	'1.0'	Numerics
*	'b2npht_pcm1'	'1.0'	Numerics
*	'b2npht_rxcg'	'1.0'	Numerics
*	'b2npht_style'	'1'	Numerics
*	'b2npmo_b2sifr_'	'1'	Physics
*	'b2npmo_ion_vlct_restrict'	'0'	Numerics
*	'b2npmo_rxcg'	'1.0e6'	Numerics
*	'b2npmo_vlct_diagno'	'1'	Output
*	'b2nxdv_style'	'1'	Numerics
*	'b2nxfc_style'	'1'	Numerics
*	'b2nxfx_style'	'1'	Numerics
*	'b2sdia_facgt'	'0.0'	Physics
*	'b2sian_phm0'	'1.0'	Physics
*	'b2siav_addvis'	'1.0'	Physics
*	'b2siav_addvis1'	'1.0'	Physics
*	'b2siav_style_qip'	'0'	Physics
*	'b2sicf_phm0'	'1.0'	Physics
*	'b2sicf_phm1'	'1.0'	Physics
*	'b2sifr_limthee'	'0.3'	Physics
*	'b2sifr_limthii'	'0.3'	Physics
*	'b2sifr_phm0'	'1.0'	Physics
*	'b2sifr_phm1'	'1.0'	Physics
*	'b2sifr_phm2'	'1.0'	Physics

*	'b2sifr_phm3'	'1.0'	Physics
*	'b2sifr_styl0'	'0'	Numerics
*	'b2sifr_styl1'	'0'	Numerics
*	'b2sigp_pressure_restriction'	'1'	Numerics
*	'b2sigp_style'	'2'	Numerics
*	'b2sihs_istyle_Joule_heating'	'1'	Physics
*	'b2sihs_phm0'	'1.0'	Physics
*	'b2sihs_phm1'	'1.0'	Physics
*	'b2sihs_phm2'	'1.0'	Physics
*	'b2sihs_phm3'	'1.0'	Physics
*	'b2sihs_phm4'	'1.0'	Physics
*	'b2sihs_phm5'	'1.0'	Physics
*	'b2sihs_phm6'	'1.0'	Physics
*	'b2sihs_phm7'	'0.0'	Physics
*	'b2sihs_phm8'	'1.0'	Physics
*	'b2sihs_rf0'	'1.0'	Numerics
*	'b2sihs_rf1'	'1.0'	Numerics
*	'b2sihs_rf2'	'1.0'	Numerics
*	'b2sihs_rf3'	'1.0'	Numerics
*	'b2sihs_rf4'	'1.0'	Numerics
*	'b2sihs_style'	'0'	Numerics
*	'b2sqcx_phm0'	'1.0'	Physics
*	'b2sqcx_styl0'	'0'	Physics
*	'b2sqel_artificial_radiation'	'0.0'	Physics
*	'b2sqel_phm0'	'1.0'	Physics
*	'b2sqel_phm1'	'1.0'	Physics
*	'b2sqel_phm2'	'1.0'	Physics
*	'b2sral_inputfile'	'0'	Run
*	'b2sral_style'	'2'	Physics
*	'b2srdt_numerics_namelist'	'0'	Run
*	'b2srdt_phm0'	'1.0'	Numerics
*	'b2srdt_phm1'	'1.0'	Numerics
*	'b2srdt_phm3'	'1.0'	Numerics
*	'b2srdt_phm4'	'0.0'	Numerics
*	'b2srdt_phm5'	'1.0'	Numerics
*	'b2srsm_diagno'	'0'	Output
*	'b2srsm_enable'	'0'	Numerics
*	'b2srst_rf0'	'1.0'	Numerics
*	'b2srst_rf1'	'1.0'	Numerics
*	'b2srst_rf2'	'1.0'	Numerics
*	'b2srst_rf3'	'1.0'	Numerics
*	'b2stbc_bc_ref'	'0.01'	Physics
*	'b2stbc_bc_ref_te'	'0.01'	Physics
*	'b2stbc_bc_ref_ti'	'0.01'	Physics
*	'b2stbc_bcene_l6_style'	'0'	Physics
*	'b2stbc_bceni_l6_style'	'0'	Physics
*	'b2stbc_bcpot_l6_step'	'50'	Numerics
*	'b2stbc_boundary_namelist'	See description	Run
*	'b2stbc_cbc'	'1.0'	Numerics
*	'b2stbc_cbsnafac'	'0.01'	Numerics
*	'b2stbc_coreregn2'	'4'	Geometry
*	'b2stbc_coreregn0'	'1'	Geometry
*	'b2stbc_diagno'	'0'	Output
*	'b2stbc_fchy_dia'	'0.0'	Numerics
*	'b2stbc_fchy_dia_coreonly'	'1'	Numerics
*	'b2stbc_fchycore'	'-1.0e30'	Physics
*	'b2stbc_fchycore_alpha'	'0.1'	Numerics
*	'b2stbc_feedback'	'0'	Run
*	'b2stbc_fheycore'	'0.0'	Physics
*	'b2stbc_fheycore_alpha'	'0.1'	Numerics
*	'b2stbc_fhiycore'	'0.0'	Physics
*	'b2stbc_fhiycore_alpha'	'0.1'	Numerics
*	'b2stbc_fhiycore_kinetic_energy'	'0'	Physics
*	'b2stbc_fix_fch_in_fhe_sheath'	'2'	Numerics
*	'b2stbc_fnaycore'	'-1.0e30'	Physics
*	'b2stbc_fnaycore_alpha'	'0.1'	Numerics
*	'b2stbc_integral_current'	'0.0'	Numerics

*	'b2stbc_isfeedback'	'0'	Physics
*	'b2stbc_istyle_cur_contr_on_S_and_N'	'2'	Numerics
*	'b2stbc_istyle_fchi'	'0'	Numerics
*	'b2stbc_iyped'	jsep/2	Physics
*	'b2stbc_ncallfeedback'	'0'	Run
*	'b2stbc_ndes'	'0.0'	Physics
*	'b2stbc_ndes_sol'	'0.0'	Physics
*	'b2stbc_neoclassical'	'0.0'	Numerics
*	'b2stbc_nepedn_sol'	'0.0'	Physics
*	'b2stbc_nesepm'	'0.0'	Physics
*	'b2stbc_nesepm_alpha'	'1.0e-3'	Numerics
*	'b2stbc_nesepm_beta'	'0.0'	Numerics
*	'b2stbc_nesepm_gamma'	'0.99'	Numerics
*	'b2stbc_nesepm_maxpuff'	'0.0'	Physics
*	'b2stbc_nesepm_minpuff'	'0.0'	Physics
*	'b2stbc_nesepm_overshoot'	'0.0'	Physics
*	'b2stbc_nesepm_pfr'	'0.0'	Physics
*	'b2stbc_nesepm_sol'	'0.0'	Physics
*	'b2stbc_pfrregno1'	'0'	Geometry
*	'b2stbc_pfrregno2'	'2'	Geometry
*	'b2stbc_private_flux_puff'	'0.0'	Physics
*	'b2stbc_secmodel'	'0'	Physics
*	'b2stbc_she0ep'	'1.0e-36'	Numerics
*	'b2stbc_sheath_drift_fix'	'1'	Numerics
*	'b2stbc_shi0ep'	'1.0e-36'	Numerics
*	'b2stbc_sna0ep'	'1.0e-36'	Numerics
*	'b2stbc_solregno'	'3'	Geometry
*	'b2stbc_type13_fac'	'1.0'	Numerics
*	'b2stbc_type13_norm'	'1.0e15'	Numerics
*	'b2stbc_type13_ref'	'1'	Physics
*	'b2stbc_type16_kinetic_energy'	'0'	Physics
*	'b2stbc_type16_ref'	'1'	Physics
*	'b2stbc_type20_ref'	'1'	Physics
*	'b2stbc_type21_ref'	'1'	Physics
*	'b2stbc_volrec'	'0.0'	Physics
*	'b2stbc_volrec_alpha'	'1.0e-3'	Numerics
*	'b2stbc_volrec_beta'	'1.0'	Numerics
*	'b2stbc_volrec_overshoot'	'0.0'	Physics
*	'b2stbm_impgyro_mod'	'0'	Numerics
*	'b2stbm_linearisation'	'1.0'	Numerics
*	'b2stbr_alpha'	'0.25'	Physics
*	'b2stbr_b2wall_netcdf'	'0'	Output
*	'b2stbr_bas_recycled_neutrals_contr'	'1.0'	Physics
*	'b2stbr_c_pumpavr'	'0.02'	Numerics
*	'b2stbr_core_sources_rescale'	'1.0'	Run
*	'b2stbr_eir_src_nhist'	'1'	Numerics
*	'b2stbr_first_flight'	'0'	Run
*	'b2stbr_first_flight_dl'	'0.001'	Numerics
*	'b2stbr_first_flight_no_of_flights'	'9'	Numerics
*	'b2stbr_first_flight_no_of_start_points'	2*(nx+2)+2*max(nncut,1)*(ny+2)	Geometry
*	'b2stbr_first_flight_table_size'	'200000'	Numerics
*	'b2stbr_neutrals_namelist'	'0'	Run
*	'b2stbr_output'	'0'	Output
*	'b2stbr_plate_model'	'0'	Physics
*	'b2stbr_plate_option'	'3'	Physics
*	'b2stbr_plate_temp'	'300.0'	Physics
*	'b2stbr_plate_thick'	'0.00'	Physics
*	'b2stbr_potential_at_guard_cell'	'1'	Numerics
*	'b2stbr_redep_alpha'	'0.00'	Physics
*	'b2stbr_refl_model'	'1'	Physics
*	'b2stbr_reflection_on'	'1'	Physics
*	'b2stbr_sna_corr'	'0.0'	Numerics
*	'b2stbr_sput_chem_alpha'	'1.0e-2'	Numerics
*	'b2stbr_sput_chem_cutoff_alpha'	'1.0'	Physics
*	'b2stbr_sput_chem_cutoff_beta'	'3.0'	Physics
*	'b2stbr_sput_chem_model'	'0'	Physics
*	'b2stbr_sput_dst'	'-1'	Physics

*	'b2stbr_sput_dst2'	'-1'	Physics
*	'b2stbr_sput_dst3'	'-1'	Physics
*	'b2stbr_sput_frac_flag'	'0'	Physics
*	'b2stbr_sput_frc'	'0.0'	Physics
*	'b2stbr_sput_mixed_alpha'	'1.0'	Physics
*	'b2stbr_sput_mixed_beta'	'1.0'	Physics
*	'b2stbr_sput_phys'	'0.0'	Physics
*	'b2stbr_sput_phys_alpha'	'1.0e-2'	Numerics
*	'b2stbr_sput_phys_col'	'3'	Physics
*	'b2stbr_sput_phys_model'	'1'	Physics
*	'b2stbr_sput_res'	'0.0'	Physics
*	'b2stbr_sput_src'	'1'	Physics
*	'b2stbr_sputter_energy_on'	'1'	Physics
*	'b2stbr_tauxmax'	'0.05'	Numerics
*	'b2stbr_therm_evap'	'0.0'	Physics
*	'b2stcx_rg0'	'1.0'	Numerics
*	'b2stcx_styl0'	'0'	Numerics
*	'b2stel_phm0'	'1.0'	Physics
*	'b2stel_rg0'	'1.0'	Numerics
*	'b2stel_rgl'	'1.0'	Numerics
*	'b2stel_rxm0'	'0.0'	Numerics
*	'b2stel_rxm1'	'0.0'	Numerics
*	'b2stel_rxm2'	'0.0'	Numerics
*	'b2stel_rxm3'	'0.0'	Numerics
*	'b2stel_rxm4'	'0.0'	Numerics
*	'b2stel_styl0'	'0'	Numerics
*	'b2tanml_anomalous'	'1.0'	Physics
*	'b2tfcc_xfac'	'1.0'	Numerics
*	'b2tfhe_PSch'	'1.0'	Physics
*	'b2tfhe_alfTeEh'	'0.0'	Physics
*	'b2tfhe_anomalous'	'1.0'	Physics
*	'b2tfhe_conduction_only'	'0'	Physics
*	'b2tfhe_dia_cur'	'1.0'	Physics
*	'b2tfhe_fch_pTe'	'1.0'	Physics
*	'b2tfhe_hybr2'	'0'	Numerics
*	'b2tfhe_lim_flux'	'0'	Physics
*	'b2tfhe_mdf'	'0'	Numerics
*	'b2tfhe_neutral'	'0.0'	Physics
*	'b2tfhe_no_current'	'0'	Numerics
*	'b2tfhe_no_hybr'	'0'	Numerics
*	'b2tfhe_prl_cur'	'1.0'	Numerics
*	'b2tfhe_stochastic'	'0.0'	Physics
*	'b2tfhe_upwind'	'0'	Numerics
*	'b2tfhe_vdia_par'	'1.0'	Physics
*	'b2tfhe_vis_par'	'1.0'	Physics
*	'b2tfhe_vis_per'	'1.0'	Numerics
*	'b2tfhe_vis_q'	'1.0'	Physics
*	'b2tfhi_hybr2'	'0'	Numerics
*	'b2tfhi_lim_flux'	'0'	Physics
*	'b2tfhi_mdf'	'0'	Numerics
*	'b2tfhi_no_hybr'	'0'	Numerics
*	'b2tfhi_upwind'	'0'	Numerics
*	'b2tfnb_PSch'	'1.0'	Physics
*	'b2tfnb_PSch_style'	See description	Numerics
*	'b2tfnb_alpha'	'0.0'	Physics
*	'b2tfnb_drift_style'	See description	Numerics
*	'b2tfnb_flux_limit_min_ti'	'0.0'	Physics
*	'b2tfnb_gamma'	'2.0'	Physics
*	'b2tfnb_mdf'	'0'	Numerics
*	'b2tfnb_no_hybr'	'0'	Numerics
*	'b2tfnb_pflux_cor'	'1'	Numerics
*	'b2tfnb_poldidr'	'1.0'	Numerics
*	'b2tfnb_poleldr'	'1.0'	Numerics
*	'b2tfnb_raddidr'	'1.0'	Numerics
*	'b2tfnb_radelldr'	'1.0'	Numerics
*	'b2tfnb_vis_per'	'1.0'	Physics
*	'b2tfnb_vis_q'	'1.0'	Physics

*	'b2tfnb_xcur'	'0.0'	Physics
*	'b2tfnb_xfrhie'	'1.0'	Numerics
*	'b2tfnb_ycur'	'1.0'	Physics
*	'b2tfrn_xvecrb'	'1.0'	Physics
*	'b2tfrn_xwdia'	'1.0'	Physics
*	'b2tiner_fch_inert_style'	See description	Numerics
*	'b2tiner_inert'	'1.0'	Physics
*	'b2tinnt_fchin_in_core'	'0'	Physics
*	'b2tlc0_alpha'	'1.0'	Physics
*	'b2tlc0_gamma'	'1.0'	Physics
*	'b2tlh0_alpha'	'1.0'	Physics
*	'b2tlh0_flux_limit_min_ti'	'0.0'	Physics
*	'b2tlh0_flux_limit_style'	'2'	Numerics
*	'b2tlh0_gamma'	'1.0'	Physics
*	'b2tlh0_hcimx_flag'	'1'	Numerics
*	'b2tlmv_style'	'1'	Physics
*	'b2tlnl_ee'	'0'	Physics
*	'b2tlnl_ei'	'0'	Physics
*	'b2tlnl_ii'	'0'	Physics
*	'b2tqca_model'	'1'	Physics
*	'b2tqca_phm0'	'1.0'	Physics
*	'b2tqce_fke_Zhdanov'	'1'	Physics
*	'b2tqce_model'	'1'	Physics
*	'b2tqin_csign_style'	'1'	Physics
*	'b2tqna_ballooning'	'0.0'	Physics
*	'b2tqna_ballooning_rescale'	'1.0'	Physics
*	'b2tqna_bb_ref'	See description (real)	Physics
*	'b2tqna_diagno'	'0'	Output
*	'b2tqna_divsol_rescale'	'1.0'	Physics
*	'b2tqna_inputfile'	'0'	Run
*	'b2tqna_ixref'	See description (integer)	Physics
*	'b2tqna_max_df0'	'1e30'	Physics
*	'b2tqna_min_df0'	'0.0'	Physics
*	'b2tqna_model_sig'	'0'	Physics
*	'b2tqna_new_df0'	'0'	Physics
*	'b2tqna_pfr_rescale'	'1.0'	Physics
*	'b2tqna_transport_namelist'	'0'	Run
*	'b2tqna_user_transport'	'0'	Physics
*	'b2tral_cvsa_style'	'1'	Physics
*	'b2tral_mode'	'1'	Numerics
*	'b2trcl_conductive_limit'	'1'	Numerics
*	'b2trcl_core_cond_limit'	'0'	Numerics
*	'b2trcl_csig_mltpl'	'1.0'	Numerics
*	'b2trcl_cthe'	'0.0'	Physics
*	'b2trcl_cthi'	'2.65'	Physics
*	'b2trcl_cvsa_mltpl'	'1.0'	Numerics
*	'b2trcl_lambda'	'-5.0'	Physics
*	'b2trcl_lluciani'	'3'	Physics
*	'b2trcl_lthf21'	'0'	Physics
*	'b2trcl_lvis21'	'0'	Physics
*	'b2treq_phm0'	'1.0'	Physics
*	'b2trno_con_e_stochastic'	'1.0'	Physics
*	'b2trno_csig_an_style'	'1'	Physics
*	'b2trno_flux_limit_to_dpa'	'1'	Numerics
*	'b2trno_pol_anom_scale'	'1.0'	Physics
*	'b2tstch_delta'	'0.0'	Physics
*	'b2tstch_sig'	'1.0'	Physics
*	'b2upht_stylec'	'0'	Numerics
*	'b2usmo_cfc0'	'1.0'	Numerics
*	'b2ux11p_acpar'	'8.0'	Numerics
*	'b2ux11p_cpu'	'0'	Output
*	'b2ux11p_mult_nonzero'	'10'	Numerics
*	'b2ux11p_mult_solvdim'	'15'	Numerics
*	'b2ux11p_mult_solvdim1'	'10'	Numerics
*	'b2ux11p_nltrsol'	'2'	Output
*	'b2ux11p_style'	'2'	Numerics
*	'b2ux5p_acpar'	'8.0'	Numerics

*	'b2ux5p_cpu'	'0'	Output
*	'b2ux5p_mult_nonzero'	'10'	Numerics
*	'b2ux5p_mult_solvdim'	'15'	Numerics
*	'b2ux5p_mult_solvdim1'	'10'	Numerics
*	'b2ux5p_nltrsol'	'2'	Output
*	'b2ux5p_style'	'2'	Numerics
*	'b2ux7p_acpar'	'8.0'	Numerics
*	'b2ux7p_cpu'	'0'	Output
*	'b2ux7p_mult_nonzero'	'10'	Numerics
*	'b2ux7p_mult_solvdim'	'15'	Numerics
*	'b2ux7p_mult_solvdim1'	'10'	Numerics
*	'b2ux7p_nltrsol'	'2'	Output
*	'b2ux7p_style'	'2'	Numerics
*	'b2ux9p_acpar'	'8.0'	Numerics
*	'b2ux9p_cpu'	'0'	Output
*	'b2ux9p_mult_nonzero'	'10'	Numerics
*	'b2ux9p_mult_solvdim'	'15'	Numerics
*	'b2ux9p_mult_solvdim1'	'10'	Numerics
*	'b2ux9p_nltrsol'	'2'	Output
*	'b2ux9p_style'	'2'	Numerics
*	'b2wdat_append'	'0'	Output
*	'b2wdat_iout'	'0'	Output
*	'b2xzdd_zero_dead_and_core'	'1'	Numerics
*	'b2yrdr_ns'	ns	Output
*	'b2ytdr_jsep1'	nx1/2	Run
*	'b2ytdr_ndepth1'	ndepth_nml	Run
*	'b2ytdr_non_commensurate'	'0'	Run
*	'b2ytdr_ns'	ns	Run
*	'b2ytdr_rescale_neutrals'	'1.0'	Run
*	'balance_average'	'0'	Output
*	'balance_netcdf'	'0'	Output
*	'eirene_ank_mods'	'0'	Numerics
*	'eirene_background'	'1'	Physics
*	'eirene_dpc_fix'	'1'	Numerics
*	'eirene_extrap'	'1'	Numerics
*	'eirene_format'	'iter'	Output
*	'eirene_ionising_core'	'0'	Physics
*	'eirene_lhalpha'	'1'	Physics
*	'eirene_lvib'	'1'	Physics
*	'eirene_mc_linearisation'	'1'	Numerics
*	'eirene_mc_output_style'	'1'	Output
*	'eirene_na_max'	'1e30'	Numerics
*	'eirene_na_min'	'0.0'	Numerics
*	'eirene_nesepm_istra'	'-1'	Physics
*	'eirene_neutr_avg'	'0'	Numerics
*	'eirene_print_minmax'	'0'	Numerics
*	'eirene_repeat_first_call'	'1'	Physics
*	'eirene_savef30'	'0'	Output
*	'eirene_savef31'	'0'	Output
*	'eirene_sheath_pot'	'1'	Physics
*	'eirene_te_max'	'1e30'	Numerics
*	'eirene_te_min'	'0.0'	Numerics
*	'eirene_ti_max'	'1e30'	Numerics
*	'eirene_ti_min'	'0.0'	Numerics
*	'eirene_ua_max'	'+c'	Numerics
*	'eirene_ua_min'	'-c'	Numerics
*	'eirene_underrelax'	'0'	Numerics
*	'eirene_use_recyceir'	'1'	Physics
*	'eirene_uub_style'	'0'	Numerics
*	'get_residuals'	'0'	Output
*	'heatdiff1D_linlog'	'1'	Numerics
*	'heatdiff1D_ratio'	'100.0'	Numerics
*	'ids_from_43'	'0'	Output
*	'ma28_nwrite'	'0'	Output
*	'my_out_digits'	6 or 15	Output
*	'neoclassical_ic'	'3'	Physics
*	'nrings_for_no_speedup_averaging'	'0'	Numerics

```

*   'set_transport_eta'           '2.0'           Physics
*   'set_transport_eta_alpha'     '0.5'           Physics
*   'set_transport_eta_ceiling'   '10.0'          Physics
*   'set_transport_eta_floor'     '0.1'           Physics
*   'set_transport_ixref'         See description (integer) Physics
*   'set_transport_iyref'         See description (integer) Physics
*   'set_transport_required_te_gradient' '5.0e4'    Physics
*   'solps_version'               'iter'          Output
*   'tallies_netcdf'              '0'             Output
* -----
*
*
*
* 1. Purpose of individual parameters
*
*   * 1.1. Geometry switches
*
*   'b2agfs_geometry'             'upgrade.geometry' Geometry
*   local_sonnet is the file name of the geometry file to be read.
*   The file will be looked for in the run directory, in the ../baserun
*   directory, in $SOLPSTOP/data.local/meshes, in $SOLPSTOP/data/meshes
*   and in $SOLPSTOP/modules/Carre/meshes/$DEVICE. To be used in b2ag.dat.
*
*   'b2mwti_jxa'                  See description (integer) Geometry
*   Cell index, on the basis mesh, of the outer midplane. Default value
*   depends on geometry:
*   Lower Single-null : jxa=rightcut(1)-(rightcut(1)-leftcut(1))/4, i.e.
*   three quarters of the way between the two cuts.
*   Upper Single-null : jxa=leftcut(1)+(rightcut(1)-leftcut(1))/4, i.e.
*   one quarter of the way between the two cuts.
*   Stellarator island : jxa=leftcut(1)+(rightcut(1)-leftcut(1))/2, i.e.
*   midway between the two cuts.
*   Double-null : jxa=(rightcut(1)+rightcut(2))/2, i.e. halfway
*   between the two outer cuts.
*   Limiter geometry : jxa=nx/2
*   Straight geometry : jxa=nx/2
*
*   'b2mwti_jxi'                  See description (integer) Geometry
*   Cell index, on the basis mesh, of the inner midplane. Default value
*   depends on geometry:
*   Lower Single-null : jxi=leftcut(1)+(rightcut(1)-leftcut(1))/4, i.e.
*   one quarter of the way between the two cuts.
*   Upper Single-null : jxi=rightcut(1)-(rightcut(1)-leftcut(1))/4, i.e.
*   three quarters of the way between the two cuts.
*   Stellarator island : jxi=leftcut(1)+(rightcut(1)-leftcut(1))/4, i.e.
*   one quarter of the way between the two cuts.
*   Double-null : jxi=(leftcut(1)+leftcut(2))/2, i.e. halfway
*   between the two inner cuts.
*   Limiter geometry : jxi=nx/4
*   Straight geometry : jxi=nx/4
*
*   'b2mwti_jsep'                 See description (integer) Geometry
*   Cell index, on the basis mesh, of the last poloidal ring before the
*   separatrix. Can only be provided for slab geometries, otherwise
*   calculated from geometry:
*   Single-null : jsep=topcut(1)-1
*   Stellarator : jsep=topcut(1)-1
*   Double-null : jsep=min(topcut(1),topcut(2))-1
*   Default for straight geometries : jsep=ny/2
*
*   'b2agmx_pbs_from_basis_mesh' '1'             Geometry
*   If pbs_from_basis_mesh.eq.0, the pbs array is computed from values on
*   the actual working mesh (old treatment).
*   If pbs_from_basis_mesh.eq.1, the pbs array is computed from values on
*   the (possibly) finer basis mesh from the geometry file and the
*   contributions from each basis cell are added to obtain the value on
*   the working mesh.

```



```

*      In both cases, mind the value of 'b2news_area_fix', which should be
*      also declared in b2ag.dat if the non-default behaviour is wanted.
*
*      'b2agmt_qc_style'          '1'                      Geometry
*      When b2agmt_qc_style.eq.1, the cosine of the angle between the left
*      face of cells and the magnetic field direction is computed using
*      information from cells to both the left and right of the face, based
*      on interpolations of the qz array (cosine at cell centers).
*      When b2agmt_qc_style.eq.2, the cosine of the angle between the left
*      face of cells and the magnetic field direction is computed assuming
*      that it is the angle between the left cell face and the bisector of
*      the poloidal direction in the left and right cells sharing the common
*      face.
*      When b2agmt_qc_style.eq.0, we only use information from the cell at
*      the right of the face. This is the treatment used up to version 3.0.6,
*      inclusively.
*
*      'b2agdr_redef_qc'          '0'                      Geometry
*      When b2agdr_redef_qc.eq.1, the cosine of the angle between the left
*      face of a cell and the magnetic field direction is interpolated from
*      the values at the cell centers.
*      When b2agdr_redef_qc.eq.0, we only use information from the cell at
*      the right of the face. This is the treatment used up to version 3.0.6,
*      inclusively.
*      This switch is superseded by b2agmt_qc_style.
*
*      'b2agdr_redef_gmtry'       '1'                      Geometry
*      When b2agdr_redef_gmtry.eq.1, geometrical quantities are adjusted so
*      as to ensure that the poloidal flux between two neighbouring flux
*      surfaces remains constant and that toroidal magnetic field times major
*      radius is the same constant for all cells independently of whether it
*      is computed at cell centers, cell faces, or cell corners.
*
*      'b2agfs_periodic_bc'       '0'                      Geometry
*      periodic_bc specifies if this is either an island or limiter geometry.
*      If periodic_bc.eq.1 then island/limiter treatment is turned on. We
*      differentiate between the two case through nncut:
*      nncut.eq.0 = limiter case
*      nncut.ge.1 = island divertor case (there should be nncut islands then)
*      The case periodic_bc.eq.-1 is used to remove tranverse physics in 1-D
*      cases.
*      If periodic_bc.eq.0 then we have the usual treatment. No other values
*      are currently allowed. To be used in b2ag.dat.
*
*      'b2agsi_isymm'             '1'                      Geometry
*      isymm specifies the type of symmetry of the geometry: isymm.eq.0
*      implies a slab geometry,
*      isymm.eq.1.or.isymm.eq.2 imply rotational symmetry about the crx=0
*      axis, and isymm.eq.3.or.isymm.eq.4 indicate rotational symmetry about
*      the cry=0 axis.
*      Other values are not allowed.
*      isymm.eq.1.or.isymm.eq.3 mean a toroidal geometry case, i.e. with a
*      toroidal (out-of-plane) magnetic field component.
*      isymm.eq.2.or.isymm.eq.4 mean a cylindrical geometry case, i.e. no
*      toroidal magnetic field component. To be used in b2ag.dat.
*
*      'b2stbc_coreregn0'         '1'                      Geometry
*      'b2stbc_coreregn2'         '4'                      Geometry
*      coreregn0 is the boundary index of the core boundary in the input
*      files b2ah.dat and b2mn.dat. For a standard single-null case,
*      coreregn0 is 1. For a straight geometry or limiter case, it is likely
*      that coreregn0 need be set to 0, depending on the actual geometry
*      details.
*      coreregn2 is the boundary index of the second core boundary in case of
*      a double-null geometry, and is not used otherwise.
*
*      'b2stbc_pfrregno1'         '0'                      Geometry

```

```

*      'b2stbc_pfrregno2'          '2'                      Geometry
*      pfrregno1 is the boundary index of the first half of the private flux
*      region boundary on which feedback is being applied (through
*      b2stbc_ndes or b2stbc_private_flux_puff) in the input files b2ah.dat
*      and b2mn.dat. pfrregno2 is the second half of the same private flux
*      region. For a standard single null, the pair should be (0,2). For a
*      standard double-null, the lower private flux region is labelled with
*      the pair (0,5), and the upper private flux region with the pair (2,3).
*      One can use only one private flux region by setting
*      pfrregno1.eq.pfrregno2.
*
*      'b2stbc_solregno'          '3'                      Geometry
*      solregno is the boundary index of the SOL North boundary in the input
*      files b2ah.dat and b2mn.dat. For a standard single-null case, solregno
*      is 3. For a standard double-null case, solregno for the outer SOL
*      North boundary would be 7.
*
*      'b2agmt_1d_width'          '1.0'                    Geometry
*      For 1-D cases, gives the width of the domain (in m) in the third
*      (toroidal) dimension.
*
*      'b2stbr_first_flight_no_of_start_points' 2*(nx+2)+2*max(nncut,1)*(ny+2) Geometry
*      When the first flight model is turned on, this number must be greater
*      than or equal to the num
*      ber of boundary cells on the mesh. Only in cases with special
*      geometries (i.e. limiters, islands, ...) need it be increased beyond
*      the default value above. This variable is only used within the first
*      flight module.
*
*      'b2agfs_geom_match_dist'    '1.0e-6'                Geometry
*      Distance used as the matching criterion when reading the geometry
*      file.
*
*      'b2agfs_Bt_adjust'          '0'                      Geometry
*      If the mesh is offset (See b2agfs_xoffset, b2agfs_yoffset below) and
*      Bt_adjust.eq.1, then the magnetic field is recomputed for toroidal
*      geometries, keeping the pitch constant but changing the toroidal field
*      magnitude according to  $B_t = B_{t0}/R$ . To be used in b2ag.dat.
*
*      'b2agfs_Bt_rescale'         '1.0'                    Geometry
*      The magnetic field will be multiplied by Bt_rescale. All components of
*      the field are scaled together. To be used in b2ag.dat.
*
*      'b2agfs_pit_rescale'        '1.0'                    Geometry
*      The magnetic field line pitch will be multiplied by pit_rescale.
*      This means that the poloidal field component is multiplied by
*      pit_rescale, while the toroidal field component is left unchanged. To
*      reverse the plasma current direction, set pit_rescale to -1.0.
*      The sign convention used is that a positive poloidal field points in
*      the direction of increasing <ix>. Be mindful however that, when
*      inverting the sign of the poloidal magnetic field, you are also
*      inverting the direction of the parallel velocity. You will then need
*      to use the 'b2mndr_inverse_ua' switch to correct for that. To be used
*      in b2ag.dat.
*
*      'b2agfs_Bt_reversal'        '0'                      Geometry
*      If Bt_reversal.eq.1, then the sign of the toroidal field component
*      only is reversed. The sign convention used is that a positive toroidal
*      field leads to a  $B_x(\text{grad}(B))$  direction pointing down, so B is out of
*      the page when looking at the usual view of the poloidal plane with the
*      center line on the left. To be used in b2ag.dat.
*
*      'b2agfs_min_pitch'          '1.0'                    Geometry
*      Minimum allowed value for the pitch angle (in degrees) at the plates.
*
*      'b2agfs_xoffset'            '0.0'                    Geometry
*      'b2agfs_yoffset'            '0.0'                    Geometry

```

```

* xoffset and yoffset are offsets of the basis mesh in the x- and
* y-direction respectively. The mesh will be translated by
* (xoffset,yoffset) if any of the two is non-zero from its position
* given in the local_sonnet (See 'b2agfs_geometry' above) file.
* To be used in b2ag.dat.
*
* 'b2agfs_xrescale'          '1.0'          Geometry
* 'b2agfs_yrescale'          '1.0'          Geometry
* Rescaling factors of the x- and y- coordinates of the basis mesh.
* To be used in b2ag.dat.
*
* 'b2agfs_nncut'             See description (integer) Geometry
* Number of topological cuts in geometry. Needed only to force a
* different value from the one computed automatically by b2ag.
*
* 'b2agfs_leftcut'           See description (integer) Geometry
* 'b2agfs_rightcut'          See description (integer) Geometry
* 'b2agfs_bottomcut'         See description (integer) Geometry
* 'b2agfs_topcut'            See description (integer) Geometry
* Parameters describing the position and extent of the first topological
* cut, i.e. corresponding to the first X-point. Needed only to force
* different values from the ones computed automatically by b2ag.
* leftcut is the poloidal index of cells directly to the RIGHT of the
* left branch of the cut.
* rightcut is the poloidal index of cells directly to the RIGHT of the
* right branch of the cut.
* bottomcut is the radial index of cells directly below the cut.
* topcut is the radial index of cells directly above the cut.
*
* 'b2agfs_leftcut2'          See description (integer) Geometry
* 'b2agfs_rightcut2'         See description (integer) Geometry
* 'b2agfs_bottomcut2'        See description (integer) Geometry
* 'b2agfs_topcut2'           See description (integer) Geometry
* Parameters describing the position and extent of the second
* topological cut, i.e. corresponding to the second X-point. Needed only
* to force different values from the ones computed automatically by
* b2ag.
* leftcut2 is the poloidal index of cells directly to the RIGHT of the
* left branch of the second cut.
* rightcut2 is the poloidal index of cells directly to the RIGHT of the
* right branch of the second cut.
* bottomcut is the radial index of cells directly below the second cut.
* topcut is the radial index of cells directly above the second cut.
*
* 'b2agdr_nxisol1'           '-2'          Geometry
* 'b2agdr_nxisol2'           '-2'          Geometry
* 'b2agdr_nyiso1'            '-2'          Geometry
* 'b2agdr_nyiso2'            '-2'          Geometry
* Range of an optional isolated region to be included in the geometry.
* The isolated region will extend over the cell range spanned by
* (nxisol1:nxisol2,nyiso1:nyiso2).
* If only some of the four coordinates are given, the region is made to
* extent from the given coordinates to the end of the grid.
* Neighbourhood arrays and region indices are automatically adjusted.
*
* * 1.2. Run switches
*
* 'b2mndr_run_number'         '1000'       Run
* 'b2mndr_shot_number'        '0'          Run
* 'b2mndr_database'           $(DEVICE)    Run
* 'b2mndr_device'             $(DEVICE)    Run
* 'b2mndr_user'               $(USER)      Run
* These switches server as identification for the simulation. They can
* be inherited from SOLPS-GUI:
* Run number : The number of the run. Must be positive and no more than
* a 4-digit integer (i.e. span from 0 to 9999).
* Shot number : Shot number identifying the run. Defaults to the last

```

```

* number found in shotnumber.history, or 0 if the file is not found.
* Database : IMAS IDS database name (supersedes Device). Defaults to
* 'solps-iter' if $DEVICE if undefined.
* User : The user who ran the simulation.
*
* 'b2aidr_read_b2fstate'      '0'                      Run
* If read_b2fstate.ne.0, b2aidr will read an existing b2fstate file and
* append to it a flat state description for additional new species being
* introduced in the run.
*
* 'b2mndr_ntim'              '1'                      Run
* Number of timesteps desired in this run. If ntim.eq.0, the code simply
* reads the input file and creates the b2fplasma and b2fstate files
* corresponding to the present plasma state.
*
* 'b2mndr_dtim'              '1.0'                    Run
* Timestep (in seconds).
*
* 'b2mndr_delta_max'         '0.0'                    Run
* 'b2mndr_delta_min'         '0.0'                    Run
* 'b2mndr_dt_change_dec'     '1.0'                    Run
* 'b2mndr_dt_change_inc'     '1.0'                    Run
* 'b2mndr_dt_max'            '1.0e+01'                Run
* 'b2mndr_dt_min'            '1.0e-30'                Run
* One of the dynamic timestep control parameters. Only active if both
* delta_max and delta_min are nonzero. The timestep is increased by a
* factor dt_change_inc or decreased by a factor dt_change_dec, within
* the bounds [dt_min, dt_max], according to whether the largest
* instantaneous change in all equations is smaller than delta_min or
* larger than delta_max, respectively.
*
* 'b2mndr_stim'              '0.0'                    Run
* stim specifies the initial time --- default 0.
* If set to a positive or zero value, this overwrites the time value
* read from b2fstat1.
* If set to a negative value, the run continues from the time read in
* b2fstat1. In that case, the tracing data is appended to the existing
* files, otherwise the tracing files are overwritten.
*
* 'b2mndr_etim'              '0.0'                    Run
* etim specifies the end time. Only active if etim > stim.
*
* 'b2mndt_nstg0'             '1'                      Run
* 'b2mndt_nstg1'             '1'                      Run
* 'b2mndt_nstg2'             '1'                      Run
* Time-dependent mode and iterative mode switches. The basic code
* timestep proceeds as follows:
* ..test input arguments
* ..compute auxiliary quantities
* ..prepare source computation
* ..do i0=1,nstg0
* ..compute log-log linearised rate coefficients
* ..do i1=1,nstg1
* ..compute source linearisation
* ..do i2=1,nstg2
* ..perform one inner iteration
* ..re-compute auxiliary quantities
* ..produce monitoring output
* ..enddo
* ..enddo
* ..enddo
* This can be completed by the the 'b2mndt_nstg_ares??' switches.
* See 'Numerics' section for details.
*
* 'b2news_no_solve'          '0'                      Run
* If no_solve.eq.1, then the code is run without actually solving any
* equations. All secondary and dependent data is computed.

```

```

* The nstg(0:2) array is overwritten to '1's. The simulation time will
* not be updated. The code will compute fluxes, sources, transport
* coefficients, etc... 'ntim' times but not update the basic plasma
* quantities. Additionally, if no_solve is set to a negative value, then
* only some equations are solved (simulation time and nstg arrays will
* behave normally).
* no_solve.eq.-1 will activate the parallel momentum equations only.
* no_solve.eq.-2 will activate the density equations only.
* no_solve.eq.-4 will activate the potential equation only. (Active only
* if poteq.eq.1, otherwise, the behaviour dictated by the poteq setting
* applies).
* no_solve.eq.-8 will activate the heat equations only. These can be
* combined. For example, no_solve.eq.-3 will activate the parallel
* momentum and particle conservation equations.
* If the no_solve switch requires an equation not to be solved, the
* settings in the corresponding solvexx arrays from
* b2.numerics.parameters are moot.
* The latter are reserved for fine-tuning numerical diagnostics.
* If wishing to toggle whether to solve or not to solve each equation
* separately, one should set, in the b2.numerics.parameters input file,
* the SOLVEMO, SOLVEMT, SOLVECO, SOLVEE, SOLVEEI, SOLVEET and SOLVEPO
* arrays, respectively, for each of the parallel momentum equations, the
* total momentum equation, each density equation, the electron and ion
* heat equations, and the total energy equation.
* The table below shows which combination of equations will be solved
* depending on the value of switch 'b2news_no_solve'
*
* -----
* | value | co | mo | he | hi | po |
* -----
* | 0 | + | + | + | + | + |
* -----
* | 1 | - | - | - | - | - |
* -----
* | -1 | - | + | - | - | - |
* -----
* | -2 | + | - | - | - | - |
* -----
* | -3 | + | + | - | - | - |
* -----
* | -4 | - | - | - | - | + |
* -----
* | -5 | - | + | - | - | + |
* -----
* | -6 | + | - | - | - | + |
* -----
* | -7 | + | + | - | - | + |
* -----
* | -8 | - | - | + | + | - |
* -----
* | -9 | - | + | + | + | - |
* -----
* | -10 | + | - | + | + | - |
* -----
* | -11 | + | + | + | + | - |
* -----
* | -12 | - | - | + | + | + |
* -----
* | -13 | - | + | + | + | + |
* -----
* | -14 | + | - | + | + | + |
* -----
*
* 'b2mndr_cpu' '0.0' Run
* CPU limit. If cpu.eq.0.0, no CPU limit is enforced. Otherwise, the
* code stops smoothly after it has been running for at least 'cpu' CPU
* seconds.
*

```

```

*      'b2mndr_elapsed'          '0.0'                Run
*      Elapsed time limit. Is elapsed.eq.0.0, no time limit is enforced.
*      Otherwise, the code stops smoothly after is has been running for at
*      least 'elapsed' wall clock seconds.
*
*      'b2mndr_savecpu'          '3600.0'             Run
*      CPU time interval after which save files plasmastate.xxxx are written.
*      These can be used, in conjunction with the b2co.exe instruction, to
*      recover a crashed run.
*
*      'b2mndr_ismain'           '1'                  Run
*      ismain identifies the index of the main plasma species.
*      It must hold that ismain is not a neutral species.
*
*      'b2news_facdrift_dec'      '0.0'                Run
*      'b2news_facdrift_inc'      '1.0'                Run
*      'b2news_facdrift_start'    '0.0'                Run
*      'b2news_facdrift_target'   '0.0'                Run
*      Ramping parameters for facdrift, which multiplies the diamagnetic
*      terms as well as the ion inertia current. The code is started on the
*      first time step with facdrift=facdrift_start. If
*      facdrift_target.ne.facdrift_start, then, on each time step, facdrift
*      is multiplied by facdrift_inc. If the code does not converge on the
*      timestep, facdrift is decreased by facdrift_dec. A facdrift profile is
*      also possible, see Numerics section for details.
*      The ion-neutral friction current requires either facdrift or fac_ExB
*      to be turned on as well.
*
*      'b2news_facExB_dec'        '0.0'                Run
*      'b2news_facExB_inc'        '1.0'                Run
*      'b2news_facExB_start'      '0.0'                Run
*      'b2news_facExB_target'     '0.0'                Run
*      Ramping parameters for facExB, which multiplies the ExB terms. Same
*      treatment as above for facdrift.
*
*      'b2news_facvis_dec'        '0.0'                Run
*      'b2news_facvis_inc'        '1.0'                Run
*      'b2news_facvis_start'      '0.0'                Run
*      'b2news_facvis_target'     '0.0'                Run
*      Ramping parameters for facvis, which multiplies the drift viscosity
*      terms. Same treatment as above for facdrift.
*
*      'b2stbc_boundary_namelist' See description      Run
*      'b2stbr_neutrals_namelist' '0'                  Run
*      'b2srdt_numerics_namelist' '0'                  Run
*      'b2tqna_transport_namelist' '0'                  Run
*      Namelist file indicators. If xxx_namelist.eq.1, then the code looks
*      for the file b2.xxx.parameters, in which it expects to find the xxx
*      namelist.
*      In the case of boundary_namelist, defaults to 1 for double-null
*      geometries, 0 otherwise.
*
*      'b2sral_inputfile'         '0'                  Run
*      Profiles file indicator. If b2sral_inputfile.eq.1, then the code looks
*      for the file b2.sources.profile in which it expects to find the
*      'profile' namelist.
*      This namelist will contain profiles of sources measured from the
*      location (ixref,iyref) (default is outer midplane separatrix), which
*      is set with the 'set_transport_i[xy]ref' switches.
*
*      'b2tqna_inputfile'         '0'                  Run
*      Profiles file indicator. If b2tqna_inputfile.eq.1, then the code looks
*      for the file b2.transport.inputfile, in which it expects to find the
*      transport namelist.
*      This namelist will contain profiles of sources transport parameters
*      measured from the location (ixref,iyref) (default is outer midplane
*      separatrix), which is set with the 'set_transport_i[xy]ref' switches.

```

```

*
* 'b2mndr_eirene'          '0'          Run
* Turns on coupling with the Eirene Monte Carlo neutral code if nonzero.
* To be used, the code must be compiled with the -DB25_EIRENE option.
*
* 'b2mndr_astra'          '0'          Run
* Turns on coupling with the ASTRA core transport code if nonzero.
* To be used, the code must be compiled with the -DASTRA option.
*
* 'b2mndr_rescale_neutrals_sources' '1.0'          Run
* Multiplier to the neutral sources (either computed by Eirene or the
* corresponding rate coefficients).
*
* 'b2mndr_rescale_neutrals' '1.0'          Run
* Multiplier to the neutral density on the first timestep.
*
* 'b2mndr_density_rescale' '1.0'          Run
* Multiplier of all densities on the first timestep.
* If set to a negative value, the rescaling (by |density_rescale|) will
* only be applied to species with zn.eq.zn_rescale.
*
* 'b2mndr_zn_rescale'      '0'          Run
* Nuclear charge number of the species to be rescaled if density_rescale
* is set to a negative value.
*
* 'b2stbr_core_sources_rescale' '1.0'          Run
* Multiplier to the totally ionised species sources at the core
* boundary.
*
* 'b2mndt_density_control' '0'          Run
* Feedback on the total heavy particle density. If density_control.ne.0,
* the sum of all densities is kept constant.
*
* 'b2stbc_feedback'        '0'          Run
* If feedback.eq.1, turns on feedback mode for the boundary conditions.
* See Physics section for more details. Equivalent to LFEEDBACK=.true.
* in the b2.boundary.parameters namelist.
*
* 'b2stbc_ncallfeedback'    '0'          Run
* Timestep index after which the feedback in b2stbc is activated.
*
* 'b2stbr_first_flight'     '0'          Run
* If first_flight.ne.0, turns on the first flight model. See Physics
* section for additional details.
*
* 'b2ytdr_ns'              ns          Run
* New number of species desired when creating a new initial state file
* using b2yt.exe. Must be specified within b2yt.dat.
*
* 'b2ytdr_jsep1'           nx1/2        Run
* Radial index of separatrix in the converted grid. Must be specified
* within b2yt.dat. Only applies when converting linear geometries.
*
* 'b2ytdr_ndepth1'         ndepth_nml    Run
* New resolution of target depth desired when creating a new initial
* state file using b2yt.exe. Must be specified within b2yt.dat.
*
* 'b2ytdr_rescale_neutrals' '1.0'          Run
* Rescaling of neutral densities by rescale_neutrals when creating a new
* initial state file using b2yt.exe. Must be specified within b2yt.dat.
*
* 'b2ytdr_non_commensurate' '0'          Run
* When set to 1, b2yt will attempt to interpolate a plasma solution and
* create new input files for a new grid which is not commensurate to the
* original grid, but still has the same topology.
* Must be specified within b2yt.dat.
*

```

```

*      * 1.3. Physics switches
*
*      'b2siav_addvis'          '1.0'          Physics
*      Multiplier to heat flux contribution to divergence of viscosity tensor
*      in the momentum equation.
*
*      'b2siav_addvis1'        '1.0'          Physics
*      When not equal to '0.0', adds contribution to divergence of viscosity
*      tensor coming from x-variations in B.
*
*      'b2siav_style_qip'      '0'            Physics
*      If style_qip.eq.1, adds a classical ion heat conductivity term to the
*      heat flux used to compute the heat viscosity current (see manual for
*      full details).
*
*      'b2npmo_b2sifr_'        '1'            Physics
*      If b2sigp_style is set to '2', this switch has no effect.
*      When set to '1', the new correct form of the friction force is used,
*      applicable for non-hydrogenic plasmas or hydrogenic mixtures.
*      The value '0' corresponds to the old SOLPS5.0 treatment. Not
*      recommended unless wanting to recover older 5.0 results.
*
*      'b2sihs_istyle_Joule_heating' '1'          Physics
*      If b2sigp_style is set to '2', this switch has no effect.
*      When set to '1', there is no radial contribution in Joule heating
*      because there are no physical reasons to take it into account.
*      The value '0' corresponds to the old SOLPS5.0 treatment.
*
*      'b2sian_phm0'           '1.0'          Physics
*      Multiplier to the parallel momentum source term associated with the
*      anomalous current. It is recommended '1.0'. Only active if both the
*      ExB and diamagnetic drifts are turned on.
*      The value '0.0' corresponds to the old SOLPS5.0 treatment.
*
*      'b2sicf_phm0'           '1.0'          Physics
*      Multiplier of the centrifugal force term. It is recommended '1.0'.
*      The value '0.0' corresponds to the old SOLPS5.0 treatment.
*
*      'b2sicf_phm1'           '1.0'          Physics
*      Multiplier of the centrifugal force correction term due to
*      linearization.
*      It is recommended '1.0'.
*      The value '0.0' corresponds to the old SOLPS5.0 treatment.
*
*      'b2tfhe_anomalous'      '1.0'          Physics
*      'b2tanml_anomalous'     '1.0'          Physics
*      Real parameter which determines anomalous current.
*      Both b2tfhe_anomalous and b2tanml_anomalous point to the same switch
*      (backward compatibility with SOLPS5.2): b2tfhe_anomalous is checked
*      first and only if it is not found is b2tanml_anomalous looked for.
*      If b2tfhe_anomalous is 0 then anomalous current is switched off
*      otherwise anomalous current is switched on, and the numerical value of
*      b2tfhe_anomalous acts as a multiplier to the anomalous electrical
*      conductivity and thermo-electric coefficients provided.
*
*      'b2news_ExB'            '0.0'          Physics
*      Real parameter which multiplies ExB flows. If b2news_ExB.eq.0 and
*      b2news_facExB_start.eq.0 then ExB flows are switched off. If
*      b2news_ExB is nonzero, then ExB flows are multiplied by that constant
*      throughout the run.
*      See also Run section on switches b2news_facExB... for more details. A
*      spatial fac_ExB profile is also possible, see Numerics section for
*      details.
*
*      'b2tiner_inert'         '1.0'          Physics
*      Real parameter which multiplies the ion inertial current.
*

```



```

*      'b2tfhe_dia_cur'          '1.0'          Physics
*      Real parameter which multiplies the diamagnetic current.
*
*      'b2tfhe_vdia_par'        '1.0'          Physics
*      Real parameter which multiplies the convective heat flux due to grad
*      B-drift of guiding centers in non-modified heat fluxes of electrons
*      and ions.
*
*      'b2tfrn_xvecrb'          '1.0'          Physics
*      Real parameter which multiplies the contribution from the ExB drift
*      velocity to the flows passed to Eirene.
*
*      'b2tfrn_xwdia'           '1.0'          Physics
*      Real parameter which multiplies the contribution from the diamagnetic
*      drift velocity to the flows passed to Eirene.
*
*      'b2tfhe_neutral'         '0.0'          Physics
*      Real parameter which multiplies the ion-neutral current.
*      If b2tfhe_neutral is 0 then the ion-neutral current is switched off
*      otherwise the ion-neutral current is switched on.
*      The ion-neutral current also requires either the diamagnetic or ExB
*      drifts to be turned on as well.
*
*      'b2tinnt_fchin_in_core'   '0'           Physics
*      Integer switch to turn off or on the ion-neutral current in the core
*      region (applies to coupled runs only). This is recommended in cases
*      where the neutral densities are very low in the core region and the
*      ion-neutral current is likely to vary widely from one iteration to the
*      next as a result of Monte Carlo noise.
*      If fchin_in_core.eq.0 (default), then fchin is set to zero in the
*      core.
*      If fchin_in_core.eq.1, then fchin is unchanged.
*
*      'b2tfhe_PSch'            '1.0'          Physics
*      Real parameter which multiplies the Pfirsch-Schlueter electron heat
*      flux and conductivity.
*
*      'b2tfhe_vis_par'         '1.0'          Physics
*      Real parameter which multiplies the current driven by parallel
*      viscosity.
*      If b2tfhe_vis_par is 0 then the viscosity-driven current is switched
*      off otherwise the viscosity-driven current is switched on.
*
*      'b2tfhe_vis_q'           '1.0'          Physics
*      Real parameter which multiplies the current driven by heat viscosity
*      effects.
*
*      'b2tfhe_stochastic'       '0.0'          Physics
*      Real parameter which turns on stochastic current.
*      If b2tfhe_stochastic is 0 then stochastic current is switched off
*      otherwise stochastic current is switched on.
*
*      'b2tstch_delta'           '0.0'          Physics
*      Width of the stochastic current layer (in meters), measured from the
*      separatrix inward, along the poloidal index ixref (given by
*      b2tqna_ixref).
*      If b2tfhe_stochastic.ne.0, then b2tstch_delta must be greater than
*      zero.
*
*      'b2tstch_sig'            '1.0'          Physics
*      Multiplier to the magnetic field line stochastic diffusion
*      coefficient, describing the stochastic conductivity.
*
*      'b2trno_con_e_stochastic' '1.0'          Physics
*      Multiplier to the stochastic conductivity.
*
*      'b2trcl_lluciani'        '3'           Physics

```

```

*   If lluciani.ne.0, then transport coefficients on cells belonging to
*   closed field lines are modified according to the Luciani model.
*   If lluciani.eq.1, the standard connection length formulation is used.
*   If lluciani.eq.2, the old SOLPS4.0 formulation is used for backward
*   compatibility.
*   If lluciani.eq.3, Spb's new form Luciani's coefficient.
*
*   'b2trcl_lthf21'           '0'           Physics
*   If lthf21.ne.0, then the heat transfer model is modified according to
*   the 21-moment approach. This should only be used for multi-fluid runs.
*
*   'b2trcl_lvis21'           '0'           Physics
*   If lvis21.ne.0, then the viscosities are modified according to the
*   21-moment approach. This should only be used for multi-fluid runs.
*   This switch should not be used along with 'b2tfhe_vis_par' to avoid
*   double-counting of classical viscosity effects.
*
*   'b2sqel_artificial_radiation' '0.0'       Physics
*   If art_rad.ne.0, then an artificial radiation loss term is added to
*   the electron cooling rate. Art_rad represents the percent fraction of
*   impurities in the plasma.
*
*   'b2stbc_fheycore'         '0.0'           Physics
*   'b2stbc_fhiycore'         '0.0'           Physics
*   'b2stbc_fhiycore_kinetic_energy' '0'       Physics
*   'b2stbc_fchycore'         '-1.0e30'       Physics
*   'b2stbc_fnaycore'         '-1.0e30'       Physics
*   'b2stbc_isfeedback'       '0'           Physics
*   'b2stbc_iyped'            jsep/2          Physics
*   'b2stbc_ndes'             '0.0'           Physics
*   'b2stbc_ndes_sol'         '0.0'           Physics
*   'b2stbc_nepedm_sol'       '0.0'           Physics
*   'b2stbc_nesepm'           '0.0'           Physics
*   'b2stbc_nesepm_overshoot' '0.0'           Physics
*   'b2stbc_nesepm_pfr'       '0.0'           Physics
*   'b2stbc_nesepm_sol'       '0.0'           Physics
*   'b2stbc_private_flux_puff' '0.0'           Physics
*   'b2stbc_volrec'           '0.0'           Physics
*   'b2stbc_volrec_overshoot' '0.0'           Physics
*   'b2stbc_nesepm_minpuff'   '0.0'           Physics
*   'b2stbc_nesepm_maxpuff'   '0.0'           Physics
*   'eirene_nesepm_istra'     '-1'           Physics
*
*   The 20 switches above are all feedback switches and require that
*   b2stbc_feedback.ne.0 (See Run section for details) or that
*   lfeedback=.true. in b2.boundary.parameters. The feedback is done as a
*   boundary condition and is under-relaxed using the b2stbc_...._alpha
*   switches (See Numerics section for details).
*   fheycore is the radial electron heat flow entering the core boundary.
*   fhiycore is the radial ion heat flow entering the core boundary.
*   fchycore is the radial current entering the core boundary. fnaycore is
*   the radial flux of species "isfeedback" entering the core boundary.
*   If fhiycore_kinetic_energy.eq.1, the parallel kinetic energy flux is
*   counted as part of the ion energy flux.
*   For double-null cases, the f.ycore variables contain the total flow
*   through both core boundaries taken together. The core boundaries are
*   identified by the coreregn0 and coreregn2 switches (see Geometry
*   section).
*   The other seven switches are density feedback switches: nesepm is the
*   outer midplane separatrix electron density. (See b2mwti_jxa switch in
*   Geometry section for setting of midplane position). This is controlled
*   through particle input of main plasma species isfeedback through the
*   core boundary (See Geometry section for the core boundary indices
*   coreregn0 and coreregn2).
*   nesepm_sol is the outer midplane separatrix electron density, but this
*   time controlled through a gas puff outside the SOL region. (See
*   b2mwti_jxa switch in Geometry section for setting of midplane
*   position). This is done through particle input of neutrals species

```

```

* isfeedback through the SOL North boundary (See Geometry section for
* the SOL North boundary index solregno). If neseqm_overshoot.gt.1.0,
* the gas puff is turned off when the midplane separatrix density
* reaches neseqm*neseqm_overshoot. Minimum and maximum values of the
* feedback gas puff can be set using b2stbc_neseqm_minpuff and
* b2stbc_neseqm_maxpuff
* nepedn_sol is similar to neseqm_sol but the feedback is exerted on the
* electron density at the top of the pedestal, along the outer midplane,
* at radial position b2stbc_iyped. ndes_sol is the total particle
* content from the homonuclear sequence of species 'isfeedback' over the
* entire simulation domain.
* It is governed, like neseqm_sol, by neseqm_overshoot and neseqm_alpha
* and corresponds to a SOL boundary feedback. volrec_sol is the total
* particle source due to volume recombination coming from Eirene. Its
* feedback is governed by volrec_overshoot and volrec_alpha and
* volrec_beta parameters (See Numerics section for the latter two).
* neseqm_pfr is the outer midplane separatrix electron density, but this
* time controlled through a gas puff outside the PFR region. (See
* b2mwti_jxa switch in Geometry section for setting of midplane
* position). This is done through particle input of neutrals species
* isfeedback through the private flux boundaries (See Geometry section
* for the PFR South boundary indices pfrregno1 and pfrregno2).
* ndes is the total particle (neutrals + ions) content of the species
* whose neutral species has index isfeedback. This is controlled through
* particle input of neutrals species isfeedback through the
* private flux boundaries (See Geometry section for the private flux
* region boundary indices pfrregno1 and pfrregno2). private_flux_puff is
* the particle flow of species isfeedback through the private flux
* region boundary. This is controlled through particle input of neutrals
* species isfeedback through the private flux boundaries (See Geometry
* section for the private flux region boundary indices pfrregno1 and
* pfrregno2).
* All but the first of these feedback schemes ("neseqm") can also be run
* though an Eirene neutral gas puff. In that case, one should set an
* appropriate gas puffing stratum in the Eirene input file input.dat, as
* well as in the b2.neutrals.parameters file. The code will assume,
* unless told otherwise through the use of "eirene_neseqm_istra", that
* the first stratum of type 'C' found in b2.neutrals.parameters is the
* stratum used for the feedback. If "eirene_neseqm_istra" is set to a
* positive number, the code will consider the feedback stratum to be the
* stratum in the "eirene_neseqm_istra"-th position in
* b2.neutrals.parameters.
*
* 'b2stbc_type13_ref'          '1'                Physics
* 'b2stbc_type16_ref'          '1'                Physics
* 'b2stbc_type20_ref'          '1'                Physics
* 'b2stbc_type21_ref'          '1'                Physics
* 'b2stbc_type16_kinetic_energy' '0'              Physics
* These switches are associated with the boundary condition of type 13,
* 16, 17, 20 or 21 for particle flux, or electron and ion energy fluxes,
* i.e. constant density or temperature giving a prescribed flux through
* a radial surface, located type13_ref or type16_ref radial steps away
* from the boundary of the computational domain, respectively, and
* applied type20_ref steps away. See code and description in b2cdcn for
* details.
* Type 21 also applies to the electric potential boundary condition.
* It scales the feedback strength according to the value of the variable
* on the ring type21_ref steps inside.
* When type16_ref.gt.1, then the position where the core flux tallies
* are computed moves along with the boundary condition.
* If type16_kinetic_energy.eq.1, then the energy flux due to kinetic
* energy transport and viscous terms is included in the ion heat flux
* component of type 16 and 17 boundary conditions.
*
* 'b2stbc_bcene_16_style'      '0'                Physics
* If bcene_16_style.eq.1, the boundary condition is enforced using the
* modified fluxes fhe_mdf. Recommended when running cases with drifts.

```

```

*
* 'b2stbc_bceni_16_style'      '0'                      Physics
* If bceni_16_style.eq.1, the boundary condition is enforced using the
* modified fluxes fhi_mdf. Recommended when running cases with drifts.
*
* 'b2stbc_bc_ref'             '0.01'                   Physics
* Under-relaxation factor used for feedback scheme in boundary condition
* types BCCON = 16, 19, 22, 23, 26, and 27.
*
* 'b2stbc_bc_ref_te'          '0.01'                   Physics
* Under-relaxation factor used for feedback scheme in boundary condition
* type BCENE = 16 (with bcene_16_style.eq.1).
*
* 'b2stbc_bc_ref_ti'          '0.01'                   Physics
* Under-relaxation factor used for feedback scheme in boundary condition
* types BCENI = 16 (with bceni_16_style.eq.1), 24, and 27.
*
* 'b2stbc_secmodel'           '0'                      Physics
* If secmodel.eq.1, then the secondary electron emission coefficient at
* the plates is computed locally according to a kinetic model, otherwise
* the default values of cbsch(7,ireg) and/or gammae are used.
*
* 'b2stbr_plate_model'        '0'                      Physics
* 'b2stbr_plate_option'       '3'                      Physics
* 'b2stbr_sput_chem_model'     '0'                      Physics
* 'b2stbr_sput_chem_cutoff_alpha' '1.0'                Physics
* 'b2stbr_sput_chem_cutoff_beta' '3.0'                Physics
* 'b2stbr_sput_mixed_alpha'    '1.0'                Physics
* 'b2stbr_sput_mixed_beta'     '1.0'                Physics
* 'b2stbr_sput_phys_model'     '1'                      Physics
* 'b2stbr_sputter_energy_on'   '1'                      Physics
* 'b2stbr_sput_res'            '0.0'                   Physics
* 'b2stbr_therm_evap'          '0.0'                   Physics
* 'b2stbr_sput_dst'            '-1'                    Physics
* 'b2stbr_sput_dst2'           '-1'                    Physics
* 'b2stbr_sput_dst3'           '-1'                    Physics
* 'b2stbr_sput_frac_flag'      '0'                      Physics
* 'b2stbr_sput_frc'            '0.0'                   Physics
* 'b2stbr_sput_phys'           '0.0'                   Physics
* 'b2stbr_sput_src'            '1'                      Physics
* 'b2stbr_sput_phys_col'       '3'                      Physics
* 'b2stbr_alpha'               '0.25'                  Physics
* 'b2stbr_plate_temp'          '300.0'                 Physics
* 'b2stbr_plate_thick'         '0.00'                  Physics
* 'b2stbr_redep_alpha'         '0.00'                  Physics
* Sputtering model switches. See the code in b2stbr and the
* b2mod_sputter module for specific implementation details and
* references.
* The default values represent the case of Graphite plates. Sput_src is
* the atomic number of the plasma species which causes chemical
* sputtering.
* Sput_dst (when .ge.0) is the species index of the destination species
* produced by physical sputtering, chemical sputtering and/or RES. If
* either of the latter two processes are included, the production
* chemical sputtering and RES rate calculations provided in the code
* assume that sput_dst points to a Carbon species.
* Sput_dst2 and sput_dst3 (when .ge.0) represent other species produced
* by wall interactions for mixed materials scenarios.
* Sput_frac_flag is the switch to turn on mixed materials scenarios
* (when sput_frac_flag.eq.1).
* Plate_model.eq.0 means the 0-D time-independent plate heating model
* while plate_model.eq.1 indicates the 1-D time-dependent plate heating
* treatment. Plate_model.eq.2 gives acces to a 2-D time-dependent model.
* Plate_option chooses the initialisation of the plate temperature
* profile. If plate_option.eq.1, the profile is set to the constant
* given in plate_temp. If plate_option.eq.2, the profile is computed to
* be the 0-D equilibrium profile. If plate_option.eq.3, the profile is

```

```

* read from results of the previous run.
* Sput_phys_model is a switch for choosing between the TRIM tables
* (model 1, default) or an empirical formula (model 0). When TRIM data
* is not available, the empirical formula is automatically used.
* Extrapolations of low and high energy ranges beyond the TRIM table
* data is done using the same physical dependencies as the empirical
* formula. Sput_chem_model is a switch for the chemical sputtering model
* used.
* If sput_chem_model.eq.0 (default), the empirical formula is used.
* If sput_chem_model.eq.1, a constant with a low energy cut-off is used.
* The cutoff occurs at approximately sput_chem_cutoff_alpha and the
* width is determined by sput_chem_cutoff_beta (the larger the value,
* the narrower the width over which the transition from 0 to 1 occurs).
* For model 0, sput_frc is a multiplier to the empirical formula, while
* for model 1, sput_frc is the constant chemical sputtering yield.
* Sput_frc is superseded by the chem_sput array from
* b2.neutrals.namelist if the latter is used.
* For neutrals species, we add a factor of *na*vbar to the particle
* flux to the plate (used to compute chemical and RES sputtering), where
* vbar is the average neutral particle speed.
* Sput_phys turns on physical sputtering when sput_phys.gt.0.0 and
* contains a (real) multiplier for the physical sputtering rate looked
* up in the TRIM tables. It is superseded by the phys_sput array in
* b2.neutrals.parameters if the latter is used.
* sput_phys_col indicates which column to use in the TRIM table; the
* default (3) corresponds to 30 degrees incidence. The columns are for
* 0 15 30 45 55 65 75 80 85
* degrees. This angle is also used in the empirical formula. Plate_temp
* is the temperature (in Kelvin) of the COOLED end of the divertor
* plates. The surface temperature is computed self-consistently (using a
* 1-D description) according to the incident heat fluxes from the
* plasma. The plate is assumed to have a thickness of plate_thick,
* measured in meters. When plate_thick.eq.0 (default), the plate surface
* temperature is assumed to be the same as plate_temp. Sput_res turns on
* radiation enhanced sublimation (RES) when .gt.0.0 and is a multiplier
* to the RES rate.
* The switch sputter_energy_on turns on the energy contribution from the
* sputtered particles (physical and chemical). If
* sputter_energy_on.eq.0, the particles are sputtered cold, with no
* energy contribution to the ion heat equation.
* If sputter_energy_on.ne.0 (default), the chemically sputtered
* particles are re-introduced into the plasma with the thermal energy
* corresponding to the surface temperature of the material, while
* physically sputtered particles are re-introduced with the TRIM-
* calculated energy sputtering yield (if the TRIM model is used).
* Therm_evap turns on thermal evaporation when .gt.0.0 and is a
* multiplier to the thermal evaporation rate.
* Redep_alpha is the multiplier to the reduction of the sputtering yield
* due to prompt redeposition. The promptly redeposited species is
* counted as eroded-then-deposited species.
*
* 'b2stbr_refl_model'          '1'              Physics
* 'b2stbr_reflection_on'       '1'              Physics
* Reflection model switches. If the recycling coefficients (particle and
* energy) are set to zero, then the code attempts to use, for surfaces
* in contact with walls, the TRIM reflection data for particle and
* energy reflection coefficients, when refl_model.eq.1. If
* refl_model.eq.0, an empirical formula is used instead. The model
* switch is subservient to reflection_on.ne.0. If reflection_on.eq.0,
* then no reflection is used.
* If reflection_on.eq.1 (default), then the incident species is
* reflected unchanged. Otherwise, the reflected flux is of the
* associated neutral species to the incident particles.
*
* 'b2mndr_coronal_model'      '0'              Physics
* Turns on the coronal model if coronal.ne.0. I.e., the reaction rates
* of all reactions are made density-independent, and the value computed

```

```

* at the lowest density (given in b2ar.dat) is used.
*
* 'b2mndr_hz' '1.0' Physics
* hz has been introduced into the new form of the parallel momentum
* balance equation. If fac_hz = 0.0 then hz = 1 and old form of
* equations is used. If fac_hz = 1.0 then new form of equations is used.
*
* 'b2stbr_bas_recycled_neutrals_contr' '1.0' Physics
* Introduced for nulling recycling energy when it is zero.
*
* 'b2tfhe_alfTeEh' '0.0' Physics
* When set to '1.0', the old form of the electron heat flux calculation
* is used and b2tfhe_fch_pTe should be set to '0.0'. It is recommended
* to use 0.0.
*
* 'b2tfhe_fch_pTe' '1.0' Physics
* When set to '1.0', the new form of the electron heat flux calculation
* is used and b2tfhe_alfTeEh should be set to '0.0'. It is recommended
* to use 1.0.
*
* 'b2tfnb_xcur' '0.0' Physics
* Xcur is a multiplier to the parallel viscosity, perpendicular
* viscosity, heat viscosity, ion inertial and anomalous current
* contributions to the ion poloidal flows (particle and energy).
*
* 'b2tfnb_ycur' '1.0' Physics
* Ycur is a multiplier to the parallel viscosity, perpendicular
* viscosity, heat viscosity, ion inertial and anomalous current
* contributions to the ion radial flows (particle and energy).
*
* 'b2tfnb_vis_per' '1.0' Physics
* vis_per is a multiplier to the perpendicular viscosity current
* contributions to the ion poloidal flows (particle and energy).
* Subservient to b2tfnb_xcur and b2tfnb_ycur.
*
* 'b2tfnb_vis_q' '1.0' Physics
* vis_q is a multiplier to the heat viscosity current contributions to
* the ion poloidal flows (particle and energy). Subservient to
* b2tfnb_xcur and b2tfnb_ycur.
*
* 'b2tqce_fke_Zhdanov' '1' Physics
* When set to '1', the Zhdanov expression is used in the electron
* thermal conductivity. It is recommended to use 'b2tqce_fke_Zhdanov'
* '1'.
* This switch is only active if, simultaneously, one has
* b2tqce_model.eq.1 and b2tfhe_fch_pTe.eq.1.0.
*
* 'b2tqna_ixref' See description (integer) Physics
* Poloidal index (on the basis mesh) of the reference surface for the
* flux-scaled transport model. The default value is the outer midplane,
* computed as such (see b2mwti_jxa in Geometry section or
* set_transport_ixref below):
* Lower Single-null : ixref=rightcut(1)-(rightcut(1)-leftcut(1))/4
* Upper Single-null : ixref=leftcut(1)+(rightcut(1)-leftcut(1))/4
* Stellarator island : ixref=leftcut(1)+(rightcut(1)-leftcut(1))/2
* Double-null : ixref=(rightcut(1)+rightcut(2))/2
* Limiter geometry : ixref=nx/2
* Straight geometry : ixref=nx/2
* The flux-scaled transport model is used in conjunction with scaling
* factors listed in column (7) of the transport coefficients description
* in input files b2ah.dat and b2mn.dat.
*
* 'b2tqna_user_transport' '0' Physics
* 'set_transport_eta' '2.0' Physics
* 'set_transport_eta_alpha' '0.5' Physics
* 'set_transport_eta_floor' '0.1' Physics
* 'set_transport_eta_ceiling' '10.0' Physics

```

```

* 'set_transport_ixref'      See description (integer) Physics
* 'set_transport_iyref'      See description (integer) Physics
* 'set_transport_required_te_gradient' '5.0e4'      Physics
* The last switch is subservient to b2tqna_user_transport, and only used
* if user_transport.ne.0. When this is the case, the electron
* perpendicular heat diffusivity and anomalous perpendicular particle
* transport coefficient are scaled so as to maintain the
* required_te_gradient (in units of eV/m), at the location (ixref,iyref)
* on the basis mesh. The default position of the reference cell is on
* the outer midplane, slightly inside the separatrix, as follows:
* |Configuration|      ixref      |      iyref
* -----+-----+-----
* |Single-null | rightcut(1)- | 2*topcut(1)/3
* | (lower)    | (rightcut(1)-leftcut(1))/4 |
* |           | |           |
* |Single-null | leftcut(1)+  | 2*topcut(1)/3
* | (upper)    | (rightcut(1)-leftcut(1))/4 |
* |           | |           |
* |Stellarator | leftcut(1)+  | 2*topcut(1)/3
* | island     | (rightcut(1)-leftcut(1))/2 |
* |           | |           |
* |Double-null | (rightcut(1)+rightcut(2))/2 | 2*min(topcut(1),
* |           | |           | topcut(2))/3
* |           | |           |
* |Limiter     | nx/2         | 2*topcut(1)/3
* |           | |           |
* |Straight    | nx/2         | ny/2
*
* The transport_eta switches are activated when user_transport.eq.3 and
* allow modifications of the electron heat conductivity according to a
* constant eta (Grad Te / Grad Ne) model. See routine set_transport_eta
* code for details.
* A model for disruption transport coefficients is available with
* user_transport.eq.7. See routine set_transport_disruption code for
* details.
* The value user_transport.eq.8 activates the neoclassical model for
* computation of the transport coefficients in the core. See routine
* set_transport_neo and subroutines called for details. Implemented from
* the NEOART package.
* The location given by ixref and iyref can also serve as the anchor
* point for the transport and/or sources profiles provided by the user
* in the b2.transport.inputfile and b2.sources.profile files.
*
* 'b2tqna_max_df0'          '1e30'          Physics
* 'b2tqna_min_df0'          '0.0'           Physics
* Upper and lower limits of the diffusivity coefficients df0 computed in
* b2tqna for each neutral species.
*
* 'b2tqna_new_df0'          '0'             Physics
* When new_df0.eq.1, the neutral diffusivity is computed according to
* the local charge exchange, ionisation and elastic collision rates,
* instead of the standard form with constant cross-sections.
*
* 'b2tqna_ballooning'       '0.0'           Physics
* 'b2tqna_ballooning_rescale' '1.0'         Physics
* 'b2tqna_bb_ref'           See description (real) Physics
* Ballooning model switches. If ballooning.ne.0, then all transport
* coefficients in cell (i) are rescaled by a factor of
* ballooning_rescale*abs(bb_ref/bb(i))**ballooning .
* The default value for bb_ref is the arithmetic average of the total
* magnetic field strength over the entire computational domain.
*
* 'b2tqna_pfr_rescale'       '1.0'           Physics
* Scaling factor for all ion and electron transport coefficients inside
* private flux regions.
*
* 'b2tqna_divsol_rescale'    '1.0'           Physics

```

```

*   Scaling factor for all ion and electron transport coefficients inside
*   divertor SOL regions.
*
*   'b2sifr_phm0'           '1.0'           Physics
*   Multiplier of the friction term between charged species (only ions in
*   case b2sigp_style.eq.2).
*
*   'b2sifr_phm1'           '1.0'           Physics
*   If b2sigp_style.eq.2, multiplier to the friction force term between
*   electrons and ions.
*   Otherwise, multiplier of the ehxp term in the older expression for the
*   thermal force.
*
*   'b2sifr_phm2'           '1.0'           Physics
*   Multiplier of the electron thermal gradient term in the thermal force
*   term and parallel current
*
*   'b2sifr_phm3'           '1.0'           Physics
*   Multiplier of the ion thermal gradient term in the thermal force term.
*
*   'b2sifr_limthee'         '0.3'           Physics
*   'b2sifr_limthii'         '0.3'           Physics
*   'b2trcl_cthe'            '0.0'           Physics
*   'b2trcl_cthi'            '2.65'          Physics
*   Parameters for the computation of the thermal force term.
*
*   'b2trcl_lambda'          '-5.0'          Physics
*   'b2tlnl_ee'              '0'             Physics
*   'b2tlnl_ei'              '0'             Physics
*   'b2tlnl_ii'              '0'             Physics
*   If lambda is positive, the Coulomb logarithm is set to lambda.
*   If lambda is negative, then abs(lambda) is the lower bound to be used
*   when computing the Coulomb logarithm according to the NRL formulae.
*   The computation of the Coulomb logarithm takes place in b2tlnl.
*   The default formula is to use the Braginskii formulation, which will
*   give the same answer for electron-electron (ee), electron-ion (ei),
*   and ion-ion (ii) collisions. Respectively, non-default values of '1',
*   '2' and '3' for b2tlnl_ee, _ei, and _ii, respectively, make use of the
*   calculation according to Wesson.
*
*   'b2tfnb_PSch'            '1.0'           Physics
*   Multiplier to the Pfirsch-Schlueter flows.
*
*   'b2news_BoRiS'           '0.0'           Physics
*   The poloidal convective heat flux contains a prefactor of (3/2+BoRiS).
*   The default gives us the internal energy equation. The value BoRiS =
*   1.0 gives us the total energy equation.
*   Only for use to benchmark with codes using the total energy equation.
*   When used, the meaning of fhe and fhi changes from internal energy
*   fluxes to total energy fluxes.
*
*   'b2tfhe_conduction_only' '0'             Physics
*   When conduction_only.eq.1, convective heat transfer terms are turned
*   off. Only to be used when benchmarking against pure conduction cases.
*
*   'b2tfnb_alpha'           '0.0'           Physics
*   'b2tfnb_gamma'           '2.0'           Physics
*   'b2tfnb_flux_limit_min_ti' '0.0'          Physics
*   Parameters for the flux limit to the convective neutral flow.
*   Alpha is a multiplier to the classical flux limit value.
*   The larger is, the weaker the flux limit is.
*   is the exponent used in the flux-limiting formula.
*   The smaller is, the stronger the flux limit is.
*   If alpha.eq.0, no flux limit is applied.
*   flux_limit_min_ti specifies the minimum ti to be used (in eV).
*
*   'b2tlc0_alpha'           '1.0'           Physics

```



```

*      'b2tlc0_gamma'          '1.0'          Physics
*      Parameters for the flux limit to dpa0 - pressure driven neutral
*      diffusion.
*      Alpha is a multiplier to the classical flux limit value.
*      is the exponent used in the flux-limiting formula. If alpha.eq.0, no
*      flux limit is applied.
*
*      'b2tlh0_alpha'          '1.0'          Physics
*      'b2tlh0_gamma'          '1.0'          Physics
*      'b2tlh0_flux_limit_min_ti' '0.0'        Physics
*      Parameters for the flux limit to the heat conductivity of the
*      neutrals. Alpha is a multiplier to the classical flux limit value.
*      The larger is, the weaker the flux limit is.
*      is the exponent used in the flux-limiting formula.
*      The smaller is, the stronger the flux limit is.
*      If alpha.eq.0, no flux limit is applied.
*      flux_limit_min_ti specifies the minimum ti to be used (in eV).
*
*      'b2tlmv_style'          '1'            Physics
*      If style = 0 then the original form of the viscosity flux limit is
*      used, otherwise the SPb flux limit is used.
*
*      'b2sihs_phm0'           '1.0'          Physics
*      Multiplier of the contribution to electron heat sources from
*      divergence(ue,ve). This term is superseded by the BoRiS switch if
*      invoked.
*
*      'b2sihs_phm1'           '1.0'          Physics
*      Multiplier of the contribution to ion heat sources from
*      divergence(ua,va). This term is superseded by the BoRiS switch if
*      invoked.
*
*      'b2sihs_phm2'           '1.0'          Physics
*      Multiplier of the contribution to ion heat sources from viscous
*      heating due to poloidal velocity differences. This term is superseded
*      by the BoRiS switch if invoked.
*
*      'b2sihs_phm3'           '1.0'          Physics
*      Multiplier of the contribution to electron heat sources from Joule
*      heating (electron-ion friction).
*
*      'b2sihs_phm4'           '1.0'          Physics
*      Multiplier of the contribution to ion heat sources from atom-atom
*      friction. This term is superseded by the BoRiS switch if invoked.
*
*      'b2sihs_phm5'           '1.0'          Physics
*      Multiplier of the contribution to electron heat sources from
*      divergence of the electrical drift. That contribution is also
*      multiplied by facExB.
*
*      'b2sihs_phm6'           '1.0'          Physics
*      Multiplier of the contribution to ion heat sources from divergence of
*      the electrical drift. That contribution is also multiplied by facExB.
*
*      'b2sihs_phm7'           '0.0'          Physics
*      Multiplier of the contribution to heat sources from friction due to
*      diamagnetic velocities. Normally already included in 'phm3' term
*      above.
*
*      'b2sihs_phm8'           '1.0'          Physics
*      Multiplier of the contribution to heat sources from viscous heating
*      due to radial velocity differences. This term is only included when
*      using b2nph9_style.eq.1 or b2npht_style.eq.1 (defaults). This term is
*      superseded by the BoRiS switch if invoked.
*
*      'b2sdia_facgt'          '0.0'          Physics
*      Multiplier of the contribution to electron and ion heat sources from

```

```

* divergence of the gradTxB heat flux. That contribution is also
* multiplied by facdrift. This term is no longer needed as it has been
* transferred from being a source to being included in the flows.
*
* 'b2sral_style'          '2'          Physics
* When set to '0' or '2', the code calls the standard b2stbc routine,
* which uses the particle flux with drift terms included in the
* expression of the electron particle flux (fne). When set to '1', the
* code calls the b2stbc_spb routine instead, which uses the particle
* flux without drift terms in fne. It is recommended '2'.
*
* 'b2sqcx_styl0'          '0'          Physics
* 'b2sqcx_phm0'          '1.0'        Physics
* phm0 : Multiplier to the charge exchange rate coefficient. Subservient
* to styl0: only used if styl0.eq.0 or is.lt.2. If styl0.ne.0, the
* charge exchange rate coefficients for species of index 2 and above are
* set to zero.
*
* 'b2sqel_phm0'          '1.0'        Physics
* Multiplier to the ionisation rate coefficient.
*
* 'b2sqel_phm1'          '1.0'        Physics
* Multiplier to the recombination rate coefficient.
*
* 'b2sqel_phm2'          '1.0'        Physics
* Multiplier to the heat loss rate coefficient.
*
* 'b2stel_phm0'          '1.0'        Physics
* Multiplier of the sources due to atomic physics.
*
* 'b2tfhe_lim_flux'      '0'          Physics
* If '0', flux limit is not applied directly to the electron heat flux
* but is applied through transport coefficients if
* 'b2trcl_conductive_limit' is '1'. It is recommended '0'.
*
* 'b2tfhi_lim_flux'      '0'          Physics
* If '0', flux limit is not applied directly to the ion heat flux but is
* applied through transport coefficients if 'b2trcl_conductive_limit' is
* '1'. It is recommended '0'.
*
* 'b2treq_phm0'          '1.0'        Physics
* Multiplier to the temperature equipartition term.
*
* 'b2tqca_phm0'          '1.0'        Physics
* Multiplier for the classical parallel viscosity.
*
* 'b2tqca_model'         '1'          Physics
* If model.eq.1, use the Balescu formulation from SOLPS5.2 classical
* parallel ion heat diffusivity.
* If model.eq.2, use the older Braginskii SOLPS4.0 model.
* Note: old option model.eq.3 removed, replaced with model.eq.1., but
* numerical treatment w.r.t. factor 4/3 according to old model.eq.3.
*
* 'b2tqce_model'         '1'          Physics
* If model.eq.1, use the fitted Balescu formulation from SOLPS5.0
* classical parallel electron heat diffusivity.
* If model.eq.2, use the older Braginskii SOLPS4.0 model.
* If model.eq.3, use the 21-moment Balescu results.
*
* 'b2tqna_model_sig'     '0'          Physics
* If '1', use constant density in core region to calculate anomalous
* conductivity sig0=dfsig*qe*ne(nmdpl,-1), nmdpl - number of midplane
* cell. If '0', use sig0=dfsig*qe*ne(x,y)
*
* 'b2tral_cvsa_style'    '1'          Physics
* If '1', only the parallel ion classical contribution to the viscosity
* is multiplied by 4/3. If '0', all components (ions and neutrals,

```

```

* parallel and perpendicular) are multiplied by 4/3.
*
* 'b2trno_csig_an_style'      '1'                      Physics
* If '1', the anomalous contributions in the parallel direction to the
* electrical conductivity csig and the thermo-electric coefficient calf
* are zeroed out.
*
* 'b2tqin_csign_style'       '1'                      Physics
* If '1', the ion-neutral conductivity is computed from hydrogen gas
* diffusivity (SOLPS5.2 treatment). If '0', it is computed from the
* local CX rate (SOLPS5.0/5.1 treatments).
*
* 'b2trno_pol_anom_scale'    '1.0'                    Physics
* If pol_anom_scale.ne.1, the poloidal anomalous transport coefficients
* are scaled by the number. To get the same treatment as setting the old
* switch (radial_only) to 1, set the new switch to 0.0.
* This multiplication is to only take place for charged species.
*
* 'eirene_lhalpha'           '1'                      Physics
* 'eirene_lvib'              '1'                      Physics
* Deprecated. Use "DEFINE_LINES" in block 12 of Eirene input file
* instead.
*
* 'eirene_repeat_first_call' '1'                      Physics
* If > 0 then repeats the first call to Eirene in eirene_mc so many
* times.
* Useful if a density feedback gas puff is being used because the flux
* passed to Eirene on its first call is ignored (instead the value in
* the Eirene input file is used).
*
* 'eirene_use_recyceir'      '1'                      Physics
* If > 0 use recyceir (non species dependent) to specify the recycling
* coefficients, else if 0 use recyc (species dependent).
*
* 'eirene_ionising_core'     '0'                      Physics
* If <> 0 then recycles the neutral flux having crossed the core
* boundary within Eirene as ions. The recycling is modulated as per the
* poloidal density distribution of the ions, and neutrals come back as
* fully-stripped ions.
* 'eirene_ionizing_core' is an alias for this switch.
* If the value = 1, then the flux is added by direct modification of the
* sources in the guard cells --- this will only work if a standard flux
* boundary condition is applied at that boundary.
* If the value is < 0, then the absolute value specifies which boundary
* in b2.boundary.parameters is to be used. This will only work for the
* type 13 boundary condition.
*
* 'eirene_background'       '1'                      Physics
* If eirene_background.eq.0, the ion velocities passed to Eirene to be
* used for the collisions are based on grad-B and ExB drifts (vadia +
* vaecrb).
* If eirene_background.eq.1, these velocities contain the full
* diamagnetic and ExB drifts (wadia + vaecrb).
* Note: recycling fluxes are always computed based on grad-B and ExB
* drifts only (and are not affected by this switch), because diamagnetic
* drift flows largely close within the sheath.
*
* 'eirene_sheath_pot'       '1'                      Physics
* If eirene_sheath_pot.eq.1, the sheath potential drop as computed by
* B2.5 (i.e. including effects of parallel currents, secondary electron
* emission, etc.) is passed to EIRENE to compute ion acceleration in the
* sheath.
* If eirene_sheath_pot.eq.0, the sheath potential drop is recomputed by
* EIRENE, usually assuming zero current and secondary electron emission.
*
* 'b2mndr_atomic_physics_rescale' '0'                Physics
* If atomic_physics_rescale is not 0, then the code will rescale the

```

```

* rates stored in b2frates according to the multipliers given in the
* b2.atomic_physics_rescale.parameters inputfile before making use of
* them.
*
* 'neoclassical_ic'          '3'                      Physics
* ..set the contribution ic in NEOART
* 0 --- classical particle flux
* 1 --- banana plateau contribution
* 2 --- Pfirsch-Schlueter contribution
* 3 --- both banana and PS
* 4 --- all contributions
* mind that B2 already calculates the classical transport !
* avoid double transport, 0+4 for cross-checks only !
*
* 1.4. Output switches
*
* 'b2mndr_b2time'          '1'                      Output
* Specifies the number of timesteps between writes of the b2time.nc
* time-dependent file. If b2time.gt.0, always writes out on the last
* timestep. If b2mndr_stim.lt.0, data from the current run is appended
* to the existing b2time.nc file, otherwise the file is overwritten.
*
* 'b2mndr_tally'           '1'                      Output
* Specifies the number of timesteps between writes of tallies. If
* tally.gt.0, always writes out on the last timestep.
*
* 'b2mndt_moitlev'         '-1'                     Output
* Controls detailed monitoring output. moitlev.eq.-1 means no output,
* moitlev.eq.0 means output once per timestep, moitlev.eq.1 means output
* once per nstg0 iteration, moitlev.eq.2 means output once per nstg1
* iteration and moitlev.eq.3 means output once per nstg2 iteration. See
* 'b2mndr_nstg?' description in Run section for more details.
*
* 'b2mndt_moqtlv'          '3'                      Output
* Controls quick trace output. moqtlv.eq.-1 means no output, moqtlv.eq.0
* means output once per timestep, moqtlv.eq.1 means output once per
* nstg0 iteration, moqtlv.eq.2 means output once per nstg1 iteration and
* moqtlv.eq.3 means output once per nstg2 iteration. See 'b2mndr_nstg?'
* description in Run section for more details.
*
* 'b2mndr_mvnum'           '0'                      Output
* Specifies the maximum number of instances at which movie data will be
* output.
*
* 'b2mndr_mvinc'           '1'                      Output
* Specifies the number of timesteps between movie output. If greater
* than 1 and the last timestep does not correspond to a planned frame,
* an extra endstate frame is added.
*
* 'b2mndr_plasnum'         '0'                      Output
* Specifies the maximum number of instances at which extra writes of
* b2fplasmf.xxxx will occur.
*
* 'b2mndr_plasinc'         '1'                      Output
* Specifies the number of timesteps between b2fplasmf.xxxx writes.
*
* 'b2mndr_cdfmovietim'     '0.0'                   Output
* Another option for movie output. Give the real-time interval between
* movie frames.
*
* 'b2mndr_ntim_save'       '0'                      Output
* Another option for plasma state file output. Give the number of B2.5
* full iterations between successive write-ups of plasmastate files, in
* addition to the ones written at regular CPU intervals. Should be used,
* at the exclusion of other plasmastate write-up frequency settings, in
* conjunction with the Leuven Monte Carlo averaging scheme.
*

```

```

*      'b2mndt_av'                '0'                Output
*      If b2mndt_av.gt.0, turns on computation of running averages.
*
*      'b2mndt_av_continue'        '1'                Output
*      If b2mndt_av_continue.gt.0, continuously perform running averages.
*
*      'b2mndt_av_ntim_batch'      '500'             Output
*      If ntim_batch.gt.0, number of iterations used to compute batch
*      averages, written out in 'b2batch.nc'.
*
*      'b2mndt_av_ntim_run'        '1000'            Output
*      If ntim_run.gt.0, number of iterations used for writing running
*      averages.
*
*      'b2mndt_av_batch_all'       '0'                Output
*      If b2mndt_av_batch_all.ge.0, produces standard output for batch
*      averages.
*      If b2mndt_av_batch_all.eq.1, entire 2D fields of batch averages and
*      variances for ne, te, ... are written. Expert use only, check
*      $SOLPSTOP/doc/SOLPS-ITER_Eirene_averaging.pdf.
*
*      'b2mndr_plasmatim'          '0.0'             Output
*      Another option for plasma state file output. Give the real-time
*      interval between successive write-ups of plasmastate files, in addition
*      to the ones written at regular CPU intervals.
*
*      'b2wdat_iout'               '0'                Output
*      If iout.eq.1, a large set of *.dat output files will be produced
*      containing the values of a variety of code quantities, evaluated at
*      the end of the B2.5 iteration.
*      If iout.eq.4, an even larger set of output files will be produced, for
*      the purposes of a full run analysis, evaluated in the individual
*      routines where the quantities are used.
*      For even more detailed debugging analysis, one should instead use
*      "procedure_name"_iout.eq.1.
*      The files and their content are fully described in the
*      Output_description.pdf file in the SOLPSTOP/doc directory and Appendix
*      G of the SOLPS-ITER manual.
*
*      'get_residuals'             '0'                Output
*      If get_residuals.eq.1, then the residuals of unsolved equations will
*      be outputted into b2ftrace. Equations are unsolved if
*      'b2news_no_solve' is nonzero or any elements of the solveco, solvemo,
*      solveee, solveei, or solvepo are set to .false. .
*      By default, get_residuals.eq.0 and the residuals for unsolved
*      equations are set to zero.
*
*      'b2wdat_append'             '0'                Output
*      If append.eq.1, the *.dat output files are appended upon every write,
*      instead of being rewritten every time.
*
*      'my_out_digits'             6 or 15             Output
*      Specifies the number of significant digits with which the *.dat files
*      are written out. Defaults to 6 in normal mode and 15 in debug mode.
*      Must be positive.
*
*      'b2mndr_old_style'          '0'                Output
*      If old_style.gt.0, old-fashioned (SOLPS4 style) output is added at the
*      end of the b2mn.prt file.
*
*      'b2mndr_av_read'            '0'                Output
*      If b2mndr_av_read.gt.0, the initial plasma is the averaged solution
*      read from the b2faveri file. Check $SOLPSTOP/doc/SOLPS-
*      ITER_Eirene_averaging.pdf for details.
*
*      'b2mndr_na_eps'             '1.0e19'           Output
*      'b2mndr_po_eps'             '1.0e+1'           Output

```

```

*      'b2mndr_te_eps'           '1.0e+1'           Output
*      'b2mndr_ti_eps'           '1.0e+1'           Output
*      'b2mndr_ua_eps'           '1.0e+4'           Output
*      The five switches above are safeguards numbers for when printing
*      changes after a time step. The change is computed as:  $\Delta X =$ 
*       $\text{abs}((X(t)-X(t-1))/(X(t)+X_{\text{eps}}))$ 
*
*      'b2mndr_trantim'           '0.0'             Output
*      Produces a numbered 'tran' file every trantim real-time seconds. An
*      endstate file is written if it falls between scheduled write-up times.
*      Only available within the -DJET environment.
*
*      'b2mwti_target_offset'      '1'              Output
*      The diagnostic values from b2time.nc use guard cell values if
*      target_offset.eq.0, and values from the neighbouring real cell if
*      target_offset.eq.1. Fluxes are not affected.
*
*      'b2mwti_2dwrite'            '0'              Output
*      Controls additional output to b2time.nc. If eq 1 then a few 2d arrays
*      (ne, Te, Ti) are written with each write to b2time.nc. If eq 2 then
*      fluxes, electric potential, kinetic energy, and fluid particle and
*      energy source terms are also included (e.g., rsana, rsahi, rgrad).
*
*      'b2mwti_ismain0'            '0'              Output
*      Index of the species used to create the 'dp3d?.last10' diagnostic
*      files.
*      Defaults to the neutral species associated with ismain. If no such
*      species is declared, the default will be set to ismain. Is expected to
*      either correspond to a neutral species or to ismain.
*      Can also be used in b2ar.dat to specify the index of the main neutral
*      CX species when resorting to the AMNS database (starting from version
*      1.3.2).
*      If ismain is also defaulted, then will be 0.
*
*      'b2mwqt_style'              '1'              Output
*      Specifies the amount of data that is written out to b2ftrace. See the
*      manual (Section on b2yq) for full details.
*
*      'tallies_netcdf'             '0'              Output
*      'b2stbr_b2wall_netcdf'       '0'              Output
*      'balance_netcdf'             '0'              Output
*      'balance_average'            '0'              Output
*      'b2mndr_cdf_default'          '0'              Output
*      If tallies_netcdf.ne.0, the file 'b2tallies.nc' is created, which
*      contains the regional tallies in CDF format. If b2mndr_stim.lt.0, data
*      from the current run is appended to the existing b2tallies.nc file,
*      otherwise the file is overwritten.
*      If b2mndt_av_ntim_batch.gt.0, the file 'b2batch.nc' is created, which
*      contains the batch averages of the tallies.
*      If b2wall_netcdf.ne.0, the file 'b2wall.nc' is created, which contains
*      the wall tallies in CDF format [written every b2wall_netcdf main
*      calls]. If b2mndr_stim.lt.0, data from the current run is appended to
*      the existing b2wall.nc file, otherwise the file is overwritten.
*      If balance_netcdf.ne.0, the file 'balance.nc' is created, which
*      contains all of the arrays required by the balance post-processing
*      routines, in CDF format.
*      If balance_average.ne.0, the balance arrays are averaged over all
*      b2mndr_ntim timesteps.
*      The NetCDF files are written using the format selected by means of the
*      b2mndr_cdf_default switch. The possible formats are:
*      0: local NetCDF library default
*      1: NetCDF classic format
*      2: 64 bit format
*      3: NetCDF4 format
*      4: NetCDF4 classic format
*      5: 64 bit data format.
*      Format availability depends on local NetCDF library implementation.

```

```

*
*      'ank_tracing'          '0'          Output
*      If ank_tracing.ge.1, additional tracing output from Andrei Kukushkin
*      is produced, in files to be found in the tracing/ directory inside the
*      run directory. The traces will be written every ank_tracing iteration.
*      If b2mndr_stim.lt.0, data from the current run is appended to the
*      existing files, otherwise the files are overwritten.
*
*      'b2stbc_diagno'        '0'          Output
*      Controls level of output in b2stbc and subservient routines.
*      Level 1 (diagno.ge.1) output includes wrong_flow warnings, transport
*      coefficients at the (nx/2,ny) basis mesh position, the solution found
*      for the given heat flux at constant temperature bc, and all feedback
*      diagnostics.
*      Level 2 output (diagno.ge.2) includes the integral current correction,
*      the boundary sources on the first call, excessive sound speeds, and
*      the iterative progress to the given heat flux at constant temperature
*      bc.
*      Level 3 output (diagno.ge.3) includes the boundary sources on each
*      call, and the boundary.prt file listing all boundary cells.
*
*      'b2stbr_output'        '0'          Output
*      Output flag for the first_flight model. If output.ge.1, the files
*      start, end, and fort.88 are created, which contain information about
*      the chords used in the model.
*
*      'b2npmo_vlct_diagno'    '1'          Output
*      Output flag to control diagnostics related to use of ion_vlct_restrict
*      switch.
*      If vlct_diagno.eq.0, the velocity restrictions are applied silently.
*      If vlct_diagno.eq.1 (default), the user only gets a count of how many
*      times the velocity restriction has been applied for each species, if
*      any.
*      If vlct_diagno.eq.2, the user gets the full details of where and how
*      large the applied velocity restriction was, if any.
*
*      'eirene_savef30'        '0'          Output
*      'eirene_savef31'        '0'          Output
*      For coupled runs, the code will create two files named fort.30 and
*      fort.31, containing the plasma geometry and plasma state,
*      respectively, for use by Eirene. This happens irrespective of the
*      setting of the two switches above. If however, the user wishes to
*      obtain these files for a non-coupled run, then these two switches can
*      be set to 1 and the files will then be produced at the end of the run.
*      If savef31.lt.0, then a fort.31.##### file is produced after every
*      B2.5 time step. Additionally, if savef31.ne.0, the fort.31 file will
*      be updated before every Eirene call for coupled runs.
*
*      'b2mndr_inverse_ua'     '0'          Output
*      If inverse_ua.eq.1, the code will produce a 'b2fstati_with-ua' file
*      that contains the same plasma information, but with the opposite sign
*      convention for the parallel velocity. The run will then proceed with
*      the new velocities with their sign inverted.
*
*      'b2yrdr_ns'             ns          Output
*      New number of species desired when reading a new atomic rates file
*      using b2yr.exe. Must be specified within b2yr.dat.
*      To be used for checking atomic rates after bundling and before any
*      plasma state files have been created or converted, i.e. when the
*      number of species in the atomic rates file b2frates does not match the
*      number of species declared in the run parameters file b2fpardf.
*
*      'b2srsm_diagno'         '0'          Output
*      Controls level of output in b2srsm. If diagno.ge.2, output will be
*      given on every call. If diagno.eq.1, output will be given on main
*      calls only.
*

```

```

*      'b2ux5p_cpu'           '0'           Output
*      'b2ux7p_cpu'           '0'           Output
*      'b2ux9p_cpu'           '0'           Output
*      'b2ux11p_cpu'          '0'           Output
*      If cpu.gt.0, prints out the time spent in the matrix solver.
*
*      'b2ux5p_nltrsol'        '2'           Output
*      'b2ux7p_nltrsol'        '2'           Output
*      'b2ux9p_nltrsol'        '2'           Output
*      'b2ux11p_nltrsol'       '2'           Output
*      Output flag for the matrix solvers. Larger numbers mean increasing
*      output level.
*
*      'eirene_mc_output_style' '1'           Output
*      If nonzero, prints sources computed by Eirene. If greater than 1,
*      prints them on every use of the recycling sources, not just when they
*      are computed.
*
*      'ma28_nwrite'           '0'           Output
*      If nwrite.gt.0, prints the content of the sparse matrix in the
*      b2_matrix file, for the first nwrite calls.
*
*      'b2news_ncallout'       '-1'          Output
*      If the iteration number is equal to ncallout, then several output
*      files 'b2ne_npmo', 'b2ne_xppb', 'b2ne_npco', 'b2ne+nppo' which detail
*      the convergence behaviour and residuals.
*
*      'b2tqna_diagno'          '0'           Output
*      If diagno.gt.0, outputs details of the calculations of neutral
*      transport coefficients when using the new_df0 model. See switch
*      b2tqna_new_df0 for more details.
*
*      'b2mndr_idout0'          'pgnl;pgmm;pzmm' Output
*      'b2mndr_idout1'          'pzmm'       Output
*      idout0 and idout1 specify the desired selection of output segments;
*      idout0 for output before the calculation and idout1 for output at the
*      conclusion of the calculation. The generic description of either of
*      these 'idout' variables follows. idout specifies the desired selection
*      of output segments for printed and graphical output. Associated with
*      each segment is a four-character word, which is identified on the
*      printed output. (The same word may be associated with several
*      segments.) A particular segment will be produced only if its
*      associated word has a match, as defined by the routine strmas, in
*      idout.
*
*      'eirene_format'          'iter'        Output
*      This switch is to be provided within b2yt.dat to indicate to the b2yt
*      converter the format of the original Eirene input.dat file to be
*      modified. The accepted values are (case-insensitive):
*      'old' for input files from SOLPS4.0 and SOLPS5.0 runs using 'old'
*      Eirene_96
*      'new' for input files from SOLPS4.0 and SOLPS5.0 runs using 'new'
*      Eirene_99
*      'facelift' for input files from SOLPS5.1 runs
*      'juelich' for input files from Juelich Eirene versions (2008 and
*      younger)
*      'iter' for input files from SOLPS4.2/4.3 and SOLPS-ITER runs
*
*      'solps_version'          'iter'        Output
*      This switch is to be provided within b2yt.dat to indicate to the b2yt
*      converter the format of the original fort.44 file to be modified. The
*      accepted values are (case-insensitive):
*      '4.3' for a fort.44 file produced from SOLPS4.3 runs
*      '5.0' for a fort.44 file produced from SOLPS5.0 runs
*      '5.1' for a fort.44 file produced from SOLPS5.1 runs
*      '5.2' for a fort.44 file produced from SOLPS5.2 runs
*      'iter' for input files from SOLPS-ITER runs (no conversion necessary)

```



```

*
* 'ids_from_43'          '0'          Output
*   This switch is to be used when running the b2_uual_write utility
*   program to obtain an IDS from the SOLPS run results.
*   If set to 1, the IDS will indicate that the results come from
*   a converted SOLPS4.3 run with no additional calculation. By default,
*   the IDS will indicate a SOLPS-ITER run.
*
* 1.5. Numerics switches
*
* 'b2news_potit'          '50'          Numerics
* 'b2news_potitmin'        '0'          Numerics
*   Maximum and minimum number of iterations in the potential equation. It
*   must hold that potitmin < potit.
*
* 'b2news_potok'          '1.0e-2'      Numerics
*   Target residual for the potential equation.
*
* 'b2news_ramp_slow'       '0'          Numerics
*   If ramp_slow.eq.0 (default), the ExB and diamagnetic drifts
*   multipliers are ramped on each call of b2news, otherwise they are only
*   changed on the first call of the nstg loop on the timestep.
*
* 'b2mndr_min_areshe'      '0.0'        Numerics
* 'b2mndr_min_areshi'      '0.0'        Numerics
* 'b2mndr_min_aresco'      '0.0'        Numerics
*   Minimum residuals for stopping a run. All criteria must be met
*   simultaneously for the run to stop. The continuity equation residual
*   applies to the "ismain" species (See "Run" section).
*
* 'b2mndt_nstg_areshe'     '0.0'        Numerics
* 'b2mndt_nstg_areshi'     '0.0'        Numerics
* 'b2mndt_nstg_aresco'     '0.0'        Numerics
*   Minimum residuals for an internal solution loop to stop.
*   Used in conjunction with (not in replacement of) the nstg(0:2)
*   numbers.
*   All nonzero criteria must be met simultaneously. The continuity
*   equation residual applies to the 'ismain' species (See 'Run' section).
*
* 'b2news_nsmin'           '0'          Numerics
* 'b2news_nsmax'           ns          Numerics
*   Allows for solving only for the species range [nsmin:nsmax-1].
*
* 'eirene_na_max'          '1e30'       Numerics
* 'eirene_na_min'          '0.0'        Numerics
* 'eirene_te_max'          '1e30'       Numerics
* 'eirene_te_min'          '0.0'        Numerics
* 'eirene_ti_max'          '1e30'       Numerics
* 'eirene_ti_min'          '0.0'        Numerics
* 'eirene_ua_max'          '+c'         Numerics
* 'eirene_ua_min'          '-c'         Numerics
*   Upper and lower bounds used when writing out data for Eirene.
*
* 'eirene_print_minmax'     '0'          Numerics
*   Print min and max values of te, ti, na & ua
*
* 'eirene_extrap'          '1'          Numerics
*   If eirene_extrap.eq.1, then guard cell plasma parameter values are
*   calculated from the neighbouring real cell values. If
*   eirene_extrap.eq.0, the guard cell values are used unchanged.
*
* 'eirene_ank_mods'         '0'          Numerics
*   If ank_mods.ne.0, then uses an additional scheme to ensure particle
*   balance as the B2 solution evolves, due to the internal iteration
*   scheme, away from the plasma background on which the Eirene sources
*   were originally computed at the beginning of the time step. The user
*   is referred to the text in $SOLPSTOP/doc/Source_Scaling_in_B2.pdf for

```

```

* a full description of the method used.
*
* 'eirene_dpc_fix'          '1'                      Numerics
* If dpc_fix.eq.1, then uses the true particle source from the stratum.
* If dpc_fix.eq.2, sets this particle source to zero. The equivalent of
* the old behaviour is dpc_fix.eq.0 and is wrong!
*
* 'b2stbr_sna_corr'        '0.0'                    Numerics
* Multiplier to the neutral source differential between puffing and
* pumping strata for code speed-up.
* See E. Kaveeva et al., Nucl. Fusion 58 (2018) 126018 for details.
*
* 'b2stbr_eir_src_nhist'   '1'                      Numerics
* If b2stbr_eir_src_nhist.gt.1, then Eirene sources are accumulated and
* a moving average is computed. Note that only one of
* b2stbr_eir_src_nhist, eirene_neutr_avg, or eirene_underrelax may be
* nonzero.
*
* 'eirene_neutr_avg'       '0'                      Numerics
* If eirene_neutr_avg.gt.0, then Eirene sources are accumulated and
* averaged until eirene_neutr_avg+1 steps, at which point the averaging
* process is reset. Note that only one of b2stbr_eir_src_nhist,
* eirene_neutr_avg, or eirene_underrelax may be nonzero.
*
* 'eirene_underrelax'      '0'                      Numerics
* If eirene_underrelax.gt.0, then an underrelaxation scheme is used for
* the Eirene sources, with an underrelaxation ratio of
* 1/eirene_underrelax. Note that only one of b2stbr_eir_src_nhist,
* eirene_neutr_avg, or eirene_underrelax may be nonzero.
*
* 'eirene_uub_style'       '0'                      Numerics
* If eirene_uub_style.eq.0, compute poloidal velocities passed to Eirene
* from ua, vadia, wadia, vaecrb arrays. If eirene_uub_style.eq.1,
* compute poloidal velocities passed to Eirene from fna and fna_eir
* fluxes.
*
* 'b2news_area_fix'        '3'                      Numerics
* When area_fix.eq.0, recovers old SOLPS5.0 behaviour.
* If area_fix.ge.1, then certain computations of poloidal velocities are
* done by dividing flows by the area normal to the flux tube as opposed
* to the area of contact.
* If area_fix.ge.2, then additionally this treatment is used for the
* parallel contact area used in determining plasma flux limiters.
* If area_fix.ge.3, then additionally this treatment is used for all
* parallel contact areas.
*
* 'b2nppo_ion_vlct_restrict' '0'                    Numerics
* If ion_vlct_restrict.ne.0, the parallel speed of ions is limited to be
* no more than 3 times the global plasma sound speed. In addition, if
* ion_vlct_restrict.eq.2, the ExB ion speeds are limited to be no more
* than twice the local ion sound speed.
*
* 'b2tfhe_vis_per'        '1.0'                    Numerics
* vis_per is a multiplier to the perpendicular viscosity current.
* If vis_per.ne.0.0, the electric potential equation is solved by the
* subroutine b2npp7 using a 7-point stencil.
* The value '1.0' is recommended for runs with drifts when perpendicular
* viscosity and corresponding perpendicular viscosity current is taken
* into account.
*
* 'b2stbc_sheath_drift_fix' '1'                    Numerics
* If sheath_drift_fix.eq.0, then the drift velocity used in the sheath
* boundary conditions is the sum of diamagnetic and ExB contributions
* (old, but likely wrong, treatment). If sheath_drift_fix.eq.1, then the
* drift velocity used in the sheath boundary conditions is the ExB
* velocity only (recommended).
*

```

```

*      'b2stbc_fix_fch_in_fhe_sheath' '2'                      Numerics
*      If fix_fch_in_fhe_sheath.eq.0, then the wrong (old) implementation of
*      the FCH contribution to FHE used;
*      If fix_fch_in_fhe_sheath.eq.1, then the wrong (second) implementation
*      of the FCH contribution to FHE used;
*      If fix_fch_in_fhe_sheath.eq.2, then the correct implementation of the
*      FCH contribution to FHE used;
*
*      'b2stbr_potential_at_guard_cell' '1'                    Numerics
*      If potential_at_guard_cell.eq.1, the electric potential used for the
*      computation of the incident energy for sputtering yields is taken as
*      the value in the guard cell. If potential_at_guard_cell.eq.0, the
*      value from the neighbouring real cell is used instead.
*
*      'b2trcl_conductive_limit' '1'                            Numerics
*      When set to '1', flux limit of the parallel electron and ion heat
*      fluxes is applied to transport coefficients.
*      It is recommended '1'. If 'b2trcl_conductive_limit' '0' then the keys
*      'b2tfhe_lim_flux' and 'b2tfhi_lim_flux' must be '0'.
*
*      'b2trcl_core_cond_limit' '0'                             Numerics
*      If core_cond_limit.eq.0, then the heat flux limits due to chvmx and
*      chvimx are not applied in the core. Switch only active if
*      'b2trcl_conductive_limit' .ne. 0.
*
*      'b2tlh0_flux_limit_style' '2'                            Numerics
*      If '0', use the SOLPS5.0 scheme for neutral heat conductivity flux
*      limits.
*      If '1', SPB form to calculate parallel and perpendicular neutral heat
*      conductivity flux limit (does not preserve symmetry, kept for backward
*      compatibility reasons).
*      If '2', modifies the SPB treatment for the flux limits to be applied
*      on the transport coefficients directly.
*      It is recommended '2'.
*
*      'b2tral_mode' '1'                                         Numerics
*      Switch to choose between various interpolation schemes for transport
*      coefficients (Does not apply to pinch velocities vla or to the
*      temperature-driven electric conductivity alf).
*      mode.eq.-1: harmonic averaging
*      mode.eq.0 : geometric averaging
*      mode.eq.1 : arithmetic averaging (SOLPS5.0 formulation)
*      mode.eq.2 : arithmetic averaging (SOLPS4.0 formulation)
*
*      'b2tfhe_no_hybr' '0'                                     Numerics
*      'b2tfhi_no_hybr' '0'                                     Numerics
*      'b2tfnb_no_hybr' '0'                                     Numerics
*      'b2tfhe_hybr2' '0'                                       Numerics
*      'b2tfhi_hybr2' '0'                                       Numerics
*      Switches to toggle between the standard 5.0 hybrid scheme
*      (no_hybr.eq.0) and the old 4.0 upwind scheme (no_hybr.eq.1).
*      These apply separately to the electron heat, ion heat and particle
*      conservation equations.
*      If hybr2.ne.0, then a mixed scheme is used where only the conductivity
*      is computed according to the 4.0 scheme.
*
*      'b2tfhe_upwind' '0'                                       Numerics
*      'b2tfhi_upwind' '0'                                       Numerics
*      If upwind.ne.0, then the SOLPS4 upwind scheme in b2tfhe_ and b2tfhi_
*      to obtain the heat fluxes.
*
*      'b2tfnb_pflux_cor' '1'                                     Numerics
*      If pflux_cor.eq.1 and fna_mdf is used, enforces that the integral
*      particle flux across the domain boundaries computed from fna is equal
*      to the same integral computed from the fna_mdf fluxes.
*
*      'b2trcl_cvsa_mltpl' '1.0'                                 Numerics

```

```

* Artificial coefficient providing faster convergence to the
* neoclassical electric field but giving a distortion of the flows in
* the SOL as a side-effect.
* Can be applied <>1 during the convergence and turned off for the final
* stage of calculations. Use with caution.
*
* 'b2trcl_csig_mltpl'          '1.0'          Numerics
* Multiplier to the electrical conductivity between guard cells along
* the core boundary. A large value helps to damp numerical instabilities
* in the potential when drifts are turned on and the electron density
* and temperature are kept constant along the core boundary.
*
* 'b2ux5p_mult_nonzero'       '10'           Numerics
* 'b2ux7p_mult_nonzero'       '10'           Numerics
* 'b2ux9p_mult_nonzero'       '10'           Numerics
* 'b2ux11p_mult_nonzero'      '10'           Numerics
* Number of expected nonzero matrix elements per matrix row.
*
* 'b2ux5p_mult_solvdim'       '15'           Numerics
* 'b2ux7p_mult_solvdim'       '15'           Numerics
* 'b2ux9p_mult_solvdim'       '15'           Numerics
* 'b2ux11p_mult_solvdim'      '15'           Numerics
* 'b2ux5p_mult_solvdim1'      '10'           Numerics
* 'b2ux7p_mult_solvdim1'      '10'           Numerics
* 'b2ux9p_mult_solvdim1'      '10'           Numerics
* 'b2ux11p_mult_solvdim1'     '10'           Numerics
* Multipliers to the number of nonzero elements in the solution matrix
* for workspace arrays in the matrix solver.
*
* 'b2ux5p_style'              '2'            Numerics
* Choose the type of 5-point stencil matrix solver. Style.eq.0 = iluter,
* Style.eq.1 = slv5pt, Style.eq.2 = MA28.
* NOTE: Only style.eq.2 will give good results. Other values are NOT
* recommended! style.eq.0 and style.eq.1 are only applicable to linear
* geometries with no cuts and no isolated regions.
*
* 'b2ux7p_style'              '2'            Numerics
* Choose the type of 7-point stencil matrix solver. Style.eq.0 = iluter,
* Style.eq.1 = slv5pt, Style.eq.2 = MA28copy3. Style.eq.3 = SDRV from
* YSMP.
* NOTE: Only style.eq.2 will give good results. Applying style.eq.3
* should be corrected and is no longer recommended. It might give slow
* convergence or even divergence of the potential equation. Other values
* are NOT recommended! style.eq.0 .or. style.eq.1 will return an error.
*
* 'b2ux9p_style'              '2'            Numerics
* Choose the type of 9-point stencil matrix solver. Style.eq.0 = iluter,
* Style.eq.1 = slv5pt, Style.eq.2 = MA28copy.
* NOTE: Only style.eq.2 will give good results. Other values are NOT
* recommended! style.eq.0 and style.eq.1 will return an error.
*
* 'b2ux11p_style'             '2'            Numerics
* Choose the type of 11-point stencil matrix solver. Style.eq.0 =
* iluter, Style.eq.1 = slv5pt, Style.eq.2 = MA28copy4.
* NOTE: Only style.eq.2 will give good results. Other values are NOT
* recommended! style.eq.0 and style.eq.1 will return an error.
*
* 'b2ux5p_acpar'              '8.0'          Numerics
* 'b2ux7p_acpar'              '8.0'          Numerics
* 'b2ux9p_acpar'              '8.0'          Numerics
* 'b2ux11p_acpar'             '8.0'          Numerics
* Parameter needed for iluter matrix solver.
*
* 'b2stbc_fchy_dia'           '0.0'          Numerics
* Multiplier to the neoclassical current and diamagnetic heat flux
* convective boundary conditions, also multiplied by facdrift.
* Adds a poloidal variation consistent with neoclassics and diamagnetic

```

```

* contributions to the heat flux boundary conditions.
* Should only be used with the 5.0 model and heat flux boundaries.
* Allows use b2stbc_integral_current if fchy_dia.eq.0, forbids it
* otherwise.
*
* 'b2stbc_fchy_dia_coreonly' '1' Numerics
* If fchy_dia_coreonly.eq.1, the neoclassical current and diamagnetic
* heat flux convective boundary conditions are only applied on the core
* boundary.
* If fchy_dia_coreonly.eq.0, those are applied to all boundaries,
* including PFR, North SOL edge and diverted island center boundary.
*
* 'b2stbc_neoclassical' '0.0' Numerics
* Real parameter which establishes boundary conditions for radial
* component of the current for North and South boundaries when these lie
* on closed flux surfaces.
* If b2stbc_neoclassical is 0 then the radial component of the current
* is zero.
* If b2stbc_neoclassical is not 0 the radial component of the current is
* given by the neoclassical value. b2stbc_neoclassical is superseded if
* facdrift.ne.0.
* This cannot be used in conjunction with b2stbc_integral_current below.
*
* 'b2stbc_cbc' '1.0' Numerics
* Multiplier to the ExB velocity for sheath boundary conditions in
* b2stbc_spb and BCMOM=13 case of b2stbc_phys.
*
* 'b2stbc_integral_current' '0.0' Numerics
* If integral_current.gt.0, corrects current sources so that there be no
* surface current at the North and South edges of the edge plasma.
* The value of integral_current multiplies the correction term added to
* the current source.
* This setting is incompatible with the normal use of diamagnetic drift
* terms (facdrift.gt.0.0 .and. b2stbc_fchy_dia.ne.0.0) or neoclassical
* boundary conditions (b2stbc_neoclassical.gt.0.0).
*
* 'b2stbr_first_flight_dl' '0.001' Numerics
* Step length (in meters) for computing the first flight model chords.
*
* 'b2stbr_tauxmax' '0.05' Numerics
* Maximum allowed fraction of source to be added or subtracted to the
* real source in the method of effective sources for code speed-up.
*
* 'b2stbr_c_pumpavr' '0.02' Numerics
* Underrelaxation parameter to compute the artificial source in the
* method of effective sources for code speed-up.
*
* 'b2stbr_first_flight_no_of_flights' '9' Numerics
* Number of chords started from each start point in the first flight
* model.
*
* 'b2stbr_first_flight_table_size' '200000' Numerics
* Workspace size given to the first flight table.
*
* 'b2stbc_she0ep' '1.0e-36' Numerics
* 'b2stbc_shi0ep' '1.0e-36' Numerics
* 'b2stbc_sna0ep' '1.0e-36' Numerics
* Small level sources introduced in all cells.
*
* 'b2stbc_bcpot_16_step' '50' Numerics
* Frequency (in number of calls to b2stbc_phys) at which the constant
* value of the potential at the boundaries where BCPOT=16 is applied
* will be recomputed.
*
* 'b2mndr_na_min' '1.0e4' Numerics
* Minimal density maintained in all cells for all species.
* It must be true that na_min is smaller than na_new, as well as any of

```

```

* the initial densities provided in b2ai.dat.
* This switch replaces 'b2mndr_na0eps' from SOLPS5.x.
*
* 'b2mndr_na_new'          '1.0e14'          Numerics
* Initial density put in all cells for all new species if not
* overwritten by initial state file. This switch replaces
* 'b2mndr_na0eps' from SOLPS5.x.
*
* 'b2news_guard_flows'     '2'               Numerics
* If guard_flows.eq.0, flows between neighbouring guard cells are
* blocked. The blocking fully applies only if b2mndt_style.eq.0.
* Otherwise, use of b2news_l_fbound should be favoured.
* If guard_flows.eq.1, flows between neighbouring guard cells are kept.
* If guard_flows.eq.2, particle flows between neighbouring guard cells
* are blocked for the density equations and the incorrect corner values
* are replaced by interpolated values. Time derivative sources in guard
* cells are turned off. It is recommended 2.
*
* 'b2news_l_fbound'        '0'               Numerics
* 'b2news_l_enebound'      '0'               Numerics
* 'b2news_l_timbound'      '0'               Numerics
* Flags to internally identify and isolate boundary cells to limit flows
* between neighbouring guard cells.
* If l_fbound.ne.0, flows between boundary conditions cells are turned
* off.
* If l_enebound.ne.0, remove non-boundary energy sources in guard cells.
* If l_timbound.ne.0, time derivative sources are not added to guard
* cells.
*
* 'b2news_fac_ref'         See description (integer) Numerics
* 'b2news_facdrift_tanh_a' '0.0'             Numerics
* 'b2news_facdrift_tanh_b' '0.0'             Numerics
* 'b2news_fac_ExB_tanh_a'  '0.0'             Numerics
* 'b2news_fac_ExB_tanh_b'  '0.0'             Numerics
* 'b2news_fac_vis_tanh_a'  '0.0'             Numerics
* 'b2news_fac_vis_tanh_b'  '0.0'             Numerics
* 'b2news_iy_nocoreExB'    '-2'             Numerics
* Parameters for spatial variation of the diamagnetic, ExB and viscosity
* drift term multipliers. These are allowed to have tanh profiles
* centered at tanh_a and of width tanh_b, where both quantities are in
* metres and measured along the radial coordinate for the cell column
* ix.eq.fac_ref (default value is jxa, see Geometry section for its
* computation). The maximum value of facdrift and fac_ExB is controlled
* by their respective _start and _target switches, see Run section for
* details.
* For consistency with the naming convention of related variables:
* b2news_facExB_tanh_a is an alias for b2news_fac_ExB_tanh_a,
* b2news_facExB_tanh_b is an alias for b2news_fac_ExB_tanh_b,
* b2news_facvis_tanh_a is an alias for b2news_fac_vis_tanh_a, and
* b2news_facvis_tanh_b is an alias for b2news_fac_vis_tanh_b.
* Additionally, for all core cell rows where iy.le.iy_nocoreExB,
* fac_ExB(ix,iy) is set to zero. This means that ExB contributions and
* flows will be zeroed out in those cells.
* For consistency with the boundary condition treatment, the value
* iy_nocoreExB.eq.-1 is NOT recommended!
*
* 'nrings_for_no_speedup_averaging' '0'          Numerics
* Number of poloidal rings in the confined region (counted from the
* separatrix towards the core boundary) where the speed-up method of the
* partial flux surface averaging (CORR_CORE_DN and CORR_CORE_DT) should
* not be used.
*
* 'b2stbc_istyle_cur_contr_on_S_and_N' '2'        Numerics
* When set to '2', SPB form of adding currents on the South core
* boundary is included by using BCPOT=12, and on the South PFR and North
* boundaries by using BCPOT=13. It is recommended '2', in conjunction
* with the BCPOT=12 and BCPOT=13 settings, respectively.

```

```

* When set to '1', SPB form of adding currents on South and North
* boundaries is done explicitly in addition to other existing boundary
* conditions for the potential equation.
* The old 5.0 calculation is recovered by using the value '0'.
*
* 'b2stbc_istyle_fchi'      '0'                      Numerics
* If '1', explicitly use the expression (bx*cs*na) of particle flux
* (fna) from boundary condition instead of fna, for boundary condition
* type BCPOT=11.
*
* 'b2news_recalculate_contributions' '1'              Numerics
* If recalculate_contributions.eq.0, turns off recomputation of sources
* when wrong_flow flag is active.
*
* 'b2news_no_b2sral_call'    '1'                      Numerics
* If no_b2sral_call.eq.0, add an additional call to recompute the
* sources in b2news_ to reproduce the behaviour from SOLPS4.
*
* 'b2news_do_2nd_b2npco_call' '0'                      Numerics
* If do_2nd_b2npco_call.eq.1, perform a second call to the density
* equation solve to improve particle balance, as done in SOLPS4.
*
* 'b2news_re_eval_pttls_fluxes' '0'                    Numerics
* If re_eval_pttls_fluxes.eq.1, the particle fluxes are recomputed at
* the end of b2news_. This is necessary for rescaling of the Eirene
* sources during coupled runs, so this switch is superceded by
* use_eirene, and also needed to reproduce SOLPS4 runs.
*
* 'b2mndt_style'            '1'                      Numerics
* If 0, b2news is called (SOLPS5.0/1) If 1, b2news_ is called (SOLPS5.2)
*
* 'b2mndt_ntim_step_out'    '1'                      Numerics
* When set to a value of 'K', residuals at each K-th step will be
* written in b2ftrace. It helps to avoid a huge b2ftrace files during
* long time run.
*
* 'b2stbc_cbsnafac'         '0.01'                    Numerics
* 'b2stbc_fchycore_alpha'   '0.1'                     Numerics
* 'b2stbc_fheycore_alpha'   '0.1'                     Numerics
* 'b2stbc_fhiycore_alpha'   '0.1'                     Numerics
* 'b2stbc_fnaycore_alpha'   '0.1'                     Numerics
* 'b2stbc_nesepm_alpha'     '1.0e-3'                  Numerics
* 'b2stbc_nesepm_beta'      '0.0'                     Numerics
* 'b2stbc_nesepm_gamma'     '0.99'                    Numerics
* 'b2stbc_volrec_alpha'     '1.0e-3'                  Numerics
* 'b2stbc_volrec_beta'      '1.0'                     Numerics
* 'b2stbr_sput_chem_alpha'   '1.0e-2'                  Numerics
* 'b2stbr_sput_phys_alpha'   '1.0e-2'                  Numerics
* Feedback relaxation parameters. See Physics section for individual
* feedback quantities.
*
* 'b2stbc_type13_fac'        '1.0'                     Numerics
* 'b2stbc_type13_norm'       '1.0e15'                  Numerics
* Additional feedback strength adjustment parameters for BCCON=13
* boundary condition, density adjusted to match a specific flux.
* The density is adjusted by a factor of:
*  $(1.0\_R8 + \text{CONPAR}(IS, IB, 2) * (DFS - FS) /$ 
*  $(\text{abs}(DFS) + \text{abs}(FS) + \text{abs}(\text{type13\_norm}) + \text{abs}(NAS * \text{type13\_fac})))$ 
* where FS is the current particle flux, DFS is the desired particle
* flux, and NAS the current volume-averaged density. The desired flux
* DFS is computed as
*  $\text{CONPAR}(IS, IB, 1) + \text{CONPAR}(IS, IB, 3)$ 
* where the first number is provided directly by the user in
* b2.boundary.parameters and the second is internally set to correspond
* to the re-entry of ionised neutrals that have crossed into the core
* for cases run with Eirene where eirene_ionising_core is activated.
* See also the description of 'eirene_ionising_core'.

```

```

*
* 'heatdiff1D_linlog'          '1'                      Numerics
* 'heatdiff1D_ratio'          '100.0'                  Numerics
* Parameters related to solving the temperature equation for the target
* plate elements in depth. If linlog.eq.1, a linear subdividing of the
* plate element is used. If linlog.eq.2, a logarithmic subdividing is
* used, with the surface layer being ratio times thinner than the last
* layer in the bulk.
* ratio must be larger than 1.
*
* 'b2tlh0_hcimx_flag'         '1'                      Numerics
* When set to -1, only the gradient to the left/bottom is used to
* compute the conductive neutral flux limits.
* When set to 0, only the gradient to the left/bottom is used to compute
* the conductive neutral flux limits, except when these do not exist
* (the other face value is used).
*
* 'b2trno_flux_limit_to_dpa'  '1'                      Numerics
* If '1', flux limit to neutrals contribution to dpa0 - the diffusion
* coefficient is applied in b2tlc0.F. It is recommended '1'.
* b2tlc0.F has the flux limit parameters and which are given by
* 'b2tlc0_alpha' and 'b2tlc0_gamma'. 'b2tfnb_alpha' and 'b2tlc0_alpha'
* cannot be different from zero simultaneously.
* 'b2tfnb_alpha' gives another form of flux limit which is applied to
* the whole particle flux.
*
* 'b2mndr_isfb'               ismain                    Numerics
* 'b2mndr_ixfb'               See description            Numerics
* 'b2mndr_iyfb'               ny/2                      Numerics
* 'b2mndr_ne_wanted'          '0.0'                    Numerics
* 'b2mndr_ne_wanted_time'     '0.0'                    Numerics
* Experimental: allows for density feedback of the number of electrons
* accompanying species 'isfb' at location (ixfb,iyfb).
* For USN topologies, the default value for ixfb is 3*nx/8.
* For all others, it is 5*nx/8.
* Note that ixfb and iyfb are real numbers!
*
* 'b2mndt_use_b2srst'         '1'                      Numerics
* Switches off the stabilization of the source coefficients.
*
* 'b2mndt_rxf'                '0.5'                    Numerics
* Main under-relaxation parameter.
*
* 'b2news_xfm0'               '1.0'                    Numerics
* 'b2news_xfm1'               '1.0'                    Numerics
* 'b2news_xfm2'               '1.0'                    Numerics
* 'b2news_xfm3'               '1.0'                    Numerics
* Under-relaxation parameters for the parallel momentum, continuity,
* potential and energy equations respectively.
*
* 'b2npco_pcm0'               '1.0'                    Numerics
* 'b2npco_pcm1'               '1.0'                    Numerics
* pcm0 specifies an additional under-relaxation factor that is applied
* to the velocity correction. It is required that
* 0.le.pcm0.and.0.le.pcm1.
*
* 'b2npco_rxg'                '1.0'                    Numerics
* rxg specifies a special under-relaxation parameter. A term of the form
* "abs(residual)/rxg" is added to the diagonal of the matrix of the
* correction equation, with the effect of limiting the computed
* correction. rxg is dimensionless and of order unity; smaller values of
* rxg imply stronger damping.
*
* 'b2npht_pcm0'               '1.0'                    Numerics
* 'b2npht_pcm1'               '1.0'                    Numerics
* pcm0 specifies an additional under-relaxation factor that is applied
* to the temperature correction.

```



```

* It is required that 0.le.pcm0.and.0.le.pcm1.
*
* 'b2npht_rxg'          '1.0'          Numerics
* rxg specifies a special under-relaxation parameter. A term of the form
* "abs(residual)/rxg" is added to the diagonal of the matrix of the
* correction equation, with the effect of limiting the computed
* correction. rxg is dimensionless and of order unity; smaller values of
* rxg imply stronger damping.
*
* 'b2npht_style'        '1'            Numerics
* 'b2nph9_style'        '1'            Numerics
* When set to '1', SPB form of the program b2sihs_ is called. It is
* recommended '1'.
*
* 'b2npmo_rxg'          '1.0e6'        Numerics
* Normalisation factor for the parallel momentum equation.
*
* 'b2news_poteq'        '1'            Numerics
* If poteq.eq.0, the potential equation is jumped over and not solved.
* If poteq.eq.2, the potential is set to  $3.1 \cdot T_e / q_e$  as per SOLPS4.0.
* If poteq.eq.1, the potential equation is solved according to the
* no_solve switch settings.
* If poteq.ne.1, then 'b2tfhe_no_current' must be set to '1'.
*
* 'b2nxdv_style'        '1'            Numerics
* When set to '1', the total friction force cancel is not calculated at
* the guard boundary cells.
* It is recommended '1'.
*
* 'b2nxfc_style'        '1'            Numerics
* style specifies the precise form of interpolation used in the
* computation of flcb and cvcb. When set to '1', SPB form of the
* transport terms in the momentum correction equation is used.
* It is recommended '1'.
*
* 'b2nxfx_style'        '1'            Numerics
* When set to '1', SPB form of an expression that occurs in the
* electron-atom thermal force is used.
* It is recommended '1'.
*
* 'b2sifr_styl0'        '0'            Numerics
* 'b2sifr_styl1'        '0'            Numerics
* Specify the type of linearisation used in the thermal force term.
*
* 'b2sigp_style'        '2'            Numerics
* When set to '1', SPB form of the pressure gradient term on the right
* hand of the momentum balance equation is used.
* When set to '2', the parallel current term contains a correction due
* to impurities, when  $Z_{\text{eff}}$  is not equal to the average plasma ion
* charge. Only usable if the potential equation is solved simultaneously
* (b2news_poteq.eq.1).
* The default value of '2' is recommended for cases with one main
* hydrogenic species (or at least where the main species is the lightest
* one) and any amount of impurities. It is not guaranteed to yield
* correct results for hydrogen isotopic mixtures or non-hydrogenic
* plasmas.
*
* 'b2sigp_pressure_restriction' '1'      Numerics
* When set to '1', a CFL-like restriction is applied to the pressure
* gradient term for minority ions and for neutrals species.
*
* 'b2xzdd_zero_dead_and_core' '1'        Numerics
* If 1 then zero passed sources in dead regions,
* If 2 zero passed sources in dead regions and core boundary cells
* If 0 then SKIP
*
* 'b2sihs_rf0'          '1.0'          Numerics

```

```

*      'b2sihs_rf1'                '1.0'                Numerics
*      'b2sihs_rf2'                '1.0'                Numerics
*      'b2sihs_rf3'                '1.0'                Numerics
*      'b2sihs_rf4'                '1.0'                Numerics
*      rf0-4 are numerical parameters, see the code. All must be non-negative
*      and of order unity; larger values are meant to provide more
*      stabilisation. Some or all may be set to zero and removed eventually.
*      I have had some experience to indicate that rf0, rf2, rf4 may have a
*      useful function; however, later changes in the code may have made all
*      these parameters superfluous.
*
*      'b2sihs_style'              '0'                  Numerics
*      style determines the form of the strange electron-atom energy transfer
*      term.
*
*      'b2srdt_phm0'               '1.0'                Numerics
*      'b2srdt_phm1'               '1.0'                Numerics
*      'b2srdt_phm3'               '1.0'                Numerics
*      'b2srdt_phm4'               '0.0'                Numerics
*      'b2srdt_phm5'               '1.0'                Numerics
*      Multipliers to the density, parallel momentum, heat, potential, and
*      electron particle time-derivative source terms, respectively.
*      'phm4' is normally zero since the potential equation contains no
*      source terms.
*
*      'b2srst_rf0'                '1.0'                Numerics
*      'b2srst_rf1'                '1.0'                Numerics
*      'b2srst_rf2'                '1.0'                Numerics
*      'b2srst_rf3'                '1.0'                Numerics
*      (rf0/1/2/3 for numerical stabilisation)
*
*      'b2stcx_rg0'                '1.0'                Numerics
*      (rg0 for numerical stabilisation; needs experiments.)
*
*      'b2stcx_styl0'              '0'                  Numerics
*      Specifies the type of linearisation used in the charge exchange
*      momentum source term.
*
*      'eirene_mc_linearisation'    '1'                  Numerics
*      Specifies the type of linearisation used in the sources derived from
*      the Monte Carlo neutrals. If linearisation.eq.0, all sources are fed
*      to the constant term, if linearisation.eq.1, positive terms go in the
*      constant term, and negative terms in the proportional term.
*      'eirene_mc_linearization' is an alias for this switch.
*
*      'b2stbm_linearisation'       '1.0'                Numerics
*      Same as above, but applied to the sources coming from an externally
*      coupled code providing plasma sources, for instance a kinetic
*      treatment of trace impurity ions. 'b2stbm_linearization' is an alias
*      for this switch.
*
*      'b2stbm_impgyro_mod'         '0'                  Numerics
*      Specifies the frequency (in units of full B2 timesteps) at which an
*      externally coupled code providing additional source (for instance a
*      kinetic treatment of trace impurity ions such as IMPGYRO) should be
*      called.
*
*      'b2stel_rg0'                '1.0'                Numerics
*      'b2stel_rgl'                '1.0'                Numerics
*      'b2stel_rxm0'               '0.0'                Numerics
*      'b2stel_rxm1'               '0.0'                Numerics
*      'b2stel_rxm2'               '0.0'                Numerics
*      'b2stel_rxm3'               '0.0'                Numerics
*      'b2stel_rxm4'               '0.0'                Numerics
*      Damping coefficient applied to the atomic physics sources.
*
*      'b2stel_styl0'              '0'                  Numerics

```

```

* Specifies the type of linearisation used in the charge exchange
* momentum source term.
*
* 'b2tfcc_xfac'          '1.0'          Numerics
* Multiplier to the pressure force term.
*
* 'b2tfhe_mdf'          '0'            Numerics
* If '1', SPB new form of calculating electron heat flux is used. It is
* recommended '1' for runs with drifts.
*
* 'b2tfhe_no_current'   '0'            Numerics
* If no_current.eq.1, all currents are set to zero. The setting
* no_current.eq.1 must be used if the potential equation is not
* explicitly solved for, i.e. for b2news_poteq.ne.1.
*
* 'b2tfhe_prl_cur'      '1.0'          Numerics
* Multiplier to the parallel current.
*
* 'b2tfhi_mdf'          '0'            Numerics
* If '1', SPB new form of calculating ion heat flux is used.
* It is recommended '1' for runs with drifts.
*
* 'b2tfnb_drift_style'  See description  Numerics
* When set to '0', drift velocities are calculated at cell centers (old
* SOLPS5.0 style).
* When set to '1', drift velocities are calculated at cell faces
* (SOLPS5.2 style).
* When set to '2', drift velocities are calculated at cell faces too,
* but a better approximation of the drift velocities and corresponding
* fluxes is used. This approximation guarantees that the divergences of
* the ExB drift velocity and diamagnetic drift fluxes are numerically
* zero in a constant B-field, which simplifies the physical analysis.
* Setting 'b2tfnb_drift_style' to '2' requires that (and may be
* activated only if) the b2fgmtry file was generated with
* 'b2agdr_redef_gmtry' set to '1' (default). If 'b2agdr_redef_gmtry' was
* set to '0', the default value for 'b2tfnb_drift_style' is '1'.
* It is recommended '2'.
*
* 'b2tfnb_PSch_style'   See description  Numerics
* Controls the numerical approximation for the so-called Pfirsch-
* Schluter fluxes of particles and internal energy (divergent part of
* diamagnetic drift in the SOLPS5.2 physics model).
* Default value is '2', unless the b2fgmtry file was generated with
* 'b2agdr_redef_gmtry' set to '0' (old style), in which case, the value
* is '1'.
*
* 'b2tiner_fch_inert_style' See description  Numerics
* Controls the numerical approximation for the inertia current.
* Default value is '2' unless the b2fgmtry file was generated with
* 'b2agdr_redef_gmtry' set to '0' (old style), in which case, the value
* is '1'.
* A value of '0' recovers the older behaviour (versions 3.0.6 or less).
*
* 'b2tfnb_poleldr'      '1.0'          Numerics
* Multiplier to the poloidal electrical drift velocity.
*
* 'b2tfnb_radeldr'      '1.0'          Numerics
* Multiplier to the radial electrical drift velocity.
*
* 'b2tfnb_poldidr'      '1.0'          Numerics
* Multiplier to the poloidal diamagnetic drift velocity.
*
* 'b2tfnb_raddidr'      '1.0'          Numerics
* Multiplier to the radial diamagnetic drift velocity.
*
* 'b2tfnb_mdf'          '0'            Numerics
* If '1', SPB new form of calculating particle flux is used.

```

```

*      It is recommended '1' for runs with drifts.
*
*      'b2tfnb_xfrhie'          '1.0'          Numerics
*      Multiplier to the Rhie and Chow upwind correction.
*
*      'b2upht_stylec'          '0'            Numerics
*      (stylec is a numerical switch, needs experiments)
*
*      'b2usmo_cfc0'            '1.0'          Numerics
*      Linearisation constant.
*
*      'b2srsn_enable'          '0'            Numerics
*      If enable.ne.0, then the sources for particle, parallel momentum,
*      electron and ion heat are rescaled, for non-boundary cells, if the
*      local timestep exceeds the physics timescale implied by those sources.
*
*      *      1.6. Atomic Physics switches
*
*      'b2ardr_fix_cx'          '1'            Atomic Physics
*      It is used to 'correct' the CX data
*      0 => do not fix
*      1 => only fix H if CX data is < 1e-40 [default]
*      2 => fix if CX data is < 1e-40
*      3 => fix H
*      4 => fix all
*      At the moment the only species with CX data in ADAS is C. Be careful
*      with options that use the fit formula other than for H.
*      See the comments in ratstr.F for the origin of the fit formula used to
*      'fix' the CX data.
*
*      'b2ardr_no_weisheit'      '0'            Atomic Physics
*      When set to 1, disables the correction of the hydrogen atomic rate
*      data with the Weisheit data using ratwei.
*      *** Use with caution! ***
*
*      'b2ardr_no_smoothing'     '0'            Atomic Physics
*      When set to 1, disables the tail smoothing for of rate coefficients to
*      represent the effect of a power-law tail on the distribution function.
*      It has been added for compatibility with SOLPS4.3
*
*      'b2ardr_fix_recomb'       '0'            Atomic Physics
*      'b2ardr_zmax_recomb'      '1'            Atomic Physics
*      When fix_recomb is changed from 0, and if the ADAS option is in use,
*      then uses the PRB file to add the contribution arising from is-->is-1
*      processes.
*      This term includes the bremsstrahlung.
*      The '0' option only contains the bremsstrahlung for the is-->is-1
*      process.
*      The correction is only applied for isonuclear sequences with nuclear
*      charge up to zmax_recomb, inclusive.
*      Use of this option is still experimental!
*
*      'b2ardr_rtnt'             '40'          Atomic Physics
*      'b2ardr_rtnn'             '32'          Atomic Physics
*      The number of intervals in the discretisation of the density (rtnn)
*      and temperature (rtnt) ranges for the atomic physics tables.
*
*      'b2mndr_dpc_mod_rates_ne_hot_frac' '0.0' Atomic Physics
*      'b2mndr_dpc_mod_rates_te_hot'      '0.0' Atomic Physics
*      Modify the atomic rates by including a hot electron population of
*      temperature te_hot (in eV) and a population fraction ne_hot_frac.
*      Still experimental.
*
*-----
*.end b2cdci

      end subroutine b2cdci

```


A.5 b2cdcn.F

```
*-----
* Generated by b2cdcn.py from b2input.xml. DO NOT EDIT THIS FILE!
*.specification

      subroutine b2cdcn ()
      implicit none

*-----
*.documentation
*
* 1. purpose
*
*   B2CDCN provides information about namelists of the B2 code.
*   No computation is performed in this routine.
*
*   The namelists are listed in no particular order. Each description
*   will begin with the file from which the code expects to read the
*   namelist and its general purpose. Then, each namelist variable
*   will be listed with its use and size.
*
*-----
*.text
*
* NAMELIST /NEUTRALS/
*   Found in b2.neutrals.parameters.
*   This namelist is used to either set parameters for the B2 neutral fluid
*   model or for the Eirene Monte Carlo neutrals run.
*   When coupling to Eirene, it is essential that the neutral boundaries
*   conditions declared here be in the same exact order than they are in the
*   Eirene input file.
*   For coupled cases, some of the parameters will become active only on the
*   second call to Eirene, while values inherited from the Eirene input file
*   will be used for the first call. In such cases, use of the switch
*   'eirene_repeat_first_call' is recommended.
*   This namelist is read if either 'b2stbr_neutrals_namelist' or
*   'b2mndr_eirene' are set to 1.
*   Some of the arrays are dimensioned by NTRACK, which is the maximum number
*   of tracked species in the mixed materials model. This parameter is set in
*   b2mod_layer and currently holds the value 4.
*   NSTRAT - integer. Number of neutral sources, or 'strata'. Must not be larger
*   than DEF_NSTRAT from $(SOLPSTOP)/include(.local)/DIMENSIONS.F file.
*   Need not include the time-dependent stratum. If a time-dependent stratum
*   is used, is incremented internally as needed. The incremented value is
*   referred to as NSTRAT below.
*   Defaults to 0.
*   RCPOS - integer array of length (NSTRAT). Position in the B2 grid of the
*   strata. Similar use as BCPOS from /BOUNDARY/ namelist.
*   Defaults to -2.
*   RCSTART - integer array of length (NSTRAT). Start coordinate on the B2 grid
*   the strata. Similar use as BCSTART from /BOUNDARY/ namelist.
*   Defaults to -2.
*   RCEND - integer array of length (NSTRAT). End coordinate on the B2 grid of
*   the strata. Similar use as BCEND from /BOUNDARY/ namelist.
*   Defaults to -2.
*   RC_LIST_SIZE - integer array of length (NSTRAT). Contains the size of the
*   recycling boundary lists. Similar use as BC_LIST_SIZE from /BOUNDARY/
*   namelist.
*   Defaults to 0.
*   RC_LIST_X - integer array of length (2*(NXD+NYD),NSTRAT). Contains the
*   X-coordinate of the cells where recycling boundaries are applied. Similar
*   use as BC_LIST_X from /BOUNDARY/ namelist.
*   Defaults to -2.
*   RC_LIST_Y - integer array of length (2*(NXD+NYD),NSTRAT). Contains the
*   Y-coordinate of the cells where recycling boundaries are applied. Similar
```

```

*      use as BC_LIST_Y from /BOUNDARY/ namelist.
*      Defaults to -2.
*  TARGSP - integer array of size (NSTRAT,NTRACK). Identifies the base
*      material(s) of this stratum wall. The number corresponds to the B2 species
*      index produced by sputtering. Check b2cdci for details. If the bulk
*      species is not sputtered, should contain -1.
*      Defaults to b2stbr_sput_dst.
*  CHEMSP - logical array of length NSTRAT. Indicates whether chemical
*      sputtering is allowed from this wall stratum.
*      Defaults to '.false.'.
*  RECYC - real*8 array of size (0:NS-1,NSTRAT). Stores the particle recycling
*      coefficients of species (is) on stratum (istra). Defaults to 1.0 for non-
*      carbon species and 0.0 for carbon species. Particles recycle into the
*      neutral species associated to their homonuclear sequence.
*      Applies to B2 neutral fluid species.
*      Also multiplies Eirene recycling fluxes if 'eirene_use_recyceir' is set
*      to 0 (see b2cdci for details).
*  MRECYC - real*8 array of size (0:NS-1,NSTRAT). Stores the parallel momentum
*      recycling coefficients of species (is) on stratum (istra). The parallel
*      momentum is carried back by the recycled neutrals. Applies only to B2
*      neutral fluid species.
*      Defaults to 0.0.
*  ERECYC - real*8 array of size (0:NS-1,NSTRAT). Stores the energy recycling
*      coefficients of species (is) on stratum (istra). Defaults to 0.3 for non-
*      carbon species and 0.0 for carbon species. The energy is carried back by
*      the recycled neutrals.
*      Applies only to B2 neutral fluid species.
*  RCIION - real*8 array of size (0:NS-1,NSTRAT). Stores the particle-into-ion
*      recycling coefficients of species (is) on stratum (istra). Particles
*      recycle into the next ionised species associated to their homonuclear
*      sequence. Applies only to B2 neutral fluid species.
*      Defaults to 0.0.
*  RECYCEIR - real*8 array of size (NSTRAT). Multiplier to the Eirene recycling
*      fluxes from stratum (istra). Only used if 'eirene_use_recyceir' is set to
*      1 (default).
*      Defaults to 1.0.
*  USERFLUXPARM - real*8 array of size (NSTRAT,2). The element (istra,1)
*      contains the strength of constant gas puff strata (type 'C') for Eirene in
*      particles/second.
*      Defaults to 0.
*  CRCSTRA - character*1 array of length (NSTRAT). Contains the type of stratum
*      for Eirene. Possible options include:
*      'N','S','W','E' - topological mesh boundaries (same use as BCCHAR in
*      /BOUNDARY/ namelist)
*      'V' - volume recombination source (related RCPOS, RCSTART, RCEND and
*      RC_LIST variables are ignored).
*      'C' - constant or feedback gas puff source (related RCPOS, RCSTART, RCEND
*      and RC_LIST variables are ignored). Gas puffs for B2 fluid neutrals are
*      handled via b2.boundary.parameters. Unless specified otherwise by use of
*      'eirene_nesepm_istra', it is the first 'C' stratum that is used for
*      feedback puff schemes. See b2cdci for details.
*      'T' - time-dependent source for Eirene (related RCPOS, RCSTART, RCEND and
*      RC_LIST variables are ignored). Long-lived Eirene neutrals are stored in
*      this stratum. See EIRENE_STEP_DT below.
*      Defaults to ' '.
*  RF_NEUT - real*8 array of size (4). Relaxation parameters multiplying the
*      Eirene particle, parallel momentum, electron energy and ion energy sources
*      respectively before use in B2.
*      Defaults to 1.0.
*  PHYS_SPUT - real*8 array of size (0:NS-1,NSTRAT). Stores the physical
*      sputtering multiplier for B2 species (is) sputtering on the wall of
*      stratum (istra). Must be non-zero for physical sputtering to occur. Only
*      applies if using B2 fluid neutral model.
*      Defaults to 0.0.
*  CHEM_SPUT - real*8 array of size (0:NS-1,NSTRAT). Stores the chemical
*      sputtering multiplier for B2 species (is) sputtering on the wall of
*      stratum (istra). Must be non-zero for chemical sputtering to occur. Only

```

```

*      applies if using B2 fluid neutral model.
*      Defaults to 0.0.
* EIRENE_STEP_CPU - real*8. Length of CPU time devoted to Eirene calls after
* the first one (in s). Defaults to the value given in input.dat.
* EIRENE_STEP_DT - real*8. Physical time (in s) the Eirene particles are to be
* followed before being passed to the time-dependent stratum.
* Defaults to 1.0e-3.
* EIRENE_MOD - integer. Frequency of Eirene calls. Eirene is called every
* EIRENE_MOD full B2 iterations.
* Defaults to 1.
* VOLRECSTART - real*8 array of size (NSTRAT). Initial volume recombination
* rate from stratum (istra) to be used for the volume recombination scaling
* regulation algorithm. Rendered obsolete by 'eirene_dpc_fix'.
* Defaults to 1.e21.
* VOLRECINC - real*8. Maximum rate of increase of the volume recombination
* source strength. The volume recombination regulation scaling scheme is
* activated if VOLRECINC is neither 0 nor 1.
* Rendered obsolete by 'eirene_dpc_fix'.
* VOLRECWT - real*8. Weight of new volume recombination strength relative to
* that of previous iteration in the volume recombination regulation scaling
* scheme.
* Rendered obsolete by 'eirene_dpc_fix'.
* Defaults to 0.1.
* SPECIES_START - integer array of size (NSTRAT). Specifies the index of the
* first B2 species involved in stratum (istra).
* Defaults to 0.
* SPECIES_END - integer array of size (NSTRAT). Specifies the index of the
* last B2 species involved in stratum (istra).
* Defaults to ns-1.
* NEUTRALS_FILENAME - character*256. Name of the next file to use for reading
* a new /NEUTRALS/ namelist.
* Defaults to b2.neutrals.parameters.
* NEUTRALS_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
* modulo(NEUTRALS_TIME_MOD), exceeds NEUTRALS_TIME_SWITCH, reads the new
* namelist from NEUTRALS_FILENAME. Also switches to the new namelist if the
* ELM count (here time/NEUTRALS_TIME_MOD) changes. Only active if
* NEUTRALS_TIME_MOD is greater than 0.
* Defaults to 0.0.
* NEUTRALS_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
* which a new NEUTRALS namelist is read.
* Defaults to 0.0.
* L_NEUTRAD - integer. If l_neutrad >= 0, then the radiated power due to the
* neutrals atoms is taken directly from the Eirene calculation, instead of
* being recomputed by B2.
* Defaults to 0.
* L_NEUTFLUX - integer. If l_neutflux >= 0, then correct treatment of the
* incident fluxes in B2 and b2plot; if < 0, then old (approximate) treatment
* Defaults to 0 for coupled cases, -1 otherwise.
* LSTRASCL - integer array of size (NSTRAT,0:natm). Indicates with which
* Eirene atomic species to scale the Eirene stratum (istra) (0 means no
* scaling, default). Atomic species 0 stands for electrons.
* B2EATCR - integer array of size (0:NS-1). Contains the index of the Eirene
* atomic species corresponding to the B2 species (is).
* Defaults to ordering of the B2 isonuclear sequences.
* B2ESPCR - integer array of size (0:NS-1). Contains the isonuclear sequence
* index of the B2 species (is).
* Defaults to ordering of the B2 isonuclear sequences.
* EB2ATCR - integer array of size (NATM). Contains the index of the B2 neutral
* fluid species corresponding to the Eirene atomic species (iatm).
* Defaults to first B2 species of each isonuclear sequence.
* EB2SPCR - integer array of size (NSPECIES). Contains the index of the first
* B2 fluid for each species.
* Defaults to first B2 species of each isonuclear sequence.
* LATMSCL - integer array of size (NATM). Contains the index of the B2.5
* isonuclear sequence with which the Eirene atomic species (IATM) should be
* scaled.
* Defaults to assuming one-to-one match between Eirene and B2.5 atomic

```



```

*   species.
*   LMOLSCL - integer array of size (NMOL). Contains the index of the Eirene
*   atomic species with which the Eirene molecular species (IMOL) should be
*   scaled.
*   Defaults to 0.
*   MLCMP - integer array of size (NATM,NMOL). Contains the number of atoms from
*   Eirene atomic species IATM within each molecule of Eirene molecular
*   species IMOL.
*   Defaults to 0.
*   LIONSCL - integer array of size (NION). Contains the index of the Eirene
*   atomic species with which the Eirene test ion species (IION) should be
*   scaled. If not provided, LIONSCL will map to LMOLSCL for molecular ions
*   that match an existing declared molecule, or to the first species listed
*   in the name of the molecular ion as declared in the Eirene input file if
*   no matching molecule is found.
*   Defaults to 0.
*   LCNS - integer array of size (NSTS). Contains the indices of Eirene non-
*   standard surfaces (block 3A in input.dat) corresponding to core
*   boundaries.
*   Defaults to 0.
*   LTNS - integer array of size (NSTS). Contains the indices of Eirene non-
*   standard surfaces (block 3A in input.dat) corresponding to target
*   boundaries.
*   Defaults to 0.
*   LSNS - integer array of size (NSTRAT*NSRFS). Contains the indices of Eirene
*   surfaces related to the recycling sources.
*   Defaults to 0.
*   KSNS - integer array of size (NSTRAT). Contains the number of Eirene
*   surfaces for each Eirene recycling stratum.
*   Defaults to 0.
*   GPFC - real*8 array of size (NATM,NSTRAT). Specifies the fraction of Eirene
*   atomic species (iatm) in one particle puffed from stratum (istra).
*   Defaults to 0.0.
*   DBG_EIR_MC - integer. Debug output control for eirene_mc routine. See code
*   for usage.
*   Defaults to 0.
*   DEBUG_FLAGS - integer array of size (100). Non-zero elements yield
*   additional print-out data for debugging B2-Eirene coupling. See code for
*   specific uses. Many slots still available for user-specific needs.
*   Defaults to 0.
*   NEUT_SCL_LIM - real*8. Maximum number by which Eirene neutral sources may be
*   scaled in either direction within the ank-mods scheme. See explaining text
*   in $SOLPSTOP/doc/Source_Scaling_in_B2.pdf for details.
*   Defaults to 2.0.
*   TRACK_INDEX - integer array of size (0:NS-1). Specifies the mixed material
*   species index related to B2 species (is).
*   Defaults to 1 for species sput_dst, 0 for all others.
*   TRACK_CHEM_SPUT - logical array of size (NTRACK). Indicates whether mixed
*   material species (itrack) participates in chemical sputtering. Defaults to
*   .true. for first tracked species if sput_dst is carbon, and to .false. for
*   all other cases.
*   CHEMICAL_EROSION_REDEP_FAC - real*8. Multiplier to the chemical sputtering
*   coefficient of carbon for computing the rate used for redeposited carbon.
*   Defaults to 1.0.
*   CHEMICAL_EROSION_BE_FAC - logical. Indicates whether the presence of
*   beryllium is to impact on the chemical sputtering yield of carbon.
*   Defaults to '.false.'.
*   CHEMICAL_EROSION_BE_FAC_A - real*8. The chemical sputtering yield of carbon
*   is multiplied by
*   (1.0-C/2*(tanh((frac-A)/B)-tanh((-A)/B)))
*   where frac is the fractional content of Be in the surface layer material.
*   Defaults to 0.2.
*   CHEMICAL_EROSION_BE_FAC_B - real*8. See above.
*   Defaults to 0.05.
*   CHEMICAL_EROSION_BE_FAC_C - real*8. See above.
*   Defaults to 0.9.
*   N_SPCSRF - integer. Number of special surfaces groups for diagnostics.

```

```

* Defaults to 0.
* L_SPCSRF - integer array of length (NLIM+NSTS). List of surface segments
* (non-default standard surfaces [NDSS] or additional surfaces in Eirene
* notation) included in the groups. Negative numbers correspond to NDSS.
* Defaults to 0.
* SPS_ID - character*8 array of length (N_SPCSRF). Labels of groups of special
* surfaces.
* I_SPCSRF - integer array of length (N_SPCSRF). Index of the first Eirene
* surface belonging to a special surface group in the L_SPCSRF list.
* Defaults to 0.
* J_SPCSRF - integer array of length (N_SPCSRF). Index of the last Eirene
* surface belonging to a special surface group in the L_SPCSRF list.
* Defaults to 0.
* SPS_ABSR - real*8 array of length (N_SPCSRF). Particle absorption on the
* surface. If positive, will supercede the setting from the Eirene input
* file: RECYCT = 1-SPS_ABSR. Ignored if negative.
* Defaults to -1.0.
* SPS_TRNO - real*8 array of length (N_SPCSRF). Surface transparency in
* positive direction. If positive, will supercede the setting from the
* Eirene input file: TRANSP(1,) = SPS_TRNO. Ignored if negative.
* Defaults to -1.0.
* SPS_TRNI - real*8 array of length (N_SPCSRF). Surface transparency in
* negative direction. If positive, will supercede the setting from the
* Eirene input file: TRANSP(2,) = SPS_TRNI. If negative, the setting from
* SPS_TRNO is used.
* Defaults to -1.0.
* SPS_MTRI - real*8 array of length (N_SPCSRF). Surface material in Eirene
* notation (e.g., 1206 for C). If positive, will supercede the setting from
* the Eirene input file: ZNML = SPS_MTRI. Ignored if negative.
* Defaults to 0.
* SPS_MTRL - character*8 array of length (N_SPCSRF). Surface material in human
* notation (e.g., 'C').
* SPS_TMPR - real*8 array of length (N_SPCSRF). Surface temperature (in eV).
* If set larger to 1.e10, will be ignored. Otherwise, will supercede the
* setting from the Eirene input file: EWALL = SPS_TMPR.
* Defaults to 1.e15.
* SPS_SPPH - real*8 array of length (N_SPCSRF). Fudge factor for physical
* sputtering. If positive, will supercede the setting from the Eirene input
* file: RECYCS = SPS_SPPH. Ignored if negative.
* Defaults to -1.0.
* SPS_SPCH - real*8 array of length (N_SPCSRF). Fudge factor for chemical
* sputtering. If positive, will supercede the setting from the Eirene input
* file: RECYCC = SPS_SPCH. Ignored if negative.
* Defaults to -1.0.
* SPS_SGRP - integer array of length (N_SPCSRF). Sputtered particle species
* flag for chemical sputtering. If positive, will supercede the setting from
* the Eirene input file: ISRC = SPS_SGRP. Ignored if negative.
* Defaults to -1.
* WRITE_NML_NEUT - logical. If .true. (default), writes the content of the
* namelist to stdout after it has been read.
* Defaults to .true..
* TIME_DEP_PUFF - logical array of length (NSTRAT). Indicates whether a
* stratum is a time-dependent gas puff. Should only be .true. for strata
* defined as gas puffs.
* Defaults to .false. for all strata.
* NGPDATA - integer data of size (NSTRAT). Number of points over which the
* time profile of the strength of gas puff (istra) is given. Defaults to 0.
* Current maximum value is 100.
* Defaults to 0.
* GPDATA - real*8 data of size (NGPDATA,2,NSTRAT). For (i,ik,istra) in
* (1:NGPDATA(istra),1:2,1:NSTRAT),
* GPDATA(i,1,istra) contains the time point (i) for stratum (istra) (in s).
* GPDATA(i,2,istra) contains the gas puff strength of stratum (istra) at
* time point (i) (in particles/s).
* The gas puff strength before the first time point is given by
* USERFLUXPARM(istra,1).
* The gas puff strength after the last time point is given by the GPDATA

```

```

*      value of the last time point.
*      Otherwise, the gas puff strength is linearly interpolated from the given
*      data.
*      Defaults to 0.0.*
* The following are active only if NO_B2_CHEM Sput is not defined among the
* compiler options and are used to override the Eirene chemical sputtering
* settings:
*
* CHEMICAL_SPUTTER_YIELD - real*8 array of size (0:NLIM+NSTS). Passed to
* Eirene. Chemical sputter yield of wall surface (ilim). Element 0
* corresponds to the default setting for all surfaces.
* Defaults to 0.0.
* FCHAR_CHEMICAL - real*8. Nuclear charge of atomic species causing the
* sputtering. If set to its default value of 0, then the chemical sputtering
* switches from this block are not used and the settings from the Eirene
* input file are used.
* Defaults to 0.
* IGASS_CHEMICAL - integer. Passed to Eirene. Eirene atomic species index of
* the sputtered particle. If igass_chemical > natmi, the data from
* chemical_sputter_yield is not used and the yield from the Eirene surface
* blocks is used instead.
* Defaults to 0.
* ITSPUT_CHEMICAL - integer. Passed to Eirene. Eirene type index of the
* sputtered particle. Atoms = 1, Ions = 4. Defaults to 0, meaning 1 eV atom
* chemical sputtering.
* Defaults to 0.
* ISSPUT_CHEMICAL - integer. Passed to Eirene. Mass*100 + nuclear charge of
* the sputtered particle. Defaults to 0, internally changed to 1206 =
* carbon.
* Defaults to 0.
*
* NAMELIST /WALL_SAVE/
* Found in b2.wall_save.parameters.
* Read if present in the run directory.
* Automatically generated by B2 but can be modified by user.
* Contains data describing the state of the walls lining the vacuum vessel.
* These data are used for smooth restarts and are not really intended for
* human use.
* NDEPTH_NML - integer. Dimension NDEPTH used for the arrays within this
* namelist. Represents the number of depth layer discretising the wall
* elements for the wall model. If using the 0-D model or the time-
* independent 1-D model, will contain 1 (default). Should not exceed the
* value of the parameter NDEPTH declared in b2mod_wall.F.
* Defaults to 1.
* IMAPX - integer array of size (NWALL). Indicates the (ix) position in the
* grid of wall element (iwall). Default ordering of the wall elements
* proceeds clockwise in physical space from the lower left corner.
* IMAPY - integer array of size (NWALL). Indicates the (iy) position in the
* grid of wall element (iwall). Default ordering of the wall elements
* proceeds clockwise in physical space from the lower left corner.
* XYMAP - integer array of size (-1:NX,-1:NY). XYMAP(ix,iy) contains the wall
* index (iwall) of the wall element located a grid locaion (ix,iy). If there
* is no wall element at this position, contains 0.
* Default ordering of the wall elements proceeds clockwise in physical
* space from the lower left corner.
* SURFACE_MATERIAL_NAME - character*6 of size (NWALL). Contains the filename
* from which to extract the surface material properties for wall element
* (iwall). The files are to be found in
* $(SOLPSTOP)/data(.local)/Surface_properties/.
* Defaults to 'C'.
* COATING_MATERIAL_NAME - character*6 of size (NWALL). Contains the filename
* from which to extract the bulk material properties of the eventual coating
* for wall element (iwall).
* Defaults to ' '.
* BULK_MATERIAL_NAME - character*6 of size (NWALL). Contains the filename from
* which to extract the bulk material properties for wall element (iwall).

```

```

* The files are to be found in $(SOLPSTOP)/data(.local)/Bulk_properties/.
* Defaults to 'C'.
* LAYER_ALLOYS - character*6 of size (NALLOYS). Contains the filename from
* which to extract the material properties for alloy (nalloy) which may be
* present in mixed materials deposited layers. Not yet operational. The
* files are to be found in $(SOLPSTOP)/data(.local)/Bulk_properties/ and
* $(SOLPSTOP)/data(.local)/Surface_properties/.
* TARGET_TEMP - real*8 array of size (NWALL,NDEPTH). Contains the temperature
* (in Kelvin) of wall element (iwall) at depth layer (idepth).
* If plate_option.eq.1, will be set to backplate_temp(iwall).
* If plate_option.eq.2 and empty, will be set to equilibrium 1-D profile
* deduced from plasma incoming fluxes.
* If plate_option.eq.3, must be set.
* INERTIAL_COOLING - logical array of size (NWALL). Indicates whether wall
* element (iwall) is inertially cooled instead of actively cooled.
* Defaults to '.false.'.
* BACKPLATE_TEMP - real*8 array of size (NWALL). Contains the temperature (in
* Kelvin) maintained by cooling at the back end of wall element (iwall).
* Defaults to b2stbr_plate_temp.
* PLATE_THICKNESS - real*8 array of size (NWALL). Contains the thickness (in
* meters) of wall element (iwall).
* Defaults to b2stbr_plate_thick.
* COATING_THICKNESS - real*8 array of size (NWALL). Contains the thickness (in
* meters) of the eventual coating on wall element (iwall).
* Defaults to 0.
* PLATE_TIME_FACTOR - real*8 array of size (NWALL). Multiplier to the time for
* the equations for temperature and composition evolution of wall element
* (iwall).
* Defaults to 1.0.
* DEPOSITION - real*8 array of size (NWALL, NTRACK). Contains the amount of
* deposited material (in atoms) from species (itrack) onto wall element
* (iwall).
* Defaults to 0.0.
* EROSION - real*8 array of size (NWALL, NTRACK). Contains the amount of eroded
* material (in atoms) of species (itrack) from wall element (iwall).
* Defaults to 0.0.
* CHEMICAL_SPUTTERING - real*8 array of size (NWALL,0:NS-1). Contains the
* chemical sputter yield of 'sput_dst' or 'sput_dst_bulk' species on wall
* element (iwall) caused by B2 species (is).
* Defaults to 0.0.
* PHYSICAL_SPUTTERING - real*8 array of size (NWALL,0:NS-1). Contains the
* physical sputter yield of 'sput_dst' or 'sput_dst_bulk' species on wall
* element (iwall) caused by B2 species (is).
* Defaults to 0.0.
* PHYSICAL_SPUTTERING_ENERGY - real*8 array of size (NWALL,0:NS-1). Contains
* the fraction of returned energy carried by sputtered particles of species
* 'sput_dst' or 'sput_dst_bulk' species from wall element (iwall) caused by
* B2 species (is).
* Defaults to 0.0.
* RES_SPUTTERING - real*8 array of size (NWALL,0:NS-1). Contains the RES
* sputter yield of 'sput_dst' or 'sput_dst_bulk' species on wall element
* (iwall) caused by B2 species (is).
* Defaults to 0.0.
* THERMAL_EVAPORATION - real*8 array of size (NWALL,0:NS-1). Contains the rate
* of thermal evaporation of species (is) (in particles/second) from wall
* element (iwall).
* Defaults to 0.0.
* BACKSCATTERING - real*8 array of size (NWALL,0:NS-1). Contains the
* backscattering fraction for incoming B2 species (is) onto wall element
* (iwall).
* Defaults to 0.0.
* BACKSCATTERING_ENERGY - real*8 array of size (NWALL,0:NS-1). Contains the
* backscattered energy fraction for incoming B2 species (is) onto wall
* element (iwall).
* Defaults to 0.0.
* PLATE_TIME - real*8 array of size (NWALL). Indicates how much simulation
* time has elapsed for wall element (iwall) (in seconds).

```

```

* Defaults to 0.0.
* PLATE_AREA - real*8 array of size (NWALL). Indicates the wall area (in
* square meters) for wall element (iwall). Defaults to the area computed
* from the B2 grid information.
* MONOLAYER_DEPOSITION - real*8 array of size (NWALL,NTRACK). Contains the
* amount of deposited material (in monolayers) from species (itrack) onto
* wall element (iwall).
* Defaults to 0.0.
* MONOLAYER_EROSION - real*8 array of size (NWALL,NTRACK). Contains the amount
* of eroded material (in monolayers) of species (itrack) from wall element
* (iwall).
* Defaults to 0.0.
* LAYER_NCONSTITUENTS - integer array of size (NWALL). Indicates the number of
* elemental constituents within the surface layer of wall element (NWALL).
* Defaults to 1.
* LAYER_NZCONSTITUENTS - integer array of size (NWALL,6+NTRACK). Contains the
* atomic numbers Z of the various elements present within the surface layer
* of wall element (iwall). Defaults to 6 for the first value, 0 otherwise.
* LAYER_RELCONSTITUENTS - real*8 array of size (NWALL,6+NTRACK). Contains the
* relative atomic abundances of the various elements present within the
* surface layer of wall element (iwall). Defaults to 1.0 for the first
* value, 0.0 otherwise.
* Replaces LAYER_NRELCONSTITUENTS.
*
* NAMELIST /B2MD_NAMELIST/
* Found in b2md.dat and read by b2md.
* Contains data useful for cataloguing the run (done from 'save_mds' and
* 'resave_mds' scripts).
* EXP - character*128. Name of the experiment being modelled.
* Defaults to 'NOT_SET'.
* SHOT - integer. Shot number identifying the run. Defaults to the last number
* found in shotnumber.history, or 0 if the file is not found.
* TIME - real*8. Time point of the experimental shot being simulated.
* Defaults to 0.0.
* COMMENT - character*128. Label for the run.
* Defaults to 'NOT_SET'.
* TIMEDEP - logical. If .true. (default), saves data from b2time.nc.
* SNAPSHOT - logical. If .true. (default), saves data from b2fplasma.
* TALLIES - logical. If .true. (default), saves data from b2tallies.nc.
* MOVIES - logical. If .true. (default), saves data from b2movies.nc.
* OVERWRITE_SHOTNUMBER - integer. Indicate the shot number to overwrite (to be
* used only when updating an already saved run with 'resave_mds' script).
* Defaults to the last number found in shotnumber.history, or 0 if the file
* is not found.
*
* NAMELIST /BOUNDARY/
* Found in b2.boundary.parameters.
* Contains data used to specify the boundary conditions.
* This namelist is read if 'b2stbc_boundary_namelist' is set to 1.
* Necessary for double-null geometries or impinging limiting structures.
* It is recommended but not strictly necessary that the boundary conditions
* declared here match in number and order any neutral boundary conditions
* declared in the b2.neutrals.parameters file.
* This description is copied from the SOLPS manual.
* NBC - integer. Number of boundary segments. Defaults to 0.
* BCCHAR - character*1 array of length (NBC). Specifies the nature of the
* boundary segment
* N = 'North' boundary
* S = 'South' boundary
* W = 'West' boundary
* E = 'East' boundary
* X = 'X' boundary used for specifying a fixed value on a row of cells
* Y = 'Y' boundary used for specifying a fixed value on a column of cells
* Defaults to ' '.
* CONPAR - real*8 array of size (0:NS-1,NBC,3). Contains parameters helping to
* define the boundary conditions for the continuity equation of species
* (is). See description of BCCON below for details.

```

```

* Defaults to 0.0.
* MOMPARG - real*8 array of size (0:NS-1,NBC,2). Contains parameters helping to
* define the boundary conditions for the parallel momentum equation of
* species (is). See description of BCMOM below for details.
* Defaults to 0.0.
* ENEPARG - real*8 array of size (NBC,2). Contains parameters helping to define
* the boundary conditions for the electron energy equation. See description
* of BCENE below for details.
* Defaults to 0.0.
* ENIPARG - real*8 array of size (NBC,2). Contains parameters helping to define
* the boundary conditions for the ion energy equation. See description of
* BCENI below for details.
* Defaults to 0.0.
* POTPAR - real*8 array of size (NBC,2). Contains parameters helping to define
* the boundary conditions for the potential equation. See description of
* BCPOT below for details.
* Defaults to 0.0.
* BCPOS - integer array, length (NBC). For North, South or X boundary
* conditions, it specifies the row index; for West, East and Y boundary
* conditions, it specifies the column index.
* Defaults to -2.
* BCSTART - integer array, length (NBC). For North, South or X boundary
* conditions, it specifies the start column index; for West, East and Y
* boundary conditions, it specifies the start row index.
* Defaults to -2.
* BCEND - integer array, length (NBC). For North, South X boundary conditions,
* it specifies the end column index; for West, East and Y boundary
* conditions, it specifies the end row index.
* Defaults to -2.
* BC_LIST_SIZE - integer array of length (NBC). Contains the size of the list
* of cells where a boundary condition is applied.
* Defaults to 0.
* BC_LIST_X - integer array of length (2*(NXD+NYD),NBC). Contains the
* X-coordinate of the cells where boundaries conditions are applied.
* Defaults to -2.
* BC_LIST_Y - integer array of length (2*(NXD+NYD),NBC). Contains the
* Y-coordinate of the cells where boundaries conditions are applied.
* Defaults to -2.
* BCCON - integer array, length (0:NS-1,NBC). Specifying the type of density
* boundary condition for each segment and species (fastest varying index is
* species); makes use of CONPAR to specify additional information, as
* indicated:
* 0 : default, no boundary condition is applied
* 1 : prescribe the value of the density, CONPAR(,,1) specifies the
* required density in m^{-3}
* 2 : prescribe the gradient of the density, CONPAR(,,1) specifies the
* required density gradient in m^{-4}
* 3 : sheath conditions, CONPAR(,,1) not used (zero gradient is used)
* 4 : prescribe the value of the density, weakly a mixed boundary
* condition, CONPAR(,,1) specifies the required density in m^{-3} and
* CONPAR(,,2) specifies the 'strength' of the boundary condition
* 5 : prescribe the particle flux per unit area, CONPAR(,,1) specifies the
* required particle flux density in m^{-2} s^{-1}
* 6 : prescribe the total particle flux for a constant density, CONPAR(,,1)
* specifies the particle flux in s^{-1}
* 7 : prescribe the density as a function of other plasma parameters [not
* yet available]
* 8 : prescribe the total particle flux with constant flux density,
* CONPAR(,,1) specifies the particle flux in s^{-1}
* 9 : prescribe the decay length for the density, CONPAR(,,1) specifies the
* gradient length in $m$ (should use type 15 instead when drifts are turned
* on)
* 10 : leakage option for density, recommended for cases with drifts,
* CONPAR(,,1) specifies the leakage factor, in _loss = C_s n_a
* 11 : particle flux feedback boundary condition, CONPAR(,,1) not used,
* derived from CBSNA(0,IS,IREG). The species used must be declared using
* the 'b2stbc_isfeedback' switch.

```

```

* 12 : particle density feedback boundary condition, as above, CONPAR(,,1)
* not used, derived from CBSNA(0,IS,IREG). The species used must be
* declared using the 'b2stbc_isfeedback' switch.
* 13 : particle density to achieve specified total flux,
* CONPAR(,,1) is the specified flux crossing the flux surface
* 'b2stbc_type13_ref' steps away from the boundary,
* CONPAR(,,2) is the strength of the feedback,
* CONPAR(,,3), when running with Eirene and the 'ionising core' switch is
* used, is set internally to match the re-entering flux of ionised neutrals
* that crossed the core boundary (one must then have 'ionising_core'.eq.-IB
* where IB is the boundary index).
* The feedback scheme can be further tweaked with the switches
* 'b2stbc_type13_norm' and 'b2stbc_type13_fac'. See code for details.
* 14 : sound speed velocity flux, CONPAR(,,1) is a multiplier to the
* outgoing sound speed C_s . To be used in conjunction with BCENE/I=15,
* BCPOT=11, and BCMOM=13. recommended for cases with drifts.
* 15 : prescribe a radial leakage velocity, CONPAR(,,1) specifies the
* leakage velocity in units of the local thermal velocity.
* 16 : particle density to achieve specified total flux, used with ASTRA
* coupling. The total desired flux is summed over all BCCON=16 core
* boundaries. CONPAR(,,1) specifies the desired particle flux in  $s^{-1}$  .
* 18 : prescribe total main ion particle flux, used with ASTRA coupling.
* 19 : particle flux feedback boundary condition, flux is summed over
* neutrals and ions, for coupling with ASTRA. The total desired flux is
* summed over all BCCON=19 core boundaries. This boundary condition type is
* applied to ions in their highest ionisation stage. CONPAR(,,1) specifies
* the desired particle flux in  $s^{-1}$  .
* 20 : constant density feedback condition, CONPAR(,,1) specifies the
* desired density in  $m^{-3}$  .
* 21 : prescribe the value of the density and add a density perturbation to
* get a solution which is as close as possible to neoclassical theory. It
* is recommended to use this boundary condition together with corresponding
* condition on ion temperature (BCENI=24). The total desired density is
* summed over all BCCON=21 core boundaries. CONPAR(,,1) specifies the
* desired density in  $m^{-3}$ ..
* 22 : feedback boundary condition: given total particle flux with constant
* average density. The poloidal density variation is added to make the
* solution as close as possible to the neoclassical value derived for pure
* H/D/T plasma. It is recommended to use this boundary condition with
* corresponding boundary condition on ion temperature (BCENI=23,24). The
* total desired flux is summed over all BCCON=22 core boundaries.
* CONPAR(,,1) specifies the desired particle flux in  $s^{-1}$  .
* 23 : feedback boundary condition: given sum of integrated neutrals and
* main ion particle fluxes with constant average density. The poloidal
* density variation is added to make the solution as close as possible to
* the neoclassical value derived for pure H/D/T plasma. It is recommended
* to use this boundary condition with corresponding boundary condition on
* ion temperature (BCENI=23,24). The total desired flux is summed over all
* BCCON=23 core boundaries. CONPAR(,,1) specifies the desired particle flux
* in  $s^{-1}$  .
* 24 : constant density feedback scaled by density on the ring
* 'bc_type21_ref' away. CONPAR(,,1) specifies the desired density in  $m^{-3}$ 
* . CONPAR(,,2) is the strength of the feedback
* 25 : feedback boundary condition: prescribe the average value of the
* density and add a density perturbation from neighbouring radial cell.
* This boundary condition is suitable for any species of a multi-species
* plasma (i.e. in the case when the condition used in BCCON=21 fails). It
* is recommended to use this boundary condition together with the
* corresponding condition on ion temperature (BCENI=26,27). CONPAR(,,1)
* specifies the desired average density in  $m^{-3}$  .
* 26 : feedback boundary condition: prescribe the total ion flux and find
* the average density. A density perturbation is taken from the
* neighbouring radial cell. This boundary condition is suitable for any
* species of a multi-species plasma (i.e. in the case when the condition
* used in BCCON=22 fails). It is recommended to use this boundary condition
* together with a corresponding condition on the ion temperature
* (BCENI=26,27). The total desired flux is summed over all BCCON=26 core

```

```

* boundaries. CONPAR(,,1) specifies the desired particle flux in  $s^{-1}$  .
* 27 : feedback boundary condition: prescribe the particle flux sum for all
* neutrals and ions belonging to a given isonuclear sequence and find the
* average density of the highest ionization stage. A density perturbation
* is taken from the neighbouring radial cell. It is recommended to use this
* boundary condition together with corresponding condition on ion
* temperature (BCENI=25,26). The total desired flux is summed over all
* BCCON=27 core boundaries. CONPAR(,,1) specifies the desired particle flux
* in  $s^{-1}$  .
* BCMOM - integer array, length NS * NBC. Specifying the type of parallel
* momentum or velocity boundary condition for each segment and species
* (fastest varying index is species); makes use of MOMPARG to specify
* additional information, as indicated
* 0 : default, no boundary condition is applied
* 1 : prescribe the value of the parallel velocity, MOMPARG(,,1) specifies
* the parallel velocity in  $m.s^{-1}$ 
* 2 : prescribe the gradient of the parallel velocity, MOMPARG(,,1)
* specifies the parallel velocity gradient in  $s^{-1}$ 
* 3 : sheath conditions, Mach number as input,
* if MOMPARG(,,2) < 0.5 , then the velocity is set to exactly
* MOMPARG(,,1) * C_{s,collective},, otherwise the velocity is set to be at
* least MOMPARG(,,1) * C_{s,collective},, species
* 4 : prescribe the value of the velocity, weakly a mixed boundary
* condition, MOMPARG(,,1) specifies the parallel velocity in  $m.s^{-1}$  and
* MOMPARG(,,2) specifies the 'strength' of the boundary condition
* 5 : prescribe the parallel momentum flux per unit area, MOMPARG(,,1)
* specifies the parallel momentum flux density in  $N.m^{-2}$ 
* 6 : prescribe the total parallel momentum flux for a constant parallel
* velocity [not yet available]
* 7 : prescribe the parallel momentum as a function of other plasma
* parameters [not yet available]
* 8 : special : limited shear, imposes zero gradient for the Mach number.
* [[Eventually intended to have MOMPARG(,,1) specify the gradient of the
* Mach number in  $m^{-1}$  ]]]
* 9 : prescribe the total parallel momentum flux with constant flux
* density, MOMPARG(,,1) specifies the parallel momentum flux in N
* 10 : prescribe the decay length for the parallel momentum, MOMPARG(,,1)
* specifies the decay length in m
* 11 : Rozhansky viscosity condition for the parallel momentum, MOMPARG(,,1)
* is not used
* 12 : condition from b2stbc_spb for the parallel momentum
* 13 : sheath boundary condition from b2stbc_spb for the parallel momentum.
* To be used in conjunction with BCENE/I=15, BCPOT=11, and BCCON=14.
* Recommended for cases with drifts.
* 14 : condition from b2stbc_spb for the parallel momentum
* 15 : prescribe the value of the parallel velocity, scaled with
* B_average/B_local
* 16 : prescribe the average value of the parallel velocity MOMPARG(,,1)
* specifies the parallel velocity in  $m.s^{-1}$ 
* 17 : leakage option for parallel momentum, MOMPARG(,,1) specifies the
* leakage factor, in  $_{loss} = C_{s,a} m_a n_a u_a$ 
* BCENE - integer array, length NBC. Specifying the type of electron energy or
* temperature boundary condition for each segment; makes use of ENEPARG to
* specify additional information, as indicated
* 0 : default, no boundary condition is applied
* 1 : prescribe the value of the electron temperature, ENEPARG(,1) specifies
* the temperature in eV
* 2 : prescribe the gradient of the electron temperature, ENEPARG(,1)
* specifies the temperature gradient in  $eV.m^{-1}$ 
* 3 : sheath conditions, electron energy transmission, ENEPARG(,1) specifies
* an additional contribution to the energy transmission coefficient in
* addition to that of the potential difference [the sound speed used
* depends on settings of MOMPARG(,ISMAIN,2)]
* 4 : prescribe the value of the electron temperature, weakly a mixed
* boundary condition, ENEPARG(,1) specifies the temperature in eV and
* ENEPARG(,2) specifies the 'strength' of the boundary condition
* 5 : prescribe the electron energy flux per unit area, ENEPARG(,1)

```



```

* specifies the energy flux density in  $\text{W.m}^{-2}$ 
* 6 : prescribe the total electron energy flux for a constant electron
* temperature, ENEPAR(,1) specifies the energy flux in W (no longer
* supported, use types [16] or [17] instead)
* 7 : prescribe the electron temperature as a function of other plasma
* parameters [not yet available]
* 8 : prescribe the total electron heat flux with constant flux density,
* ENEPAR(,1) specifies the energy flux in W
* 9 : prescribe the decay length for the electron temperature, ENEPAR(,1)
* specifies the decay length in m (can also use type [19] instead)
* 10 : feedback option for core, ENEPAR(,1) not used, derived from
* cbshe(0,coreregno)
* 11 : not used
* 12 : sheath conditions, electron energy transmission coefficient,
* ENEPAR(,1) specifies an energy transmission factor,  $\delta_e$  in  $Q_e =$ 
*  $\delta_e \frac{e}{T_e}$ 
* 13 : prescribe the electron energy flux per unit area proportional to
* temperature, ENEPAR(,1) specifies the energy flux density per temperature
* in  $\text{W.m}^{-2} \cdot \text{J.J}^{-1}$  (the temperature here in J)
* 14 : leakage option for electron energy, ENEPAR(,1) specifies the leakage
* factor, in  $_{\text{loss}} = C_s \cdot n_e T_e$ 
* 15 : sheath boundary condition, from b2stbc_spb, recommended when using
* drifts (see Section C.7.4 of manual for details). Linked to using
* BCCON=14 and BCMOM=13 for all ion species, BCENI=15, and BCPOT=11.
* 16 : feedback boundary condition with constant temperature, ENEPAR(,1)
* specifies the power flux in W across the flux surface with index
* 'b2stbc_type16_ref' (default=-1), ENEPAR(,2) should be something like 0.1
* and specifies the strength of the feedback. Also see type [17] below.
* Available if bcene_16_style=0 (default). If bcene_16_style=1, integrated
* electron heat flux with constant electron temperature, summed over all
* core boundaries with BCENE=16.
* 17 : feedback boundary condition with constant shared temperature for
* both electrons and ions, with ENEPAR(,1) + ENIPAR(,1) giving the total
* power flux in W across the flux surface with index 'b2stbc_type16_ref'
* (default=-1), ENEPAR(,2) should be something like 0.1 and specifies the
* strength of the feedback. Replaces [16] for high densities and large
* values of 'b2stbc_type16_ref'.
* 18 : fractional drop condition. Not yet working.
* 19 : same as [9] but to be used when simultaneously setting BCCON=1 on
* the same boundary.
* 20 : feedback boundary condition with constant temperature, as per type
* [16] but with 'type20_' switches and a different feedback scheme.
* 21 : constant temperature feedback scaled by temperature on the ring
* bc_type21_ref away
* ENEPAR(,1) specifies the desired electron temperature in eV.
* ENEPAR(,2) is the strength of the feedback
* 22 : radial leakage condition for the electron temperature. ENEPAR(,1)
* specifies the leakage velocity in units of the electron thermal velocity.
* A temperature gradient such that the diffusive flux is set to match this
* leakage is imposed.
* BCENI - integer array, length NBC. Specifying the type of ion energy or
* temperature boundary condition for each segment; makes use of ENIPAR to
* specify additional information, as indicated
* 0 : default, no boundary condition is applied
* 1 : prescribe the value of the ion temperature, ENIPAR(,1) specifies the
* temperature in eV
* 2 : prescribe the gradient of the ion temperature, ENIPAR(,1) specifies
* the temperature gradient in  $\text{eV.m}^{-1}$ 
* 3 : sheath conditions, ion energy transmission, ENIPAR(,1) specifies the
* contribution to the energy transmission coefficient [the sound speed used
* depends on settings of MOMP(,ISM(,2))]
* 4 : prescribe the value of the ion temperature, weakly a mixed boundary
* condition, ENIPAR(,1) specifies the temperature in eV and ENIPAR(,2)
* specifies the 'strength' of the boundary condition
* 5 : prescribe the ion energy flux per unit area, ENIPAR(,1) specifies the
* energy flux density in  $\text{W.m}^{-2}$ 
* 6 : prescribe the total ion energy flux for a constant ion temperature,

```

```

* ENIPAR(,1) specifies the energy flux in W (no longer supported, use types
* [16] or [17] instead)
* 7 : prescribe the ion temperature as a function of other plasma
* parameters [not yet available]
* 8 : prescribe the total ion heat flux with constant flux density,
* ENIPAR(,1) specifies the energy flux in W
* 9 : prescribe the decay length for the ion temperature, ENIPAR(,1)
* specifies the decay length in m (can also use type [19] instead)
* 10 : feedback option for core, ENIPAR(,1) not used, derived from
* cbshi(0,ISMAIN,coreregno)
* 11 : sheath conditions, ion energy transmission coefficient, ENIPAR(,1)
* specifies an energy transmission factor,  $\delta_i$  in  $Q_i = \delta_i T_i$ 
*  $\sum_a n_a C_{s,a}$ 
* 12 : sheath conditions, ion energy transmission coefficient, ENIPAR(,1)
* specifies an energy transmission factor,  $\delta_i$  in  $Q_i = \delta_i T_i$ 
*  $\sum_a n_a$ 
* 13 : prescribe the ion energy flux per unit area proportional to
* temperature, ENIPAR(,1) specifies the energy flux density per temperature
* in  $W.m^{-2}.J.J^{-1}$  (the temperature here in J)
* 14 : leakage option for ion energy, ENIPAR(,1) specifies the leakage
* factor,  $\text{in\_loss} = C_{sTT,i}$ 
* 15 : sheath boundary condition, from b2stbc_spb, recommended when using
* drifts (see Section C.7.5 of manual for details). Linked to using
* BCCON=14 and BCMOM=13 for all ion species, along with BCENE=15 and
* BCPOT=11.
* 16 : feedback boundary condition with constant temperature, ENIPAR(,1)
* specifies the power flux in W across the flux surface with index
* 'b2stbc_type16_ref' (default=-1), ENIPAR(,2) should be something like 0.1
* and specifies the strength of the feedback. Also see type [17] below.
* Available if bceni_16_style=0 (default). If bceni_16_style=1, integrated
* ion heat flux with constant ion temperature, summed over all core
* boundaries with BCENI=16.
* ENIPAR(,1) specifies the power flux in W
* 17 : feedback boundary condition with constant shared temperature for
* both electrons and ions, see BCENE=17 above for description.
* 18 : fractional drop condition. Not yet working.
* 19 : same as [9] but to be used when simultaneously setting BCCON=1 on
* the same boundary.
* 20 : feedback boundary condition with constant temperature, as per type
* [16] but with 'type20_' switches and a different feedback scheme.
* 21 : from b2stbc_spb
* 22 : radial leakage condition for the ion temperature. ENIPAR(,1)
* specifies the leakage velocity in units of the collective ion thermal
* velocity. A temperature gradient such that the diffusive flux is set to
* match this leakage is imposed.
* 23 : prescribe the poloidally averaged value of the ion temperature and
* introduce a poloidal variation as close as possible to neoclassical
* solution. It is recommended to use this boundary condition together with
* corresponding condition on ion density (BCCON=21,22,23). The average is
* taken over all core boundaries with BCENI=23. ENIPAR(,1) specifies the
* temperature in eV
* 24 : feedback boundary condition with prescribed total ion flux, constant
* poloidally averaged ion temperature and a poloidal variation as close as
* possible to neoclassical solution. It is recommended to use this boundary
* condition together with corresponding condition on ion density
* (BCCON=21,22,23). The flux is summed over all core boundaries with
* BCENI=24. ENIPAR(,1) specifies the energy flux in W
* 25 : constant temperature feedback scaled by temperature on the ring
* bc_type21_ref away. ENIPAR(,1) specifies the desired ion temperature in
* eV. ENIPAR(,2) is the strength of the feedback
* 26 : prescribe the poloidally averaged value of the ion temperature and
* introduce a poloidal variation in a simplified manner. This boundary
* condition is suitable for any plasma composition (i.e. when BCENI=23
* fails). It is recommended to use this boundary condition together with
* corresponding condition on ion density (BCCON = 25,26,27). The average is
* taken over all core boundaries with BCENI=26. ENIPAR(,1) specifies the
* temperature in eV

```

```

*      27 : feedback boundary condition with prescribed total ion heat flux,
*      constant poloidally averaged ion temperature and a poloidal variation in
*      a simplified manner. This boundary condition is suitable for any plasma
*      composition (i.e. when BCENI=24 fails). It is recommended to use this
*      boundary condition together with corresponding condition on ion density
*      (BCCON = 25,26,27). The flux is summed over all core boundaries with
*      BCENI=27. ENIPAR(,1) specifies the energy flux in W
*  BCPOT - integer array, length NBC. Specifying the type of electric potential
*      or current boundary condition for each segment; makes use of POTPAR to
*      specify additional information, as indicated
*      0 : default, no boundary condition is applied
*      1 : prescribe the value of the potential, POTPAR(,1) specifies the
*      potential in V
*      2 : prescribe the gradient of the potential, POTPAR(,1) specifies the
*      potential gradient in  $V.m^{-1}$ 
*      3 : sheath conditions,
*      POTPAR(,2) used for biasing [see code for details]
*      4 : prescribe the value of the potential weakly a mixed boundary
*      condition, POTPAR(,1) specifies the potential in V and POTPAR(,2)
*      specifies the 'strength' of the boundary condition
*      5 : prescribe the current flux density per unit area, POTPAR(,1)
*      specifies the electric current flux density in  $A.m^{-2}$ 
*      6 : prescribe the total current flux density for a constant potential
*      [not yet available]
*      7 : prescribe the potential as a function of other plasma parameters [not
*      yet available]
*      8 : prescribe the total electric current with constant flux density,
*      POTPAR(,1) specifies the electric current in A
*      9 : prescribe the decay length for the potential, POTPAR(,1) specifies
*      the decay length in m
*      10 : feedback option for core [not yet tested!!!!!!!] (based on using
*      cbsch(0,coreregno))
*      11 : sheath conditions, electron energy transmission from b2stbc_spb,
*      POTPAR(,2) specifies the bias potential in V. Recommended for use in
*      cases with drifts, along with BCENE/I=15, BCCON=14, and BCMOM=13.
*      12 : Imposes the currents due to drifts for the South core boundary. Must
*      be used in conjunction with istyle_cur_contr_on_S_and_N.eq.2
*      13 : Imposes the currents due to drifts for the South private flux and
*      North boundaries. Must be used in conjunction with
*      istyle_cur_contr_on_S_and_N.eq.2
*      16 : Constant electric potential feedback on imposed total current. The
*      current prescribed is given by the sum of the POTPAR(IB,1) (in A) over
*      all the BCPOT=16 boundaries.
*      Still experimental, will not work for drift cases.
*      21 : constant potential feedback scaled by potential on the ring
*      'bc_type21_ref' away.
*      POTPAR(,1) specifies the desired potential in V.
*      POTPAR(,2) is the strength of the feedback
*  GAMMAI - real*8. Adiabatic coefficient multiplying the ion temperature when
*      computing the plasma sound speed.
*      Defaults to 1.0.
*  GAMMAE - real*8. Secondary electron emission coefficient.
*      Defaults to 0.5.
*  LBNDUSR - logical. If .true. will also call b2stbc_bas boundary condition
*      routine after the end of the physics boundary condition routine (governed
*      by the data from b2ah.dat and b2mn.dat).
*      Defaults to '.false.'.
*  LFEEDBACK - logical. Indicates whether a feedback scheme is used. Obsolete.
*      Superseded by 'b2stbc_feedback'.
*      Defaults to '.false.'.
*  NNISO - integer. Number of dead (or isolated) regions.
*      Defaults to 0.
*  NIISO - real*8 array of size (0:NS-1). Density of species (is) in  $m^{-3}$  to
*      impose in isolated regions.
*  TEISO - real*8. Electron temperature (in eV) to impose in isolated regions.
*      Defaults to 1.0.
*  TIISO - real*8. Ion temperature (in eV) to impose in isolated regions.

```

```

* Defaults to 1.0.
* PHIISO - real*8. Electric potential (in V) to impose in isolated regions.
* Defaults to 0.0.
* NXISO1 - integer array of size NNISO. Column number of bottom left corner of
* the dead region (iiso).
* Defaults to -2.
* NXISO2 - integer array of size NNISO. Column number of top right corner of
* the dead region (iiso).
* Defaults to -2.
* NYISO1 - integer array of size NNISO. Row number of bottom left corner of
* the dead region (iiso).
* Defaults to -2.
* NYISO2 - integer array of size NNISO. Row number of top right corner of the
* dead region (iiso).
* Defaults to -2.
* BOUNDARY_FILENAME - character*256. Name of the next file to use for reading
* a new /BOUNDARY/ namelist.
* Defaults to b2.boundary.parameters.
* BOUNDARY_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
* modulo(BOUNDARY_TIME_MOD), exceeds BOUNDARY_TIME_SWITCH, reads the new
* namelist from BOUNDARY_FILENAME. Also switches to the new namelist as the
* ELM count (here time/BOUNDARY_TIME_MOD) changes.
* Only active if BOUNDARY_TIME_MOD is greater than 0.
* Defaults to 0.0.
* BOUNDARY_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
* which a new /BOUNDARY/ namelist is read.
* Defaults to 0.0.
* LCBS - integer array of size (NBC). Indices of the core boundary segments in
* B2 for passing to EIRENE. Defaults to the list of 'S' boundaries in
* regions 1 and 5.
* WRITE_NML_BND - logical. If .true. (default), writes the content of the
* namelist to stdout after it has been read.
* Defaults to .true..
*
* NAMELIST /FEEDBACK/
* Found in b2.feedback.save.parameters.
* Read if present in the run directory.
* Used internally by B2 to regulate feedback loops and restarts.
* Not really intended for human use.
* See b2cdci and SOLPS manual for description of the feedback schemes.
* SAVED_CBSHE_CORE - real*8. Last value used for the core electron energy
* radial flux feedback.
* Defaults to 0.0.
* SAVED_CBSHI_CORE - real*8. Last value used for the core ion energy radial
* flux feedback.
* Defaults to 0.0.
* SAVED_CBSNA_CORE - real*8. Last value used for the core particle radial flux
* feedback. Corresponds to 'isfeedback' B2 species.
* Defaults to 0.0.
* SAVED_CBSCH_CORE - real*8. Last value used for the core radial electric
* current feedback.
* Defaults to 0.0.
* SAVED_CBSNA_PFR1 - real*8. Last value used for the radial inner PFR boundary
* flux feedback. Corresponds to 'isfeedback' B2 species.
* Defaults to 0.0.
* SAVED_CBSNA_PFR2 - real*8. Last value used for the radial outer PFR boundary
* flux feedback. Corresponds to 'isfeedback' B2 species.
* Defaults to 0.0.
* SAVED_CBSNA_SOL - real*8. Last value used for the radial particle flux
* feedback in the SOL. Corresponds to 'isfeedback' B2 species.
* Defaults to 0.0.
* SAVED_NA_FEEDBACK_ACTUATOR - real*8 array of size (NSPECIES). Last value
* used for the feedback control actuator for each isonuclear sequence.
* Defaults to 0.0.
*
* NAMELIST /FEEDBACK_CONTROL/
* Found in b2.feedback_control.parameters.

```

```

*   Allows for more complex feedback schemes.
*   This namelist is read if 'b2stbc_feedback' is set to '1' in the b2mn.dat
*   file or if LFEEDBACK=.true. in b2.boundary.parameters (and that latter file
*   is also read).
* VACUUM_COMMUNICATION - Integer. If > 0, allows for a
*   communication of particle fluxes across vacuum regions. This option only
*   applies to neutrals. The density boundary condition is based on the
*   difference between the average pressure and the local pressure.
*   Defaults to 0.
* VACUUM_COMMUNICATION_NREG - Integer array of size (NVAC). Number of
*   communicating vacuum regions.
*   Defaults to 0.
* VACUUM_COMMUNICATION_METHOD - Integer array of size (NVAC). Option for
*   resorbing the pressure difference.
*   1: Try to set a flux.  $\text{Corr} = (\text{beta} \cdot \text{ave\_pressure} - \text{pressure}) / \text{temp} \cdot \text{alpha}$ 
*   2: Try to set a density based on pressure equality.
*    $\text{Corr} = \text{beta} \cdot \text{pressure} / \text{Ti}$ 
*   Defaults to 0.
* VACUUM_COMMUNICATION_IY - Integer array of size (NVACREG,NVAC). Radial index
*   of the ring on which the neutral pressure is computed for region IREG.
*   Defaults to -2.
* VACUUM_COMMUNICATION_IX1 - Integer array of size (NVACREG,NVAC). Poloidal
*   lower bound of the range over which the neutral pressure is computed for
*   region IREG.
*   Defaults to -2.
* VACUUM_COMMUNICATION_IX2 - Integer array of size (NVACREG,NVAC). Poloidal
*   upper bound of the range over which the neutral pressure is computed for
*   region IREG.
*   Defaults to -2.
* VACUUM_COMMUNICATION_ALPHA - Real*8 array of size (0:NSPECIES-1,NVAC).
*   Parameter for setting the pressure correction. See above.
*   Defaults to 0.0.
* VACUUM_COMMUNICATION_BETA - Real*8 array of size (0:NSPECIES-1,NVAC).
*   Parameter for setting the pressure correction. See above.
*   Defaults to 1.0.
* NA_FEEDBACK_TARGET - Real*8 array of size (0:NSPECIES-1). Sets the target
*   value of the quantity associated with species (ISPECIES) for the feedback
*   scheme defined by NA_FEEDBACK_CHOICE(ISPECIES).
*   Defaults to 0.0.
* NA_FEEDBACK_TIME - Real*8 array of size (0:NSPECIES-1). Sets the time of
*   reference (in s) for the feedback of species (ISPECIES).
*   Defaults to 0.0.
* NA_FEEDBACK_CHOICE - Integer array of size (0:NSPECIES-1).
*   Choice of quantity on which the feedback is computed:
*   0: no action
*   1: local species density (averaged over the rectangle of
*   cells[IX1:IX2,IY1:IY2]). Summed over all charge states of that species.
*   2: local electron density (averaged over the rectangle of
*   cells[IX1:IX2,IY1:IY2]).
*   3: outer midplane separatrix electron density.
*   4: total particle content for that species summed over the rectangle of
*   cells[IX1:IX2,IY1:IY2].
*   5: total ion content for that species (not including neutrals) summed
*   over the rectangle of cells[IX1:IX2,IY1:IY2].
*   6: neutral particle flux through the core boundary.
*   7: relative average concentration of this species at the separatrix.
*   8: relative average concentration of this species averaged over the
*   rectangle of cells[IX1:IX2,IY1:IY2].
*   9: inner midplane separatrix electron density.
*   Defaults to 0.
* NA_FEEDBACK_OPTION - Integer array of size (0:NSPECIES-1). Option for
*   computing the new feedback quantity.
*   0: no action
*   1: rescale slowed by na_feedback_alpha
*   2: pure rescale
*   3: rescaling slowed by tanh_log
*   4: rescale done according to SOLPS4 formula

```

```

*      The target waveform for the particle content is
*       $N = C + V \cdot (\text{time} - T)$ 
*      and the current puffing rate  $S$  is adjusted
*       $S \rightarrow \max(0, \min(X, S + D))$ , where  $D = F \cdot (N - \langle N \rangle) / dt + (\langle N \rangle_{\text{prev}} - \langle N \rangle) / dt_{\text{prev}}$ 
*      5: rescale done according to SOLPS4 formula:  $N = C \cdot \exp((\text{time} - T) \cdot V)$ 
*      6: rescale slowed by na_feedback_alpha (SOLPS4 style)
*      Defaults to 0.
* NA_FEEDBACK_ACTUATOR - Integer array of size (0:NSPECIES-1). Choice for the
*      actuator used for the feedback.
*      0: no action
*      1: gas puff via boundary condition
*      2: rescale na
*      3: core boundary flux condition
*      Defaults to 0.
* NA_FEEDBACK_ALPHA - Real*8 array of size (0:NSPECIES-1). Factor by which the
*      rescaling is slowed. Rescaling factor is :
*      Option 1:  $(1 + \alpha \cdot \text{target} / \text{current}) / (1 + \alpha)$ 
*      Option 3:  $2 \cdot (\tanh(\log(x) / \beta) \cdot \log(\alpha) / \log(2))$ 
*      Options 4 and 5: Corresponds to parameter F
*      Defaults to 0.001.
* NA_FEEDBACK_BETA - Real*8 array of size (0:NSPECIES-1). Factor by which the
*      rescaling is slowed. See above. Options 4 and 5: Corresponds to parameter
*      V (ffb_rtvn)
*      Defaults to 1.0.
* NA_FEEDBACK_CONST - Real*8 array of size (0:NSPECIES-1). Options 4 and 5:
*      Corresponds to parameter C. If negative, C is
*      computed as the initial total particle content of the sequence.
*      Defaults to 0.0.
* NA_FEEDBACK_IX1 - Integer array of size (0:NSPECIES-1). Lower poloidal bound
*      for the region over which the quantity used for the feedback of species
*      (ISPECIES) is computed, if applicable.
*      Defaults to -2.
* NA_FEEDBACK_IX2 - Integer array of size (0:NSPECIES-1). Upper poloidal bound
*      for the region over which the quantity used for the feedback of species
*      (ISPECIES) is computed, if applicable.
*      Defaults to -2.
* NA_FEEDBACK_IY1 - Integer array of size (0:NSPECIES-1). Lower radial bound
*      for the region over which the quantity used for the feedback of species
*      (ISPECIES) is computed, if applicable.
*      Defaults to -2.
* NA_FEEDBACK_IY2 - Integer array of size (0:NSPECIES-1). Upper poloidal bound
*      for the region over which the quantity used for the feedback of species
*      (ISPECIES) is computed, if applicable.
*      Defaults to -2.
* NA_FEEDBACK_IB - Integer array of size (0:NSPECIES-1). Index of boundary
*      condition through which the feedback is being applied.
*      Defaults to -1.
* NA_FEEDBACK_PUFF_MIN - Real*8 array of size (0:NSPECIES-1). Minimum gas puff
*      being applied.
*      Defaults to 0.0.
* NA_FEEDBACK_PUFF_MAX - Real*8 array of size (0:NSPECIES-1). Maximum gas puff
*      being applied.
*      Defaults to 0.0.
* NA_FEEDBACK_OVERSHOOT - Real*8 array of size (0:NSPECIES-1). If the density
*      is larger than target*overshoot, the gas puff is turned off.
*      Defaults to 0.0.
*
* NAMELIST /PROFILE/
*      Found in b2.sources.profile.
*      Allows the user to set external sources to use within the code.
*      Unless specifying a divertor heat source, the source will be located in the
*      core and main SOL regions only.
*      Used if 'b2sral_inputfile' is set to 1.
* NSDATA - integer array of size (NKIND_DATA, NKIND_SOURCE, 0:NS). Number of
*      points over which the source profile of (kind_data, kind_source, is) is
*      defined. Should not exceed NY+2.

```

```

*      If KIND_DATA=1, the data is expressed as a profile in physical distance
*      from the separatrix (in metres) along the outer midplane.
*      The user can change this default reference location by use of the
*      'set_transport_i[xy]ref' switches.
*      If KIND_DATA=2, the data is expressed as a profile in flux distance from
*      the separatrix (not yet available).
*      Defaults to 0.
* SDATA - real*8 data of size (2,NY+2,NKIND_SOURCE,0:NS). For (i,ir,ik,is) in
*      (1:2,1:NY+2,1:NKIND_SOURCE,0:NS),
*      SDATA(1,ir,,:) contains the radial location of the profile point (ir).
*      SDATA(2,ir,,:) contains the source profile value at point (ir).
*      KIND_SOURCE=1 means particle source of species (is) (in particles/m3)
*      KIND_SOURCE=2 means parallel momentum source for species (is) (in
*      kg.m/s/m3)
*      KIND_SOURCE=3 means electron heat source (in Watts/m3)
*      KIND_SOURCE=4 means ion heat source (in Watts/m3)
*      KIND_SOURCE=5 means electric charge source (in Coulombs/m3)
*      KIND_SOURCE=6 means non-ambipolar electron particle source (in e/m3)
*      Defaults to 0.0.
* NXDATA - integer array of size (NKIND_DATA,NKIND_SOURCE,0:NS). Number of
*      points over which the axial profile of (kind_data,kind_source,is) is
*      defined. Should not exceed NX+2. Defaults to 0.
*      If KIND_DATA=1, the data is expressed as a profile in physical distance,
*      here connection length, rescaled from 0.0 to 1.0.
*      For closed field lines, the reference location for the zero of distance
*      is set by use of the 'set_transport_i[xy]ref' switches.
*      If KIND_DATA=2, the data is expressed as a profile in (ix) cell index,
*      again normalized from 0.0 to 1.0 to match the [0:nx-1] interval.
* XDATA - real*8 data of size (2,NY+2,NKIND_SOURCE,0:NS). Multiplier to the
*      poloidal source profile in the axial direction. Same convention for
*      KIND_SOURCE as above.
*      Defaults to 1.0.
* DIVHEAT - real*8. Additional divertor ion heat source (in Watts/m3)
*      Defaults to 0.0.
* SOURCES_FILENAME - character*256. Name of the next file to use for reading a
*      new /PROFILE/ namelist. Quantities not present in the new file will be
*      inherited from the old one.
*      Defaults to b2.sources.profile.
* SOURCES_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
*      modulo(SOURCES_TIME_MOD), exceeds SOURCES_TIME_SWITCH, reads the new
*      namelist from SOURCES_FILENAME. Also switches to the new namelist if the
*      ELM count (here time/SOURCES_TIME_MOD) changes. Only active if
*      SOURCES_TIME_MOD is greater than 0.
*      Defaults to 0.0.
* SOURCES_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
*      which a new /PROFILE/ namelist is read.
*      Defaults to 0.0.
*
* NAMELIST /TRANSPORT/
*      Found in b2.transport.inputfile.
*      Allows the user to set transport coefficients profiles.
*      Only used is 'b2tqna_inputfile' is set to 1.
*      Overwrites b2ah.dat, b2mn.dat and b2.transport.parameters.
*      See SOLPS manual for details.
* NDATA - integer array of size (NKIND_DATA,NCOEF,0:NS). Number of points over
*      which the source profile of (kind_data,kind_coef,is) is defined. Should
*      not exceed NY+2.
*      If KIND_DATA=1, the data is expressed as a profiles in physical distance
*      from the separatrix (in metres).
*      If KIND_DATA=2, the data is expressed as a profile in flux distance from
*      the separatrix (not yet available).
*      Defaults to 0.
* TDATA - real*8 data of size (3,NY+2,NKIND_COEFF,0:NS). For (i,ir,ik,is) in
*      (1:3,1:NY+2,1:NKIND_COEFF,0:NS), TDATA(1,ir,,:) contains the radial
*      location of the profile point (ir).
*      By default, these are measured at the outer midplane, as distance to the
*      separatrix in metres. The user can change this by setting the

```

```

*      'set_transport_i[xy]ref' switches to choose a different location for the
*      distance reference.
*      TDATA(2,ir,,:) contains the transport profile value at point (ir).
*      TDATA(3,ir,,:) contains the ELM transport profile value at point (ir).
*      KIND_COEFF=1 means density-driven particle diffusivity for species (is)
*      KIND_COEFF=2 means pressure-driven particle diffusivity for species (is)
*      KIND_COEFF=3 means ion heat diffusivity for species (is)
*      KIND_COEFF=4 means electron heat diffusivity
*      KIND_COEFF=5 means poloidal pinch velocity for species (is)
*      KIND_COEFF=6 means radial pinch velocity for species (is)
*      KIND_COEFF=7 means viscosity for species (is)
*      KIND_COEFF=8 means field-driven radial current conductivity
*      KIND_COEFF=9 means temperature-driven radial current conductivity
*      Defaults to 0.0.
* ADDSPEC - integer array of size (NS,NKIND_COEFF,0:NS). If
*      ADDSPEC(is,ikind,spec).ge.0, then the transport coefficient profile of
*      type (ikind) from species (spec) is also used for species with index
*      ADDSPEC(is,ikind,spec).
*      Defaults to -5.
* TRANSPORT_FILENAME - character*256. Name of the next file to use for reading
*      a new /TRANSPORT/ namelist.
*      Defaults to b2.transport.inputfile.
* TRANSPORT_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
*      modulo(TRANSPORT_TIME_MOD), exceeds TRANSPORT_TIME_SWITCH, reads the new
*      namelist from TRANSPORT_FILENAME. Also switches to the new namelist if the
*      ELM count (here time/TRANSPORT_TIME_MOD) changes. Only active if
*      TRANSPORT_TIME_MOD is greater than 0.
*      Defaults to 0.0.
* TRANSPORT_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
*      which a new /TRANSPORT/ namelist is read.
*      Defaults to 0.0.
* REGION_FLAGS - logical array of size (NREG,NKIND_COEFF). If
*      region_flags(ireg,ikind) is .true. (default), then the transport
*      parameters profiles for region (ireg) and kind (ikind) are used.
*      Defaults to .true..
* NO_PFLUX - logical. If .true., transport coefficients profiles are not
*      implemented in the private flux regions.
*      Defaults to '.false.'.
* NO_DIV - logical. If .true., transport coefficients profiles are not
*      implemented in the divertor regions.
*      Defaults to '.false.'.
* POLOIDAL_SCALING - logical array of size (10). If .true., then the transport
*      coefficients profiles from the current b2.transport.inputfile are
*      increased by a factor of 1.0+Gaussian where Gaussian is a Gaussian profile
*      in the poloidal direction of amplitude SCALING_STRENGTH extending from
*      SCALING_IX_BEGIN to SCALING_IX_END inclusively. The profile has a decay
*      length of SCALING_WIDTH (in units of the number of poloidal cells).
*      Defaults to '.false.'.
* SCALING_STRENGTH - real*8 array of size 10. See above.
*      Defaults to 0.
* SCALING_WIDTH - real*8 array of size 10. See above. Defaults to about 1/3 of
*      the interval over which the scaling is to be done.
*      Defaults to 1/3 interval.
* SCALING_IX_BEGIN - integer array of size 10. See above.
*      Defaults to -2.
* SCALING_IX_END - integer array of size 10. See above.
*      Defaults to -2.
* ELM_TIME_BEGIN - real*8. Time (in seconds) modulo ELM_TIME_PERIOD at which
*      the ELM phase begins and the ELM data must be used.
*      Defaults to 0.0.
* ELM_TIME_END - real*8. Time (in seconds) modulo ELM_TIME_PERIOD at which the
*      ELM phase ends and the ELM data is no longer used.
*      Defaults to 0.0.
* ELM_TIME_PERIOD - real*8. Indicates the real frequency of simulated ELMS.
*      See above for usage.
*      If zero, no ELM profiles are used.
*      Defaults to 0.0.

```



```

* ELM_IX_BEGIN - integer. Poloidal position at which the ELM profile starts to
*   be applied.
*   Defaults to -2.
* ELM_IX_END - integer. Poloidal position at which the ELM profile ends being
*   applied.
*   Defaults to -2.
*
* NAMELIST /NEUTRALS_SAVE/
*   Found in b2.neutrals_save.parameters.
*   Read if file is present.
*   Used internally for regulation and rescaling of volume recombination
*   sources and restarts. Superseded by use of 'eirene_dpc_fix' switch. See
*   code in eirene_mc and description in b2cdci for details.
*   SAVED_VOLREC - real*8 array of size (NSTRAT). Contains the last value of the
*   strength of volume recombination sources from stratum (istra).
*   Defaults to 0.0.
*
* NAMELIST /NUMERICS/
*   Found in b2.numerics.parameters.
*   Used to specify time multipliers per equation and per region for the B2
*   solver.
*   Active is 'b2srdt_numerics_namelist' is set to 1 or greater.
*   It is recommended to use the same multiplier for all equations within one
*   region.
*   DTCO - real*8 array of size (0:NS-1,0:NREG). Multiplier to the time used in
*   solving the continuity equation of species (is) in region (ireg).
*   Defaults to 1.0.
*   DTMO - real*8 array of size (0:NS-1,0:NREG). Multiplier to the time used in
*   solving the parallel momentum equation of species (is) in region (ireg).
*   Defaults to 1.0.
*   DTEE - real*8 array of size (0:NREG). Multiplier to the time used in solving
*   the electron heat equation in region (ireg).
*   Defaults to 1.0.
*   DTEI - real*8 array of size (0:NREG). Multiplier to the time used in solving
*   the ion heat equation in region (ireg).
*   Defaults to 1.0.
*   SOLVECO - logical array of size (0:NS-1,0:NREG). Indicates whether the
*   continuity equation for species (is) is to be solved in region (ireg).
*   Subservient to 'no_solve' switch.
*   Defaults to .true..
*   SOLVEMT - logical array of size (0:NREG). Indicates whether the total
*   parallel momentum equation is to be solved in region (ireg). For the total
*   parallel momentum equation to be solved in a region, all momentum
*   equations in the range defined by 'b2news_nsmin' and 'b2news_nsmax' must
*   have SOLVEMO true in that region. Subservient to 'no_solve' switch. It is
*   important to note that the 'b2news_no_solve' switch allows for solving or
*   not all momentum equations together as a whole, along with the total
*   equation.
*   Defaults to .true..
*   SOLVEMO - logical array of size (0:NS-1,0:NREG). Indicates whether the
*   parallel momentum equation for species (is) is to be solved in region
*   (ireg). Subservient to 'no_solve' switch.
*   Defaults to .true..
*   SOLVEMT - logical array of size (0:NREG). Indicates whether the total
*   parallel momentum equation is to be solved in region (ireg). For the total
*   parallel momentum equation to be solved in a region, all momentum
*   equations in the range defined by 'b2news_nsmin' and 'b2news_nsmax' must
*   have SOLVEMO true in that region. Subservient to 'no_solve' switch.
*   It is important to note that the 'b2news_no_solve' switch allows for
*   solving or not all momentum equations together as a whole, along with the
*   total equation.
*   Defaults to .true..
*   SOLVEPO - logical array of size (0:NREG). Indicates whether the potential
*   energy equation is to be solved in region (ireg). Subservient to
*   'no_solve' switch.
*   Defaults to .true..
*   SOLVEEE - logical array of size (0:NREG). Indicates whether the electron

```

```

*     energy equation is to be solved in region (ireg). Subservient to
*     'no_solve' switch.
*     Defaults to .true..
* SOLVEEI - logical array of size (0:NREG). Indicates whether the ion energy
* equation is to be solved in region (ireg). Subservient to 'no_solve'
* switch.
* Defaults to .true..
* SOLVEET - logical array of size (0:NREG). Indicates whether the total energy
* equation is to be solved in region (ireg). For the equation to be solved
* in region (ireg), both SOLVEEE and SOLVEEI must be true in that region.
* Subservient to 'no_solve' switch.
* It is important to note that the 'b2news_no_solve' switch allows to solve
* or not to solve the electron and ion heat equations together with total
* energy equation, as a group.
* Defaults to .true..
* TIME_FACTOR_REQUIRED - real*8. Minimum time scale of evolution allowed for
* all equations. Only active if 'b2srsm_enable' is set to 1.
* Defaults to 0.1.
* CORE_DT_SUPPRESSION - real*8. De-multiplier to the timestep in the core.
* Only active if less than 1. Should be larger than 0. Applies fully to the
* innermost core ring of cells (IY .eq. -1). See CORE_DT_FACTOR for further
* use.
* Defaults to 1.0.
* CORE_DT_FACTOR - real*8. Multiplier to the timestep in the core. Only active
* is less than 1. Should be larger than 0. Multiplies each successive core
* ring of cells (increasing IY) by CORE_DT_FACTOR, until the local time step
* multiplier is equal to 1.
* Defaults to 1.0.
* CORR_CORE_DN - real*8 array of size (0:NS-1). Pressure correction speed-up
* parameter _a, acting on the density contribution from species a. Will
* apply in the core region, except for the nrings outer surfaces. See
* nrings_for_no_speedup_averaging for details.
* See Pressure_correction_speed-up.pdf in $SOLPSTOP/doc for a full
* description. Should be roughly equal to corr_core_dt below. Does not
* apply to neutral species.
* Defaults to 1.0.
* CORR_CORE_DT - real*8. Pressure correction speed-up parameter _T, acting on
* the temperature contributions. Will apply in the core region, except for
* the nrings outer surfaces. See nrings_for_no_speedup_averaging for
* details.
* See Pressure_correction_speed-up.pdf in $SOLPSTOP/doc for a full
* description. Should be roughly equal to corr_core_dn above.
* Defaults to 1.0.
* NUMERICS_FILENAME - character*256. Name of the next file to use for reading
* a new /NUMERICS/ namelist.
* Defaults to b2.numerics.namelist.
* NUMERICS_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
* modulo(NUMERICS_TIME_MOD), exceeds NUMERICS_TIME_SWITCH, reads the new
* namelist from NUMERICS_FILENAME. Also switches to the new namelist as the
* ELM count (here time/NUMERICS_TIME_MOD) changes. Only active if
* NUMERICS_TIME_MOD is greater than 0.
* Defaults to 0.0.
* NUMERICS_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
* which a new /NUMERICS/ namelist is read.
* Defaults to 0.0.
* WRITE_NML_NUM - logical. If .true. (default), writes the content of the
* namelist to stdout after it has been read.
* Defaults to .true..
* ADD_TE_CORR_TO_PO - logical array of size (0:NREG). If .true. (default),
* adds dte(ix,iy)/qe to the potential correction after the internal energy
* balance equations are solved, where dte(ix,iy) is the electron temperature
* correction on the time step. Individually set for each region index.
* Defaults to .true..
*
* NAMELIST /TRANSPORT_MODELS/
* Found in b2.transport_models_save.parameters.
* Read if file is present.

```

```

*   ETA_HCE_MULT - real*8 array of size (-1:NY). Used by the user specified
*   set_transport_eta transport model. See code for details.
*   Defaults to 1.0.
*
* NAMELIST /TRANSPORT/
*   Found in b2.transport.parameters.
*   Allows user to choose between different anomalous transport models.
*   Read if 'b2tqna_transport_namelist' is set to 1.
*   Overwrite data from b2ah.dat and b2mn.dat.
*   Does not apply to fluid neutral species.
* FLAG_DNA - integer. Transport model flag for density-driven diffusion.
* FLAG_DPA - integer. Transport model flag for pressure-driven diffusion.
* FLAG_VLA - integer. Transport model flag for pinch velocity.
* FLAG_VSA - integer. Transport model flag for viscosity.
* FLAG_HCI - integer. Transport model flag for ion heat diffusivity.
* FLAG_HCE - integer. Transport model flag for electron heat diffusivity.
* FLAG_SIG - integer. Transport model flag for field-driven current radial
*   conductivity.
* FLAG_ALF - integer. Transport model flag for temperature-driven current
*   radial conductivity.
*   All flags follow:
*   FLAG=0: Use model from b2ah.dat and b2mn.dat. Default.
*   FLAG=1: Constant, value set to PARM.
*   FLAG=2: 1/N model, multiplied by PARM.
*   FLAG=3: Bohm model, multiplied by PARM.
*   FLAG=4: flux-scaled model, multiplied by PARM.
*
* PARM_DNA - real*8 array of size (0:NS-1). Parameter for the density-driven
*   particle diffusion coefficient for species (is).
* PARM_DPA - real*8 array of size (0:NS-1). Parameter for the pressure-driven
*   particle diffusion coefficient for species (is).
* PARM_VLA - real*8 array of size (0:NS-1). Parameter for the anomalous radial
*   pinch velocity for species (is).
* PARM_VSA - real*8 array of size (0:NS-1). Parameter for the viscosity for
*   species (is).
* PARM_HCI - real*8 array of size (0:NS-1). Parameter for the heat diffusivity
*   coefficient for species (is).
* PARM_HCE - real*8. Parameter for the electron heat diffusivity coefficient.
* PARM_SIG - real*8. Parameter for the anomalous radial field-driven current
*   conductivity.
* PARM_ALF - real*8. Parameter for the anomalous radial temperature-driven
*   current conductivity.
* TRANSPORT_FILENAME - character*256. Name of the next file to use for reading
*   a new /TRANSPORT/ namelist.
*   Defaults to b2.transport.parameters.
* TRANSPORT_TIME_MOD - real*8. When the B2 run simulation time, in seconds,
*   modulo(TRANSPORT_TIME_MOD), exceeds TRANSPORT_TIME_SWITCH, reads the new
*   namelist from TRANSPORT_FILENAME. Also switches to the new namelist if the
*   ELM count (here time/TRANSPORT_TIME_MOD) changes.
*   Only active if TRANSPORT_TIME_MOD is greater than 0.
*   Defaults to 0.0.
* TRANSPORT_TIME_SWITCH - real*8. Time (in seconds) within an ELM cycle after
*   which a new /TRANSPORT/ namelist is read.
*   Defaults to 0.0.
* CFLME - real*8. Multiplier to the electron heat flux limit.
*   Defaults to Default inherited from b2mn.dat.
* CFLMI - real*8. Multiplier to the ion heat flux limit.
*   Defaults to Default inherited from b2mn.dat.
* CFLMV - real*8. Multiplier to the viscous heat flux limit.
*   Defaults to Default inherited from b2mn.dat.
* CFLAL - real*8. Multiplier to the thermo-electric coefficient flux limit.
*   Defaults to Default inherited from b2mn.dat.
* CFLAB - real*8. Multiplier to the friction force flux limit.
*   Defaults to Default inherited from b2mn.dat. For all flux limit
*   multipliers, setting them to zero turns off the corresponding flux
*   limiter.
*

```

```

* WRITE_NML_TRANSP - logical. If .true. (default), writes the content of the
*   namelist to stdout after it has been read.
*   Defaults to .true..
* VOUT_CNV - real*8 array of size (0:NS-1). Value of the "blob" convection
*   velocity for species (is) at the outer grid edge (in m/s).
*   Defaults to 0.0.
* PW0_CNV - real*8 array of size (0:NS-1). Exponent in radial profile of the
*   "blob" velocity for species (is).
*   Defaults to 1.0.
* PW1_CNV - real*8 array of size (0:NS-1). First exponent in poloidal profile
*   of the "blob" velocity for species (is).
*   Defaults to 5.0.
* PW2_CNV - real*8 array of size (0:NS-1). Second exponent in a poloidal
*   profile of the "blob" velocity for species (is).
*   Defaults to 2.0. The four *_CNV parameters are only activated if
*   'b2tqna_user_transport' is set to '6'.
*
*
* NAMELIST /USER/
*   Found in b2.user.parameters.
*   Contains additional data for user-specific diagnostics.
*   This version provided by Andrei Kukushkin for ITER cases.
* LHETRGETS - integer array of size (NLIM). List of surface indices (EIRENE
*   notation) which are used for calculation of helium enrichment.
*   Defaults to Eirene recycling target surfaces defined in the LTNS array.
* LPFRB_I - integer. B2 x-cell index corresponding to the first edge of the
*   bypass between the inner and outer divertor (Western divertor, edge
*   closest to target). If non-positive, counted backwards from the X-point
*   location in the lower PFR, from the inner upper target in the upper PFR,
*   from the lower outer target in the outer SOL, and from the inner upper
*   target in the inner SOL.
*   Defaults to 0.
* LPFRB_O - integer. B2 x-cell index corresponding to the first edge of the
*   bypass between the inner and outer divertor (Eastern divertor, edge
*   closest to target). If non-positive, counted backwards from the outer
*   lower target in the lower PFR, from the lower outer target in the outer
*   SOL, from the upper outer target in the upper PFR, and from the inner
*   upper target in the inner SOL.
*   Defaults to 0.
* LPFRT_I - integer. B2 x-cell index corresponding to the second edge of the
*   bypass between the inner and outer divertor (Western divertor, edge
*   furthest to target). If non-positive, counted backwards from the X-point
*   location in the PFR, from the outer target in the outer SOL for a SN, from
*   the top targets in the SOL for a DN.
*   Defaults to 0.
* LPFRT_O - integer. B2 x-cell index corresponding to the second edge of the
*   bypass between the inner and outer divertor (Eastern divertor, edge
*   furthest to target). If non-positive, counted backwards from the X-point
*   location in the PFR, from the outer target in the outer SOL for a SN, from
*   the top targets in the SOL for a DN.
*   Defaults to 0.
* J_HE_AT - integer. Species index of the helium atoms in Eirene. The code
*   attempts to find a match by default.
* J_NE_AT - integer. Species index of the neon atoms in Eirene. The code
*   attempts to find a match by default.
* J_H_AT - integer array of size (3). Species indices of the hydrogen isotopes
*   in Eirene. If using only one hydrogen species, only the first element
*   needs to be provided. Otherwise the 3 elements correspond to H/D/T. The
*   code attempts to find a match by default.
* L_H_MOL - integer array of size (NMOL,3). Number of hydrogen isotope nuclei
*   for molecules in Eirene. If using only one hydrogen species, only the
*   first element needs to be provided. Otherwise the 3 elements correspond to
*   H/D/T. The code attempts to find a match by default.
* FUSION_POWER - real*8. Fusion power occurring in core (including neutrons, in
*   Megawatts).
*   Defaults to 0.0.
* SPMP_HE_TO_D - real*8. Ratio of He to DT pumping speeds (typically, 0.8).

```

```

* Defaults to 1.0.
* LPFRS_PMP - integer. Location of the pump. 0 no pump at all (default), 1 -
* SN of lower DN PFR, 2 - (outer) SOL, 3 - upper DN PFR, 4 - inner SOL for
* DN
* Defaults to 0.
* NPFRGRP - integer. Actual number of surface groups for PFR flows.
* Defaults to 0.
* LPFRGRP - integer array of size (NLIM). List of surface segments for PFR
* flows.
* IPFRGRP - integer array of size (NPFRGRP). First positions in this list for
* each group.
* Defaults to 0.
* JPFRGRP - integer array of size (NPFRGRP). Last positions in this list for
* each group.
* Defaults to 0.
* GPFRGRP - character*8 array of size (NPFRGRP). Labels for surface groups for
* PFR flows.
* Defaults to ' '.
* NNTRGRP - integer. Actual number of surface groups for neutral data.
* LNTRGRP - integer array of size (NLIM). List of surface segments for neutral
* data.
* INTRGRP - integer array of size (NNTRGRP). First positions in this list for
* each group.
* JNTRGRP - integer array of size (NNTRGRP). Last positions in this list for
* each group.
* GNTRGRP - character*8 array of size (NNTRGRP). Labels for the neutral data
* surface groups.
* SPMP_NOM - real*8. Nominal pumping speed.
* Defaults to 0.0.
* FILEDATA - logical array of size 10. Switches to turn on/off the tracing
* files controlled by the ank_tracing switch. The files are defined in
* b2mod_diag, in order, starting with element 2 of the filedata array:
* test.trc, residuals.trc, sources.trc, blnn.trc, blne.trc, integral.trc,
* user.trc, blnm.trc, sepdata.trc.
* If there is no b2.user.parameters file present, the flag for user.trc is
* set to .false..
* If the tracing files are to be appended but a reading error occurs when
* opening them, the corresponding filedata element is overwritten to
* .false..
* Defaults to .true..
* USER_FILENAME - character*80. Filename where /USER/ namelist is stored.
* Defaults to b2.user.parameters.
*
* NAMELIST /SPUTTER_SAVE/
* Found in b2.sputter_save.parameters.
* Read if file is present.
* Used to regulate the temporal evolution of sputtering and restarts.
* Not intended for human use.
* NX_SP - integer. Array dimension NX used in this namelist.
* Defaults to 'nx'.
* NY_SP - integer. Array dimension NY used in this namelist.
* Defaults to 'ny'.
* NS_SP - integer. Array dimension NS used in this namelist.
* Defaults to ns.
* SPUTTER_YIELD - real*8 array of size (-1:NX,-1:NY,0:NS-1,1:2). Contains the
* chemical sputtering yield of species (is) at position (ix,iy) in element
* (:,:,1) and the physical sputtering yield in element (:,:,2).
* Defaults to 0.0.
* SPUTTER_YIELD2 - real*8 array of size (-1:NX,-1:NY,0:NS-1,1:2). Contains the
* energy chemical sputtering yield of species (is) at position (ix,iy) in
* element (:,:,1) and the energy physical sputtering yield in
* element (:,:,2).
* Defaults to 0.0.
*
* NAMELIST /ATOMIC_PHYSICS_RESCALE/
* Found in b2.atomic_physics_rescale.parameters.
* Read if 'b2mndr_atomic_physics_rescale' is set to 1.

```

```

*   Contains rescaling multipliers for atomic physics rates.
*   RESCALE_SA - real*8 array of size(0:NS-1). Scaling factors for rtas:
*       ionisation rates of species (is).
*       Defaults to 1.0.
*   RESCALE_RA - real*8 array of size(0:NS-1). Scaling factors for rtra:
*       recombination rates of species (is).
*       Defaults to 1.0.
*   RESCALE_QA - real*8 array of size(0:NS-1). Scaling factors for rtqa:
*       electron cooling rates of species (is).
*       Defaults to 1.0.
*   RESCALE_CX - real*8 array of size(0:NS-1). Scaling factors for rtcx: charge
*       exchange rates of species (is).
*       Defaults to 1.0.
*   RESCALE_RD - real*8 array of size(0:NS-1). Scaling factors for rtrd: line
*       radiation rates of species (is).
*       Defaults to 1.0.
*   RESCALE_BR - real*8 array of size(0:NS-1). Scaling factors for rtbr:
*       bremsstrahlung radiation rates of species (is).
*       Defaults to 1.0.
*
*-----
*.end b2cdcn

      end subroutine b2cdcn

```

Appendix B

The physics of B2.5

An extensive review of B2-Eirene has been published as [15]. It is asked that authors using SOLPS simulation results cite that work when referring to the code. This document, only some short sections of which are detailed below, can be found as part of the SOLPS package at $\{\text{SOLPSTOP}\}/\text{doc}/\text{CPP-2006.pdf}$.

The equations are written on a curvilinear orthogonal coordinate system. The x -coordinate varies along flux surfaces and the y -coordinate varies perpendicular to flux surfaces (see figure B.1), z is the toroidal direction.

The metric coefficients are $h_x = 1/\|\nabla x\|$, $h_y = 1/\|\nabla y\|$ and $\sqrt{g} = h_x h_y h_z$. One can replace h_z by $2\pi R$, where R is the local major radius. Components of a vector are the projections on unit basis vectors; they are not the contravariant or covariant components ($b_x = B_x/B$, $b_z = B_z/B$). Ion velocities $V_\perp$ (velocity perpendicular both to magnetic field B and the y -axis) and V_y are determined from the two components of the momentum balance equation for ions perpendicular to a magnetic field. Using the Braginskii expression for the friction force, one obtains:

$$V_\perp = V_\perp^{(a)} - \frac{1}{enB} \frac{1}{h_y} \frac{\partial n T_i}{\partial y} + V_\perp^{(in)} + V_\perp^{(vis)} + V_\perp^{(s)} \quad (\text{B.1})$$

$$V_y = V_y^{(a)} + \frac{B_z}{enB^2} \frac{1}{h_x} \frac{\partial n T_i}{\partial x} + V_y^{(in)} + V_y^{(vis)} + V_y^{(s)} \quad (\text{B.2})$$

Velocities $V_{\perp,y}^{(a)}$ are the sum of the $\vec{E} \times \vec{B}$ drifts and diffusive and thermo-diffusive velocities. This part corresponds to the ambipolar flux. It is the same both for ions and electrons. The second term in the r.h.s. of Eqs. (B.1) and (B.2) corresponds to diamagnetic flow of ions, while the other terms are caused by inertial, viscous and ion-neutral friction forces. These non-ambipolar terms are connected with currents: $\vec{V}^{(in)} = \vec{j}^{(in)}/en$, $\vec{V}^{(vis)} = \vec{j}^{(vis)}/en$, $\vec{V}^{(s)} = \vec{j}^{(s)}/en$.

In the presence of anomalous transport (D is the classical diffusion coefficient, $D = \frac{T_e + T_i}{eb} \frac{\nu_{ei}}{\omega_{ce}}$, with electron cyclotron frequency ω_{ce} and electron-ion collision frequency ν_{ei})

$$V_\perp^{(a)} = -\frac{1}{B} \frac{1}{h_y} \frac{\partial \Phi}{\partial y} - \frac{D}{T_e + T_i} \frac{b_z}{h_x} \left(\frac{\partial p}{n \partial x} - \frac{3}{2} \frac{\partial T_e}{\partial x} \right) - D_n^{(AN)} \frac{1}{h_x n} \frac{\partial n}{\partial x} - D_p^{(AN)} \frac{1}{h_x n} \frac{\partial p_i}{\partial x} + V_\perp^{(AN)} \quad (\text{B.3})$$

$$V_y^{(a)} = \frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x} - \frac{D}{T_e + T_i} \frac{1}{h_y} \left(\frac{\partial p}{n \partial y} - \frac{3}{2} \frac{\partial T_e}{\partial y} \right) - D_n^{(AN)} \frac{1}{h_y n} \frac{\partial n}{\partial y} - D_p^{(AN)} \frac{1}{h_y n} \frac{\partial p_i}{\partial y} + V_y^{(AN)} \quad (\text{B.4})$$

The poloidal velocity is a sum $V_x = b_z V_\perp + b_x V_\parallel$.

In the particle balance the fact that the diamagnetic velocity is almost divergence free is taken into account. Therefore, instead of the diamagnetic velocity it is possible to introduce an effective velocity $\vec{V}^{(dia)}$, which gives the same

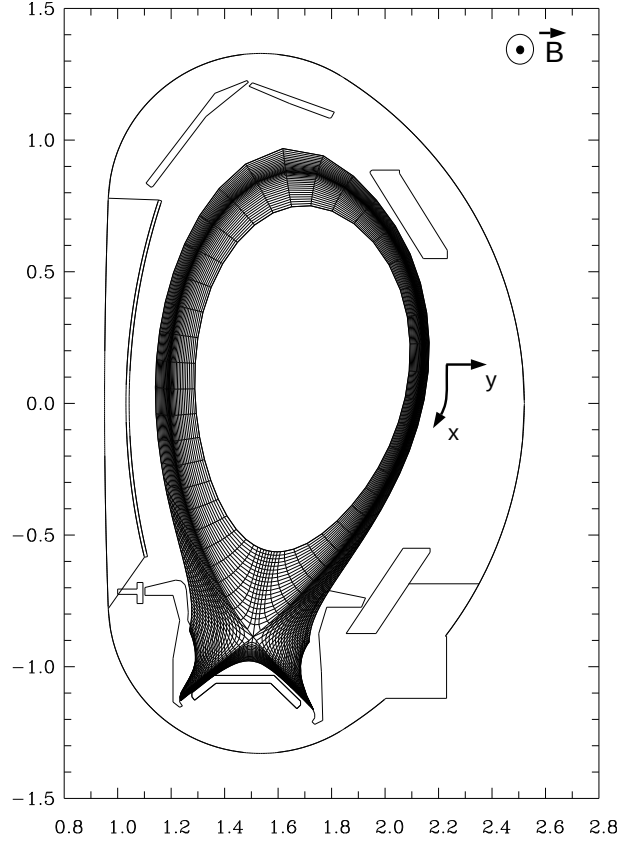


Figure B.1: Coordinate system: x is the poloidal coordinate, y is the radial coordinate orthogonal to the flux surfaces. The coordinate $\perp$ corresponds to the direction perpendicular both to the magnetic field and to the y -axis. The directions of magnetic field and plasma current correspond to normal operation conditions of ASDEX-Upgrade (ion ∇B drift directed towards the X-point).

contribution to the divergence of the particle flux:

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n (b_x V_{\parallel} + b_z V_{\perp}^{(0)}) \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n V_y^{(0)} \right) = S^n \quad (\text{B.5})$$

$$V_{\perp}^{(0)} = V_{\perp}^{(a)} + V_{\perp}^{(in)} + V_{\perp}^{(vis)} + V_{\perp}^{(s)} + \tilde{V}_{\perp}^{(dia)} \quad (\text{B.6})$$

$$V_y^{(0)} = V_y^{(a)} + V_y^{(in)} + V_y^{(vis)} + V_y^{(s)} + \tilde{V}_y^{(dia)} \quad (\text{B.7})$$

$$\tilde{V}_{\perp}^{(dia)} = \frac{T_i B_z}{e} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \quad (\text{B.8})$$

$$\tilde{V}_y^{(dia)} = -\frac{T_i B_z}{e} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \quad (\text{B.9})$$

The velocity $\tilde{V}^{(dia)}$ is an effective velocity. It represents vertical guiding centre drift of ions caused by ∇B .

Combining the inertia and gyroviscosity terms in the Braginskii equations, one finds

$$\begin{aligned}
m_i \left[\frac{\partial n V_{\parallel}}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n \left(V_{\perp}^{(0)} b_z + V_{\parallel} b_x \right) V_{\parallel} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n V_y^{(0)} V_{\parallel} \right) \right] = \\
- \frac{b_x}{h_x} \frac{\partial n T_i}{\partial x} - b_x \frac{e n}{h_x} \frac{\partial \Phi}{\partial x} + F_k + \frac{4}{3} b_x B^{3/2} \frac{\partial}{h_x \partial x} \left(\frac{\eta_0 b_x}{B^2} \frac{\partial \left(\sqrt{B} \left(V_{\parallel} + \frac{B}{B_x} \left(V_x^{(dia)} + V_x^{(E)} \right) \right) \right)}{h_x \partial x} \right) \\
+ B^{3/2} b_x \frac{\partial}{h_x \partial x} \left(\frac{b_x}{\nu_{ii} B^2} \frac{\partial \left(\sqrt{B} q_{i\parallel}^{(0)} \right)}{h_x \partial x} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y^2} \eta_2 \frac{\partial V_{\parallel}}{\partial y} \right) \\
+ \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x^2} \eta_2 \frac{\partial V_{\parallel}}{\partial x} \right) + S_{i\parallel}^m + R_{ie\parallel}.
\end{aligned} \tag{B.10}$$

The diamagnetic terms in the l.h.s. of (B.10) are almost cancelled [16, 17, 18, 19], so that velocities normal to the flux surface in the second and third term, as in the particle balance equation, are determined by (B.1) and (B.2). In [19] the divergent part of diamagnetic velocity has been neglected. Here, it is kept, because the corresponding flux is still relatively large and can be comparable with the diffusive flux even for anomalous diffusion coefficients. The authors of [16, 17, 18] also kept this flux. In contrast to [16, 17, 18] there is an additional term of the same kind in the r.h.s. of Eq. (B.10):

$$F_k = -m_i \left(\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n V_{\parallel} \tilde{V}_{\perp}^{(dia)} b_z \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n V_{\parallel} \tilde{V}_y^{(dia)} \right) \right) \tag{B.11}$$

This parallel force is the Coriolis force, connected with the diamagnetic drift velocity and toroidal (parallel) rotation. This force can be obtained in a different way by averaging the drift kinetic equation [20] and is of the same order as the inertia terms.

Parallel viscosity is represented by the fourth and fifth terms in the r.h.s. The former is the Balescu parallel viscosity with coefficient $\eta_0 = \frac{4}{3} 0.96 \sqrt{2} n T_i / \nu_{ii}$. A detailed derivation is given at the end of this section, see B.0.1. The latter corresponds to the parallel viscosity driven by parallel ion heat flux $q_{i\parallel}^{(0)} = -\kappa_{i,\parallel} b_x \frac{1}{h_x} \frac{\partial T_i}{\partial x}$. This viscosity is absent in the Braginskii equations, however, in a tokamak, where parallel particle flux multiplied by temperature is of the order of the ion heat flux, these two viscosities are of the same order. On closed flux surfaces far from the separatrix, the flux-surface-averaged parallel viscosity coincides with the well-known neoclassical expression for the Pfirsch-Schlüter regime [21]. Thus, parallel viscosity in Eq. (B.10) provides smooth transition to neoclassical theory. The classical perpendicular viscosity coefficient can be replaced by an anomalous value as in [16, 17, 18]: $\eta_2 = n m_i D_n^{(AN)}$. Perpendicular viscosity and inertia terms with the anomalous particle flux are responsible for the transport of the parallel momentum in radial and poloidal directions. The radial transport can be rather important despite the fact that $\eta_2 \ll \eta_0$, since the scale in the radial direction is rather small.

The term $S_{i\parallel}^m$ represents the parallel momentum loss due to ion-neutral friction and parallel momentum input from neutral beam injection. $R_{ie\parallel}$ is the ion-electron friction.

The parallel momentum balance for electrons has the standard form:

$$j_{\parallel} = \sigma_{\parallel} \left(\frac{b_x}{e} \frac{1}{h_x} \left(\frac{\partial n T_e}{\partial x} + 0.71 \frac{\partial T_e}{\partial x} \right) - \frac{b_x}{h_x} \frac{\partial \Phi}{\partial x} \right). \tag{B.12}$$

Expressions for $j_{\perp}$ and j_y are obtained from radial and poloidal projections of the total momentum balance equation (the sum of electron and ion momentum balance equations). The resulting current is a sum of contributions from pressure gradient (diamagnetic current) $\vec{j}^{(dia)}$, inertia and gyroviscosity $\vec{j}^{(in)}$, viscosity $\vec{j}^{(vis)}$ and ion-neutral friction $\vec{j}^{(s)}$. Similar to the particle continuity equation, the effective current $\vec{j}$ is introduced, which gives the same contribution

to $\nabla \vec{j}$, as the ordinary current. This gives the current continuity equation

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{j}_x \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{j}_y \right) = 0. \quad (\text{B.13})$$

$$\tilde{j}_x = b_z \tilde{j}_\perp + b_x \tilde{j}_\parallel. \quad (\text{B.14})$$

$$\vec{j} = \vec{j}^{(dia)} + \vec{j}^{(in)} + \vec{j}^{(vis)} + \vec{j}^{(s)} + \vec{j}_\parallel. \quad (\text{B.15})$$

The effective diamagnetic current can be chosen in a form

$$\tilde{j}_\perp^{(dia)} = \frac{1}{b_z} \frac{n(T_e + T_i) B_z}{h_y} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right), \quad (\text{B.16})$$

$$\tilde{j}_y^{(dia)} = -\frac{n(T_e + T_i) B_z}{h_x} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right). \quad (\text{B.17})$$

Current driven by parallel viscosity should be taken into account in Eqs. (B.13), (B.14) and (B.15). This current is a small correction to the diamagnetic current Eqs. (B.16) and (B.17). However, on closed flux surfaces, where the pressure is almost constant, the diamagnetic current averaged over the flux surfaces tends to zero. The resulting remaining part can become comparable to the parallel viscosity-driven current, which is necessary to satisfy the condition that the averaged net radial current should be zero. The terms, which contain the plasma viscosity, produce almost divergence-free currents. Similar to the case of diamagnetic currents, it can be reduced to

$$\tilde{j}_\perp^{(vis\parallel)} = -\frac{B_x \eta_0}{3\sqrt{B}} \frac{\partial \left(\sqrt{B} \left(V_\parallel + b_x \left(V_\perp^{(dia)} + V_\perp^{(E)} \right) \right) \right)}{h_x \partial x} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \quad (\text{B.18})$$

$$\tilde{j}_y^{(vis\parallel)} = b_z \frac{B_x \eta_0}{3\sqrt{B}} \frac{\partial \left(\sqrt{B} \left(V_\parallel + b_x \left(V_\perp^{(dia)} + V_\perp^{(E)} \right) \right) \right)}{h_x \partial x} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \quad (\text{B.19})$$

The current produced by components of viscosity tensor connected with heat fluxes is:

$$\tilde{j}_\perp^{(visq)} = -\frac{0.24 B_x}{\sqrt{B} \nu_{ii}} \frac{\partial \left(q_{i\parallel}^{(0)} \sqrt{B} \right)}{h_x \partial x} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \quad (\text{B.20})$$

$$\tilde{j}_y^{(visq)} = b_z \frac{0.24 B_x}{\sqrt{B} \nu_{ii}} \frac{\partial \left(q_{i\parallel}^{(0)} \sqrt{B} \right)}{h_x \partial x} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \quad (\text{B.21})$$

The second part of the current contains contributions from both inertia and gyroviscosity forces. In contrast to the parallel momentum balance, the cancellation is not complete (as has been noticed in [19]). Expressions for this part of the current are

$$\begin{aligned} j_\perp^{(in)} = & -\frac{m_i}{B} \left[\frac{\partial n V_y}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n \left(V_\perp^{(0)} b_z + V_\parallel b_x \right) V_y \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n V_y^{(0)} V_y \right) \right] \\ & - \frac{2b_x}{B} \frac{1}{h_x} \frac{\partial}{\partial x} \left(\eta_3 \frac{\partial V_\parallel}{h_x \partial x} \right) + \frac{m_i}{B} n V_\parallel^2 \frac{\partial \ln h_z}{h_y \partial y} \end{aligned} \quad (\text{B.22})$$

$$j_y^{(in)} = \frac{m_i}{B} \left[\frac{\partial n V_\perp}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n (V_\perp^{(0)} b_z + V_\parallel b_x) V_\perp \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n V_y^{(0)} V_\perp \right) \right] - \frac{2b_x}{B} \frac{1}{h_x} \frac{\partial}{\partial x} \left(\eta_3 \frac{\partial V_\parallel}{h_y \partial y} \right) - \frac{m_i}{B} n V_\parallel^2 \frac{\partial \ln h_z}{h_x \partial x} \quad (\text{B.23})$$

where $\eta_3 = \frac{nT_i}{2\omega_{ci}}$, $\eta_1 = \frac{\eta_2}{4}$.

The last terms proportional to the gradient of B correspond to the vertical guiding centre drift of ions caused by the centrifugal force. This current can be simply added to the diamagnetic current. Other terms seem to be less important.

The contribution from perpendicular viscosity is

$$j_\perp^{(vis\perp)} = \frac{1}{B} \left[\frac{1}{h_x} \frac{\partial}{\partial x} \left(\eta_1 \frac{\partial V_y}{h_x \partial x} \right) + \frac{1}{h_y} \frac{\partial}{\partial y} \left(\eta_1 \frac{\partial V_y}{h_y \partial y} \right) + \frac{1}{h_x h_y} \frac{\partial \eta_1}{\partial x} \frac{\partial V_\perp}{\partial y} - \frac{1}{h_x h_y} \frac{\partial \eta_1}{\partial y} \frac{\partial V_\perp}{\partial x} \right] + \frac{4b_x}{B} \frac{1}{h_x h_y} \left(\frac{\partial \eta_1}{\partial x} \frac{\partial V_\parallel}{\partial y} - \frac{\partial \eta_1}{\partial y} \frac{\partial V_\parallel}{\partial x} \right) \quad (\text{B.24})$$

$$j_y^{(vis\perp)} = -\frac{1}{B} \left[\frac{1}{h_y} \frac{\partial}{\partial y} \left(\eta_1 \frac{\partial V_\perp}{h_y \partial y} \right) + \frac{1}{h_x} \frac{\partial}{\partial x} \left(\eta_1 \frac{\partial V_\perp}{h_x \partial x} \right) - \frac{1}{h_x h_y} \frac{\partial \eta_1}{\partial x} \frac{\partial V_y}{\partial y} + \frac{1}{h_x h_y} \frac{\partial \eta_1}{\partial y} \frac{\partial V_y}{\partial x} \right]. \quad (\text{B.25})$$

Here, the largest term is the first one in the r.h.s. of the expression for the radial current, since the radial derivatives are the largest. This current corresponds to a pure cylindrical approximation. It is responsible, for example, for closing currents near flush-mounted probes [22]. It was taken into account in [18, 23]. However, in the problem of global current structure in the edge plasma, the currents driven by parallel viscosity seem to be more important.

The total current caused by viscosity is

$$\vec{j}^{(vis)} = \vec{j}^{(vis\parallel)} + \vec{j}^{(vis\perp)} + \vec{j}^{(visq)}. \quad (\text{B.26})$$

The contribution from ion neutral friction is

$$j_\perp^{(s)} = \frac{S_{iy}^m}{B}. \quad (\text{B.27})$$

$$j_y^{(s)} = -\frac{S_{i\perp}^m}{B}, \quad (\text{B.28})$$

where S_{iy}^m and $S_{i\perp}^m$ are the momentum sinks due to ion-neutral friction.

$$S_{i\perp}^m = -n\mu_{iN}\nu_{iN} (V_\perp - V_{\perp N}), \quad (\text{B.29})$$

$$S_{iy}^m = -n\mu_{iN}\nu_{iN} (V_y - V_{yN}), \quad (\text{B.30})$$

where $\mu_{iN} = m_i/2$ is the effective mass, $\nu_{iN} = n_N \langle V_{iN} \sigma_{cx} \rangle$ is the ion-neutral collision frequency, n_N is the neutral density and $\vec{V}_{iN}$ is the neutral mean velocity. For the ion velocity used above, only electric and diamagnetic drifts are left, neglecting smaller terms, to obtain

$$\tilde{j}_x^{(s)} = \tilde{j}_\perp^{(s)} b_z = -\sigma_{iN} b_z^2 \frac{\partial \Phi}{h_x \partial x} - \sigma_{iN} b_z^2 \frac{1}{en} \frac{\partial n T_i}{h_x \partial x} + \sigma_{iN} B_z V_{yN}, \quad (\text{B.31})$$

$$\tilde{j}_y^{(s)} = -\sigma_{iN} \frac{\partial \Phi}{h_y \partial y} - \sigma_{iN} \frac{1}{en} \frac{\partial n T_i}{h_y \partial y} - \sigma_{iN} B V_{\perp N}, \quad (\text{B.32})$$

where σ_{iN} is the ion-neutral perpendicular conductivity

$$\sigma_{iN} = \frac{nm_i \langle V_{iN} \sigma_{cx} \rangle n_N}{2B^2} . \quad (\text{B.33})$$

The rate of charge-exchange $\langle V_{iN} \sigma_{cx} \rangle$ can be expressed through the mobility K_0 measured in the absence of a magnetic field for room temperature and atmospheric pressure for deuterium [24]:

$$K_0 = \frac{2e}{m_i \langle V_{iN} \sigma_{cx} \rangle n_N^{(0)}} , \quad (\text{B.34})$$

with $n_N^{(0)} = 2.687 \cdot 10^{25} m^{-3}$ and $K_0 = 11.2 \cdot 10^{-4} \frac{m^2}{sV}$. Assuming that the charge-exchange cross-section is approximately constant one gets

$$\langle V_{iN} \sigma_{cx} \rangle = 3.2 \cdot 10^{-15} \sqrt{\frac{T_i}{0.026}} \frac{m^3}{s} , \quad (\text{B.35})$$

where T_i is measured in eV. A similar expression, which gives approximately the same result, is [25]:

$$\langle V_{iN} \sigma_{cx} \rangle = \frac{32\sigma_{cx}}{3\sqrt{\pi}} \sqrt{\frac{T_i}{m_i}} \frac{m^3}{s} . \quad (\text{B.36})$$

In the energy balance equations appears a special problem, which is the possible inclusion of an anomalous radial current. The basic guidelines are as follows:

- The expression must be as simple as possible, and the most convenient format is an Ohm-like expression;
- The system of equations must be informed about this non-classical contribution, but
- The classical core of the equations must not be disturbed.

Apparently, the radial current would enter the electron energy equation via the Joule term, but this manipulation is inappropriate since one cannot be sure that this term directly heats the electrons, which is the case in a purely classical setup. The proper thing to do is to keep only the non-radial classical contribution in the Joule term, and construct additional heat exchange terms containing the radial current. In this way, the system decides how to thermodynamically allocate the "alien" heat term, among electrons and ion species.

For this reason, one cannot simply use the standard Ohm's law in the radial direction with anomalous coefficients, or any other prescribed expression whatsoever, as a constructive procedure is necessary for simultaneously producing the radial current as well as the heat exchange terms. At this point it has to be mentioned that this peculiarity arises only in the multi-fluid case, whereas in the single species case these heat exchange terms can indeed be constructed even using a prescribed expression for the radial current.

The trick for carrying out this program is to correct the electron friction density $\vec{R}^e$ by adding the term $n_e e \tilde{\eta}_\perp \vec{J}_\perp$ similar to the classical one, only using the *ad hoc* parameter $\tilde{\eta}_\perp$.

Since $\vec{R}^e = -\sum_a \vec{R}^a$, one has a modified total ion momentum equation at hand which in conjunction with the pressure equilibrium (not affected by the additional term) $\vec{J} \times \vec{B} = \nabla P$, gives the diffusive part of the radial current, namely

$$J_y^{diff} = \frac{\tilde{\eta}_\perp e n_e}{B^2} \frac{\partial P_i}{\partial y} \quad (\text{B.37})$$

where P_i is the total ion pressure. This dependency is also found in [16]. Moreover, an additional dependence on the radial electric field can potentially be added without interfering with the procedure.

The final step, namely the construction of the heat exchange terms, is performed by claiming that the well known relation [26] (adapted for the multi-fluid case)

$$Q_e = -\sum_a \left(Q_a + (\vec{u}^a - \vec{u}^e) \cdot \vec{R}^a \right) , \quad (\text{B.38})$$

which stems from the fundamental null space of the collision operator, is still true while carrying the modified friction. Claiming this, after performing the proper manipulations in the energy equations, one reveals the *ad hoc* contribution as the exchange term that one wants, which reads $\sum_a (\vec{u}^a - \vec{u}^e)_\perp n_a \frac{Z_a^2}{Z_{eff}} e \tilde{\eta}_\perp \vec{J}_\perp$ added in the total ion energy equation and subtracted from the electron equation. Note that in the single species case, this term boils down to $\tilde{\eta}_\perp J_\perp^2$ removing energy from the electrons, depositing it to the ions, a feature predicted by neoclassical theory [27].

In the final form of the energy balance equations, the convective heat flux and the diamagnetic heat flux are combined as in [16, 17, 18, 19], see also [22].

The energy balance equation for the electrons is

$$\begin{aligned} \frac{3}{2} \frac{\partial n T_e}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \right) + \frac{n T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} (V_\parallel - j_\parallel / en) b_x \right) \\ = Q_e + n T_e B \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right), \end{aligned} \quad (\text{B.39})$$

where

$$\begin{aligned} \tilde{q}_{ex} = \frac{3}{2} n T_e \left(-\frac{b_z}{B h_y} \frac{\partial \Phi}{\partial y} + b_x V_\parallel - b_x j_\parallel / en + b_z V_\perp^{(AN)} \right) \\ + \frac{5}{2} n T_e b_z \left(-\frac{D}{T_e + T_i} \frac{b_z}{h_x} \left(\frac{\partial p}{n \partial x} - \frac{3}{2} \frac{\partial T_e}{\partial x} \right) - D_n^{(AN)} \frac{1}{h_x n} \frac{\partial n}{\partial x} - D_p^{(AN)} \frac{1}{h_x n} \frac{\partial p_i}{\partial x} \right) + n V_{e\perp}^{(AN)} \\ - 0.71 b_x j_\parallel T_e / e - \kappa_{e\parallel} \frac{b_x^2}{h_x} \frac{\partial T_e}{\partial x} - \kappa_{e\perp} b_z^2 \frac{1}{h_x} \frac{\partial T_e}{\partial x} \\ + \frac{3}{2} \frac{T_e}{e B} \frac{\nu_{ei}}{\omega_{ce}} \frac{b_z^2}{h_x} \frac{\partial n (T_e + T_i)}{\partial x} - \frac{5}{2} n T_e^2 \frac{B_z}{e} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right), \end{aligned} \quad (\text{B.40})$$

$$\begin{aligned} \tilde{q}_{ey} = \frac{3}{2} n T_e \left(\frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x} + V_y^{(AN)} \right) \\ + \frac{5}{2} n T_e \left(-\frac{D}{T_e + T_i} \frac{1}{h_y} \left(\frac{\partial p}{n \partial y} - \frac{3}{2} \frac{\partial T_e}{\partial y} \right) - D_n^{(AN)} \frac{1}{h_y n} \frac{\partial n}{\partial y} - D_p^{(AN)} \frac{1}{h_y n} \frac{\partial p_i}{\partial y} \right) + n V_{ey}^{(AN)} \\ - \kappa_{e\perp} \frac{1}{h_y} \frac{\partial T_e}{\partial y} + \frac{3}{2} \frac{T_e}{e B} \frac{\nu_{ei}}{\omega_{ce}} \frac{1}{h_y} \frac{\partial n (T_e + T_i)}{\partial y} + \frac{5}{2} n T_e^2 \frac{B_z}{e} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right), \end{aligned} \quad (\text{B.41})$$

$$Q_\Delta = \frac{3 m_e}{m_i} n \nu_{ei} (T_e - T_i), \quad (\text{B.42})$$

$$Q_u = R_{ie\parallel} j_\parallel / en, \quad (\text{B.43})$$

$$Q_{AN} = \sum_a (\vec{u}^a - \vec{u}^e)_\perp n_a \frac{Z_a^2}{Z_{eff}} e \tilde{\eta}_\perp \vec{J}_\perp, \quad (\text{B.44})$$

$$Q_e = -Q_\Delta - Q_u - Q_{AN}. \quad (\text{B.45})$$

Here $\kappa_{e\parallel}$ is the Balescu (not Braginskii as stated in the CPP review paper [28] or in `\${SOLPSTOP}/doc/CPP_2006.pdf`) heat conductivity with a possible heat flux limit [29]. The leading numerical coefficient is then $\frac{5}{2} \frac{2.16z}{1+0.27z}$ as opposed to 3.16 for the Braginskii formula [30]. The perpendicular heat conductivity $\kappa_{e\perp}$ can be either classical or anomalous. In the last term of the energy balance equation, it is taken into account that the $\vec{E} \times \vec{B}$ drift velocity is almost divergence-free.

The ion energy balance equation is

$$\begin{aligned} & \frac{3}{2} \frac{\partial n T_i}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right) + \frac{n T_i}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} V_{\parallel} b_x \right) \\ &= Q_{\Delta} + Q_{AN} + \frac{\eta_0}{3} \left(2b_x \frac{\partial V_{\parallel}}{h_x \partial x} \right)^2 + n T_i B \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right), \end{aligned} \quad (\text{B.46})$$

where

$$\begin{aligned} \tilde{q}_{ix} &= \frac{3}{2} n T_i \left(-\frac{b_z}{B h_y} \frac{\partial \Phi}{\partial y} + b_x V_{\parallel} + b_z V_{\perp}^{(AN)} \right) \\ &+ \frac{5}{2} n T_i b_z \left(-\frac{D}{T_e + T_i} \frac{b_z}{h_x} \left(\frac{\partial p}{n \partial x} - \frac{3}{2} \frac{\partial T_e}{\partial x} \right) - D_n^{(AN)} \frac{1}{h_x n} \frac{\partial n}{\partial x} - D_p^{(AN)} \frac{1}{h_x n} \frac{\partial p_i}{\partial x} \right) + n V_{i\perp}^{(AN)} \\ &+ \frac{5}{2} n T_i b_z \tilde{V}_{\perp}^{(dia)} - \kappa_{i\parallel} \frac{b_x^2}{h_x} \frac{\partial T_i}{\partial x} - \kappa_{i\perp} b_z^2 \frac{1}{h_x} \frac{\partial T_i}{\partial x}, \end{aligned} \quad (\text{B.47})$$

$$\begin{aligned} \tilde{q}_{iy} &= \frac{3}{2} n T_i \left(\frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x} + V_y^{(AN)} \right) \\ &+ \frac{5}{2} n T_i \left(-\frac{D}{T_e + T_i} \frac{1}{h_y} \left(\frac{\partial p}{n \partial y} - \frac{3}{2} \frac{\partial T_e}{\partial y} \right) - D_n^{(AN)} \frac{1}{h_y n} \frac{\partial n}{\partial y} - D_p^{(AN)} \frac{1}{h_y n} \frac{\partial p_i}{\partial y} \right) + n V_{iy}^{(AN)} \\ &- \kappa_{e\perp} \frac{1}{h_y} \frac{\partial T_i}{\partial y} + \frac{5}{2} n T_i \tilde{V}_y^{(dia)}. \end{aligned} \quad (\text{B.48})$$

As for the electrons, the parallel heat conductivity is classical with a possible heat flux limit, while the perpendicular contribution can be either classical or anomalous.

There is discussion in the literature on how to take into account anomalous terms in the energy balance equations. One point of view (see e.g. [31]) is to cancel the contribution from anomalous particle fluxes in the terms $p_i \nabla \cdot \vec{V}_i$ (or $\vec{V}_i \cdot \nabla p_i$ if equations are written in a different form). In contrast, according to the derivation [19], where a kinetic equation with fluctuating term has been integrated, the anomalous fluxes should be kept in all terms of the energy balance equations. This variant was used in UEDGE [16, 17, 18]. In B2.5 only the contribution from anomalous radial particle flux to the convective heat flux are left in the expressions for $\tilde{q}_{e,i}$. In the term $p_i \nabla \cdot \vec{V}_i$ the anomalous particle flux is absent.

B.0.1 Derivation of parallel viscosity term in momentum equation

To evaluate the parallel viscosity contribution we should start from the general expressions, which can be transformed and simplified in arbitrary Cartesian coordinate system assuming only parallel velocity. The Braginskii description for parallel viscosity in Cartesian coordinates is:

$$\pi_{\alpha\beta} = \frac{3}{2} \eta_0 \left(b_{\alpha} b_{\beta} - \frac{1}{3} \delta_{\alpha\beta} \right) \left(b_{\mu} b_{\nu} - \frac{1}{3} \delta_{\mu\nu} \right) W_{\mu\nu}$$

where

$$W_{\mu\nu} = \frac{\partial V_{\mu}}{\partial x_{\nu}} + \frac{\partial V_{\nu}}{\partial x_{\mu}} - \frac{2}{3} \delta_{\mu\nu} \frac{\partial V_i}{\partial x_i}, \quad \vec{b} = \vec{B}/B.$$

The parallel momentum balance component is:

$$\vec{b} \cdot (\nabla \cdot \overleftrightarrow{\pi}) = b_\alpha \frac{\partial \pi_{\alpha\beta}}{\partial x_\beta} = b_\alpha \frac{\partial}{\partial x_\beta} \left[\frac{3}{2} \eta_0 \left(b_\alpha b_\beta - \frac{1}{3} \delta_{\alpha\beta} \right) \left(b_\mu b_\nu - \frac{1}{3} \delta_{\mu\nu} \right) W_{\mu\nu} \right]$$

We can evaluate the scalar value separately:

$$\begin{aligned} \pi_0 &= \frac{3}{2} \eta_0 W_{\parallel\parallel} = \frac{3}{2} \eta_0 \left(b_\mu b_\nu - \frac{1}{3} \delta_{\mu\nu} \right) W_{\mu\nu} \\ &= \frac{3}{2} \eta_0 \left(b_\mu b_\nu - \frac{1}{3} \delta_{\mu\nu} \right) \left(\frac{\partial V_\mu}{\partial x_\nu} + \frac{\partial V_\nu}{\partial x_\mu} - \frac{2}{3} \delta_{\mu\nu} \frac{\partial V_i}{\partial x_i} \right) = 3 \eta_0 \left(b_\mu b_\nu - \frac{1}{3} \delta_{\mu\nu} \right) \left(\frac{\partial V_\mu}{\partial x_\nu} - \frac{1}{3} \delta_{\mu\nu} \frac{\partial V_i}{\partial x_i} \right) \\ &= 3 \eta_0 \left(b_\mu b_\nu \frac{\partial V_\mu}{\partial x_\nu} - \frac{1}{3} \frac{\partial V_i}{\partial x_i} \right) = 3 \eta_0 \left(b_\mu b_\nu \frac{\partial V_{\parallel} b_\mu}{\partial x_\nu} - \frac{1}{3} \frac{\partial V_{\parallel} b_\nu}{\partial x_\nu} \right) \\ &= 3 \eta_0 \left(\frac{B_\nu}{B} \frac{\partial V_{\parallel}}{\partial x_\nu} - \frac{1}{3} B_\nu \frac{\partial V_{\parallel}/B}{\partial x_\nu} \right) = \eta_0 \frac{B_\nu}{B} \left(2 \frac{\partial V_{\parallel}}{\partial x_\nu} + \frac{V_{\parallel}}{B} \frac{\partial B}{\partial x_\nu} \right) \\ &= 2 \eta_0 \frac{B_\nu}{B} \frac{1}{\sqrt{B}} \frac{\partial V_{\parallel} \sqrt{B}}{\partial x_\nu} = 2 \eta_0 \frac{1}{\sqrt{B}} \vec{b} \cdot \nabla (V_{\parallel} \sqrt{B}) \end{aligned}$$

The projection of the viscosity divergence on the parallel direction is then:

$$\begin{aligned} \vec{b} \cdot (\nabla \cdot \overleftrightarrow{\pi}) &= b_\alpha \frac{\partial}{\partial x_\beta} \left[\pi_0 \left(b_\alpha b_\beta - \frac{1}{3} \delta_{\alpha\beta} \right) \right] = \frac{\partial \pi_0 b_\alpha}{\partial x_\alpha} - \frac{1}{3} b_\alpha \frac{\partial \pi_0}{\partial x_\alpha} = B_\alpha \frac{\partial \pi_0/B}{\partial x_\alpha} - \frac{1}{3} b_\alpha \frac{\partial \pi_0}{\partial x_\alpha} \\ &= \frac{2}{3} b_\alpha \frac{\partial \pi_0}{\partial x_\alpha} - \pi_0 \frac{b_\alpha}{B} \frac{\partial B}{\partial x_\alpha} = \frac{2}{3} B^{3/2} b_\alpha \frac{\partial \pi_0 B^{-3/2}}{\partial x_\alpha} = \frac{2}{3} B^{3/2} \vec{b} \cdot \nabla (\pi_0 B^{-3/2}) \end{aligned}$$

Then the parallel viscosity in the parallel momentum balance is:

$$\vec{b} \cdot (\nabla \cdot \overleftrightarrow{\pi}) = \frac{4}{3} B^{3/2} \vec{b} \cdot \nabla \left(\frac{\eta_0}{B^2} \vec{b} \cdot \nabla (V_{\parallel} \sqrt{B}) \right), \text{ where } V_{\parallel} = \vec{b} \cdot \vec{V}$$

For **SOLPS-ITER**, the orthogonal coordinates (x, r, z) are more convenient for further evaluations than the non-orthogonal coordinates (x, y, z) . In these coordinates, it is readily seen that $\vec{b} \cdot \nabla f = b_x \frac{\partial f}{h_x \partial x}$ for any scalar function f . The viscosity can be finally written as:

$$\vec{b} \cdot (\nabla \cdot \overleftrightarrow{\pi}) = \frac{4}{3} b_x B^{\frac{3}{2}} \frac{\partial}{h_x \partial x} \left(\frac{\eta_0 b_x}{B} \frac{\partial (B^{\frac{1}{2}} V_{\parallel})}{h_x \partial x} \right)$$

B.1 Boundary conditions with drifts and currents

Whether one deals with the diamagnetic or $\mathbf{E} \times \mathbf{B}$ perpendicular drift velocity, special care must be taken to avoid unphysical flows at the plates. The standard boundary condition is to require that the flow be at least sonic at the sheath entrance [32], which translates into:

$$V_{\parallel} \geq C_s - \frac{b_z}{b_x} V_{\perp}, \quad (\text{B.49})$$

where C_s is the local sound speed and $\frac{b_z}{b_x}$ is the inverse of the field line pitch. However, for very small pitch angles, as in stellarators, the $V_{\perp}$ term may overpower the sound velocity [33] and then the boundary condition would require

a plasma flow exiting the plate. Therefore, the pitch angle at the plates is limited to be no less than 1 degree (user parameter, see switch "b2agfs_min_pitch" in Appendix A.4), motivated by the engineering limits met when attempting to align the divertor tiles with the magnetic field. The solution procedure of the code was also modified so as ensure that no unphysical flows were being created, and solves the parallel momentum equation using the updated electric potential.

Boundary conditions for the current continuity equation at the plates correspond to sheath current-voltage characteristics

$$j_x = en \left(b_x c_s - b_x \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e\Phi}{T_e} \right) (1 - \gamma_e) \right), \quad (\text{B.50})$$

where γ_e is the secondary electron emission coefficient. At the inner (core) flux surface the currents are set either to the divergent part of the diamagnetic current or to zero. Both variants give the same result and both guarantee that the net current integrated over the inner flux surface should be zero. At the walls, the same conditions on the normal current components were imposed as for the inner core.

The electron and ion heat fluxes to the target plates are

$$\tilde{q}_{ex} = b_x \frac{n}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e\Phi}{T_e} \right) (1 - \gamma_e) \left(T_e \frac{1 + \gamma_e}{1 - \gamma_e} + e\Phi \right), \quad (\text{B.51})$$

$$\tilde{q}_{ix} = \frac{3}{2} n T_i c_s b_x. \quad (\text{B.52})$$

The question of kinetic boundary conditions including drifts is not yet finally clarified and will need further work from both kinetic and fluid modelling [34, 32].

B.2 Problems for a fluid description of the edge plasma

To apply a fluid model for the description of the edge, a specific ordering of lengths is necessary.

The necessary condition for a fluid model is that all characteristic lengths (characteristic length of the sheath, which is a microscopic length scale of about 10 Debye lengths; electron and ion mean free path ranging from sub-cm to meters; neutral mean free path, which is centimeters to meters; heat conduction length which is typically five times the mean free path length due to the dominant contribution from tail electrons or ions) are smaller than the parallel connection length (order of several tens of meters). This is usually fulfilled in edge plasmas with the following possible exceptions:

1. at low temperatures, e.g. near the target in the divertor, steeper gradients result in non-local heat flux contributions. In general, a non-local description might be necessary (weighted mean of temperature profile over five mean free lengths of electrons or ions). However, a simpler correction is sometimes satisfactory and is used in the code (see below).
2. at high temperatures (midplane, near separatrix). Corrections for flux limit and for periodic systems on closed field lines are necessary due to the fact that the heat conduction mean free path is larger than the connection length.
3. the sheath cannot be resolved by a fluid model and is represented by boundary conditions at the sheath entrance deduced from kinetic models.

Simple heat flux and momentum flux (viscosity) limits are available. The heat flux limit is formulated as

$$\kappa_{\parallel} = \frac{\kappa_{SH}}{1 + |q_{SH}/q_{fl}|}, \quad (\text{B.53})$$

with the classical Spitzer–Härm heat flux

$$q_{SH} = -\kappa_{\parallel,SH} \cdot \frac{\partial T_e}{\partial x} \quad (\text{B.54})$$

and the flux limit

$$q_{fl} = \alpha n_e T_e^{3/2} \sqrt{m_e} \quad (\text{B.55})$$

($\alpha \approx 0.2$ from comparison with kinetic calculations for electrons; the situation for ions is less clear).

The momentum flux limit is done the same way

$$\eta_{\parallel} = \frac{\eta_{cl,a}}{1 + |q_{cl,a}/q_{fl,a}|}, \quad (\text{B.56})$$

with the classical momentum flux

$$q_{cl,a} = -\eta_{\parallel,a} \cdot \frac{\partial u_{\parallel,a}}{\partial x} \quad (\text{B.57})$$

and the flux limit

$$q_{fl,a} = \alpha p_a \quad (\text{B.58})$$

(with $\alpha = 0.5$). Unfortunately, for this correction even less kinetic work exists than for the heat flux limits.

Additionally, long mean free path corrections of classical heat conduction and viscosity (in the core) are applied. For periodic systems and small perturbations one gets [35]:

$$q_{\parallel,eff} = q_{SH} / \left(h_{\parallel}^2 \lambda_{HC}^2 + 1 \right), \quad (\text{B.59})$$

with wavenumber of periodicity

$$h_{\parallel} = 2\pi/\lambda_{\parallel}, \quad (\text{B.60})$$

$$\lambda_{\parallel} = 2\pi Rq \quad (\text{B.61})$$

and mean free path of tail electrons [36]

$$\lambda_{HC}(\text{m}) = 7.5 \cdot 10^{16} \cdot T_e^2(\text{eV})/n_e(\text{m}^{-3}). \quad (\text{B.62})$$

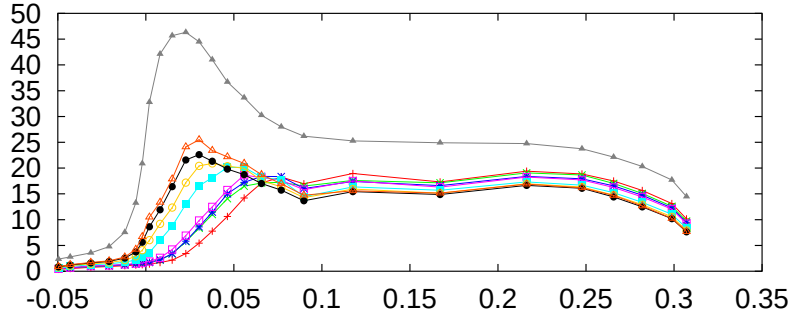
Fokker-Planck computer simulations of electron heat transport in laser plasmas [37] show that a better description is obtained for a linear correction:

$$q_{\parallel,eff} = q_{SH} / (h_{\parallel} \lambda_{HC} + 1). \quad (\text{B.63})$$

Applying these limits for the core, convergence behaviour of the code improves significantly.

A more correct kinetic limit would imply a non-local heat transport (and viscosity,...) description. The parallel heat flux at a point should be determined by classical fluxes at points up to typically five mean free paths away. As an additional feature, one would get pre-heating effects by fast electrons, maybe important for the ELMs. Work is necessary to generalize this to the SOL [38, 39] and this has to be cross-checked with kinetic calculations [40]. Also, numerical problems due to this non-local description have to be solved [41].

In general, if these kinetic corrections become important, quantitative conclusions are questionable, because the fluid model is no longer strictly valid. Considerations of particle, energy and momentum conservation still guarantee a



Fri Aug 24 15:02:28 2001



Figure B.2: Impact of ion flux limits for a 10^{19} m^{-3} upstream density case.

qualitatively correct description. The flux-limiters are designed to give the correct answer in the fully collisional and in the fully collisionless regime.

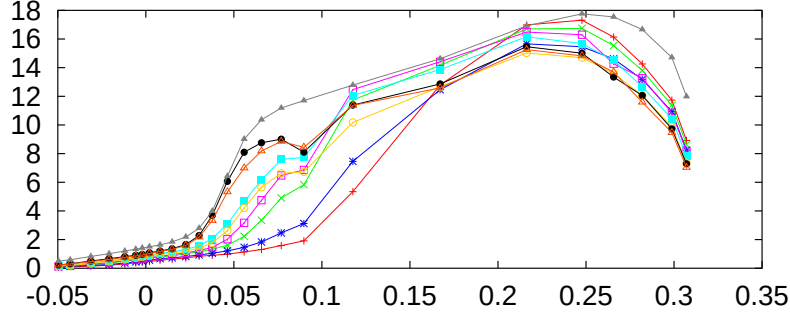
Parametric studies varying the various flux limiters should be done, particularly for predictive work. Specific experimental scenarios would be required to calibrate the flux limiters. However, complete experimental data sets including electron and ion temperatures and densities at the target plate and upstream are necessary. Fortunately, at higher densities these corrections are not important. Furthermore, one should note, that mapping of target data to the midplane and the extraction of anomalous radial transport coefficients from such a procedure is questionable in the regime where flux limiters are important [42].

As an example, SOLPS5.0 calculations for JET using deuterium, carbon and helium are shown in Figs. B.2 and B.3. The predicted electron temperatures at the outer target are shown for two different upstream densities. In each graph, the ion energy flux limit is varied and the sensitivity of the target profiles to this variation is obvious.

B.3 Neutral models

B.3.1 Fluid neutrals

As an alternative to the complete kinetic Monte Carlo description offered by `Eirene`, a simpler and faster neutral fluid model is also available as `B2.5` standalone. The surface sources of neutrals are then given by user-defined recycling models, possibly enhanced with local sputtering models when impurities are considered (see e.g. [43]). The volume sources of neutrals are given by plasma recombination rates computed by the `B2.5` atomic physics packages. The neutral species is then treated just as the other plasma species, with its own particle and momentum transport equations in the parallel direction, equivalent to Eqs. (B.5) and (B.10). The neutrals are assumed to have the same common temperature as all other ion species and the neutral energy balance equation becomes simply a component of Eq. (B.46). Of course, terms due to charged particle motions or currents do not appear for neutral species.



Fri Aug 24 15:03:59 2001



Figure B.3: Impact of ion flux limits for a $2 \times 10^{19} \text{ m}^{-3}$ upstream density case.

The neutral species transport coefficients are usually given by the following equations:

$$D_{n,0}^{(AN)} = 0, \quad (\text{B.64})$$

$$D_{p,0}^{(AN)} = C + \sqrt{\frac{T_i}{m_N}} \frac{1}{(\sigma_{CX} n_i + \sigma_{in} n_e)}, \quad (\text{B.65})$$

where m_N is the mass of the neutral species, and C , σ_{CX} and σ_{in} are three constants given by the user, representing hydrogen gas diffusion, and the charge-exchange and ionization cross-sections, respectively. As an option, one may choose to use cross-sections from atomic databases instead of constants, with their appropriate parameter dependencies. The diffusion coefficients can also be bounded to avoid unphysically large or small values.

Additionally, the heat and particle fluxes carried by neutrals can be flux-limited, following the formalism laid out in [44]. On the one hand, we flux-limit the neutral contribution to the ion heat diffusivity, involved in the conductive heat flux computation. On the other hand, we also flux-limit the convective particle and energy fluxes of neutrals. Because these limits apply for neutral species, both the parallel and perpendicular directions are to be considered. The basic formula used is:

$$X \Rightarrow \frac{X}{\left[1 + \left|\frac{X}{\alpha F}\right|^\gamma\right]^{(1/\gamma)}} \quad (\text{B.66})$$

where X is the quantity to limit and F is the limiting value. Typically, the γ coefficient is set to $\gamma = 1$ or 2 ; if $\gamma = 1$ is used, then the absolute value of the flux ratio is taken. The value of α is of order unity; the precise value is unimportant, but the goal is to limit the flux to a value of approximately $n\bar{v}/4$, where $\bar{v} = \sqrt{(8T/\pi m)} = 2v_T/\sqrt{\pi}$. One could also use v_T rather than $\bar{v}$, but the difference is not very significant; in fact, flux limits of $\sim n\bar{v}/2$ are often chosen, since too severe flux limits sometimes make the solutions more difficult to obtain.

For the thermal diffusivities, the individual species diffusivities are flux-limited, then multiplied by the corresponding density (ion or neutral) and then added up in a term which gets multiplied by the (common) ion temperature gradient. For the neutrals, the heat conductivity is

$$\chi_N = n_N \lambda_{CX} v_T, \quad (\text{B.67})$$

where $\lambda_{CX} = 1/(\sigma_{CX}n_i)$ is the neutral CX mean-free-path, n_N is the neutral density, and v_T is the (neutral, ion) thermal velocity. Thus, this conductivity gets limited as

$$\chi_N \Rightarrow \frac{n_N \lambda_{CX} v_T}{\left[1 + \left| \frac{n_N \lambda_{CX} v_T \nabla T_i}{\alpha_2 v_T n_N T_i} \right|^{\gamma_2} \right]^{(1/\gamma_2)}} \quad (\text{B.68})$$

where $\alpha_2 \sim 1$ again. Some cancellations can be made in the equation, but all terms are included for clarity. Again, either $\gamma_2 = 1$ or $\gamma_2 = 2$ could be used.

In general, both of these terms can be important for limiting the heat flux, as the neutral convective velocity shows up as a convective heat flux. In fact, because neutral gradients can be larger than T_i gradients, the convective velocity corrections might be the most important. The impact of the fluid model and associated flux limits, in comparison with the `Eirene` Monte Carlo model, is covered in detail with examples in [42].

B.3.2 First-flight model

As an intermediate approximation between the fluid and kinetic neutral transport models, a so-called "first-flight" model has been implemented. In this model, the first trajectory flight of neutrals launched from the computational domain boundary (at the plates or the wall boundaries) is followed until the neutral undergoes ionization or charge-exchange, and then passed on to the neutral fluid model. In practice, as an initial computational step, one defines a number of chords (user-specified, usually 9) radiating from each edge boundary segment into the plasma, and computes the length travelled by each chord within each plasma cell it crosses. Then, at each time step, the locally injected neutral flux (from recycling, sputtering, or the like) is launched along each chord (with a normalized intensity determined by some pre-determined angular dependence) and considered to be attenuated within each intersected cell according to the local CX and ionization mean free paths, while the balancing plasma sources and sinks in that cell are computed. The effect of such a model is also shown in [42].

B.3.3 Kinetic neutrals: B2.5-Eirene coupling

The coupling procedure between the plasma fluid and Monte Carlo neutral parts of the SOLPS package is done by means of passing arrays in both directions, namely the `braeir` and `eirbra` modules. The plasma fluid code delivers a plasma background on which the neutral trajectories are computed (also available as files `fort.30` for the plasma geometry and `fort.31` for the plasma solution), and the associated particle, parallel momentum and energy sources are deduced (returned *via* the `fort.44` file). Then, in order to translate the total energy sources from the Monte Carlo code into the internal energy sources needed for the plasma fluid solution, we apply the following conversion rules, where the E superscript represents the energy sources in `Eirene` (and old B2) and H the heat sources in B2.5:

$$\text{total energy} \Rightarrow S_a^E = \int \frac{1}{2} m_a v_a^2 S_a d^3v, \quad (\text{B.69})$$

$$\text{internal energy} \Rightarrow S_a^H = \int \frac{1}{2} m_a (v_a - u_a)^2 S_a d^3v, \quad (\text{B.70})$$

$$\rightarrow S_a^H = S_a^E - u_a \cdot S_a^{mom} + \frac{1}{2} m_a u_a^2 S_a^{particle}. \quad (\text{B.71})$$

The derivation is as follows. The energy source rate from `Eirene` is computed as:

$$S_i^E = \sum_a S_a^E = \sum_a \sum_{Ea} \iint \frac{1}{2} m_a v_a^2 d^3v d^3x$$

where the $\sum_a$ is taken over all species, and $\sum_{Ea}$ over the newly ionised particles of species a coming from the `Eirene` strata, and the spatial integral is taken over the cell volume. We also define the particle and parallel momentum source rates:

$$S_i^{particle} = \sum_a S_{Ea}^{particle} = \sum_a \sum_{Ea} \int 1 \cdot d^3x$$

$$\vec{S}_i^{mom} = \sum_a \vec{S}_{Ea}^{mom} = \sum_a \sum_{Ea} \int m_a \vec{v}_a d^3x$$

We need S_i^H , which, neglecting all other terms, is defined by:

$$\frac{\partial}{\partial t} \left(\int \frac{3}{2} n_i T_i d^3x \right) = \frac{\partial}{\partial t} \left(\sum_a \int \frac{3}{2} n_a T_a d^3x \right) = S_i^H$$

Thus,

$$\frac{3}{2} \sum_a \int (n_a^{t+\Delta t} T_a^{t+\Delta t} - n_a^t T_a^t) d^3x = S_i^H \Delta t$$

$$\frac{3}{2} \sum_a [(N_a^t + S_a^{particle} \Delta t) T_a^{t+\Delta t} - N_a^t T_a^t] = S_i^H \Delta t$$

where $N_a = \int n_a d^3x$ is the cell particle content for species a .

For a drifting Maxwellian distribution with drift velocity $\vec{u}_a$, the temperature (in energy units) is defined as:

$$\frac{3}{2} T_a = \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a)^2 \right\rangle$$

In B2.5, we assume $T_a = T_i$ and use $n_i = \sum_a n_a$.

So we can write:

$$\sum_a (N_a^t + S_a^{particle} \Delta t) \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a)^2 \right\rangle^{t+\Delta t} - \sum_a N_a^t \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a)^2 \right\rangle^t = S_i^H \Delta t \quad (\text{B.72})$$

We define:

$$\vec{u}_a^{t+\Delta t} = \langle \vec{v}_a \rangle^{t+\Delta t} = \frac{N_a^t \vec{u}_a^t + \frac{1}{m_a} \vec{S}_a^{mom} \Delta t}{N_a^t + S_a^{particle} \Delta t} \quad (\text{B.73})$$

$$\delta \vec{u}_a = \vec{u}_a^{t+\Delta t} - \vec{u}_a^t = \frac{\frac{1}{m_a} \vec{S}_a^{mom} \Delta t - \vec{u}_a^t S_a^{particle} \Delta t}{N_a^t + S_a^{particle} \Delta t} \quad (\text{B.74})$$

And make use of:

$$\begin{aligned}
\left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a)^2 \right\rangle^{t+\Delta t} &= \left\langle \frac{1}{2} m_a (\vec{v}_a - (\vec{u}_a^t + \delta \vec{u}_a))^2 \right\rangle^{t+\Delta t} \\
&= \left\langle \frac{1}{2} m_a \left(v_a^2 - 2\vec{v}_a \cdot \vec{u}_a^t - 2\vec{v}_a \cdot \delta \vec{u}_a + u_a^{t^2} + 2\delta \vec{u}_a \cdot \vec{u}_a^t + (\delta u_a)^2 \right) \right\rangle^{t+\Delta t} \\
&= \left\langle \frac{1}{2} m_a \left[(\vec{v}_a - \vec{u}_a^t)^2 + (-2\vec{v}_a \cdot \delta \vec{u}_a + 2\delta \vec{u}_a \cdot \vec{u}_a^t + (\delta u_a)^2) \right] \right\rangle^{t+\Delta t} \\
&= \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \right\rangle^{t+\Delta t} - m_a \langle \vec{v}_a \cdot \delta \vec{u}_a \rangle^{t+\Delta t} + m_a \langle \vec{u}_a^t \cdot \delta \vec{u}_a \rangle^{t+\Delta t} + \frac{1}{2} m_a (\delta u_a)^2 \\
&= \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \right\rangle^{t+\Delta t} - m_a \delta \vec{u}_a \cdot (\vec{u}_a^{t+\Delta t} - \vec{u}_a^t) + \frac{1}{2} m_a (\delta u_a)^2 \\
&= \left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \right\rangle^{t+\Delta t} - \frac{1}{2} m_a (\delta u_a)^2
\end{aligned}$$

Writing out the first average explicitly:

$$\begin{aligned}
&\left\langle \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \right\rangle^{t+\Delta t} \\
&= \frac{\iint n_a^t \left(\frac{m_a}{2\pi T_a^t} \right)^{\frac{3}{2}} \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \exp \left(-\frac{1}{2} m_a \frac{(\vec{v}_a - \vec{u}_a^t)^2}{T_a^t} \right) d^3 \vec{v}_a d^3 x + \sum_a \frac{1}{2} m_a (\vec{v}_a - \vec{u}_a^t)^2 \Delta t}{N_a^t + S_a^{particle} \Delta t} \\
&= \frac{1}{N_a^t + S_a^{particle} \Delta t} \left[\frac{3}{2} N_a^t T_a^t + \sum_a \int \frac{1}{2} m_a v_a^2 \Delta t d^3 x + \frac{1}{2} m_a u_a^{t^2} S_a^{particle} \Delta t - \vec{u}_a^t \cdot \vec{S}_a^{mom} \Delta t \right] \\
&= \frac{1}{N_a^t + S_a^{particle} \Delta t} \left[\frac{3}{2} N_a^t T_a^t + S_a^E \Delta t + \frac{1}{2} m_a u_a^{t^2} S_a^{particle} \Delta t - \vec{u}_a^t \cdot \vec{S}_a^{mom} \Delta t \right]
\end{aligned}$$

Putting all this back into Eq. (B.72), we obtain:

$$\begin{aligned}
&\sum_a \left\{ \left[\frac{3}{2} N_a^t T_a^t + S_a^E \Delta t + \frac{1}{2} m_a u_a^{t^2} S_a^{particle} \Delta t - \vec{u}_a^t \cdot \vec{S}_a^{mom} \Delta t \right] - (N_a^t + S_a^{particle} \Delta t) \frac{1}{2} m_a (\delta u_a)^2 \right\} \\
&- \sum_a \frac{3}{2} N_a^t T_a^t = S_i^H \Delta t
\end{aligned}$$

Thus:

$$S_i^H = \sum_a S_a^E - \sum_a \vec{u}_a^t \cdot \vec{S}_a^{mom} + \sum_a \frac{1}{2} m_a u_a^{t^2} S_a^{particle} - \sum_a \frac{N_a^t + S_a^{particle} \Delta t}{\Delta t} \frac{1}{2} m_a (\delta u_a)^2 \quad (\text{B.75})$$

In the code, we previously had only the first three terms in the R.H.S. of the final equation above. But for very strong sources, dominating the background plasma, the higher order correction is necessary.

In some cases, it may also be essential to ensure excellent particle conservation. During a coupled run timestep, as the B2 internal iterations make the plasma solution evolve away from the plasma background originally used for the computation of the Eirene sources above, it may become necessary to rescale these. A scheme to do so was developed by A. Kukushkin for ITER applications [45] and can be activated using the 'eirene_ank_mods' switch. See the listing in Appendix A.4 and the additional description in the `\${SOLPSTOP}\doc\Source.Scaling-in-B2.pdf` file.

B.4 Classical transport: Braginskii, Balescu and Zhdanov formulations

In **SOLPS 4.x** the classical transport coefficients were derived from Braginskii [46]. In **SOLPS 5.x** these have been replaced with expressions from Balescu [47], completed in **SOLPS-ITER** with expressions derived from the work of Zhdanov [48], sometimes fitted as a function of the local Z_{eff} . Table B.1 summarizes the differences between the Braginskii and Balescu models. It is possible in the code to use either model, or even the full Balescu 21-moment expressions, as well as a Zhdanov fit to the table published in the Braginskii paper as a function of Z_{eff} . See the use of the switches 'b2tqce_model', 'b2tqca_model' and 'b2sigp_style' as described in Appendix A.4.

quantity	functional dependence	numerical prefactor		
		Braginskii	fits to Balescu	fits to Zhdanov
	default in	SOLPS 4.x	SOLPS 5.x	SOLPS-ITER
$\kappa_{e\parallel}$	$\frac{T_e^{5/2} n_e}{\ln \Lambda \sum_a Z_a^2 n_a}$	3.16	$\frac{5}{2} \frac{2.16 Z_{eff}}{1+0.27 Z_{eff}}$	$\frac{\left(3.9 + \frac{2.3}{Z_{eff}}\right)}{0.31 + \frac{1.2}{Z_{eff}} + \frac{0.41}{Z_{eff}^2}}$
$\kappa_{i\parallel}$	$\frac{T_i^{5/2}}{\ln \Lambda} \sum_a \frac{n_a}{Z_a^2}$	$\frac{3.9}{\sum_b Z_b^2 n_b \sqrt{\frac{2m_a m_b}{m_a + m_b}}}$	$\frac{3.95}{\sqrt{m_a} \sum_b Z_b^2 n_b}$	
$\eta_{a\parallel}$	$\frac{T_i^{5/2}}{\ln \Lambda} \frac{n_a}{Z_a^2}$	$\frac{1.37}{\sum_b \frac{Z_b^2 n_b}{\sqrt{\frac{m_a + m_b}{2}}}}$	$\frac{\sqrt{2} 0.96 \sqrt{m_a}}{\sum_b Z_b^2 n_b}$	
$\sigma_{\parallel}$	$\frac{n_e T_e^{3/2}}{\ln \Lambda \sum_a Z_a^2 n_a}$	$\frac{4.95 Z_{eff}}{1+1.54 Z_{eff}}$		$\frac{(1+0.24 Z_{eff})(1+0.93 Z_{eff})}{(1+2.56 Z_{eff})(1+0.29 Z_{eff})}$
$\alpha_{e\parallel}$	$\frac{n_e T_e^{3/2}}{\ln \Lambda \sum_a Z_a^2 n_a}$	$-\sqrt{\frac{5}{2}} \frac{1.25 Z_{eff}}{1+0.41 Z_{eff}}$		$-\frac{1.56 Z_{eff} \left(\frac{1+1.4 Z_{eff}}{1+2.56 Z_{eff}}\right) \left(\frac{1+0.52 Z_{eff}}{1+0.29 Z_{eff}}\right) \frac{1}{Z_{eff} + \frac{\sqrt{2}}{2}}}{\left(\frac{1+0.24 Z_{eff}}{1+2.56 Z_{eff}}\right) \left(\frac{1+0.93 Z_{eff}}{1+0.29 Z_{eff}}\right)}$

Table B.1: Comparison of classical transport coefficients according to the Braginskii, Balescu and Zhdanov formulations

Note that this table does not contain the full prefactors, but only the numerically changing pieces between the models.

In figures B.4 to B.6, we illustrate the dependencies of the electronic classical transport coefficients as a function of Z_{eff} , comparing the **SOLPS 5.x** and **SOLPS-ITER** fit used by default (to the 29-moment Balescu result), the exact 21-moment Balescu expressions, the Zhdanov fit to Braginskii and the **SOLPS 4.x** values.

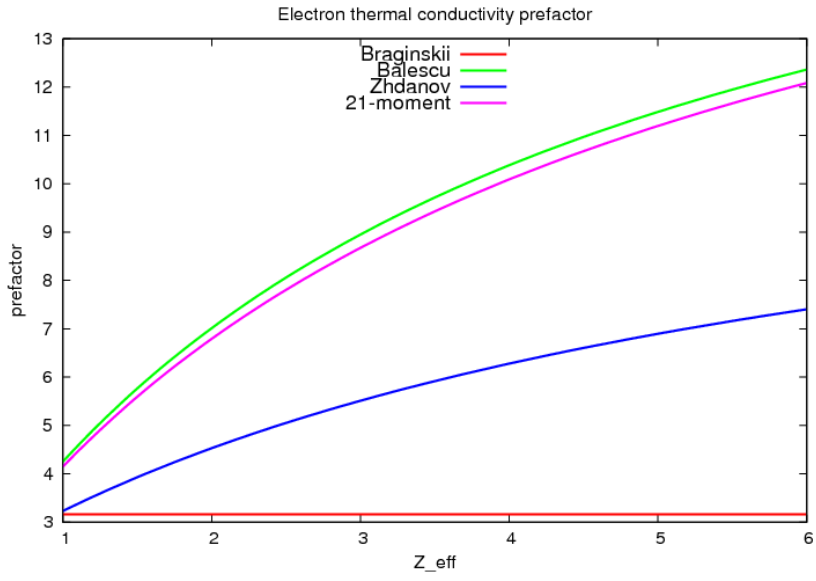


Figure B.4: Dependence of the various models for the $\kappa_{e||}$ prefactor as a function of Z_{eff} .

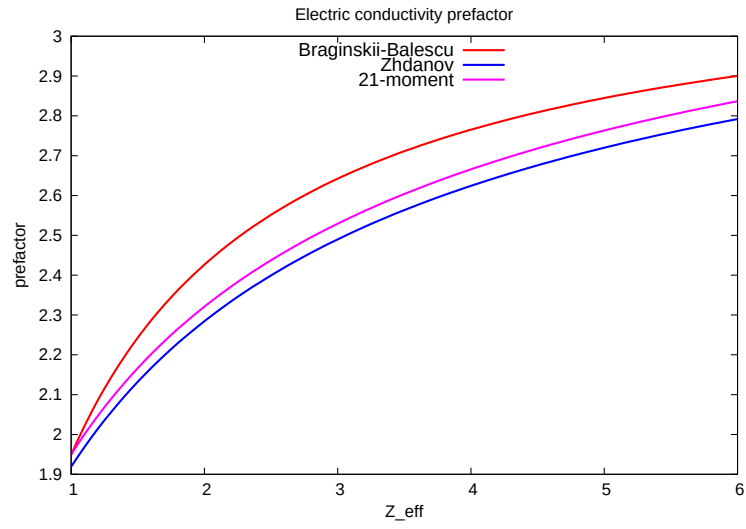


Figure B.5: Dependence of the various models for the parallel electron conductivity $\sigma_{||}$ prefactor as a function of Z_{eff} .

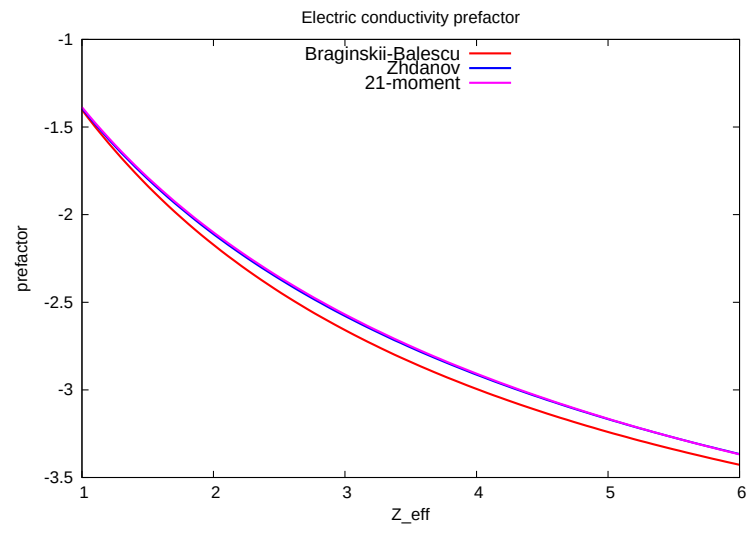


Figure B.6: Dependence of the various models for the thermo-electric coefficient $\alpha_{e||}$ prefactor as a function of Z_{eff} .

Appendix C

Model equations for B2.5

The version of the equations shown in this Appendix represents the best available description of the standard recommended settings for the **SOLPS-ITER** physics model, built upon the original work on **SOLPS5.2** from the St. Petersburg group [49]. This Appendix is not meant as an exhaustive registry of the capabilities of the code, for which the reader is encouraged to peruse Chapters 3 and 5 as well as the list of switches in Appendix A.4 and of namelist variables in Appendix A.5.

Some equations include terms which are multiplied by zero. This is meant to indicate that such terms are not included in the standard **SOLPS-ITER** physics model, but that switches exist that allow for their inclusion.

C.1 Sign conventions

The radial coordinate y increases as one crosses the separatrix from the core (or private flux) region into the main SOL region (and so is minimum at the magnetic axis). The poloidal coordinate x increases in the *clockwise* direction when viewing the poloidal plane with the central axis line on its left. The toroidal magnetic field component B_z is taken as positive if the toroidal magnetic field is directed *clockwise* when viewing the tokamak from above. The poloidal magnetic field component B_x follows the same convention as the x component and is positive when directed in the direction of increasing poloidal index, i.e. *clockwise*. This corresponds, following the conventions laid out in [50], to a *COCOS* index of 6.

Is it important to know, that, in the equations below, the parallel velocity $V_{\parallel}$ is a true parallel velocity, i.e. it changes sign if the poloidal magnetic field component does, namely: $V_{\parallel} = \vec{V} \cdot \vec{B}/|B|$. This applies to all species, including electrons, meaning $V_{\parallel a}$ and $V_{\parallel e}$. However, the poloidal projections of the fluxes and currents, such as Γ_{ax} , are positive when they are colinear with $\vec{e}_x$, i.e. point in the direction of increasing x . The exception to this is the parallel current, $j_{\parallel}$. In the equations below, it has the same sign convention as the parallel velocity. In the actual code, because it is treated as a flux, it follows the sign convention of the fluxes. This therefore means that the code does not follow $j_{\parallel}$, but rather $j_{\parallel} \cdot |B_x|/B_x$.

C.2 Continuity equations for particle conservation

$$\frac{\partial n_a}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \right) = S_a^n, a = 0, 1, \dots, ns - 1 \quad (\text{C.1})$$

$$\begin{aligned}\tilde{\Gamma}_{ax} = & \left(b_x V_{\parallel a} + V_{ax}^{(E)} + V_{ax}^{(AN)} + V_a^{corr-dpc} \right) n_a + 0 \cdot \frac{\delta_{1,a}}{e} \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) \\ & - D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} - \tilde{D}_{p,ax} \frac{1}{h_x} \frac{\partial p_a}{\partial x} - \tilde{\Gamma}_{ax}^{(PSch)} \quad (C.2)\end{aligned}$$

$$\begin{aligned}\tilde{\Gamma}_{ay} = & \left(V_{ay}^{(E)} + V_{ay}^{(AN)} \right) n_a + \frac{\delta_{1,a}}{e} \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) \\ & - D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} - \tilde{D}_{p,ay} \frac{1}{h_y} \frac{\partial p_a}{\partial y} - \tilde{\Gamma}_{ay}^{(PSch)} \quad (C.3)\end{aligned}$$

$$\Gamma_{ax}^{(PSch)} = D_{axy} \frac{1}{h_y} \frac{\partial n_a T_i}{\partial y} \quad \Gamma_{ay}^{(PSch)} = -D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} \quad (C.4)$$

$$\tilde{\Gamma}_{ax}^{(PSch)} = \begin{cases} \Gamma_{ax}^{(PSch)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{\Gamma_{ax}^{(PSch)}}{1 + \frac{\left[\Gamma_{ax}^{(PSch)} \right]}{\frac{\sqrt{g}}{h_x} n_a V_{Ta}}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad \tilde{\Gamma}_{ay}^{(PSch)} = \begin{cases} \Gamma_{ay}^{(PSch)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{\Gamma_{ay}^{(PSch)}}{1 + \frac{\left[\Gamma_{ay}^{(PSch)} \right]}{\frac{\sqrt{g}}{h_y} n_a V_{Ta}}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad (C.5)$$

$$V_{Ta} = |b_x| \sqrt{\frac{8T_i}{\pi m_a}} \quad (C.6)$$

$$\begin{aligned}\Gamma_{ax} = & \left(b_x V_{\parallel a} + \tilde{U}_{ax}^{(dia)} + V_{ax}^{(AN)} + V_a^{corr-dpc} \right) n_a + \\ & 0 \cdot \frac{\delta_{1,a}}{e} \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) - D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} - \tilde{D}_{p,ax} \frac{1}{h_x} \frac{\partial p_a}{\partial x} \quad (C.7)\end{aligned}$$

$$\begin{aligned}\Gamma_{ay} = & \left(\tilde{U}_{ay}^{(dia)} + V_{ay}^{(AN)} \right) n_a + \frac{\delta_{1,a}}{e} \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) \\ & - D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} - \tilde{D}_{p,ay} \frac{1}{h_y} \frac{\partial p_a}{\partial y} \quad (C.8)\end{aligned}$$

$$\begin{aligned}\Gamma_{ax}^{Cor} = & \left(b_x V_{\parallel a} + \tilde{U}_{ax}^{(2dia)} + V_{ax}^{(AN)} + V_a^{corr-dpc} \right) n_a + \\ & 0 \cdot \frac{\delta_{1,a}}{e} \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) - D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} - \tilde{D}_{p,ax} \frac{1}{h_x} \frac{\partial p_a}{\partial x} \quad (C.9)\end{aligned}$$

$$\begin{aligned}\Gamma_{ay}^{Cor} = & \left(\tilde{U}_{ay}^{(2dia)} + V_{ay}^{(AN)} \right) n_a + \\ & \frac{\delta_{1,a}}{e} \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) - D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} - \tilde{D}_{p,ay} \frac{1}{h_y} \frac{\partial p_a}{\partial y} \quad (C.10)\end{aligned}$$

$$\Gamma_{ax}^{(\text{no drift})} = \left(b_x V_{\parallel a} + V_{ax}^{(AN)} + 0 \cdot V_a^{corr-dpc} \right) n_a \quad (\text{C.11})$$

$$\Gamma_{ay}^{(\text{no drift})} = \left(V_{ay}^{(AN)} \right) n_a, \text{ really } 0 \text{ as } V_{ay}^{(AN)} = 0 \quad (\text{C.12})$$

where

$$V_{ax}^{(E)} = \begin{cases} -\frac{B_z}{B^2} \frac{1}{h_y} \frac{\partial \Phi}{\partial y}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad V_{ay}^{(E)} = \begin{cases} \frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (\text{C.13})$$

$$\tilde{V}_{ax}^{(dia)} = \begin{cases} \frac{T_i B_z}{z_a e} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right), & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad \tilde{V}_{ay}^{(dia)} = \begin{cases} -\frac{T_i B_z}{z_a e} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right), & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (\text{C.14})$$

$$\tilde{\tilde{V}}_{ax}^{(dia)} = \begin{cases} \tilde{V}_{ax}^{(dia)}, & (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{\tilde{V}_{ax}^{(dia)}}{1 + \frac{|\tilde{V}_{ax}^{(dia)}|}{V_{Ta}}}, & (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad \tilde{\tilde{V}}_{ay}^{(dia)} = \begin{cases} \tilde{V}_{ay}^{(dia)}, & (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{\tilde{V}_{ay}^{(dia)}}{1 + \frac{|\tilde{V}_{ay}^{(dia)}|}{V_{Ta}}}, & (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad (\text{C.15})$$

$$\Gamma_{ax}^{(Eir)} = \left(b_x V_{\parallel a} + V_{ax}^{(E)} + \tilde{W}_{ax}^{(dia)} + V_{ax}^{(AN)} + V_a^{corr-dpc} \right) n_a + 0 \cdot \frac{\delta_{1,a}}{e} \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) - D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} \quad (\text{C.16})$$

$$\Gamma_{ay}^{(Eir)} = \left(V_{ay}^{(E)} + \tilde{W}_{ya}^{(dia)} + V_{ay}^{(AN)} \right) n_a + \frac{\delta_{1,a}}{e} \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) - D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} \quad (\text{C.17})$$

where

$$W_{ax}^{(dia)} = \begin{cases} -\frac{B_z}{z_a e B^2 n_a} \frac{1}{h_y} \frac{\partial n_a T_i}{\partial y}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad W_{ay}^{(dia)} = \begin{cases} \frac{B_z}{z_a e B^2 n_a} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (\text{C.18})$$

$$\tilde{\tilde{W}}_{ax}^{(dia)} = \begin{cases} W_{ax}^{(dia)}, & (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{W_{ax}^{(dia)}}{1 + \frac{|W_{ax}^{(dia)}|}{V_{Ta}}}, & (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad \tilde{\tilde{W}}_{ay}^{(dia)} = \begin{cases} W_{ay}^{(dia)}, & (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{W_{ay}^{(dia)}}{1 + \frac{|W_{ay}^{(dia)}|}{V_{Ta}}}, & (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad (\text{C.19})$$

$$U_{ax}^{(dia)} = \tilde{V}_{ax}^{(dia)} + V_{ax}^{(E)} \quad U_{ay}^{(dia)} = \tilde{V}_{ay}^{(dia)} + V_{ay}^{(E)} \quad (\text{C.20})$$

$$U_{ax}^{(2dia)} = 2\tilde{V}_{ax}^{(dia)} + V_{ax}^{(E)} \quad U_{ay}^{(2dia)} = 2\tilde{V}_{ay}^{(dia)} + V_{ay}^{(E)} \quad (\text{C.21})$$

$$\tilde{U}_{ax}^{(dia)} = \begin{cases} U_{ax}^{(dia)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{U_{ax}^{(dia)}}{1 + \frac{U_{ax}^{(dia)}}{V_{Ta}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad \tilde{U}_{ay}^{(dia)} = \begin{cases} U_{ay}^{(dia)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{U_{ay}^{(dia)}}{1 + \frac{U_{ay}^{(dia)}}{V_{Ta}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad (\text{C.22})$$

$$\tilde{U}_{ax}^{(2dia)} = \begin{cases} U_{ax}^{(2dia)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{U_{ax}^{(2dia)}}{1 + \frac{U_{ax}^{(2dia)}}{V_{Ta}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad \tilde{U}_{ay}^{(2dia)} = \begin{cases} U_{ay}^{(2dia)}, (z_{n,a} = 1 \text{ and } z_a \neq 1) \text{ or } z_a = 0 \\ \frac{U_{ay}^{(2dia)}}{1 + \frac{U_{ay}^{(2dia)}}{V_{Ta}}}, (z_{n,a} \neq 1 \text{ or } z_a \neq 1) \text{ and } z_a \neq 0 \end{cases} \quad (\text{C.23})$$

$$E_{k,a} = \frac{m_a}{2} V_{\parallel a}^2 \quad (\text{C.24})$$

$$W_{a\perp}^{(dia)} = \begin{cases} -\frac{1}{z_a e B n_a} \frac{1}{h_y} \frac{\partial n_a T_i}{\partial y}, z_a \neq 0 \\ 0, z_a = 0 \end{cases} \quad W_{ay}^{(dia)} = \begin{cases} \frac{B_z}{z_a e B^2 n_a} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x}, z_a \neq 0 \\ 0, z_a = 0 \end{cases} \quad (\text{C.25})$$

$$V_{a\perp}^{(E \times B)} = \begin{cases} -\frac{1}{B} \frac{1}{h_y} \frac{\partial \Phi}{\partial y}, z_a \neq 0 \\ 0, z_a = 0 \end{cases} \quad V_{ay}^{(E \times B)} = \begin{cases} \frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x}, z_a \neq 0 \\ 0, z_a = 0 \end{cases} \quad (\text{C.26})$$

$$n_e = \sum_{a=0}^{ns-1} z_a n_a, \quad n_i = \sum_{a=0}^{ns-1} n_a \quad (\text{C.27})$$

$$p_a = n_a (z_a T_e + T_i), \quad p_e = n_e T_e \quad (\text{C.28})$$

$$V_a^{corr.dpc} = \frac{c^{corr.dpc}}{16} |b_x| \sqrt{\frac{p_a}{m_a n_a}} \frac{\partial^2}{\partial x^2} \left(\frac{1}{p_a} \frac{\partial p_a}{\partial x} \right) \quad (\text{C.29})$$

and $\delta_{a,b}$ is the Kronecker delta (1 if $a = b$, 0 otherwise). Here, it is used as shortcut to identify the one hydrogenic species with $z_a = 1$ and $z_{n,a} = 1$. If the code contains multiple hydrogenic species, an error will be reported to avoid double-counting of the current contributions.

C.2.1 Source terms in density equations

$$S_a^n = S_{I_a}^n + S_{I_{a+1}}^n + S_{R_a}^n + S_{R_{a-1}}^n + S_{CX_a}^n + S_{CX_{a-1}}^n + S_a^{\text{Eirene}} \quad (\text{C.30})$$

z_a — the atomic charge of species a , $z_{n,a}$ — the nuclear charge of species a

$$S_{I_a}^n = \begin{cases} -r_{I_a} n_e n_a, & a = 0, \dots, ns-2, z_a < z_{a+1} \text{ and } z_{n,a} = z_{n,a+1} \end{cases} \quad (\text{C.31})$$

$$S_{I_{a+1}}^n = \begin{cases} r_{I_a} n_e n_a, & a = 0, \dots, ns-2, z_a < z_{a+1} \text{ and } z_{n,a} = z_{n,a+1} \\ 0 \end{cases} \quad (\text{C.32})$$

$$r_{I_a} = \exp \left(r_{0,I_a} + r_{1,I_a} \ln \left(\frac{T_e}{e} \right) \right) \quad (C.33)$$

$$S_{R_a}^n = \begin{cases} -r_{R_a} n_e n_a, & a = 1, \dots, ns-1, z_{a-1} < z_a \text{ and } z_{n-1,a} = z_{n,a} \\ 0 & \end{cases} \quad (C.34)$$

$$S_{R_{a-1}}^n = \begin{cases} r_{R_a} n_e n_a, & a = 1, \dots, ns-1, z_{a-1} < z_a \text{ and } z_{n-1,a} = z_{n,a} \\ 0 & \end{cases} \quad (C.35)$$

$$r_{R_a} = \exp \left(r_{0,R_a} + r_{1,R_a} \ln \left(\frac{T_e}{e} \right) \right) \quad (C.36)$$

$$S_{CX_a}^n = - \sum_{\substack{b=0 \\ b \neq a-1}}^{nsCX} r_{CX_{ab}} n_a n_b \quad (C.37)$$

$$S_{CX_{a-1}}^n = \sum_{\substack{b=0 \\ b \neq a-1}}^{nsCX} r_{CX_{ab}} n_a n_b \quad (C.38)$$

$$(C.39)$$

$$S_{CX_b}^n = - \sum_{\substack{a=0 \\ b \neq a-1}}^{ns-1} r_{CX_{ab}} n_a n_b \quad (C.40)$$

$$S_{CX_{b+1}}^n = \sum_{\substack{a=0 \\ b \neq a-1}}^{ns-1} r_{CX_{ab}} n_a n_b \quad (C.41)$$

$$r_{CX_{ab}} = \exp \left(r_{0,CX_{ab}} + r_{1,CX_{ab}} \ln \left(\frac{m_p T_i}{m_b e} \right) \right) \quad (C.42)$$

C.2.2 Velocities

$$V_a^{corr.dpc} = \frac{c^{corr.dpc}}{16} |b_x| \sqrt{\frac{p_a}{m_a n_a}} \frac{\partial^2}{\partial x^2} \left(\frac{1}{p_a} \frac{\partial p_a}{\partial x} \right) \quad (C.43)$$

$$V_{ax}^{(E)} = \begin{cases} -c_{E \times B} \frac{B_z}{B^2} \frac{1}{h_y} \frac{\partial \Phi}{\partial y}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (C.44)$$

$$V_{ay}^{(E)} = \begin{cases} c_{E \times B} \frac{B_z}{B^2} \frac{1}{h_x} \frac{\partial \Phi}{\partial x}, & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (C.45)$$

$$\tilde{V}_{ax}^{(dia)} = \begin{cases} c_{dia} \frac{T_i B_z}{z_a e} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right), & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (C.46)$$

$$\tilde{V}_{ay}^{(dia)} = \begin{cases} -c_{dia} \frac{T_i B_z}{z_a e} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right), & z_a \neq 0 \\ 0, & z_a = 0 \end{cases} \quad (C.47)$$

$$V_{ax}^{(AN)} = 0$$

$$V_{ay}^{(AN)} = c_{V_a^{(AN)},0} + c_{V_a^{(AN)},1} \frac{10^{20}}{n_e} + c_{V_a^{(AN)},2} D_{f,a} + c_{V_a^{(AN)},3} \frac{T_e}{16e|B|} \quad (C.48)$$

$$a = 0, c_{V_a^{(AN)},0} = 0, c_{V_a^{(AN)},1} = 0, c_{V_a^{(AN)},2} = 0, c_{V_a^{(AN)},3} = 0$$

$$a = 1, c_{V_a^{(AN)},0} = 0, c_{V_a^{(AN)},1} = 0, c_{V_a^{(AN)},2} = 0, c_{V_a^{(AN)},3} = 0$$

C.2.3 Particle transport coefficients

$$\begin{cases} D_{axy} = \left(\frac{1}{B^2} - \frac{1}{\langle B^2 \rangle} \right) \frac{B_z}{z_a e}, & z_a \neq 0 \\ D_{axy} = 0, & z_a = 0 \end{cases}, \text{ where } \langle B^2 \rangle = \frac{\iint B^2 \sqrt{g} dx dy}{\iint_{core} \sqrt{g} dx dy} \quad (C.49)$$

$$\begin{cases} D_{ayx} = \left(\frac{1}{B^2} - \frac{1}{\langle B^2 \rangle} \right) \frac{B_z}{z_a e}, & z_a \neq 0 \\ D_{ayx} = 0, & z_a = 0 \end{cases} \quad (C.49)$$

$$D_{n,a} = D_{n,a}^{(AN)} \quad (C.50)$$

$$D_{n,a}^{(AN)} = \left(c_{D_{n,a},0} + c_{D_{n,a},1} \frac{10^{20}}{n_e} + c_{D_{n,a},2} D_{f,a} + c_{D_{n,a},3} \frac{T_e}{16e|B|} \right) f_D(x, y) \quad (C.51)$$

$$f_D(x, y) = \begin{cases} 1, & z_a = 0 \\ \alpha_{1,D}, & -\delta_{1,D} \leq y \leq 0, x \in \text{Core}, z_a \neq 0 \\ \alpha_{2,D}, & 0 \leq y \leq \delta_{2,D}, x \in \text{Core} \\ \alpha_{3,D}, & x, y \in \text{PFR} \cup \text{SOL} \end{cases} \quad (C.52)$$

$$a = 0, \quad c_{D_{n,a},0} = 0, \quad c_{D_{n,a},1} = 0, \quad c_{D_{n,a},2} = 0, \quad c_{D_{n,a},3} = 0$$

$$a = 1, \quad c_{D_{n,a},0} = 0, \quad c_{D_{n,a},1} = 0, \quad c_{D_{n,a},2} = 1, \quad c_{D_{n,a},3} = 0$$

$$D_{f,a} = c_{D_{f,a},0} + c_{D_{f,a},1} \frac{10^{20}}{n_i} + \frac{c_{D_{f,a},2} \sqrt{\frac{T_i}{m_a}}}{c_{D_{f,a},3} n_i + c_{D_{f,a},4} n_e},$$

a	$c_{D_{f,a},0}$	$c_{D_{f,a},1}$	$c_{D_{f,a},2}$	$c_{D_{f,a},3}$	$c_{D_{f,a},4}$	$c_{D_{f,a},5}$	$c_{D_{f,a},6}$	$c_{D_{f,a},7}$
0	10^2	0	1	$5 \cdot 10^{-20}$	$1.5 \cdot 10^{-18}$	0	0	0
1	0.3	0	0	—	—	0	0	0

$$\tilde{D}_{p,ax} = \begin{cases} \frac{D_{p,a}}{\left(1 + \left(\frac{\sqrt{g}}{h_x} D_{p,a} \left| \frac{1}{h_x} \frac{\partial p_a}{\partial x} \right| \right)^{\gamma_1} \right)^{\frac{1}{\gamma_1}}}, & V_{Ti} = \sqrt{\frac{8T_i}{\pi m_a}}, \quad z_a = 0 \\ D_{p,a}, & z_a \neq 0 \end{cases} \quad (C.53)$$

$$\tilde{D}_{p,ay} = \begin{cases} \frac{D_{p,a}}{\left(1 + \left(\frac{\sqrt{g}}{h_y} D_{p,a} \left| \frac{1}{h_y} \frac{\partial p_a}{\partial y} \right| \right)^{\gamma_1} \right)^{\frac{1}{\gamma_1}}}, & V_{Ti} = \sqrt{\frac{8T_i}{\pi m_a}}, \quad z_a = 0 \\ D_{p,a}, & z_a \neq 0 \end{cases} \quad (C.54)$$

$$D_{p,a} = D_{p,a}^{(AN)} \quad (C.55)$$

$$D_{p,a}^{(AN)} = \frac{1}{z_a T_e + T_i} \left(c_{D_{p,a},0} + c_{D_{p,a},1} \frac{10^{20}}{n_e} + c_{D_{p,a},2} D_{f,a} + c_{D_{p,a},3} \frac{T_e}{16e|B|} \right)$$

$$a = 0, c_{D_{p,a},0} = 0, c_{D_{p,a},1} = 0, c_{D_{p,a},2} = 1, c_{D_{p,a},3} = 0$$

$$a = 1, c_{D_{p,a},0} = 0, c_{D_{p,a},1} = 0, c_{D_{p,a},2} = 0, c_{D_{p,a},3} = 0$$

C.3 Parallel momentum conservation equations

The standard form of the parallel momentum conservation equation, Eq. (C.56), as implemented in the code for the default choice of `b2sigp_style` set to '2', is:

$$\begin{aligned} m_a \frac{\partial n_a V_{\parallel a}}{\partial t} + \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \Gamma_{ax}^m \right) + \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial y} \left(\frac{h_z \sqrt{g}}{h_y} \Gamma_{ay}^m \right) + \frac{b_x}{h_x} \frac{\partial n_a T_i}{\partial x} + Z_a e n_a \frac{b_x}{h_x} \frac{\partial \Phi}{\partial x} = \\ = S_{a\parallel}^m + S_{CF_a}^m + S_{fr_a}^m + S_{Therm_a}^m + S_{I_a}^m + S_{R_a}^m + S_{CX_a}^m + S_{AN_a}^m + S_{EIRENE_a}^m \end{aligned} \quad (C.56)$$

However, one can also use the formulation given as Eq. (C.57), by setting `b2sigp_style` set to '1':

$$\begin{aligned} m_a \frac{\partial n_a V_{\parallel a}}{\partial t} + \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \Gamma_{ax}^m \right) + \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial y} \left(\frac{h_z \sqrt{g}}{h_y} \Gamma_{ay}^m \right) + \frac{b_x}{h_x} \frac{\partial n_a T_i}{\partial x} + \frac{b_x}{h_x} \frac{Z_a n_a \partial n_e T_e}{n_e \partial x} = \\ = S_{a\parallel}^m + S_{CF_a}^m + S_{fr_a}^m + S_{Therm_a}^m + S_{I_a}^m + S_{R_a}^m + S_{CX_a}^m + S_{AN_a}^m + S_{EIRENE_a}^m \end{aligned} \quad (C.57)$$

The treatment using Eq. (C.56) is recommended for cases with a single main hydrogenic ion (or at least where the main species is the lightest one) and an arbitrary amount of impurities. It also requires the current continuity equation (C.181) to be solved simulatenously for Φ (i.e. 'b2news_poteq' set to '1'). For cases with hydrogenic ion mixtures or non-hydrogenic plasmas, the use of Eq. (C.57) is preferred.

C.3.1 Sources for parallel momentum equations

$$S_a^m = S_{a\parallel}^m + S_{CF_a}^m + S_{fr_a}^m + S_{Therm_a}^m + S_{I_a}^m + S_{R_a}^m + S_{CX_a}^m + S_{AN_a}^m \quad (C.58)$$

$$S_{a\parallel}^m = - \left(\nabla \cdot \frac{\leftrightarrow \pi_a}{\pi_a} \right)_{\parallel} = \begin{cases} - \left(\nabla \cdot \frac{\leftrightarrow \pi_a^{V_{\parallel}}}{\pi_a} \right)_{\parallel}^{res} - \left(\nabla \cdot \frac{\leftrightarrow \pi_a^{q_{\parallel}}}{\pi_a} \right)_{\parallel} & z_a = 1 \ \& \ z_{n,a} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (C.59)$$

$$\left(\nabla \cdot \frac{\leftrightarrow \pi_a^{V_{\parallel}}}{\pi_a} \right)_{\parallel}^{res} = \begin{cases} - \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \frac{4 \tilde{\eta}_{ax}^{(CL)}}{3} \frac{\partial \ln \left(h_z B^{\frac{1}{2}} \right)}{B^{\frac{1}{2}} h_x \partial x} \right) B^{\frac{1}{2}} V_{\parallel a}, & z_a = 1 \ \& \ z_{n,a} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (C.60)$$

$$\left(\nabla \cdot \frac{\leftrightarrow \pi_a^{q_{\parallel}}}{\pi_a} \right)_{\parallel} = \begin{cases} - B^{\frac{3}{2}} b_x \frac{\partial}{h_x \partial x} \left(\alpha^{NEO} \frac{b_x \tilde{r}_a}{B^2} \frac{\partial \left(B^{\frac{1}{2}} \tilde{q}_a^{(0)} \right)}{h_x \partial x} \right), & z_a = 1 \ \& \ z_{n,a} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (C.61)$$

$$S_{CF_a}^m = m_a b_x n_a V_{\parallel a}^2 \frac{1}{h_z h_x} \frac{\partial h_z}{\partial x} \quad (\text{C.62})$$

$S_{AN_a}^m$ is a correcting source due to the radial anomalous current (here assumed to be carried by hydrogenic ions only):

$$S_{AN_a}^m = \begin{cases} 0, & z_a \neq 1 \text{ or } z_{n,a} \neq 1 \\ -j_y^{(AN)} \frac{B B_x}{B_z}, & z_a = 1 \text{ and } z_{n,a} = 1 \end{cases} \quad (\text{C.63})$$

Parallel momentum source due to ionization

$$S_{I_a}^m = S_{I_a}^q + S_{I_{a+1}}^q \quad (\text{C.64})$$

$$S_{I_a}^q = \begin{cases} -r_{I_a} n_e n_a \left(\frac{m_a + m_{a+1}}{2} \right) V_{\parallel a}, & a = 0, \dots, ns-2, \quad z_a < z_{a+1} \text{ and } z_{n,a} = z_{n,a+1} \\ 0 & \end{cases} \quad (\text{C.65})$$

$$S_{I_{a+1}}^q = \begin{cases} r_{I_a} n_e n_a \left(\frac{m_a + m_{a+1}}{2} \right) V_{\parallel a}, & a = 0, \dots, ns-2, \quad z_a < z_{a+1} \text{ and } z_{n,a} = z_{n,a+1} \\ 0 & \end{cases} \quad (\text{C.66})$$

Parallel momentum source due to recombination

$$S_{R_a}^m = S_{R_a}^q + S_{R_{a-1}}^q \quad (\text{C.67})$$

$$S_{R_a}^q = \begin{cases} -r_{R_a} n_e n_a \left(\frac{m_a + m_{a-1}}{2} \right) V_{\parallel a}, & a = 1, \dots, ns-1, \quad z_{a-1} < z_a \text{ and } z_{n,a-1} = z_{n,a} \\ 0 & \end{cases} \quad (\text{C.68})$$

$$S_{R_{a-1}}^q = \begin{cases} r_{R_a} n_e n_a \left(\frac{m_a + m_{a-1}}{2} \right) V_{\parallel a}, & a = 1, \dots, ns-1, \quad z_{a-1} < z_a \text{ and } z_{n,a-1} = z_{n,a} \\ 0 & \end{cases} \quad (\text{C.69})$$

$$S_{CX_a}^m = S_{CX_a}^q + S_{CX_{a-1}}^q + S_{CX_b}^q + S_{CX_{b+1}}^q \quad (\text{C.70})$$

$$S_{CX_a}^q = - \sum_{\substack{b=0 \\ b \neq a-1}}^{ns_{CX}} r_{CX_{ab}} n_a n_b \left(\frac{m_a + m_{a-1}}{2} \right) V_{a\parallel} \quad (\text{C.71})$$

$$S_{CX_{a-1}}^q = \sum_{\substack{b=0 \\ b \neq a-1}}^{ns_{CX}} r_{CX_{ab}} n_a n_b \left(\frac{m_a + m_{a-1}}{2} \right) V_{a\parallel} \quad (\text{C.72})$$

$$S_{CX_b}^q = - \sum_{\substack{a=0 \\ b \neq a-1}}^{ns-1} r_{CX_{ab}} n_a n_b \left(\frac{m_b + m_{b+1}}{2} \right) V_{b\parallel} \quad (\text{C.73})$$

$$S_{CX_{b+1}}^q = \sum_{\substack{a=0 \\ b \neq a-1}}^{ns-1} r_{CX_{ab}} n_a n_b \left(\frac{m_b + m_{b+1}}{2} \right) V_{b\parallel} \quad (\text{C.74})$$

If using Eq. (C.56), the thermal and friction forces have two parts:

$$S_{Therm,a}^m = S_{Therm,ea}^m + S_{Therm,ia}^m \quad (C.75)$$

$$S_{fr,a}^m = S_{fr,ea}^m + S_{fr,ia}^m \quad (C.76)$$

The electron contributions are:

$$S_{Therm,ea}^m = -\alpha_{ex} c_e^{(1)} \frac{n_a z_a^2 m_e}{b_x e \zeta_e} \frac{1}{c_{\sigma_{\parallel}}^{(Luciani)}} \frac{1}{h_x} \frac{\partial T_e}{\partial x}, \quad z_a \neq 0 \quad (C.77)$$

$$S_{fr,ea}^m = \frac{b_x^2 e^2 z_a^2}{\sigma_{\parallel} z_{eff}} n_e n_a (V_{e\parallel} - V_{a\parallel}), \quad z_a \neq 0 \quad (C.78)$$

and the ion contributions are:

$$S_{Therm,ia}^m = \sum_{\substack{b=0 \\ b \neq a}}^{ns-1} c_{\alpha_{ab}}^{(F \text{ lim})} z_a^2 z_b^2 \frac{m_p}{\zeta_p e} \frac{n_a n_b}{n_a + n_b} \alpha_{ab} \frac{1}{h_x} \frac{\partial T_i}{\partial x}, \quad z_a \neq 0 \quad (C.79)$$

$$S_{fr,ia}^m = \sum_{\substack{b=0 \\ b \neq a}}^{ns-1} R_{ab}^{(V_{\parallel})} = \sum_{\substack{b=0 \\ b \neq a}}^{ns-1} \frac{1}{\zeta_p} m_p \sqrt{\frac{\frac{m_a}{m_p} \frac{m_b}{m_p}}{\frac{m_a}{m_p} + \frac{m_b}{m_p}}} z_a^2 z_b^2 n_a n_b K_{ab}^{(V_{\parallel})} (V_{b\parallel} - V_{a\parallel}) \quad (C.80)$$

where

$$\alpha_{ab} = \frac{\zeta_p e (n_a + n_b)}{\sqrt{m_p}} \sqrt{\frac{\frac{m_a}{m_p} \frac{m_b}{m_p}}{\frac{m_a}{m_p} + \frac{m_b}{m_p}}} \left(\frac{K_{ab}^{(Therm)}}{\kappa_b} - \frac{K_{ba}^{(Therm)}}{\kappa_a} \right) b_x, \quad z_a \neq 0$$

$$\zeta_e = \frac{3}{4\sqrt{2\pi}} \frac{\sqrt{m_e} T_e^{\frac{3}{2}}}{\ln \Lambda} \left(\frac{4\pi\epsilon_0}{e^2} \right)^2, \quad \tau_e = \frac{\zeta_e}{z_{eff} n_e}$$

$$\zeta_p = \frac{3}{4\sqrt{2\pi}} \frac{\sqrt{m_p} T_i^{\frac{3}{2}}}{\ln \Lambda} \left(\frac{4\pi\epsilon_0}{e^2} \right)^2, \quad \tau_a = \frac{\zeta_p}{z_a^2 z_{eff} n_e} \sqrt{\frac{m_a}{m_p}}$$

$$c_e^{(1)} = \frac{(1+0.24z_{eff})(1+0.93z_{eff})}{(1+2.56z_{eff})(1+0.29z_{eff})}, \quad c_e^{(2)} = 1.56 \frac{(1+1.4z_{eff})(1+0.52z_{eff})}{(1+2.56z_{eff})(1+0.29z_{eff})}$$

$$\kappa_a = \sum_{r=0}^{ns-1} z_a^2 z_r^2 n_r \sqrt{m_p} \sqrt{\frac{\frac{m_a}{m_p} \frac{m_r}{m_p}}{\frac{m_a}{m_p} + \frac{m_r}{m_p}}}$$

$$K_{rs}^{(Therm)} = \begin{cases} c_{imp}^{(2)}, & s \neq r, & s = s_{main}, & z_r \neq 0, & z_s \neq 0 \\ 0, & r \neq r_{main}, & s \neq s_{main}, & z_r \neq 0, & z_s \neq 0 \\ 0, & r = r_{main}, & \forall s \end{cases}$$

$$K_{rs}^{(V_{\parallel})} = \begin{cases} c_{imp}^{(1)}, & r = r_{main}, & s \neq r, & z_r \neq 0, & z_s \neq 0 \\ c_{imp}^{(1)}, & s = s_{main}, & r \neq s, & z_r \neq 0, & z_s \neq 0 \\ 1, & r \neq r_{main}, & s \neq s_{main}, & z_r \neq 0, & z_s \neq 0 \\ 0, & r = s, & \text{or} & z_r = 0, & z_s = 0 \end{cases}$$

$$c_{imp}^{(1)} = \frac{(1+0.24z_{eff,imp})(1+0.93z_{eff,imp})}{(1+2.56z_{eff,imp})(1+0.29z_{eff,imp})}, \quad c_{imp}^{(2)} = 1.56 \frac{(1+1.4z_{eff,imp})(1+0.52z_{eff,imp})}{(1+2.56z_{eff,imp})(1+0.29z_{eff,imp})}$$

$$z_{eff,imp} = \frac{\sum_{a=0}^{ns-1} n_a z_a^2}{n_{a_{main}} z_{a_{main}}^2}, \quad c_{\sigma_{\parallel}}^{(Luciani)} = c_{\kappa_{ex}}^{(Luciani)}$$

$$c_{\alpha_{ab}}^{(F\lim)} = \begin{cases} 1, & \text{inside the separatrix} \\ \frac{1}{1 + \frac{|\alpha_{ab}|}{c_{flab} j_{ab, \max}} \frac{\partial T_i}{\partial x}}, & \text{outside the separatrix,} \end{cases} \quad j_{ab, \max} = e K_{ab}^{(V_{\parallel})} \sqrt{\frac{T_i}{m_p}} (n_a + n_b), \quad c_{flab} = c_{flim_4} = 0.5$$

Otherwise, when using Eq. (C.57), the thermal and friction force term expressions are:

$$S_{Therm,a}^m = n_a \left(\frac{z_a}{n_e} - \frac{z_a^2}{n_{e2}} \right) b_x e \left[\hat{E}_x - \frac{n_e}{e} (c_{the,a} \Delta T_e + c_{thi,a} \Delta T_i) \right],$$

$$S_{fr,a}^m = \sum_{\substack{b=0 \\ b \neq a, z_a \neq 0, z_b \neq 0}}^{ns-1} \frac{1}{\tau_p} m_p \sqrt{\frac{\frac{m_a}{m_p} \frac{m_b}{m_p}}{\frac{m_a}{m_p} + \frac{m_b}{m_p}}} z_a^2 z_b^2 n_b n_a (V_{b\parallel} - V_{a\parallel}) \quad (C.81)$$

where

$$\tau_p = \frac{3}{4\sqrt{2}\pi} \frac{\sqrt{m_p} T_i^{\frac{3}{2}}}{\ln \Lambda} \left(\frac{4\pi\epsilon_0}{e^2} \right)^2, \text{ where the Coulomb logarithm } \ln \Lambda \text{ can be chosen as}$$

$$\ln \Lambda = \begin{cases} \max\{5; 23.4 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 3.45 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} \leq 50 \\ \max\{5; 25.3 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 2.30 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} > 50 \end{cases} \quad \begin{matrix} \text{Braginsky} \\ \text{formula} \end{matrix} \quad (C.82)$$

Wesson formula for ion-ion collisions

$$n_{e2} = \sum_{a=0}^{ns-1} z_a^2 n_a, \quad l_{t_e} = 0.3; \quad l_{t_i} = 0.3; \quad z_{\lambda_e} = \frac{1.5 \cdot 10^{16}}{n_e} \left(\frac{T_e}{e} \right)^2, \quad z_{\lambda_i} = \frac{1.5 \cdot 10^{16}}{n_{e2}} \left(\frac{T_i}{e} \right)^2$$

$$\Delta T_e = \begin{cases} \left| \min \left\{ \left| \frac{\partial T_e}{h_x \partial x} \right|, \frac{l_{t_e} T_e}{z_{\lambda_e} |b_x|} \right\} \right|, & \frac{\partial T_e}{h_x \partial x} \geq 0, \quad l_{t_e} \leq 0 \\ - \left| \min \left\{ \left| \frac{\partial T_e}{h_x \partial x} \right|, \frac{l_{t_e} T_e}{z_{\lambda_e} |b_x|} \right\} \right|, & \frac{\partial T_e}{h_x \partial x} < 0, \quad l_{t_e} > 0 \end{cases} \quad (C.83)$$

$$\Delta T_i = \begin{cases} \left| \min \left\{ \left| \frac{\partial T_i}{h_x \partial x} \right|, \frac{l_{t_i} T_i}{z_{\lambda_i} |b_x|} \right\} \right|, & \frac{\partial T_i}{h_x \partial x} \geq 0, \quad l_{t_i} \leq 0 \\ - \left| \min \left\{ \left| \frac{\partial T_i}{h_x \partial x} \right|, \frac{l_{t_i} T_i}{z_{\lambda_i} |b_x|} \right\} \right|, & \frac{\partial T_i}{h_x \partial x} < 0, \quad l_{t_i} > 0 \end{cases} \quad (C.84)$$

$c_{the,a} = 0, a = 0, \dots, ns-1$ and $c_{thi,a} = 2.65, a = 0, \dots, ns-1$ if the 21-moment approximation is not used.

C.3.2 Parallel momentum fluxes

$$\Gamma_{ax}^m = \begin{cases} m_a V_{\parallel a} \Gamma_{ax}^{Cor} + \frac{4}{3} \tilde{\eta}_{ax}^{(CL)} \frac{\partial \ln h_z}{h_x \partial x} V_{\parallel a} - \eta_{ax} \frac{\partial V_{\parallel a}}{h_x \partial x}, & z_a = 1 \text{ \& } z_{n,a} = 1 \\ m_a V_{\parallel a} \Gamma_{ax}^{Cor} - \eta_{ax} \frac{\partial V_{\parallel a}}{h_x \partial x}, & \text{otherwise} \end{cases} \quad (C.85)$$

$$\Gamma_{ay}^m = m_a V_{\parallel a} \Gamma_{ay}^{Cor} - \eta_{ay} \frac{\partial V_{\parallel a}}{h_y \partial y} \quad (C.86)$$

C.3.3 Parallel momentum transport coefficients

$$\eta_{ax} = \eta_a^{(AN)} + \frac{4}{3} \tilde{\eta}_{ax}^{(CL)} \quad (C.87)$$

$$\eta_a^{(AN)} = m_a n_a \nu_a^{visc} = \begin{cases} \left(c_{\eta_a,0} + c_{\eta_a,1} \frac{10^{20}}{n_e} + c_{\eta_a,2} D_{f,a} + c_{\eta_a,3} \frac{T_e}{16e|B|} \right) m_a n_a f_{\eta_a}(x, y), & z_a \neq 0 \\ \left(c_{\eta_a,0} + c_{\eta_a,1} \frac{10^{20}}{n_e} + c_{\eta_a,2} D_{f,a} + c_{\eta_a,3} \frac{T_e}{16e|B|} \right) m_a n_a, & z_a = 0 \end{cases} \quad (C.88)$$

$$f_{\eta_a}(x, y) = \begin{cases} 1, & z_a = 0 \\ \alpha_{1,\eta_a}, & -\delta_{1,\eta_a} \leq y \leq 0, \quad x \in \text{Core}, z_a \neq 0 \\ \alpha_{2,\eta_a}, & 0 \leq y \leq \delta_{2,\eta_a}, \quad x \in \text{Core} \\ \alpha_{3,\eta_a}, & x, y \in \text{PFR} \cup \text{SOL} \end{cases} \quad (C.89)$$

$$a = 0, \quad c_{\eta_a,0} = 0, \quad c_{\eta_a,1} = 0, \quad c_{\eta_a,2} = 1, \quad c_{\eta_a,3} = 0$$

$$a = 1, \quad c_{\eta_a,0} = 0, \quad c_{\eta_a,1} = 0, \quad c_{\eta_a,2} = 1, \quad c_{\eta_a,3} = 0$$

$$\tilde{\eta}_{ax}^{(CL)} = c_{\eta_{ax}^{(CL)}} \tilde{\eta}_{ax}^{(CL)} \quad (C.90)$$

$$c_{\eta_{ax}}^{(CL)} = \begin{cases} 1, & z_a \neq 0 \text{ and only for inner flux surfaces} \\ \frac{1}{1 + \frac{\frac{\sqrt{g}}{h_x} \tilde{\eta}_{ax}^{(CL)} \frac{\partial V_{|a}}{h_x \partial x}}{V_{m_a}^{(\max)}}}, & \text{for all other cases} \end{cases} \quad (C.91)$$

$$V_{m_a}^{(\max)} = c_{flim_2} \frac{\sqrt{g}}{h_x} |b_x| n_a T_i, \quad c_{flim_2} = \frac{3}{8} \quad (C.92)$$

$$\tilde{\eta}_{ax}^{(CL)} = c_{\eta_{ax}^{(CL)}}^{Luciani} \tilde{\eta}_{ax}^{(CL)} \quad (C.93)$$

If 'b2trcllluciani' is set to '1', the Luciani flux limiter is given by:

$$c_{\eta_{ax}}^{Luciani} = \begin{cases} \frac{1}{1 + \frac{7.5 \cdot 10^{16} \left(\frac{T_i}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{for ions on inner flux surfaces} \\ 1, & \text{for all other cases} \end{cases}$$

For the plateau and banana transport regimes, the correction, with 'b2trcllluciani' set to '3' (default), is given by:

$$c_{\eta_{ax}}^{Luciani} = \begin{cases} \frac{1}{1 + \frac{15.12 \cdot 10^{16} \left(\frac{T_i}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}} \cdot \frac{1}{1 + \frac{15.12 \cdot 10^{16} \cdot \varepsilon^{3/2} \left(\frac{T_i}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{for ions on inner flux surfaces} \\ 1, & \text{for all other cases} \end{cases}$$

$$c(y) = \oint h_x |b_x|^{-1} dx, \quad \varepsilon = \frac{(\oint h_x dx)^2}{\oint \sqrt{g} / h_y dx}$$

$$\eta_{ax}^{(CL)} = \begin{cases} 0, & z_a = 0 \\ \eta \sqrt{2} T_i n_a \tau_a b_x^2, & z_a \neq 0 \end{cases}, \quad \eta = 0.96 \quad (C.94)$$

$$\tau_a = \frac{3}{4\sqrt{2\pi}} \frac{\sqrt{m_a} T_i^{\frac{3}{2}}}{z_a^2 z_{eff} n_e \ln \Lambda} \left(\frac{4\pi\epsilon_0}{e^2} \right)^2, \quad (\text{C.95})$$

where the Coulomb logarithm $\ln \Lambda$ can be chosen as

$$\ln \Lambda = \begin{cases} 12 \\ \max\{5; 23.4 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^{16}} \right) + 3.45 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} \leq 50 \\ \max\{5; 25.3 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^{16}} \right) + 2.30 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} > 50 \\ \max\{5; 17.3 - 0.5 \cdot \ln \left(\frac{n_e}{10^{20}} \right) + 1.5 \cdot \ln \left(\frac{T_i}{e \cdot 10^3} \right)\} & \text{Wesson formula} \end{cases} \quad \begin{matrix} \text{Braginsky} \\ \text{formula} \\ \text{for ion-ion collisions} \end{matrix}$$

$$\eta_{ay} = \eta_a^{(AN)} + \eta_{ay}^{(CL)}, \eta_{ay}^{(CL)} = 0$$

$$\tilde{q}_a^{(0)} = q^{(1)} \arctan \left(\frac{q_a^{(0)}}{q^{(1)}} \right), \quad q^{(1)} = c_{q1} n_i T_i \sqrt{\frac{T_i + T_e}{m_i}}$$

$c_{q1} = 0.5$ or 'b2siav_cqip1' set to '0.1' in b2mn.dat

The three-point smoothing procedure is imposed twice on $\tilde{q}^{(0)}$ along the x direction.

In case 'b2siav_style_qip' is set to '0' in b2mn.dat:

$$q_a^{(0)} = q_{a\parallel}^{(0)} + \frac{B}{B_x} q_{ax}^{(dia)} = \begin{cases} 0 & \text{outside the separatrix} \\ -\frac{5}{2} \frac{B_z B}{B_x \langle B^2 \rangle} \frac{n_a T_i}{e} \frac{\partial T_i}{h_y \partial y} & \text{inside the separatrix} \end{cases} \quad (\text{C.96})$$

In case 'b2siav_style_qip' is set to '1' in b2mn.dat:

$$q_a^{(0)} = q_{a\parallel}^{(0)} + \frac{B}{B_x} q_{ax}^{(dia)} = \begin{cases} 0 & \text{outside the separatrix} \\ -\tilde{k}_{ix}^{(CL)} b_x \frac{\partial T_i}{h_x \partial x} - \frac{5}{2} \frac{B_z B}{B_x \langle B^2 \rangle} \frac{n_a T_i}{e} \frac{\partial T_i}{h_y \partial y} & \text{inside the separatrix} \end{cases} \quad (\text{C.97})$$

$$\tilde{\tau}_a = \frac{\tau_a^{(Br)}}{1 + \frac{15 \cdot 12 \cdot 10^{16} \left(\frac{T_i}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}} \cdot \frac{1}{1 + \frac{15 \cdot 12 \cdot 10^{16} \cdot \epsilon^{3/2} \left(\frac{T_i}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}}, \quad \tau_a^{(Br)} = \frac{3}{4\sqrt{\pi}} \frac{\sqrt{m_a} T_i^{\frac{3}{2}}}{z_a^2 z_{eff} n_e \ln \Lambda} \left(\frac{4\pi\epsilon_0}{e} \right)^2$$

This correction is compatible **only** with plateau and banana correction ('b2trcl_lluciani' set to '3').

$\alpha^{NEO} = \frac{8}{15} \left(\frac{-0.17 + 1.05\nu_*^{1/2} + 2.7\nu_*^2\epsilon^3}{1 + 0.7\nu_*^{1/2} + \nu_*^2\epsilon^3} - 1 \right)$ gives the correct coefficient before the temperature gradient in the neo-classical electric field in the plateau and banana regime

$$\nu_* = \frac{c(y)}{2\pi\epsilon^{3/2}\tau_a\sqrt{T_i/m_a}}, \quad c(y) = \oint h_x |b_x|^{-1} dx, \quad \epsilon = \frac{(\oint h_x dx)^2}{\oint \sqrt{g}/h_y dx}$$

C.4 Electron energy equations

$$\begin{aligned} & \frac{3}{2} \frac{\partial n_e T_e}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \right) + \frac{n_e T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} b_x V_{e\parallel} \right) = \\ & = Q_e + c_{E \times B} n_e T_e B_z \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) - \frac{j_y^{(ST)}}{e n_e} \frac{\partial n_e T_e}{h_y \partial y} + Q_{Fei} + Q_e^{\text{EIRENE}} \end{aligned} \quad (\text{C.98})$$

C.4.1 Electron energy sources

$$S_e^{heat} = Q_e + c_{E \times B} n_e T_e B_z \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) - \frac{j_y^{(ST)}}{en_e} \frac{\partial n_e T_e}{h_y \partial y} + Q_{Fei} - \frac{n_e T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} b_x V_{e\parallel} \right) + Q_e^{\text{EIRENE}} \quad (\text{C.99})$$

$$Q_e = -Q_\Delta - Q_R$$

$$Q_\Delta = c_{eqp} \ln \Lambda e^{\frac{3}{2}} \frac{\sum_{a=0}^{ns-1} \frac{z_a^2 m_p}{m_a} n_a}{n_i} \frac{n_e n_i}{T_e^{\frac{3}{2}}} (T_e - T_i), \quad c_{eqp} = 4.8 \cdot 10^{-15}, \quad (\text{C.100})$$

where $\ln \Lambda$ can be chosen as

$$\ln \Lambda = \begin{cases} 12 \\ \max\{5; 23.4 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 3.45 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} \leq 50 \\ \max\{5; 25.3 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 2.30 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} > 50 \\ \max\{5; 15.2 - 0.5 \cdot \ln \left(\frac{n_e}{10^{20}} \right) + \ln \left(\frac{T_e}{e \cdot 10^3} \right)\} \end{cases} \quad \begin{array}{l} \text{Braginsky formula} \\ \text{Wesson formula for electron-ion collisions} \end{array} \quad (\text{C.101})$$

$$Q_R = \sum_{a=0}^{ns-1} r_{Q_a} n_e n_a \text{ from b2stel} \quad (\text{C.102})$$

$$r_{Q_a} = r_{Q_a}^{\text{ion}} + r_{Q_a}^{\text{rec}} + r_{Q_a}^{\text{brm}} + r_{Q_a}^{\text{line}} \quad (\text{C.103})$$

contains contributions from ionization, recombination, bremsstrahlung and line radiation.

$$Q_{Fei} = \begin{cases} V_{e\parallel} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} (S_{fr,ea}^m + S_{Therm,ea}^m), \text{ when Eq. (C.56) is activated ('b2sigp_style' set to '2')} \\ j_x \hat{E}_x + 0 \cdot j_y \hat{E}_y, \text{ when Eq. (C.57) is activated} \end{cases} \quad (\text{C.104})$$

C.4.2 Poloidal electron energy fluxes

$$\tilde{q}_{ex} = \tilde{f}_{ex} T_e - \kappa_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x} + 0 \cdot \alpha_{ex} T_e \hat{E}_x - q_{ex}^{P.Sch} \quad (\text{C.105})$$

$$\tilde{f}_{ex} = \frac{3}{2} \Gamma_{ex} + V_{ex}^{(str)} n_e - \tilde{c}_e^{(2)} \frac{j_x^{(\parallel)}}{e} \quad (\text{C.106})$$

$$q_{ex}^{P.Sch} = -\frac{5}{2} D_{exy} \frac{1}{h_y} \frac{\partial n_e T_e^2}{\partial y} \quad (\text{C.107})$$

$$D_{exy} = \left(\frac{1}{B^2} - \frac{1}{\langle B^2 \rangle} \right) \frac{B_z}{e}$$

$$D_{eyx} = \left(\frac{1}{B^2} - \frac{1}{\langle B^2 \rangle} \right) \frac{B_z}{e}, \text{ where } \langle B^2 \rangle = \frac{\iint_{core} B^2 \sqrt{g} dx dy}{\iint_{core} \sqrt{g} dx dy}$$

$$q_{ex} = \left[\frac{3}{2} \Gamma_{ex} - \tilde{c}_e^{(2)} \frac{j_x^{(\parallel)}}{e} - \frac{5}{2} n_e T_e \frac{B_z}{e} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) + V_{ex}^{(str)} n_e \right] T_e - \kappa_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x} + 0 \cdot \alpha_{ex} T_e \hat{E}_x \quad (C.108)$$

$$\Gamma_{ex} = \sum_{a=0}^{ns-1} z_a \Gamma_{ax}^{(he)} - \frac{j_x^{(\parallel)}}{e} \quad (C.109)$$

$$\Gamma_{ax}^{(he)} = \left(b_x V_{a\parallel} + V_{ax}^{(E)} + \frac{5}{3} V_a^{(AN)} + V_a^{corr-dpc} \right) n_a - \frac{5}{3} \left[D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} + \tilde{D}_{p,ax} \frac{1}{h_x} \frac{\partial p_a}{\partial x} \right] \quad (C.110)$$

$$q_{ex}^{(Eir)} = \left[\frac{5}{2} \Gamma_{ex}^{(Eir)} + V_{ex}^{(str)} n_e - \tilde{c}_e^{(2)} \frac{j_{\parallel}}{e} \right] T_e - \kappa_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x} \quad (C.111)$$

$$q_{ey}^{(Eir)} = \left[\frac{5}{2} \Gamma_{ey}^{(Eir)} + V_{ey}^{(str)} n_e \right] T_e - \kappa_{ey} \frac{1}{h_y} \frac{\partial T_e}{\partial y} \quad (C.112)$$

where

$$\Gamma_{ex}^{(Eir)} = \left(b_x V_{\parallel e} + V_{ex}^{(E)} + V_{ex}^{(dia)} + V_{ex}^{(AN)} \right) n_e - D_{n,a} \frac{1}{h_x} \frac{\partial n_e}{\partial x} \quad (C.113)$$

$$\Gamma_{ey}^{(Eir)} = \left(V_{ey}^{(E)} + V_{ey}^{(dia)} + V_{ey}^{(AN)} \right) n_e - D_{n,a} \frac{1}{h_y} \frac{\partial n_e}{\partial y}$$

$$V_{ex}^{(dia)} = \frac{B_z}{e B^2 n_e} \frac{\partial n_e T_e}{h_y \partial y} \quad (C.114)$$

$$V_{ey}^{(dia)} = - \frac{B_z}{e B^2 n_e} \frac{\partial n_e T_e}{h_x \partial x} \quad (C.115)$$

$$V_{ex}^{(str)} = 0$$

The parameter $\tilde{c}_e^{(2)}$ is given by $\tilde{c}_e^{(2)} = \frac{c_e^{(2)}}{(z_{eff} + \frac{\sqrt{2}}{2})} = 1.56 \frac{(1+1.4z_{eff})(1+0.52z_{eff})}{(1+2.56z_{eff})(1+0.29z_{eff})} \frac{1}{(z_{eff} + \frac{\sqrt{2}}{2})}$ when 'b2sigp-style' is set to '2' (default) or by its limiting value at $z_{eff} = 1$ of 0.71 otherwise.

C.4.3 Electron energy transport

$$\kappa_{ex} = \kappa_e^{(AN)} + \tilde{\kappa}_{ex}^{(CL)} \quad (C.116)$$

$$\kappa_e^{(AN)} = \chi_e^{(AN)} n_e \quad (C.117)$$

$$\chi_e^{(AN)} = \left(c_{he,0} + c_{he,1} \frac{10^{20}}{n_e} + c_{he,3} \frac{T_e}{16e|B|} \right) f_{he}(x, y) \quad (C.118)$$

$$c_{he,0} = 1.2, \quad c_{he,1} = 0, \quad c_{he,3} = 0$$

$$f_{h_e}(x, y) = \begin{cases} \alpha_{1, h_e}, & -\delta_{1, h_e} \leq y \leq 0, & x \in \text{Core} \\ \alpha_{2, h_e}, & 0 \leq y \leq \delta_{2, h_e}, & x \in \text{Core} \\ \alpha_{3, h_e}, & x, y \in \text{PFR} \cup \text{SOL} \end{cases}$$

$$\tilde{\kappa}_{ex}^{(CL)} = c_{\kappa_{ex}}^{(F \text{ lim})} \tilde{\kappa}_{ex}^{(CL)} \quad (\text{C.119})$$

$$c_{\kappa_{ex}}^{F \text{ lim}} = \begin{cases} 1, & \text{only for inner flux surfaces} \\ \frac{1}{1 + \frac{\sqrt{g} \tilde{\kappa}_{ex}^{(CL)} \left| \frac{\partial T_e}{\partial x} \right|}{V_{h_e}^{(\max)} n_e T_e}}, & \text{for other flux surfaces} \end{cases}$$

$$V_{h_e}^{(\max)} = c_{flim0} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{T_e}{m_e}} \quad (\text{C.120})$$

$$\tilde{\kappa}_{ex}^{(CL)} = c_{\kappa_{ex}}^{(Luciani)} \kappa_{ex}^{(CL)} b_x^2 \quad (\text{C.121})$$

$c_{flim0} = 0.15$ is the electron heat flux limiter coefficient.

The classical electron thermal diffusivity is expressed as: $\kappa_{ex}^{(CL)} = \frac{T_e}{m_e} f_{k_e}(z_{eff}) n_e \tau_e$ (Zhdanov formulation, default, 'b2tqce_fke_Zhdanov' set to '1'), with $f_{k_e}(z_{eff}) = \frac{3.9 + 2.3/z_{eff}}{0.31 + 1.2/z_{eff} + 0.41/z_{eff}^2}$, or $\kappa_{ex}^{(CL)} = \frac{5}{2} \frac{T_e}{m_e} f_{k_e}(z_{eff}) n_e \tau_e$ (Balescu formulation, 'b2tqce_fke_Zhdanov' set to '0'), with $f_{k_e}(z_{eff}) = \frac{2.16 z_{eff}}{1 + 0.27 z_{eff}}$ and $z_{eff} = \frac{1}{n_e} \sum_{a=0}^{ns-1} z_a^2 n_a$.

If the plateau and banana transport regimes correction is applied (default, 'b2trcllluciani' set to '3'), one has:

$$c_{\kappa_{ex}}^{(Luciani)} = \begin{cases} 0.58 \varepsilon^{\frac{3}{2}} / K_1, & \text{only for inner flux surfaces} \\ 1, & \text{for other flux surfaces} \end{cases}$$

If 'b2trcllluciani' is set to '1', one gets instead:

$$c_{\kappa_{ex}}^{(Luciani)} = \begin{cases} \frac{1}{1 + \frac{7.5 \cdot 10^{16} \left(\frac{T_e}{e} \right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{only for inner flux surfaces} \\ 1, & \text{for all other cases} \end{cases}$$

If 'b2trcllluciani' is set to '2', the **SOLPS 4.x** formulation is used:

$$c_{\kappa_{ex}}^{(Luciani)} = \begin{cases} \frac{1}{1 + \frac{0.3 \cdot 10^{16} \left(\frac{T_e}{e} \right)^2}{\sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{only for inner flux surfaces} \\ 1, & \text{for all other cases} \end{cases}$$

$$K_1 = \frac{1}{1 + 2.01 \cdot \nu_{*e}^{1/2} + 1.53 \cdot \nu_{*e}} + \frac{0.52 \cdot \varepsilon^3 \cdot \nu_{*e}}{1 + 0.89 \cdot \varepsilon^{3/2} \cdot \nu_{*e}}$$

$$\nu_{*e} = \frac{c(y)}{2\pi \varepsilon^{3/2} \tau_e \sqrt{T_e/m_e}} \quad (\text{C.122})$$

$$\tau_e = \frac{3}{4\sqrt{2}\pi} \frac{\sqrt{m_e T_e^{\frac{3}{2}}}}{\ln \Lambda \sum_{a=0}^{ns-1} z_a^2 n_a} \left(\frac{4\pi \varepsilon_0}{e^2} \right)^2 \quad (\text{C.123})$$

where the Coulomb logarithm $\ln \Lambda$ can be chosen as

$$\ln \Lambda = \begin{cases} 12 \\ \max\{5; 23.4 - 1.15 \cdot \log_{10}\left(\frac{n_e}{10^6}\right) + 3.45 \cdot \log_{10}\left(\frac{T_e}{e}\right)\}, & \frac{T_e}{e} \leq 50 \\ \max\{5; 25.3 - 1.15 \cdot \log_{10}\left(\frac{n_e}{10^6}\right) + 2.30 \cdot \log_{10}\left(\frac{T_e}{e}\right)\}, & \frac{T_e}{e} > 50 \\ \max\{5; 14.9 - 0.5 \cdot \ln\left(\frac{n_e}{10^{20}}\right) + \ln\left(\frac{T_e}{e \cdot 10^3}\right)\} \end{cases} \quad \begin{array}{l} \text{Braginsky formula} \\ \text{Wesson formula for electron-electron collisions} \end{array}$$

$$c(y) = \begin{cases} \int \frac{h_x(x,y)}{|b_x|} dx, & \text{for inner flux surfaces} \\ 0, & \text{for other} \end{cases}$$

Electron parallel velocity

$$V_{e\parallel} = \frac{1}{en_e} \left[\sum_{a=0}^{ns-1} z_a en_a V_{a\parallel} - \frac{j_x^{(\parallel)}}{b_x} \right] \quad (\text{C.124})$$

$$j_x^{(\parallel)} = \sigma_{\parallel} \left(\frac{1}{en_e h_x} \frac{\partial n_e T_e}{\partial x} - \frac{1}{h_x} \frac{\partial \Phi}{\partial x} \right) - \alpha_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x} - \frac{eb_x}{z_{eff}} \sum_{a=0}^{ns-1} n_a V_{a\parallel} (z_a^2 - z_a z_{eff}) \quad (\text{C.125})$$

The last term in Eq. (C.125) is activated with the switch `b2sigp_style` set to '2'.

$$\hat{E}_x = \frac{1}{en_e h_x} \frac{\partial n_e T_e}{\partial x} - \frac{1}{h_x} \frac{\partial \Phi}{\partial x} \quad (\text{C.126})$$

$$\sigma_{\parallel} = \sigma_{\parallel}^{(AN)} + \tilde{\sigma}_{\parallel}^{(CL)} \quad (\text{C.127})$$

$$\sigma_{\parallel}^{(AN)} = 0$$

$$\tilde{\sigma}_{\parallel}^{(CL)} = c_{\kappa_{ex}^{(CL)}}^{(Luciani)} \sigma_{\parallel}^{(CL)} \quad (\text{C.128})$$

The classical parallel conductivity is given as:

$$\sigma_{\parallel}^{(CL)} = b_x^2 \frac{e^2}{m_e} f_{\sigma_e}(z_{eff}) n_e \tau_e \quad (\text{C.129})$$

where the electron form factor is $f_{\sigma_e}(z_{eff}) = \frac{1}{c_e^{(1)}}$ when 'b2sigp_style' is set to '2' (Zhdanov formulation), and $f_{\sigma_e}(z_{eff}) = \frac{4.95 z_{eff}}{1+1.54 z_{eff}}$ otherwise (Balescu formulation). The parameter $c_e^{(1)}$ is given by:

$$c_e^{(1)} = \frac{(1+0.24 z_{eff})(1+0.93 z_{eff})}{(1+2.56 z_{eff})(1+0.29 z_{eff})}. \text{ See also Section B.4.}$$

$$\alpha_{ex} = \alpha_x^{(AN)} + \tilde{\alpha}_x^{(CL)} \quad (\text{C.130})$$

$$\alpha_x^{(AN)} = 0,$$

The classical thermo-electric coefficient $\alpha_x^{(CL)}$ is given as:

$$\alpha_x^{(CL)} = -b_x^2 \frac{e}{m_e} \frac{\tilde{c}_e^{(2)}}{c_e^{(1)}} n_e \tau_e z_{eff} \quad (\text{C.131})$$

when 'b2sigp_style' is set to '2' (Zhdanov formulation, default), and $\alpha_x^{(CL)} = b_x^2 \frac{e}{m_e} \sqrt{\frac{5}{2}} f_{\alpha}(z_{eff}) n_e \tau_e$ otherwise (Balescu formulation), and $f_{\alpha}(z_{eff}) = -\frac{1.25 z_{eff}}{1+0.41 z_{eff}}$. See also Section B.4.

A conductive flux limit is applied to the thermo-electric coefficient, as follows. If 'b2sigp_style' is set to '2' (default):

$$\tilde{\alpha}_x^{(CL)} = c_{\alpha_{ex}}^{(F \text{ lim})} \tilde{\alpha}_x^{(CL)} \quad (\text{C.132})$$

$$\tilde{\alpha}_x^{(CL)} = c_{\kappa_{ex}}^{(Luciani)} \alpha_x^{(CL)} \quad (\text{C.133})$$

$$c_{\alpha_{ex}}^{(F \text{ lim})} = \begin{cases} 1, & \text{inside the separatrix} \\ \frac{1}{1 + \left| \frac{\alpha_x^{(CL)} \frac{1}{h_x} \frac{\partial T_e}{\partial x}}{c_{flal} j_{e \text{ max}}} \right|}, & \text{outside the separatrix} \end{cases}$$

$$j_{e \text{ max}} = |b_x| e n_e \sqrt{\frac{T_e}{m_e}}, \quad c_{flal} = c_{flim3} = 0.5$$

If 'b2sigp_style' is set to '1':

$$\tilde{\alpha}_x^{(CL)} = c_{\kappa_{ex}}^{(F \text{ lim})} \tilde{\alpha}_x^{(CL)} \quad (\text{C.134})$$

$$\tilde{\alpha}_x^{(CL)} = c_{\kappa_{ex}}^{(Luciani)} \alpha_x^{(CL)} \quad (\text{C.135})$$

C.4.4 Radial electron energy fluxes

$$\tilde{q}_{ey} = \tilde{f}_{ey} T_e - \kappa_{ey} \frac{1}{h_y} \frac{\partial T_e}{\partial y} + 0 \cdot \alpha_{ey} T_e \hat{E}_y - q_{ey}^{P.Sch} \quad (\text{C.136})$$

$$\tilde{f}_{ey} = \frac{3}{2} \left(\Gamma_{ey} - \frac{5}{3} \frac{j_y^{(ST)}}{e} \right) + V_{ey}^{(str)} n_e$$

$$q_{ey}^{P.Sch} = \frac{5}{2} D_{eyx} \frac{1}{h_x} \frac{\partial n_e T_e^2}{\partial x} \quad (\text{C.137})$$

$$q_{ey} = \left[\frac{3}{2} \Gamma_{ey} - \frac{5}{2} \frac{j_y^{(ST)}}{e} + \frac{5}{2} n_e T_e \frac{B_z}{e} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) + V_{ey}^{(str)} n_e \right] T_e - \kappa_{ey} \frac{1}{h_y} \frac{\partial T_e}{\partial y} + 0 \cdot \alpha_{ey} T_e \hat{E}_y \quad (\text{C.138})$$

$$\Gamma_{ey} = \sum_{a=0}^{ns-1} z_a \Gamma_{ay}^{(he)} \quad (\text{C.139})$$

$$\Gamma_{ay}^{(he)} = \left(\frac{5}{3} V_a^{(AN)} + V_{ay}^{(E)} \right) n_a - \frac{5}{3} \left[D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} + \tilde{D}_{p,ay} \frac{1}{h_y} \frac{\partial p_a}{\partial y} \right] \quad (\text{C.140})$$

$$V_{ey}^{(str)} = 0$$

$$\hat{E}_y = \frac{1}{e n_e h_y} \frac{\partial n_e T_e}{\partial y} - \frac{1}{h_y} \frac{\partial \Phi}{\partial y} \quad (\text{C.141})$$

$$\kappa_{ey} = \kappa_e^{(AN)} + \kappa_{ey}^{(CL)} + \kappa_e^{(ST)} \quad (\text{C.142})$$

$$\kappa_{ey}^{(CL)} = 0$$

$$\kappa_e^{(ST)} = \begin{cases} 1.6\alpha^{(ST)}\sigma^{(ST)}T_e/e^2, & y \in [-\Delta^{(ST)}, 0], \\ 0, & y \notin [-\Delta^{(ST)}, 0] \end{cases}, \quad \alpha^{(ST)} \approx 2 \quad (\text{C.143})$$

$$\alpha_{ey} = \alpha_y^{(AN)} + \alpha_y^{(CL)}$$

$$\alpha_y^{(AN)} = \left(c_{\alpha,0} + c_{\alpha,1} \frac{10^{20}}{n_e} + c_{\alpha,3} \frac{T_e}{16e|B|} \right) n_e \sqrt{\frac{e}{T_e}} \quad (\text{C.144})$$

$$c_{\alpha,0} = 10^{-3}, \quad c_{\alpha,1} = 0, \quad c_{\alpha,3} = 0$$

$$\alpha_y^{(CL)} = 0$$

C.5 Ion energy equations

$$\begin{aligned} \frac{3}{2} \frac{\partial n_i T_i}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right) + \sum_{a=0}^{ns-1} \frac{n_a T_i}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} V_{a\parallel} b_x \right) = \\ = Q_{\Delta} + Q_{F_{ab}} + c_{E \times B} T_i B_z \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a + \\ + \sum_{fluid\ species} \left(\eta_{ax} \left(\frac{\partial V_{a\parallel}}{h_x \partial x} \right)^2 + \eta_a^{(AN)} \left(\frac{\partial V_{a\parallel}}{h_y \partial y} \right)^2 \right) + Q_I^{(i)} + Q_R^{(i)} + Q_i^{\text{EIRENE}}, \quad (\text{C.145}) \end{aligned}$$

The 'sum over fluid species' above means the species index a ranges from 0 to $ns - 1$ for **B2.5** standalone runs and only over charged species for coupled **SOLPS-ITER** runs.

C.5.1 Ion energy sources

$$\begin{aligned} S_i^{heat} = Q_{\Delta} + Q_{F_{ab}} + c_{E \times B} T_i B_z \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a + \\ + \sum_{fluid\ species} \left(\eta_{ax} \left(\frac{\partial V_{a\parallel}}{h_x \partial x} \right)^2 + \eta_a^{(AN)} \left(\frac{\partial V_{a\parallel}}{h_y \partial y} \right)^2 \right) + Q_I^{(i)} + Q_R^{(i)} - \sum_{a=0}^{ns-1} \frac{n_a T_i}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} V_{a\parallel} b_x \right) \quad (\text{C.146}) \end{aligned}$$

When Eq. (C.56) is used ('b2sigp_style' set to '2'), the contribution from friction between charged species is given as:

$$Q_{F_{ab}} = - \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} V_{a\parallel} (S_{fr,a}^m + S_{Therm,a}^m) \quad (\text{C.147})$$

Otherwise, it is:

$$Q_{F_{ab}} = \sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} \sum_{\substack{a=b+1 \\ z_a \neq 0}}^{ns-1} \frac{1}{4\tau_p} m_p \sqrt{\frac{\frac{m_a}{m_p} \frac{m_b}{m_p}}{\frac{m_a}{m_p} + \frac{m_b}{m_p}}} z_a^2 z_b^2 n_a n_b (V_{a\parallel} - V_{b\parallel})^2 \quad (C.148)$$

Atom heat source due to ionization

$$Q_I^{(i)} = \sum_{a=0}^{ns-2} Q_{I_a}^{(i)} \quad (C.149)$$

$$Q_{I_a}^{(i)} = \begin{cases} r_{I_a} n_e n_a \left(\frac{m_a + m_{a+1}}{4} \right) (V_{\parallel a} - V_{\parallel a+1})^2, & z_a < z_{a+1} \text{ \& } z_n = z_{n+1} \\ 0 & \end{cases} \quad (C.150)$$

Atom heat source due to recombination

$$Q_R^{(i)} = \sum_{a=1}^{ns-1} Q_{R_a}^{(i)} \quad (C.151)$$

$$Q_{R_a}^{(i)} = \begin{cases} r_{R_a} n_e n_a \left(\frac{m_a + m_{a-1}}{4} \right) (V_{\parallel a} - V_{\parallel a-1})^2, & z_{a-1} < z_a \text{ \& } z_{n-1} = z_n \\ 0 & \end{cases} \quad (C.152)$$

$\tau_p = \frac{3}{4\sqrt{2\pi}} \frac{\sqrt{m_p T_i^{\frac{3}{2}}}}{\ln \Lambda} \left(\frac{4\pi\epsilon_0}{e^2} \right)^2$, where the Coulomb logarithm $\ln \Lambda$ can be chosen as

$$\ln \Lambda = \begin{cases} \max\{5; 23.4 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 3.45 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} \leq 50 \\ \max\{5; 25.3 - 1.15 \cdot \log_{10} \left(\frac{n_e}{10^6} \right) + 2.30 \cdot \log_{10} \left(\frac{T_e}{e} \right)\}, & \frac{T_e}{e} > 50 \end{cases} \quad \begin{array}{l} \text{Braginsky formula} \\ \text{Wesson formula for ion-ion collisions} \end{array}$$

C.5.2 Poloidal ion energy fluxes

$$\begin{aligned} \tilde{q}_{ix} &= \tilde{f}_{ix} T_i - \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x} - q_{ix}^{P.Sch} = \\ &= \frac{3}{2} \Gamma_{ix} T_i + V_{ix}^{(str)} n_i T_i + \frac{3}{2} \left(\frac{j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)}}{e} \right) T_i - \\ &\quad - \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x} - q_{ix}^{P.Sch} \end{aligned} \quad (C.153)$$

$$\tilde{f}_{ix} = \frac{3}{2} \Gamma_{ix} + V_{ix}^{(str)} n_i + \frac{3}{2} \left(\frac{j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)}}{e} \right)$$

$$q_{ix}^{P.Sch} = \frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} D_{axy} \frac{1}{h_y} \frac{\partial n_a T_i^2}{\partial y} \quad (C.154)$$

$$q_{ix} = \left[\frac{3}{2} \Gamma_{ix} + \frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a \tilde{V}_{ax}^{(dia)} + V_{ix}^{(str)} n_i + \frac{3}{2} \left(\frac{j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis||)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)}}{e} \right) \right] T_i - \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x} \quad (C.155)$$

$$\Gamma_{ix} = \sum_{a=0}^{ns-1} \Gamma_{ax}^{(he)} \quad (C.156)$$

$$q_{ix}^{(Eir)} = \frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \Gamma_{ax}^{(Eir)} T_i - \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x} \quad (C.157)$$

C.5.3 Poloidal ion energy transport coefficients

$$\kappa_{ix} = \kappa_{ix}^{(AN)} + \tilde{\kappa}_{ix}^{(CL)} \quad (C.158)$$

$$\kappa_{ix}^{(AN)} = \sum_{a=0}^{ns-1} \tilde{\kappa}_{ax}^{(AN)} \quad (C.159)$$

$$\tilde{\kappa}_{ax}^{(AN)} = \begin{cases} \kappa_a^{(AN)}, & z_a \neq 0 \\ \frac{\kappa_a^{(AN)}}{\left(1 + \left(\frac{\frac{\sqrt{g}}{h_x} \kappa_a^{(AN)} \left| \frac{\partial T_i}{h_x \partial x} \right|}{\frac{\sqrt{g}}{h_x} \alpha V_{h_a} n_a T_i} \right)^{\gamma_2} \right)^{\frac{1}{\gamma_2}}}, & z_a = 0 \end{cases} \quad (C.160)$$

$$V_{h_a} = \sqrt{\frac{2T_i}{m_a}} \quad (C.161)$$

$$\kappa_a^{(AN)} = \chi_a^{(AN)} n_a \quad (C.162)$$

$$\chi_a^{(AN)} = \left(c_{h_a,0} + c_{h_a,1} \frac{10^{20}}{n_e} + c_{h_a,2} D_{f,a} + c_{h_a,3} \frac{T_e}{16e|B|} \right) f_{h_a}(x, y) \quad (C.163)$$

$$a = 0, \quad c_{h_a,0} = 0, \quad c_{h_a,1} = 0, \quad c_{h_a,2} = 1.4, \quad c_{h_a,3} = 0$$

$$a = 1, \quad c_{h_a,0} = 0, \quad c_{h_a,1} = 0, \quad c_{h_a,2} = 1.2, \quad c_{h_a,3} = 0$$

$$f_{h_a}(x, y) = \begin{cases} 1, & z_a = 0 \\ \alpha_{1,h_a}, & -\delta_{1,h_a} \leq y \leq 0, \quad x \in \text{Core}, z_a \neq 0 \\ \alpha_{2,h_a}, & 0 \leq y \leq \delta_{2,h_a}, \quad x \in \text{Core} \\ \alpha_{3,h_a}, & x, y \in \text{PFR} \cup \text{SOL} \end{cases}$$

$$\tilde{\kappa}_{ix}^{(CL)} = c_{\kappa_{ix}}^{(F \text{ lim})} \tilde{\kappa}_{ix}^{(CL)} \quad (C.164)$$

$$c_{\kappa_{ix}}^{F \text{ lim}} = \begin{cases} 1, & \text{for inner flux surfaces} \\ \frac{1}{1 + \frac{\frac{\sqrt{g}}{h_x} \tilde{\kappa}_{ix}^{(CL)} \left| \frac{\partial T_i}{h_x \partial x} \right|}{V_{h_i}^{(\max)} n_i T_i}}, & \text{for other flux surfaces} \end{cases} \quad (C.165)$$

$$V_{h_i}^{(\max)} = c_{flim1} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{T_i}{m_p} \frac{\sum_{a=0}^{ns-1} n_a \sqrt{\frac{m_p}{m_a}}}{n_i}} \quad (C.166)$$

$c_{flim1} = 0.15$ is the ion heat flux limiter coefficient.

$$\tilde{\kappa}_{ix}^{(CL)} = c_{\kappa_{ix}^{(CL)}}^{(Luciani)} \kappa_{ix}^{(CL)} \quad (C.167)$$

If 'b2trcl_luciani' is set to '1', one gets:

$$c_{\kappa_{ix}^{(CL)}}^{(Luciani)} = \begin{cases} \frac{1}{1 + \frac{7.5 \cdot 10^{16} \left(\frac{T_i}{e}\right)^2}{c(y) \sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{only for inner flux surfaces} \\ 1, & \text{for other flux surfaces} \end{cases} \quad (C.168)$$

If 'b2trcl_luciani' is set to '2', the **SOLPS 4.x** formulation is used:

$$c_{\kappa_{ix}^{(CL)}}^{(Luciani)} = \begin{cases} \frac{1}{1 + \frac{0.3 \cdot 10^{16} \left(\frac{T_i}{e}\right)^2}{\sum_{a=0}^{ns-1} z_a^2 n_a}}, & \text{only for inner flux surfaces} \\ 1, & \text{for other flux surfaces} \end{cases} \quad (C.169)$$

If the plateau and banana transport regimes correction is applied (default, 'b2trcl_luciani' set to '3'), one has:

$$c_{\kappa_{ix}^{(CL)}}^{(Luciani)} = \begin{cases} 1.6\varepsilon^{\frac{3}{2}}/K_2, & \text{only for inner flux surfaces} \\ 1, & \text{for other flux surfaces} \end{cases} \quad (C.170)$$

$$K_2 = \frac{0.66 + 1.88 \cdot \sqrt{\varepsilon} - 1.54 \cdot \varepsilon}{1 + 1.03 \cdot \nu_*^{1/2} + 0.31 \cdot \nu_*} + \frac{1.17 \cdot \varepsilon^3 \cdot \nu_*}{1 + 0.74 \cdot \varepsilon^{3/2} \cdot \nu_*}$$

$$\kappa_{ix}^{(CL)} = \frac{5}{2} b_x^2 \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \frac{T_i}{m_a} n_a f_{k_i} \tau_a, \quad f_{k_i} = 2.253 \quad (C.171)$$

For integrated modeling workflows ('b2trcl_integral_modeling' set to '1'), in the region of the overlap with ASTRA in the inner part of the core:

$$\tilde{\kappa}_{ix}^{(CL)} = \frac{25}{4} \left(n_1 T_i \frac{B_z}{B_x h_y} \right)^2 \frac{h_{y(\text{outer midplane})}}{e} \left[\oint \frac{\sqrt{g}}{B^2} \left(1 - \frac{B^2}{\langle B^2 \rangle} \right)^2 dx \right] \times K_{\text{ASTRA}} \quad (C.172)$$

$$\text{Here } \langle B^2 \rangle = \frac{\oint B^2 \sqrt{g} dx}{\oint \sqrt{g} dx}$$

From ASTRA we get T_i/e and $\oint q_i^{NEO} ds$ as a functions of $R_{(\text{outer midplane})}$ and calculate

$$K_{\text{ASTRA}} = \frac{-1}{\oint q_i^{NEO} ds} \frac{d(T_i/e)}{dR_{(\text{outer midplane})}} \quad (C.173)$$

as a flux surface function in B2.5.

$$V_{ix}^{(str)} = 0$$

C.5.4 Radial ion energy fluxes

$$\tilde{q}_{iy} = \tilde{f}_{iy} T_i - \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y} - q_{iy}^{P.Sch} = \left[\frac{3}{2} \Gamma_{iy} + V_{iy}^{(str)} n_i + \frac{3}{2} \frac{(j_y^{(AN)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + j_y^{(in)})}{e} \right] T_i - \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y} - q_{iy}^{P.Sch} \quad (C.174)$$

$$\tilde{f}_{iy} = \frac{3}{2} \Gamma_{iy} + V_{iy}^{(str)} n_i + \frac{3}{2} \frac{(j_y^{(AN)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + j_y^{(in)})}{e}$$

$$q_{iy}^{P.Sch} = -\frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i^2}{\partial x} \quad (C.175)$$

$$q_{iy} = \left[\frac{3}{2} \Gamma_{iy} + \frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a \tilde{V}_{ay}^{(dia)} + V_{iy}^{(str)} n_i + \frac{3}{2} \frac{(j_y^{(AN)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + j_y^{(in)})}{e} \right] T_i - \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y} \quad (C.176)$$

$$\Gamma_{iy} = \sum_{a=0}^{ns-1} \Gamma_{ay}^{(he)}$$

$$q_{iy}^{(Eir)} = \frac{5}{2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \Gamma_{ay}^{(Eir)} T_i - \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y} \quad (C.177)$$

$$V_{iy}^{(str)} = 0$$

C.5.5 Radial ion energy transport coefficients

$$\kappa_{iy} = \kappa_{iy}^{(AN)} + \kappa_{iy}^{(CL)} \quad (C.178)$$

$$\kappa_{iy}^{(AN)} = \sum_{a=0}^{ns-1} \tilde{\kappa}_{ay}^{(AN)} \quad (C.179)$$

$$\tilde{\kappa}_{ay}^{(AN)} = \begin{cases} \kappa_a^{(AN)}, & z_a \neq 0 \\ \frac{\kappa_a^{(AN)}}{\left(1 + \left(\frac{\sqrt{g}}{h_y} \kappa_a^{(AN)} \left| \frac{\partial T_i}{h_y \partial y} \right| \right)^{\gamma_2}\right)^{\frac{1}{\gamma_2}}}, & z_a = 0 \end{cases} \quad (C.180)$$

$$\kappa_{iy}^{(CL)} = 0$$

C.6 Current continuity equation

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} j_x \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} j_y \right) = 0 \quad (C.181)$$

C.6.1 Poloidal currents

$$j_x = j_x^{(AN)} + \tilde{j}_x^{(dia)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + j_x^{(vis\perp)} + \tilde{j}_x^{(visq)} + \tilde{j}_x^{(s)} + j_x^{(\parallel)} \quad (C.182)$$

$$j_x^{(AN)} = -\sigma^{(AN)} \frac{\partial \Phi}{h_x \partial x} \quad (C.183)$$

$$\tilde{j}_x^{(dia)} = B_z \frac{\sum_{a=0}^{ns-1} n_a (z_a T_e + T_i)}{h_y} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \quad (C.184)$$

$$\tilde{j}_x^{(in)} = \frac{B_z}{2} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a V_{a\parallel}^2 \quad (C.185)$$

$$\tilde{j}_x^{(vis\parallel)} = -\frac{b_z B_x}{3B^{\frac{1}{2}}} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \tilde{\eta}_{ax}^{(CL)} \frac{\partial (\sqrt{B} V_{a\parallel})}{h_x \partial x} \quad (C.186)$$

$$j_x^{(vis\perp)} = 0$$

$$\tilde{j}_x^{(visq)} = -0.24 \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \alpha^{NEO} \tilde{\tau}_a b_z \frac{B_x}{B^{\frac{1}{2}}} \frac{\partial}{h_y \partial y} \left(\frac{1}{B^2} \right) \frac{\partial (\tilde{q}_a^{(0)} B^{\frac{1}{2}})}{h_x \partial x} \quad (C.187)$$

For definition of $\tilde{q}_a^{(0)}$, see Eqs. (C.96) and (C.97).

$$\tilde{j}_x^{(s)} = \sum_{\substack{a_0=0 \\ z_{a_0}=0 \ \& \ z_{n,a_0}=1}}^{ns-1} \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \left(-\sigma_{aa_0} b_z^2 \frac{\partial \Phi}{h_x \partial x} - \sigma_{aa_0} b_z^2 \frac{1}{z_a e n_a} \frac{\partial n_a T_i}{h_x \partial x} + \sigma_{aa_0} B_z V_{a_0 y} \right) \quad (C.188)$$

$$\sigma_{aa_0} = \begin{cases} \frac{m_a n_a \langle V_{aa_0} \sigma_{ex} \rangle n_{a_0}}{2B^2}, & z_{a_0} = 0 \ \& \ z_{n,a_0} = 1, z_a \neq 0 \ \& \ z_{n,a} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (C.189)$$

$$\langle V_{aa_0} \sigma_{ex} \rangle = 3.2 \cdot 10^{-15} \sqrt{\frac{T_i}{0.026e}} \quad (C.190)$$

$$V_{a_0 y} = \frac{\Gamma_{a_0 y}}{n_{a_0}}$$

$j_x^{(\parallel)}$ - see above

C.6.2 Radial currents

$$j_y = j_y^{(AN)} + \tilde{j}_y^{(dia)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + \tilde{j}_y^{(s)} + j_y^{(ST)} \quad (C.191)$$

$$j_y^{(AN)} = -\sigma^{(AN)} \frac{\partial \Phi}{h_y \partial y} \quad (C.192)$$

$$\tilde{j}_y^{(dia)} = -B_z \frac{\sum_{a=0}^{ns-1} n_a (z_a T_e + T_i)}{h_x} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) \quad (C.193)$$

$$j_y^{(in)} = -\frac{B_z}{2} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a V_{a\parallel}^2 \quad (C.194)$$

$$\tilde{j}_y^{(vis\parallel)} = b_z \frac{B_x}{3B^{\frac{1}{2}}} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \tilde{\eta}_{ax}^{(CL)} \frac{\partial (\sqrt{B} V_{a\parallel})}{h_x \partial x} \quad (C.195)$$

not agreed with parallel momentum balance?

$$\tilde{j}_y^{(vis\perp)} = - \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \frac{1}{B\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g} \eta_a^{(AN)}}{h_y} \frac{\partial V_{a\perp}}{h_y \partial y} \right) \quad (C.196)$$

$$V_{a\perp} = -\frac{1}{B h_y} \frac{\partial \Phi}{\partial y} - \frac{1}{z_a e B h_y n_a} \frac{\partial n_a T_i}{\partial y} \quad (C.197)$$

$$\tilde{j}_y^{(visq)} = 0.24 \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \alpha^{NEO} \tilde{\tau}_a b_z \frac{B_x}{B^{\frac{1}{2}}} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \frac{\partial (\tilde{q}_a^{(0)} B^{\frac{1}{2}})}{h_x \partial x} \quad (C.198)$$

For definition of $\tilde{q}_a^{(0)}$, see Eqs. (C.96) and (C.97).

$$\tilde{j}_y^{(s)} = \sum_{\substack{a_0=0 \\ z_{a_0}=0 \ \& \ z_{n,a_0}=1}}^{ns-1} \sum_{\substack{a=0 \\ z_a=1 \ \& \ z_{n,a}=1}}^{ns-1} \left(-\sigma_{aa_0} \frac{\partial \Phi}{h_y \partial y} - \sigma_{aa_0} \frac{1}{z_a e n_a} \frac{\partial n_a T_i}{h_y \partial y} - \sigma_{aa_0} B_z V_{a_0 x} \right) \quad (C.199)$$

$$j_y^{(ST)} = \begin{cases} -\sigma^{(ST)} \left(\frac{1}{h_y} \frac{\partial \Phi}{\partial y} - \frac{T_e}{e n_e} \frac{1}{h_y} \frac{\partial n_e}{\partial y} - \frac{0.5}{e} \frac{1}{h_y} \frac{\partial T_e}{\partial y} \right), & y \in [-\Delta^{(ST)}, 0] \\ 0, & y \notin [-\Delta^{(ST)}, 0] \end{cases} \quad (C.200)$$

$$V_{a_0 x} = \frac{\Gamma_{a_0 x}}{n_{a_0}}$$

σ_{aa_0} see above

C.6.3 Conductivities

$$\sigma^{(ST)} = \begin{cases} c_{\sigma^{(ST)}} e n_e, & y \in [-\Delta^{(ST)}, 0], \quad c_{\sigma^{(ST)}} > c_{\sigma_y^{(AN)}} \\ 0, & y \notin [-\Delta^{(ST)}, 0] \end{cases} \quad (C.201)$$

$$\sigma_y = \sigma^{(AN)} + \sigma_y^{(CL)}$$

$\sigma^{(AN)}$ depends on 'b2tqna_model_sig'.

If 'b2tqna_model_sig' is '0', then:

$$\sigma^{(AN)} = \left(c_{\sigma_a,0} + c_{\sigma_a,1} \frac{10^{20}}{n_e} + c_{\sigma_a,3} \frac{T_e}{16e|B|} + c_{\sigma_a,4} e^{-c_{\sigma_a,5} \left(\frac{y}{L(x)} \right)^2} \right) e n_e, \quad (C.202)$$

otherwise:

$$\sigma^{(AN)} = \left(c_{\sigma_a,0} + c_{\sigma_a,1} \frac{10^{20}}{n_e} + c_{\sigma_a,3} \frac{T_e}{16e|B|} + c_{\sigma_a,4} e^{-c_{\sigma_a,5} \left(\frac{y}{L(x)} \right)^2} \right) e \tilde{n}_e \quad (C.203)$$

where $\tilde{n}_e$ is the electron density at the location where the innermost flux surface crosses the outer midplane.

$$L(x) = \int_0^{y_{out}} h_y(x, y) dy$$

$$c_{\sigma_a,0} = 10^{-6} \div 10^{-3}, c_{\sigma_a,1} = 0, c_{\sigma_a,3} = 0$$

$$\sigma_y^{(CL)} = 0$$

C.7 Typical boundary conditions

In this Section, the boundaries are denoted as S_0 for the PFR boundaries, S_1 for the core boundaries, E for the East boundaries (right targets), W for the West boundaries (left targets) and N for the outer main wall boundary.

C.7.1 Particle balance equations

$$S_{n_a} = S_{n_a,0} + S_{n_a,1} n_a$$

South. Private region

If $z_a = 0$

For leakage boundary condition: $BCCON = 10$

$$\left. \frac{\sqrt{g}}{h_y} \Gamma_{ay} \right|_{S_0} = \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a + \sum_{a=0}^{ns-1} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_{S_0} \quad (C.204)$$

$$p_{n_a,a,1} = -10^{-2}, \quad r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1$$

$\tilde{\Gamma}_{ay}$ was replaced by Γ_{ay}

c_{nsr} — neutral sources rescale coefficient, 1 — when working with B2.5 standalone and 10^{-10} when coupling to Eirene

$p_{n_a,a,1}$ is the radial leakage velocity, in units of the local species sound speed. Negative velocities are directed out of the plasma.

If $z_a \neq 0$

For decay length boundary condition: BCCON = 9

$$\left. \frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \right|_{S_0} = \begin{cases} \left[-\frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) n_a \right] \Big|_{S_0}, & \text{mdf\_fnb} = 0 \\ \left[-\frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) n_a + \frac{\sqrt{g}}{h_y} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} + \frac{\sqrt{g}}{h_y} V_{ay}^{(E)} n_a \right] \Big|_{S_0}, & \text{mdf\_fnb} = 1 \end{cases} \quad (\text{C.205})$$

$$S_{n_a,0}^{no.mdf} = 0 \quad (\text{C.206})$$

$$S_{n_a,1}^{no.mdf} = -\frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) \quad (\text{C.207})$$

$p_{n_a,a,1}$ is the density radial decay length of species a , in m .

For leakage boundary conditions: BCCON = 10

$$\left. \frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \right|_{S_0} = \begin{cases} \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a + \min \left\{ 0, \frac{\sqrt{g}}{h_y} n_a V_{ay}^{(E)} \right\} \right] \Big|_{S_0}, & \text{mdf\_fnb} = 0 \\ \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a + \frac{\sqrt{g}}{h_y} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} + \min \left\{ 0, \frac{\sqrt{g}}{h_y} n_a V_{ay}^{(E)} \right\} \right] \Big|_{S_0}, & \text{mdf\_fnb} = 1 \end{cases} \quad (\text{C.208})$$

$p_{n_a,a,1} = -10^{-3}$ is the radial leakage velocity, in units of the species local sound speed. Negative velocities are directed out of the plasma.

$$S_{n_a,0}^{no.mdf} = 0 \quad (\text{C.209})$$

$$S_{n_a,1}^{no.mdf} = p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a \quad (\text{C.210})$$

Imposing radial leakage: BCCON = 15

When using drifts, one should use BCCON = 10 instead.

$$\left. -\frac{\sqrt{g}}{h_y^2} D_{n,ay} \frac{\partial n_a}{\partial y} \right|_{S_0} = \left[c_{n_a,a,2} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a \right] \Big|_{S_0} \quad (\text{C.211})$$

$c_{n_a,a,2} = -10^{-3}$ is the radial leakage velocity, in units of the species local thermal velocity. Negative velocities are directed out of the plasma.

South 1. Core plasma

If $z_a = 0$

Imposing total particle flux: BCCON = 8

$$\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_{S_1} = 0$$

$$\frac{\sqrt{g}}{h_y} \Gamma_{ay} \Big|_{S_1} = 0$$

If $z_a \neq 0$

Imposing a constant density : BCCON = 1

$$\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_{S_1} = \frac{\sqrt{g}}{h_y} \left(v_{big} n_a|_{S_1} - v_{big} n_a \right) \Big|_{S_1}, \quad v_{big} = 10^8 \sqrt{\frac{100 \text{eV}}{m_p}}, \quad n_a|_{S_1} \text{ is the desired density in } m^{-3} \quad (\text{C.212})$$

Imposing the total particle flux with constant flux density: BCCON = 8

$$\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_{S_1} = p_{n,a,0} \frac{\frac{\sqrt{g}}{h_y}}{\sum_i \left(\frac{\sqrt{g}}{h_y} \right)_{i,-1}} \Big|_{S_1} + \frac{\sqrt{g}}{h_y} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} \Big|_{S_1} + \frac{\sqrt{g}}{h_y} V_{ya}^{(E)} n_a \Big|_{S_1}$$

$p_{n,a,0}$ is the desired total particle flux, per unit time.

North

If $z_a = 0$

For leakage boundary condition: BCCON = 10

$$-\frac{\sqrt{g}}{h_y} \Gamma_{ay} \Big|_N = \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a + \sum_{a=0}^{ns-1} r_{rec,a,0} c_{nsr} \max \left\{ 0, \frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_N \quad (\text{C.213})$$

$$p_{n_a,a,1} = -10^{-2}, \quad r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1$$

c_{nsr} — neutral sources rescale coefficient, 1 — when working with B2.5 standalone and 10^{-10} when coupling to Eirene

$p_{n_a,a,1}$ is the radial leakage velocity, in units of the local species sound speed. Negative velocities are directed out of the plasma.

$\tilde{\Gamma}_{ay}$ was replaced by Γ_{ay}

If $z_a \neq 0$

For decay length boundary condition: BCCON = 9

$$-\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_N = \begin{cases} \left[\frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) n_a \right] \Big|_N, & mdf\_fnb = 0 \\ \left[\frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) n_a - \frac{\sqrt{g}}{h_y} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} - \frac{\sqrt{g}}{h_y} V_{ay}^{(E)} n_a \right] \Big|_N, & mdf\_fnb = 1 \end{cases} \quad (\text{C.214})$$

$$S_{n_a,0}^{no.mdf} = 0 \quad (C.215)$$

$$S_{n_a,1}^{no.mdf} = \frac{1}{p_{n_a,a,1}} \frac{\sqrt{g}}{h_y} \left(D_{n,a}^{(AN)} + D_{p,a}^{(AN)} (z_a T_e + T_i) \right) \quad (C.216)$$

$p_{n_a,a,1}$ is the density radial decay length of species a , in m .

For leakage boundary conditions: BCCON = 10

$$-\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_N = \begin{cases} \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a - \max \left\{ 0, \frac{\sqrt{g}}{h_y} n_a V_{ay}^{(E)} \right\} \right] \Big|_N, & mdf\_fnb = 0 \\ \left[p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a - \frac{\sqrt{g}}{h_y} D_{ayx} \frac{1}{h_x} \frac{\partial n_a T_i}{\partial x} - \max \left\{ 0, \frac{\sqrt{g}}{h_y} n_a V_{ay}^{(E)} \right\} \right] \Big|_N, & mdf\_fnb = 1 \end{cases} \quad (C.217)$$

$p_{n_a,a,1} = -10^{-3}$ is the radial leakage velocity, in units of the species local sound speed. Negative velocities are directed out of the plasma.

$$S_{n_a,0}^{no.mdf} = 0 \quad (C.218)$$

$$S_{n_a,1}^{no.mdf} = p_{n_a,a,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a \quad (C.219)$$

Imposing radial leakage: BCCON = 15

When using drifts, one should use BCCON = 10 instead.

$$\frac{\sqrt{g}}{h_y^2} D_{n,ay} \frac{\partial n_a}{\partial y} \Big|_N = \left[c_{n_a,a,2} \frac{\sqrt{g}}{h_y} \sqrt{\frac{z_a T_e + T_i}{m_a}} n_a \right] \Big|_N \quad (C.220)$$

$c_{n_a,a,2} = -10^{-3}$ is the radial leakage velocity, in units of the species local thermal velocity. Negative velocities are directed out of the plasma.

North with puffing

If $z_a = 0$

Imposing a particle flux per unit area: BCCON = 5

$$-\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_N = p_{n,a,0} \frac{\sqrt{g}}{h_y} \Big|_N + \sum_{a=0}^{ns-1} \left[r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_N \quad (C.221)$$

c_{nsr} — neutral sources rescale coefficient, 1 — when working with B2.5 standalone and 10^{-10} when coupling to Eirene

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1$$

$p_{n,a,0}$ is the desired total particle flux density, in $m^{-2}.s^{-1}$.

$\tilde{\Gamma}_{ay}$ was replaced by Γ_{ay}

or

Imposing a total particle flux with constant flux density: BCCON = 8

$$-\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \Big|_N = \frac{p_{n,a,0}}{\sum_{i=0}^{Nx-1} \left(\frac{\sqrt{g}}{h_y} \right)_{i,Ny}} \frac{\sqrt{g}}{h_y} \Big|_N + \sum_{a=0}^{ns-1} \left[r_{rec,a,0} c_{nsr} \max \left\{ 0, \frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_N$$

c_{nsr} — neutral sources rescale coefficient, 1 — when working with B2.5 standalone and 10^{-10} when coupling to Eirene

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1$$

$p_{n,a,0}$ is the desired total particle flux, per unit time.

$\tilde{\Gamma}_{ay}$ was replaced by Γ_{ay}

West

If $z_a = 0$

Imposing a given particle flux per unit area: BCCON = 5

$$\frac{\sqrt{g}}{h_x} \Gamma_{ax} \Big|_W = \left[p_{n,a,1} \frac{\sqrt{g}}{h_x} + \sum_{a=0}^{ns-1} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_x} \Gamma_{ax} \right\} \right] \Big|_W \quad (\text{C.222})$$

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1, \quad p_{n,a,a,1} = 0$$

$$\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax} \Big|_W = \left[p_{n,a,1} \frac{\sqrt{g}}{h_x} + \sum_{a=0}^{ns-1} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_x} \Gamma_{ax} \right\} \right] \Big|_W \quad (\text{C.223})$$

$\tilde{\Gamma}_{ax}$ was replaced by Γ_{ax}

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1, \quad p_{n,a,a,1} = 0$$

If $z_a \neq 0$

Sheath density boundary condition: BCCON = 3

$$\frac{\sqrt{g}}{h_x} \Gamma_{ax} \Big|_W = 10^{20} \frac{\sqrt{g}}{h_x} \frac{\partial n_a}{\partial x} \Big|_W$$

Imposing a sound speed outgoing flux: BCCON = 14

To be used in conjunction with BCENE = 15, BCENI = 15, BCMOM = 13, and BCPOT = 11.

$$\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax} \Big|_W = \left[-p_{n,a,1} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{\sum_{b=0}^{ns-1} p_b}{\sum_{b=0}^{ns-1} m_b n_b}} n_a - \frac{\sqrt{g}}{h_x} D_{axy} \frac{1}{h_y} \frac{\partial n_a T_i}{\partial y} \right] \Big|_W$$

$$p_{n,a,1} = +1$$

East

If $z_a = 0$

Imposing a given particle flux per unit area: BCCON = 5

$$-\frac{\sqrt{g}}{h_x}\Gamma_{ax}\Big|_E = \left[p_{n,a,1}\frac{\sqrt{g}}{h_y} + \sum_{a=0}^{ns-1} r_{rec,a,0}c_{nsr} \max\left\{0, \frac{\sqrt{g}}{h_x}\Gamma_{ax}\right\} \right] \Big|_E \quad (\text{C.224})$$

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1, \quad p_{n,a,1} = 0$$

$$-\frac{\sqrt{g}}{h_x}\tilde{\Gamma}_{ax}\Big|_E = \left[\sum_{a=0}^{ns-1} r_{rec,a,0}c_{nsr} \max\left\{0, \frac{\sqrt{g}}{h_x}\Gamma_{ax}\right\} \right] \Big|_E \quad (\text{C.225})$$

$\tilde{\Gamma}_{ax}$ was replaced by Γ_{ax}

$$r_{rec,0,0} = 0, \quad r_{rec,1,0} = 1$$

If $z_a \neq 0$

Sheath density boundary condition: BCCON = 3

$$-\frac{\sqrt{g}}{h_x}\Gamma_{ax}\Big|_E = -10^{20} \frac{\sqrt{g}}{h_x} \frac{\partial n_a}{\partial x} \Big|_E$$

Imposing a sound speed outgoing flux: BCCON = 14

To be used in conjunction with BCENE = 15, BCENI = 15, BCMOM = 13, and BCPOT = 11.

$$-\frac{\sqrt{g}}{h_x}\tilde{\Gamma}_{ax}\Big|_E = \left[-p_{n,a,1}\frac{\sqrt{g}}{h_x}|b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} n_a + \frac{\sqrt{g}}{h_x} D_{axy} \frac{1}{h_y} \frac{\partial n_a T_i}{\partial y} \right] \Big|_E$$

$$p_{n,a,1} = +1$$

C.7.2 Parallel momentum balance for ions

$$S_{m_a} = S_{m_a,0} + S_{m_a,1}u_{\parallel a} + S_{m_a,2}m_a n_a + S_{m_a,3}m_a n_a u_{\parallel a}$$

South. Private region

If $z_a = 0$

Weakly imposing a velocity: BCMOM = 4

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_0} = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_{S_0}$$

$p_{m,a,1} = 0$ is the imposed velocity, in $m.s^{-1}$, and $p_{m,a,2} = 10^5$ is the strength with which this velocity is imposed.

We had this b.c.

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_0} = \left[c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} + c_{m,a,6} \max \left(0, -h_z \frac{\sqrt{g}}{h_y} \frac{\Gamma_{ay}^{Cor}}{n_a} \right) m_a n_a V_{\parallel a} \right] \Big|_{S_0}, \text{ delete term with max !}$$

$$\Gamma_{ay}^{Cor} = \Gamma_{ay} + \Gamma_{ay}^{(dia)}$$

$$c_{m,a,3} = -10^{10}, \text{ we have } c_{m,a,6} = -1, \text{ but not } c_{m,a,6} = 0$$

$$\text{Inserted into b2stbc\_phys for BCMOM=12 as } c_{m,a,3} \rightarrow p_{m,a,2} = -10^{10}, \text{ and } c_{m,a,6} \rightarrow p_{m,a,1} = -1$$

$$\text{If } z_a \neq 0$$

$$\text{Weakly imposing a velocity: BCMOM} = 4$$

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_0} = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_{S_0} \quad (\text{C.226})$$

$$p_{m,a,1} = 0 \text{ is the imposed velocity, in } m.s^{-1}, \text{ and } p_{m,a,2} = 10^5 \text{ is the strength with which this velocity is imposed.}$$

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_0} = \left[c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} + c_{m,a,6} \max \left(0, -h_z \frac{\sqrt{g}}{h_y} \frac{\Gamma_{ay}^{Cor}}{n_a} \right) m_a n_a V_{\parallel a} \right] \Big|_{S_0} \quad (\text{C.227})$$

$$\text{delete term with max!}$$

$$\Gamma_{ay}^{Cor} = \Gamma_{ay} + \Gamma_{ay}^{(dia)}$$

$$c_{m,a,3} = 0, c_{m,a,6} = -1$$

South 1. Core

$$\text{If } z_a = 0$$

$$\text{Weakly imposing a velocity: BCMOM} = 4$$

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_1} = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_{S_1}$$

$$p_{m,a,1} = 0 \text{ is the imposed velocity, in } m.s^{-1}, \text{ and } p_{m,a,2} = 10^5 \text{ is the strength with which this velocity is imposed.}$$

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_1} = c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} \Big|_{S_1}$$

$$c_{m,a,3} = -10^{10}$$

$$\text{Inserted into b2stbc\_phys for BCMOM=12 as } c_{m,a,3} \rightarrow p_{m,a,2} = -10^{10}, c_{m,a,6} \rightarrow p_{m,a,1} = 0$$

$$\text{If } z_a \neq 0$$

$$\text{Weakly imposing a velocity: BCMOM} = 4$$

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_1} = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_{S_1}$$

$$p_{m,a,1} = 0 \text{ is the imposed velocity, in } m.s^{-1}, \text{ and } p_{m,a,2} = 10^5 \text{ is the strength with which this velocity is imposed.}$$

$$\text{b2stbc\_spb style: BCMOM} = 14$$

$$\begin{aligned} h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_1} &= c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} \Big|_{S_1} + \left[h_z \frac{\sqrt{g}}{h_y} m_a \langle n_a \rangle (-2) \frac{\langle T_i \rangle B_z}{e z_a} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \langle V_{\parallel a} \rangle c_{drift} + c_{corflux} h_z \frac{\sqrt{g}}{h_y} \right] \Big|_{S_1} \\ &\quad + c_{m,a,1} \max \left(0, -h_z \frac{\sqrt{g}}{h_y} \frac{\Gamma_{ay}^{Cor}}{n_a} \right) m_a n_a V_{\parallel a} \Big|_{S_1} \end{aligned}$$

$$c_{m,a,1} = 0, c_{m,a,3} = -10^{10}$$

Prescribe the value of the parallel velocity as a function of the poloidal coordinate: BCMOM = 15

$$h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_{S_1} = v_{big} n_{e0} m_a h_z \frac{\sqrt{g}}{h_y} \left[\frac{\langle B \rangle}{B} p_{m,a,1} - V_{\parallel a} \right] \Big|_{S_1}$$

where $\langle B \rangle = \frac{\oint_{S_1} B \sqrt{g} dx}{\oint_{S_1} \sqrt{g} dx}$, the integration is done over the innermost magnetic surface, $v_{big} = 10^8 \sqrt{\frac{100\text{eV}}{m_p}}$, $n_{e0} = 10^{20}$, $p_{m,a,1}$ is the given value of the average parallel velocity, in $m.s^{-1}$.

North

If $z_a = 0$

Weakly imposing a velocity: BCMOM = 4

$$-h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_N = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_N$$

$p_{m,a,1} = 0$ is the imposed velocity, in $m.s^{-1}$, and $p_{m,a,2} = 10^5$ is the strength with which this velocity is imposed.

$$-h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_N = \left[c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} + c_{m,a,6} \max \left(0, h_z \frac{\sqrt{g}}{h_y} \frac{\Gamma_{ay}^{Cor}}{n_a} \right) m_a n_a V_{\parallel a} \right] \Big|_N$$

delete term with max !

$c_{m,a,3} = -10^{10}$, we have $c_{m,a,6} = -1$, but not $c_{m,a,6} = 0$

Inserted into b2stbc_phys for BCMOM=12 as $c_{m,a,3} \rightarrow p_{m,a,2} = -10^{10}$, $c_{m,a,6} \rightarrow p_{m,a,1} = -1$

If $z_a \neq 0$

Weakly imposing a velocity: BCMOM = 4

$$-h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_N = \left[p_{m,a,1} p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} - p_{m,a,2} h_z \frac{\sqrt{g}}{h_y} V_{\parallel a} \right] \Big|_N$$

$p_{m,a,1} = 0$ is the imposed velocity, in $m.s^{-1}$, and $p_{m,a,2} = 10^5$ is the strength with which this velocity is imposed.

$$-h_z \frac{\sqrt{g}}{h_y} \Gamma_{ay}^m \Big|_N = \left[c_{m,a,3} h_z \frac{\sqrt{g}}{h_y} m_a n_a V_{\parallel a} + c_{m,a,6} \max \left(0, h_z \frac{\sqrt{g}}{h_y} \frac{\Gamma_{ay}^{Cor}}{n_a} \right) m_a n_a V_{\parallel a} \right] \Big|_N$$

$c_{m,a,3} = 0$, $c_{m,a,6} = -1$

West

If $z_a = 0$, $p_{m,a,1} = 0$

If $z_a \neq 0$, $p_{m,a,1} = 1$

Imposing the b2stbc_spb style boundary condition on the velocity: BCMOM = 13

To be used in conjunction with BCENE = 15, BCENI = 15, BCCON = 14, and BCPOT = 11.

$$h_z \frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_W = \left[p_{m,a,1} \frac{u_{big}}{\epsilon^2} V_{West} m_a n_a - \frac{u_{big}}{\epsilon^2} h_z \frac{\sqrt{g}}{h_x} |b_x| m_a n_a V_{\parallel a} \right] \Big|_W$$

with $u_{big} = 10^{10} \sqrt{\frac{100\text{eV}}{m_p}}$ and $\epsilon = 10^{-10}$.

$$V_{West} = -h_z \frac{\sqrt{g}}{h_x} \frac{|b_x|}{b_x} V_{ax}^{(E \times B)} - h_z \frac{\sqrt{g}}{h_x} b_x \sqrt{\frac{\sum_{a=0}^{ns-1} p_a}{\sum_{a=0}^{ns-1} m_a n_a}} \quad \text{for } z_a \neq 0$$

If $z_a = 0$

Imposing a specified parallel velocity: BCMOM = 1

$$\frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_W = \left[u_{big} n_{e00} \frac{m_a}{m_p} \frac{\sqrt{g}}{h_x} p_{m,a,1} - u_{big} n_{e00} \frac{m_a}{m_p} \frac{\sqrt{g}}{h_x} V_{\parallel a} \right] \Big|_W$$

$p_{m,a,1} = 0$ is the imposed velocity in $m.s^{-1}$, $n_{e00} = 10^{20}$.

If $z_a \neq 0$

Imposing a sheath velocity boundary condition: BCMOM = 3

If $p_{m,a,2} < 0.5$:

$$\frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_W = \left[-\frac{u_{big}}{\epsilon^2} n_a m_a \frac{\sqrt{g}}{h_x} u_{p_{out}} - \frac{u_{big}}{\epsilon^2} n_a m_a \frac{\sqrt{g}}{h_x} V_{\parallel a} \right] \Big|_W$$

$p_{m,a,1} = 1, p_{m,a,2} = 0$

$$u_{p_{out}} = \max \left\{ 0, \frac{u_{out}}{b_x} + \frac{u_a^{(dia)}}{b_x} \right\}, u_{u_{out}} = p_{m,a,1} c_s |b_x|, c_s = \sqrt{\frac{\sum_{a=0}^{ns-1} p_a}{\sum_{a=0}^{ns-1} m_a n_a}} \quad \text{for } z_a \neq 0$$

East

If $z_a = 0$, $p_{m,a,1} = 0$

If $z_a \neq 0$

Imposing the b2stbc_spb style boundary condition for parallel velocity: BCMOM = 13

To be used in conjunction with BCENE = 15, BCENI = 15, BCCON = 14, and BCPOT = 11.

$p_{m,a,1} = 1$

$$-h_z \frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_E = \left[p_{m,a,1} \frac{u_{big}}{\epsilon^2} V_{East} m_a n_a - \frac{u_{big}}{\epsilon^2} h_z \frac{\sqrt{g}}{h_x} |b_x| m_a n_a V_{\parallel a} \right] \Big|_E$$

$$V_{East} = -h_z \frac{\sqrt{g}}{h_x} \frac{|b_x|}{b_x} V_{ax}^{(E \times B)} + h_z \frac{\sqrt{g}}{h_x} b_x \sqrt{\frac{\sum_{a=0}^{ns-1} p_a}{\sum_{a=0}^{ns-1} m_a n_a}} \quad \text{for } z_a \neq 0$$

If $z_a = 0$

Imposing a specified parallel velocity: BCMOM = 1

$$-\frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_E = \left[u_{big} n_{e00} \frac{m_a}{m_p} \frac{\sqrt{g}}{h_x} p_{m,a,1} - u_{big} n_{e00} \frac{m_a}{m_p} \frac{\sqrt{g}}{h_x} V_{\parallel a} \right] \Big|_E$$

$p_{m,a,1} = 0$ is the imposed parallel velocity in $m.s^{-1}$.

If $z_a \neq 0$

Imposing a sheath velocity boundary condition: BCMOM = 3

If $p_{m,a,2} < 0.5$:

$$-\frac{\sqrt{g}}{h_x} \Gamma_{ax}^m \Big|_E = \left[-\frac{u_{big}}{\epsilon^2} n_a m_a \frac{\sqrt{g}}{h_x} u_{p_{out}} - \frac{u_{big}}{\epsilon^2} n_a m_a \frac{\sqrt{g}}{h_x} V_{\parallel a} \right] \Big|_E$$

$$p_{m,a,1} = 1, p_{m,a,2} = 0$$

$$u_{p_{out}} = \max \left\{ 0, \frac{u_{u_{out}}}{b_x} - \frac{u_a^{(dia)}}{b_x} \right\}, u_{u_{out}} = p_{m,a,1} c_s |b_x|, c_s = \sqrt{\frac{\sum_{a=0}^{ns-1} p_a}{\sum_{a=0}^{ns-1} m_a n_a}} \Big|_{z_a \neq 0}$$

C.7.3 Current continuity equation

$$S_{ch} = S_{ch,0} + S_{ch,1}\Phi + S_{ch,2}n_e + S_{ch,3}n_e\Phi$$

South. Private region

$$\frac{\sqrt{g}}{h_y} j_y \Big|_{S_0} = 0$$

$$\frac{\sqrt{g}}{h_y} j_y \Big|_{S_0} = \begin{cases} 0, & facdrift = 0 \\ \frac{\sqrt{g}}{h_y} \left(\tilde{j}_y^{(dia)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + \tilde{j}_y^{(s)} \right) \Big|_{S_0}, & facdrift \neq 0 \end{cases}$$

South 1. Core

$$\frac{\sqrt{g}}{h_y} j_y \Big|_{S_1} = \begin{cases} 0, & facdrift = 0 \\ \frac{\sqrt{g}}{h_y} \left(\langle \tilde{j}_y^{(dia)} \rangle \right) \Big|_{S_1}, & facdrift \neq 0 \end{cases}$$

$$\frac{\sqrt{g}}{h_y} j_y \Big|_{S_1} = \begin{cases} 0, & facdrift = 0 \\ \frac{\sqrt{g}}{h_y} \left(\langle \tilde{j}_y^{(dia)} \rangle + \langle j_y^{(in)} \rangle \right) \Big|_{S_1}, & facdrift \neq 0 \end{cases}$$

$$\langle \tilde{j}_y^{(dia)} \rangle = -\frac{B_z \left\langle \sum_{a=0}^{ns-1} n_a (z_a T_e + T_i) \right\rangle}{h_x} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right)$$

$$\langle j_y^{(in)} \rangle = -\frac{B_z}{2} \frac{\partial}{h_x \partial x} \left(\frac{1}{B^2} \right) \sum_{a=0}^{ns-1} m_a n_a \langle V_{a\parallel}^2 \rangle$$

North

$$\frac{\sqrt{g}}{h_y} j_y \Big|_N = \begin{cases} 0, & facdrift = 0 \\ \frac{\sqrt{g}}{h_y} \left(\langle \tilde{j}_y^{(dia)} \rangle \right) \Big|_N, & facdrift \neq 0 \end{cases}$$

$$-\frac{\sqrt{g}}{h_y} j_y \Big|_N = \begin{cases} 0, & facdrift = 0 \\ -\frac{\sqrt{g}}{h_y} \left(\tilde{j}_y^{(dia)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + j_y^{(vis\perp)} + \tilde{j}_y^{(visq)} + \tilde{j}_y^{(s)} \right) \Big|_N, & facdrift \neq 0 \end{cases}$$

West

Imposing a sound speed velocity flux: BCPOT = 11

To be used in conjunction with BCENE = 15, BCENI = 15, BCMOM = 13, and BCCON = 14.

$$\left. \frac{\sqrt{g}}{h_x} j_x \right|_W = \frac{\sqrt{g}}{h_x} e \left[- \sum_{a=0}^{ns-1} z_a \left(p_{n,a,1} |b_x| \sqrt{\frac{\sum_{b=0}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} n_a \right) + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_W$$

if 'b2stbc_istyle_fchi' is '1', or

$$\left. \frac{\sqrt{g}}{h_x} j_x \right|_W = \frac{\sqrt{g}}{h_x} e \left[\sum_{a=0}^{ns-1} z_a \Gamma_{ax} + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_W$$

if 'b2stbc_istyle_fchi' is '0'.

Imposing a sheath potential boundary condition: BCPOT = 3

$$\left. \frac{\sqrt{g}}{h_x} j_x \right|_W = \frac{\sqrt{g}}{h_x} e \left[\sum_{b=0}^{ns-1} z_a \Gamma_{ax} + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_W$$

East

Imposing a sound speed velocity flux: BCPOT = 11

To be used in conjunction with BCENE = 15, BCENI = 15, BCMOM = 13, and BCCON = 11.

$$\left. -\frac{\sqrt{g}}{h_x} j_x \right|_E = \frac{\sqrt{g}}{h_x} e \left[- \sum_{a=0}^{ns-1} z_a \left(p_{n,a,1} |b_x| \sqrt{\frac{\sum_{b=0}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} n_a \right) + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_E$$

if 'b2stbc_istyle_fchi' is '1', or

$$\left. -\frac{\sqrt{g}}{h_x} j_x \right|_E = \frac{\sqrt{g}}{h_x} e \left[- \sum_{a=0}^{ns-1} z_a \Gamma_{ax} + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_E$$

if 'b2stbc_istyle_fchi' is '0'.

Imposing a sheath potential boundary condition: BCPOT = 3

$$\left. -\frac{\sqrt{g}}{h_x} j_x \right|_E = \frac{\sqrt{g}}{h_x} e \left[- \sum_{a=0}^{ns-1} z_a \Gamma_{ax} + (1 - \gamma_e) |b_x| n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(-\frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \right] \Big|_E$$

C.7.4 Energy balance equation for electrons

$$S_{he} = S_{he,0} + S_{he,1} T_e + S_{he,2} n_e + S_{he,3} n_e T_e$$

South. Private region

Imposing the b2stbc_spb style boundary condition: BCENE = 11

$$\left. \frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \right|_{S_0} = \left[\left(c_{he,2} \frac{\sqrt{g}}{h_y} \sqrt{\frac{T_e}{m_e}} \right) T_e n_e + c_{he,7} \max \left(0, -\frac{\sqrt{g}}{h_y} \tilde{f}_{ey} \right) T_e - q_{ex}^{P.Sch} \right] \Big|_{S_0}$$

$$c_{he,2} = -10^{-4}, c_{he,7} = -1, c_{he,2} \rightarrow p_{he,2} = -10^{-4}, c_{he,7} \rightarrow p_{he,1} = -1$$

Imposing a decay length for the electron temperature: BCENE = 22

$$-\frac{\sqrt{g}}{h_y} \kappa_{ey} \frac{\partial T_e}{h_y \partial y} \Big|_{S_0} = \left[\left(p_{he,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{T_e}{m_e}} \right) T_e n_e \right] \Big|_{S_0}$$

Imposing a decay length for the electron temperature: BCENE=9

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \Big|_{S_0} = \begin{cases} \left[\frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,0}^{no.mdf} T_e - \frac{\sqrt{g}}{h_y} \frac{\kappa_e^{(AN)}}{p_{he,1}} T_e + \frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,1}^{no.mdf} n_a T_e \right] \Big|_{S_0}, & mdf\_fhe = 0 \\ \left[\frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,0}^{no.mdf} T_e - \frac{\sqrt{g}}{h_y} \frac{\kappa_e^{(AN)}}{p_{he,1}} T_e + \frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,1}^{no.mdf} n_a T_e - q_{ey}^{P.Sch} + \frac{\sqrt{g}}{h_y} \frac{3}{2} V_{ey}^{(E)} n_e T_e \right] \Big|_{S_0}, & mdf\_fhe = 1 \end{cases}$$

South 1. Core

In b2stbc_spb:

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \Big|_{S_1} = \left[\left(c_{he,0} \frac{\sqrt{g}}{h_y} + c_{he,1} \frac{\sqrt{g}}{h_y} T_e \right) \right] \Big|_{S_1}$$

$$c_{he,0} = 4 \cdot 10^{13}, c_{he,a,1} = -10^{30},$$

Imposing a total electron energy flux: BCENE = 8

$$\frac{\sqrt{g}}{h_y} q_{ey} \Big|_{S_1} = p_{he,1} \frac{\frac{\sqrt{g}}{h_y}}{\sum_{core} \left(\frac{\sqrt{g}}{h_y} \right)} \Big|_{S_1}$$

$p_{he,1}$ is the imposed total electron energy flux, in W .

North

Imposing the b2stbc_spb style boundary condition: BCENE = 11

$$-\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \Big|_N = \left[\left(p_{he,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{T_e}{m_e}} \right) T_e n_e + p_{he,2} \max \left(0, \frac{\sqrt{g}}{h_y} \tilde{f}_{ey} \right) T_e - q_{ex}^{P.Sch} \right] \Big|_N$$

$$p_{he,1} = -10^{-4}, p_{he,2} = -1$$

Imposing a decay length for electron temperature: BCENE = 22

$$\frac{\sqrt{g}}{h_y} \kappa_{ey} \frac{\partial T_e}{h_y \partial y} \Big|_N = \left[\left(p_{he,1} \frac{\sqrt{g}}{h_y} \sqrt{\frac{T_e}{m_e}} \right) T_e n_e \right] \Big|_N$$

Imposing a decay length for electron temperature: BCENE = 9

$p_{he,1}$ is the radial electron temperature decay length, in m .

$$-\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \Big|_N = \begin{cases} \left[\frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,0}^{no.mdf} T_e - \frac{\sqrt{g}}{h_y} \frac{\kappa_e^{(AN)}}{p_{he,1}} T_e + \frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,1}^{no.mdf} n_a T_e \right] \Big|_N, & mdf\_fhe = 0 \\ \left[\frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,0}^{no.mdf} T_e - \frac{\sqrt{g}}{h_y} \frac{\kappa_e^{(AN)}}{p_{he,1}} T_e + \frac{5}{2} \sum_{a=0}^{ns-1} z_a S_{n_a,1}^{no.mdf} n_a T_e + q_{ey}^{P.Sch} - \frac{\sqrt{g}}{h_y} \frac{3}{2} V_{ey}^{(E)} n_e T_e \right] \Big|_N, & mdf\_fhe = 1 \end{cases}$$

West

Imposing the b2stbc_spb style boundary condition for electron temperature: BCENE = 15

To be used in conjunction with BCENI = 15, BCCON = 14, BCMOM = 13, and BCPOT = 11.

If 'bcene.15_style' is set to '0'

$$\left. \frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right|_W = \left[- \left(\frac{1+\gamma_e}{1-\gamma_e} T_e + e\Phi \right) \max \left(0, \frac{\sqrt{g}}{h_x} \sum_{a=0}^{ns-1} z_a p_{n,a,1} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} n_a + \frac{\sqrt{g}}{h_x} \frac{j_{||}}{e} \right) - \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_W$$

If 'bcene.15_style' is set to '1':

$$\left. \frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right|_W = \left[- \frac{\sqrt{g}}{h_x} |b_x| \left((1+\gamma_e) T_e + (1-\gamma_e) e\Phi \right) n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(- \frac{e(\Phi - p_{\Phi,2})}{T_e} \right) - \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_W$$

Imposing the sheath boundary condition for electron temperature: BCENE = 3

If $p_{m,a_{main},2} < 0.5$:

If $p_{h_e,1} > 0$ then

$$\left. \frac{\sqrt{g}}{h_x} q_{ex} \right|_W = \left[\left(p_{h_e,1} + \frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \frac{\sqrt{g}}{h_x} \nu_{e_{out}} T_e - \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_W$$

$p_{h_e,1} = 0.9$, $p_{\Phi,2}$ is the applied electric potential at the target, in V .

$$\nu_{e_{out}} = \min \left\{ 0, -u_{u_{out}} n_e - \frac{j_x}{e} \right\}, \quad u_{u_{out}} = p_{m,a_{main},1} c_s |b_x|, \quad c_s = \sqrt{\frac{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_a}{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a}}$$

East

Imposing the b2stbc_spb style boundary condition for electron temperature: BCENE = 15

To be used in conjunction with BCENI = 15, BCCON = 14, BCMOM = 13, and BCPOT = 11.

If 'bcene.15_style' is set to '0':

$$\left. - \frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right|_E = \left[- \left(\frac{1+\gamma_e}{1-\gamma_e} T_e + e\Phi \right) \max \left(0, \frac{\sqrt{g}}{h_x} \sum_{a=0}^{ns-1} z_a p_{n,a,1} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} n_a - \frac{\sqrt{g}}{h_x} b_x \frac{j_{||}}{e} \right) + \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_E$$

If 'bcene.15_style' is set to '1':

$$\left. - \frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right|_E = \left[- \frac{\sqrt{g}}{h_x} |b_x| \left((1+\gamma_e) T_e + (1-\gamma_e) e\Phi \right) n_e \frac{1}{\sqrt{2\pi}} \sqrt{\frac{T_e}{m_e}} \exp \left(- \frac{e(\Phi - p_{\Phi,2})}{T_e} \right) + \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_E$$

Imposing the sheath boundary condition for electron temperature: BCENE = 3

If $p_{m,a_{main},2} < 0.5$:

If $p_{h_e,1} > 0$ then

$$\left. - \frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right|_E = - \left[\left(p_{h_e,1} + \frac{e(\Phi - p_{\Phi,2})}{T_e} \right) \frac{\sqrt{g}}{h_x} \nu_{e_{out}} T_e + \frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right] \Big|_E$$

$p_{h_e,1} = 0.9$, $p_{\Phi,2}$ is the applied electric potential at the target, in V .

$$\nu_{e_{out}} = \max \left\{ 0, u_{u_{out}} n_e - \frac{j_x}{e} \right\}, \quad u_{u_{out}} = p_{m,a_{main},1} c_s |b_x|, \quad c_s = \sqrt{\frac{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_a}{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a}}$$

C.7.5 Energy balance equation for ions

$$S_{h_i} = S_{h_i,0} + S_{h_i,1}T_i + S_{h_i,2}n_i + S_{h_i,3}n_iT_i$$

South. Private region

Imposing a decay length for the ion temperature: BCENI = 9

$p_{h_i,1}$ is the radial ion temperature decay length, in m .

$e_{rec,0} = 0$ and $e_{rec,1} = 0.3$ are energy recycling coefficients.

$r_{rec,0,0} = 0$ and $r_{rec,1,0} = 1$ are particle recycling coefficients.

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \Big|_{S_0} = \begin{cases} \left[\frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,0}^{no,mdf} T_i - \frac{\sqrt{g}}{h_y} \frac{\kappa_i^{(AN)}}{p_{h_i,1}} T_i + \frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,1}^{no,mdf} n_a T_i + \right. \\ \left. + \sum_{a=0}^{ns-1} \max \left\{ e_{rec,a} \left(\frac{3}{2} + BoRiS \right) (T_i + z_a (\Phi - p_{\Phi,2})) , \left(\frac{3}{2} + BoRiS \right) T_{surf k_B} \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_{S_0}, \text{ } mdf.fhi = 0 \\ \left[\frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,0}^{no,mdf} T_i - \frac{\sqrt{g}}{h_y} \frac{\kappa_i^{(AN)}}{p_{h_i,1}} T_i + \frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,1}^{no,mdf} n_a T_i - q_{iy}^{P.Sch} + \frac{\sqrt{g}}{h_y} \frac{3}{2} \sum_{a=0}^{ns-1} V_{ay}^{(E)} n_a T_i + \right. \\ \left. + \sum_{a=0}^{ns-1} \max \left\{ e_{rec,a} \left(\frac{3}{2} + BoRiS \right) (T_i + z_a (\Phi - p_{\Phi,2})) , \left(\frac{3}{2} + BoRiS \right) T_{surf k_B} \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_{S_0}, \text{ } mdf.fhi = 1 \end{cases}$$

Imposing a constant ion temperature: BCENI = 21

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \Big|_{S_0} = p_{h_i,2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(\frac{\sqrt{g}}{h_y} n_a \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) T_i \Big|_{S_0} + p_{h_i,1} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \tilde{f}_{iy} \right\} T_i \Big|_{S_0}$$

$$p_{h_i,1} = -1, p_{h_i,2} = -10^{-2}$$

Imposing a decay length for the ion temperature: BCENI = 22

$$-\frac{\sqrt{g}}{h_y} \kappa_{iy} \frac{\partial T_i}{h_y \partial y} \Big|_{S_0} = p_{h_i,1} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(\frac{\sqrt{g}}{h_y} n_a \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) T_i \Big|_{S_0}$$

$p_{h_i,1}$ is the imposed ion temperature decay length, in m .

b2stbc_spb boundary condition:

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \Big|_{S_0} = \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(c_{h_i,a,2} \frac{\sqrt{g}}{h_y} \frac{n_a}{n_i} \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) n_i T_i \Big|_{S_0} + c_{h_i,a,7} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \tilde{f}_{iy} \right\} T_i \Big|_{S_0}$$

$$c_{h_i,a,2} = -10^{-2}$$

South 1. Core

Imposing a total ion energy flux: BCENI = 8

$$\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \Big|_{S_1} = \left[p_{h_i,1} \frac{\frac{\sqrt{g}}{h_y}}{\sum_{core} \left(\frac{\sqrt{g}}{h_y} \right)} - q_{iy}^{P.Sch} + \frac{3}{2} \frac{(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + \tilde{j}_y^{(visq)} + j_y^{(vis\perp)})}{e} T_i \right] \Big|_{S_1}$$

$p_{h_i,1}$ is the imposed total ion energy flux, in W .

b2stbc_spb boundary condition:

$$\left. \frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right|_{S_1} = \sum_{a=0}^{ns-1} \left[\left(c_{h_i,a,0} \frac{\sqrt{g}}{h_y} \frac{n_a}{n_i} + c_{h_i,a,1} \frac{\sqrt{g}}{h_y} \frac{n_a}{n_i} T_i \right) \right] \Big|_{S_1}$$

$$c_{h_i,a,0} = 4 \cdot 10^{13}, c_{h_i,a,1} = -10^{30}, a = 0.1.$$

North

Imposing an ion temperature decay length: BCENE = 9

$p_{h_i,1}$ is the radial ion temperature decay length, in m .

$e_{rec,0} = 0$ and $e_{rec,1} = 0.3$ are energy recycling coefficients.

$r_{rec,0,0} = 0$ and $r_{rec,1,0} = 1$ are particle recycling coefficients.

$$-\left. \frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right|_N = \begin{cases} \left[\frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,0}^{no.mdf} T_i - \frac{\sqrt{g}}{h_y} \frac{\kappa_i^{(AN)}}{p_{h_i,1}} T_i + \frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,1}^{no.mdf} n_a T_i + \right. \\ \left. + \sum_{a=0}^{ns-1} \max \left\{ e_{rec,a} \left(\frac{3}{2} + BoRis \right) \left(T_i + z_a \left(\Phi - p_{\Phi,2} \right) \right), \left(\frac{3}{2} + BoRis \right) T_{surf} k_B \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_N, & mdf.fhi = 0 \\ \left[\frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,0}^{no.mdf} T_i - \frac{\sqrt{g}}{h_y} \frac{\kappa_i^{(AN)}}{p_{h_i,1}} T_i + \frac{5}{2} \sum_{a=0}^{ns-1} S_{n_a,1}^{no.mdf} n_a T_i - q_{iy}^{P.Sch} + \frac{\sqrt{g}}{h_y} \frac{3}{2} \sum_{a=0}^{ns-1} V_{ay}^{(E)} n_a T_i + \right. \\ \left. + \sum_{a=0}^{ns-1} \max \left\{ e_{rec,a} \left(\frac{3}{2} + BoRis \right) \left(T_i + z_a \left(\Phi - p_{\Phi,2} \right) \right), \left(\frac{3}{2} + BoRis \right) T_{surf} k_B \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \Gamma_{ay} \right\} \right] \Big|_N, & mdf.fhi = 1 \end{cases}$$

Boundary condition from b2stbc_spb: BCENI = 21

$$-\left. \frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right|_N = p_{h_i,2} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(\frac{\sqrt{g}}{h_y} n_a \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) T_i \Big|_N + p_{h_i,1} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \tilde{f}_{iy} \right\} T_i \Big|_N$$

$$p_{h_i,1} = -1, p_{h_i,2} = 10^{-2}$$

Prescribing a decay length for the ion temperature: BCENI = 22

$$\left. \frac{\sqrt{g}}{h_y} k_{iy} \frac{\partial T_i}{\partial y} \right|_N = p_{h_i,1} \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(\frac{\sqrt{g}}{h_y} n_a \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) T_i \Big|_N$$

$p_{h_i,1}$ is the imposed ion temperature decay length, in m .

Boundary condition in b2stbc_spb:

$$-\left. \frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right|_N = \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left(c_{h_i,a,2} \frac{\sqrt{g}}{h_y} \frac{n_a}{n_i} \sqrt{\frac{z_a T_e + T_i}{m_a}} \right) n_i T_i \Big|_N + c_{h_i,a,7} \max \left\{ 0, -\frac{\sqrt{g}}{h_y} \tilde{f}_{iy} \right\} T_i \Big|_N$$

$$c_{h_i,a,2} = -10^{-2}$$

West

Boundary condition from b2stbc_spb: BCENI = 15

To be used in conjunction with BCENE = 15, BCCON = 14, BCMOM = 13, and BCPOT = 11.

$$\left. \frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right|_W = -p_{h_i,1} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} T_i \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_{n,a,1} n_a - \frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch} \Big|_W$$

$$p_{h_i,1} = +1.5$$

Boundary condition from b2stbc_spb:

$$\left. \frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right|_W = \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left[\left(c_{h_i,a,\tau} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} \frac{n_a}{n_i}} \right) n_i T_i - \frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch} \right] \Big|_W$$

Imposing a sheath ion boundary condition: BCENI = 3

If $p_{m,a_{main},2} < 0.5$:

$$\left. \frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right|_W = \left[\sum_{z_a \neq 0} \left(\frac{\sqrt{g}}{h_x} \frac{p_{h_i,2}}{2} u_{u_{out}} m_a n_a \left(u_{p_{out}}^2 + \left(u_{a,x}^{(dia)} \right)^2 \right) T_i - \frac{\sqrt{g}}{h_x} p_{h_i,1} u_{u_{out}} n_a T_i \right) + \sum_{a=0}^{ns-1} \max \left\{ r_{rec,a,1} \left(\frac{3}{2} + BoRiS \right) (T_i + z_a (\Phi - p_{\Phi,2})), \left(\frac{3}{2} + BoRiS \right) T_{surf} k_B \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, -\frac{\sqrt{g}}{h_x} \Gamma_{ax} \right\} \right] \Big|_W$$

$$u_{a,x}^{(dia)} = V_{ax}^{(E)} + \tilde{V}_{xa}^{(dia)}$$

$$u_{p_{out}} = \max \left\{ 0, \frac{u_{u_{out}}}{b_x} + \frac{u_{1,x}^{(dia)}}{b_x} \right\}$$

$$u_{u_{out}} = p_{m,a_{main},1} c_s |b_x|, c_s = \sqrt{\frac{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_a}{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a}},$$

$$p_{m,a_{main},1} = 1.0$$

If neutrals_namelist.eq.1.or.use_eirene.ne.0 then $e_{rec,0} = 0, e_{rec,1} = 0.3, r_{rec,0,0} = 0, r_{rec,1,0} = 1$

else $r_{rec,0,0} = 0, r_{rec,1,0} = 1, r_{rec,0,1} = 0, r_{rec,1,1} = 0.3$

$p_{h_i,1} = 2.5$ should be $p_{h_i,1} = 1.5$

$p_{h_i,2} = 0$ without drifts

$p_{h_i,2} = 10^{-3}$ with drifts, should be $p_{h_i,2} = 0$

East

Imposing a b2stbc_spb style boundary condition: BCENI = 15

To be used in conjunction with BCENE = 15, BCCON = 14, BCMOM = 13, and BCPOT = 11.

$$\left. -\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right|_E = -p_{h_i,1} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} T_i \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_{n,a,1} n_a + \frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch} \Big|_E$$

$$p_{h_i,1} = +1.5$$

Boundary condition from b2stbc_spb:

$$-\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \Big|_E = \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} \left[- \left(c_{hi,a,7} \frac{\sqrt{g}}{h_x} |b_x| \sqrt{\frac{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} p_b}{\sum_{\substack{b=0 \\ z_b \neq 0}}^{ns-1} m_b n_b}} \frac{n_a}{n_i} \right) n_i T_i + \frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch} \right] \Big|_E$$

Imposing a sheath ion boundary condition: BCENI = 3

If $p_{m,a_{main},2} < 0.5$:

$$-\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \Big|_E = \left[\sum_{z_a \neq 0} \left(-\frac{\sqrt{g}}{h_x} \frac{p_{hi,2}}{2} u_{u_{out}} m_a n_a \left(u_{p_{out}}^2 + \left(u_{a,x}^{(dia)} \right)^2 \right) - \frac{\sqrt{g}}{h_x} p_{hi,1} u_{u_{out}} n_a T_i \right) + \sum_{a=0}^{ns-1} \max \left\{ r_{rec,a,1} \left(\frac{3}{2} + BoRiS \right) (T_i + z_a (\Phi - p_{\Phi,2})), \left(\frac{3}{2} + BoRiS \right) T_{surf k_B} \right\} r_{rec,a,0} c_{nsr} \max \left\{ 0, \frac{\sqrt{g}}{h_x} \Gamma_{ax} \right\} \right] \Big|_E,$$

$$u_{a,x}^{(dia)} = V_{ax}^{(E)} + \tilde{V}_{ax}^{(dia)}$$

$$u_{p_{out}} = \max \left\{ 0, \frac{u_{u_{out}}}{b_x} - \frac{u_{1,x}^{(dia)}}{b_x} \right\}$$

$$u_{u_{out}} = p_{m,a_{main},1} c_s |b_x|, c_s = \sqrt{\frac{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} p_a}{\sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} m_a n_a}}, p_{m,a_{main},1} = 1.0$$

If neutrals_namelist.eq.1.or.use_eirene.ne.0 then $e_{rec,0} = 0, e_{rec,1} = 0.3, r_{rec,0,0} = 0, r_{rec,1,0} = 1$

else $r_{rec,0,0} = 0, r_{rec,1,0} = 1, r_{rec,0,1} = 0, r_{rec,1,1} = 0.3$

$p_{hi,1} = 2.5$, should be $p_{hi,1} = 1.5$

$p_{hi,2} = 0$, without drifts

$p_{hi,2} = 10^{-3}$ with drifts, should be $p_{hi,2} = 0$,

C.8 Total energy balance

The aim of this section is to derive an equation which resembles the total energy balance equation, i.e. has the form

$$\frac{\partial \varepsilon}{\partial t} + (\nabla \cdot \vec{q}^{\text{ENERGY}}) = (\vec{j} \cdot \vec{E}) + V_{e\parallel} R_{e\parallel}^{\text{INELASTIC}} + \sum_a V_{a\parallel} R_{a\parallel}^{\text{INELASTIC}} + Q_e^{\text{ATOMIC}} + Q_i^{\text{ATOMIC}} + Q^{\text{res}} \quad (\text{C.228})$$

from the code equations written in the sections above. Then it is explained how to compute each term of the equation (C.228), including the residual term Q^{res} , making use of the quantites available from the code output activated by the switch `b2wdat_iout = 4` (see Appendix G).

The equation (C.228) may be derived from the code equations, following [46], as

$$\text{Eq. (C.98)} + \text{Eq. (C.145)} + \sum_a V_{a\parallel} \cdot (\text{Eq. (C.56)}) - \sum_a \frac{m_a V_{a\parallel}^2}{2} \cdot (\text{Eq. (C.1)}) \quad (\text{C.229})$$

The term $\{V_{e\parallel} \cdot (\text{electron parallel force balance equation})\}$ does not appear in the sum (C.229), because **SOLPS-ITER** neglects electron inertia, so the corresponding equation does not appear explicitly in the B2.5 equation system and there is no term involving the electron kinetic energy $\frac{m_e u_e^2}{2}$. One should make use of equations (C.124) and (C.125) instead. Similarly, the electron continuity equation (or its equivalent (C.181)) does not enter the sum (C.229) either.

The kinetic energy of perpendicular motion is neglected as well for the same reason — the inertia terms are neglected in the perpendicular force balance equations (even if these equations are implicitly taken into account).

Thus the total energy density is

$$\varepsilon = \sum_a \frac{m_a V_{a\parallel}^2}{2} + \frac{3}{2} \sum_a n_a T_i + \frac{3}{2} n_e T_e. \quad (\text{C.230})$$

The energy flux

$$\vec{q}^{\text{ENERGY}} = \vec{q}_e^{\text{ENERGY}} + \vec{q}_i^{\text{ENERGY}} \quad (\text{C.231})$$

should be associated with the motion of guiding centers in its convective part, BUT it should NOT contain $\vec{\Gamma}_a$, $\vec{q}_e$ and $\vec{q}_i$, since these are modified fluxes where divergence-free components have been analytically removed for numerical stability reasons.

The explicit expressions may be rewritten in components:

$$q_{ex}^{\text{ENERGY}} = \frac{5}{2} \left[\sum_a z_a \Gamma_{ax}^{\text{ENERGY}} - \frac{j_x^{(\parallel)}}{e} - n_e T_e \frac{B_z}{e} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right] T_e - \kappa_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x} - \tilde{c}_e^{(2)} \frac{j_x^{(\parallel)}}{e} T_e \quad (\text{C.232})$$

$$q_{ey}^{\text{ENERGY}} = \frac{5}{2} \left[\sum_a z_a \Gamma_{ay}^{\text{ENERGY}} + n_e T_e \frac{B_z}{e} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) \right] T_e - \kappa_{ey} \frac{1}{h_y} \frac{\partial T_e}{\partial y} \quad (\text{C.233})$$

$$q_{ix}^{\text{ENERGY}} = \sum_a \left(\Gamma_{ax}^{\text{ENERGY}} + n_a \tilde{V}_{ax}^{(dia)} + \frac{j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + \tilde{j}_x^{(vis\perp)} + \tilde{j}_x^{(visq)}}{e} \right) \times \left(\frac{m_a n_a V_{a\parallel}^2}{2} + \frac{5}{2} n_a T_i \right) - \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x} \quad (\text{C.234})$$

$$q_{iy}^{\text{ENERGY}} = \sum_a \left(\Gamma_{ay}^{\text{ENERGY}} + n_a \tilde{V}_{ay}^{(dia)} + \frac{j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + \tilde{j}_y^{(vis\perp)} + \tilde{j}_y^{(visq)}}{e} \right) \times \left(\frac{m_a V_{a\parallel}^2}{2} + \frac{5}{2} T_i \right) - \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y} \quad (\text{C.235})$$

$$\Gamma_{ax}^{\text{ENERGY}} = \left(b_x V_{\parallel a} + V_x^{(E)} + V_{ax}^{(AN)} + V_a^{corr.dpc} \right) n_a - D_{n,a} \frac{1}{h_x} \frac{\partial n_a}{\partial x} - \tilde{D}_{p,ax} \frac{1}{h_x} \frac{\partial p_a}{\partial x} \quad (\text{C.236})$$

$$\Gamma_{ay}^{\text{ENERGY}} = \left(V_y^{(E)} + V_{ay}^{(AN)} \right) n_a - D_{n,a} \frac{1}{h_y} \frac{\partial n_a}{\partial y} - \tilde{D}_{p,ay} \frac{1}{h_y} \frac{\partial p_a}{\partial y} \quad (\text{C.237})$$

$R_{e\parallel}$ and $R_{a\parallel}$ are contributions from inelastic collisions to the parallel projection of friction forces acting on electrons and ions correspondingly. They are

$$R_{e\parallel} = 0; \quad (C.238)$$

$$R_{a\parallel} = S_{I_a}^m + S_{R_a}^m + S_{CX_a}^m + S_{AN_a}^m + S_{EIRENE_a}^m \quad (C.239)$$

Q_e^{ATOMIC} and Q_i^{ATOMIC} are electron and ion energy sources and sinks due to inelastic collisions.

$$Q_e^{\text{ATOMIC}} = Q_R + Q_e^{\text{EIRENE}} \quad (C.240)$$

$$Q_i^{\text{ATOMIC}} = Q_I^{(i)} + Q_R^{(i)} + Q_i^{\text{EIRENE}} \quad (C.241)$$

The residual term Q^{res} consists of residuals coming from the inaccuracies of code equations and from numerical approximation. They can be grouped according to their origin as follows:

$$Q^{\text{res}} = Q_{\text{visc}} + Q_{\text{inert}}^{\text{res}} + \left(\left[\vec{j} - \left(\vec{j} \cdot \vec{b} \right) \vec{b} - \vec{j}_{\text{dia}} \right] \cdot \vec{E} \right) + Q_{j\perp}^{\text{res}} + Q_{\text{corr\_dpc}}^{\text{res}} + Q_{i,nTu\parallel}^{\text{res}} + Q_{e,dia}^{\text{res}} + Q_{i,dia}^{\text{res}} \quad (C.242)$$

The viscosity residual Q_{visc} comes from the fact that the term $\left(\vec{\pi}_a \cdot \vec{V}_a \right)$ does not appear in the code equations and cannot be extracted from the code output, therefore it is excluded from the total energy flux $\vec{q}^{\text{ENERGY}}$. Instead Q_{visc} appears in the right hand side in the form:

$$Q_{\text{visc}} = \sum_a \left[\frac{4}{3} \tilde{\eta}_{ax}^{(CL)} \left(\frac{\partial V_{a\parallel}}{h_x \partial x} \right)^2 + \eta_a^{(AN)} \left[\left(\frac{\partial V_{a\parallel}}{h_x \partial x} \right)^2 + \left(\frac{\partial V_{a\parallel}}{h_y \partial y} \right)^2 \right] \right] - \sum_a \left[V_{a\parallel} \left(\left(\nabla \cdot \vec{\pi}_a^{V\parallel} \right)_{\parallel} + \left(\nabla \cdot \vec{\pi}_a^{\perp} \right)_{\parallel} + \left(\nabla \cdot \vec{\pi}_a^q \right)_{\parallel} \right) \right] \quad (C.243)$$

which is in fact only the approximation to

$$\sum_a \left(\nabla \cdot \left(\vec{\pi}_a \cdot \vec{V}_a \right) \right) - \sum_a \left(\vec{b} \cdot V_{a\parallel} - \left(\nabla \cdot \vec{\pi}_a \right) \right)$$

Here

$$\left(\nabla \cdot \vec{\pi}_a^{V\parallel} \right)_{\parallel} = \frac{4}{3} \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \tilde{\eta}_{ax}^{(CL)} \frac{\partial \ln h_z}{h_x \partial x} V_{\parallel a} - \tilde{\eta}_{ax}^{(CL)} \frac{\partial V_{\parallel a}}{h_x \partial x} \right) - \frac{4}{3} \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \tilde{\eta}_{ax}^{(CL)} \frac{\partial \ln \left(h_z B^{\frac{1}{2}} \right)}{B^{\frac{1}{2}} h_x \partial x} \right) B^{\frac{1}{2}} V_{\parallel a}, \quad z_a = 1 \ \& \ z_{n,a} = 1 \quad (C.244)$$

$$\left(\nabla \cdot \vec{\pi}_a^{\perp} \right)_{\parallel} = - \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \eta_a^{(AN)} \frac{\partial V_{\parallel a}}{h_x \partial x} \right) - \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial y} \left(\frac{h_z \sqrt{g}}{h_y} \eta_a^{(AN)} \frac{\partial V_{\parallel a}}{h_y \partial y} \right) \quad (C.245)$$

The inertia residual $Q_{\text{inert}}^{\text{res}}$ is the difference between what we want and what we have regarding the convective flux of the ion parallel kinetic energy:

$$\begin{aligned}
Q_{\text{inert}}^{\text{res}} = & \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left[\frac{\sqrt{g}}{h_x} \sum_a \left(\Gamma_{ax}^{\text{ENERGY}} + n_a \tilde{V}_{ax}^{(dia)} + \frac{j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + \tilde{j}_x^{(vis\perp)} + \tilde{j}_x^{(visq)}}{e} \right) \frac{m_a V_{a\parallel}^2}{2} \right] + \\
& \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left[\frac{\sqrt{g}}{h_y} \sum_a \left(\Gamma_{ay}^{\text{ENERGY}} + n_a \tilde{V}_{ay}^{(dia)} + \frac{j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + \tilde{j}_y^{(vis\perp)} + \tilde{j}_y^{(visq)}}{e} \right) \frac{m_a V_{a\parallel}^2}{2} \right] - \\
& \sum_a V_{a\parallel} \left[\frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial x} \left(\frac{h_z \sqrt{g}}{h_x} \Gamma_{ax}^{Cor} \right) + \frac{1}{h_z \sqrt{g}} \frac{\partial}{\partial y} \left(\frac{h_z \sqrt{g}}{h_y} \Gamma_{ay}^{Cor} \right) - S_{CF_a}^m \right] + \\
& \sum_a \frac{m_a V_{a\parallel}^2}{2} \left[\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay} \right) \right] \quad (\text{C.246})
\end{aligned}$$

This residual comes not only from numerical approximations, but also from the inaccuracies in the definition of $(\vec{b} \cdot (\nabla \cdot m_a n_a \vec{V}_a \vec{V}_a))$.

The Joule heating term $(\vec{j} \cdot \vec{E})$ requires a separate discussion.

It may be noted that, analytically:

$$\begin{aligned}
j_{\parallel} E_{\parallel} = & \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n_e T_e b_x V_{e\parallel} \right) - \frac{n_e T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} b_x V_{e\parallel} \right) + V_{e\parallel} \sum_a (S_{fr,ea}^m + S_{Therm,ea}^m) - \\
& \frac{b_x}{h_x} \frac{\partial \Phi}{\partial x} \sum_a Z_a e n_a V_{a\parallel} \quad (\text{C.247})
\end{aligned}$$

and

$$(\vec{j}_{\text{dia}} \cdot \vec{E}) = (\nabla(n_e T_e + n_i T_i) \cdot \vec{V}^{(E)}) \quad (\text{C.248})$$

No other currents contribute to the total energy balance Eq. (C.228), which is why the term $([\vec{j} - (\vec{j} \cdot \vec{b}) \vec{b} - \vec{j}_{\text{dia}}] \cdot \vec{E})$ appears in the expression for the residual. Practically, it is only possible to compute the right hand sides of Eqs. (C.247) and (C.248), since terms like $j_{\parallel} E_{\parallel}$, $(\vec{j}_{\text{dia}} \cdot \vec{E})$ or $([\vec{j} - (\vec{j} \cdot \vec{b}) \vec{b} - \vec{j}_{\text{dia}}] \cdot \vec{E})$ are not available in the code.

The term

$$\begin{aligned}
Q_{j_{\perp}}^{\text{res}} = & \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left[\frac{\sqrt{g}}{h_x} \frac{T_i}{e} \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + \tilde{j}_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) \right] + \\
& \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left[\frac{\sqrt{g}}{h_y} \frac{T_i}{e} \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + \tilde{j}_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) \right] \quad (\text{C.249})
\end{aligned}$$

appears due to the approximate treatment of the ion perpendicular force balance equations.

The term

$$Q_{\text{corr.dpc}}^{\text{res}} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left[\frac{\sqrt{g}}{h_x} \sum_a (n_e T_e + n_a T_i) V_a^{\text{corr.dpc}} \right] \quad (\text{C.250})$$

is physically meaningless, but it may give a notable contribution to the total energy balance residual in cases, for example, where it compensates numerical oscillations in the parallel velocity.

The rest of the residual terms are analytically zero, but may be non-zero due to numerical discretization errors:

$$Q_{i,nTu_{\parallel}}^{\text{res}} = \sum_a \left[\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n_a T_i b_x V_{a\parallel} \right) - \frac{n_a T_i}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} b_x V_{a\parallel} \right) - \frac{b_x V_{a\parallel}}{h_x} \frac{\partial n_a T_i}{\partial x} \right] \quad (\text{C.251})$$

$$Q_{e,dia}^{\text{res}} = \frac{5}{2} \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left[-\frac{\sqrt{g}}{h_x} n_e T_e^2 \frac{B_z}{eh_y} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right] + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left[\frac{\sqrt{g}}{h_y} n_e T_e^2 \frac{B_z}{eh_x} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) \right] + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} q_{ey}^{P.Sch} \right) \quad (\text{C.252})$$

$$Q_{i,dia}^{\text{res}} = \frac{5}{2} \sum_a \left[\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} n_a T_i \tilde{V}_{ax}^{(dia)} \right) + \frac{5}{2} \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} n_a T_i \tilde{V}_{ay}^{(dia)} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} q_{iy}^{P.Sch} \right) \right] \quad (\text{C.253})$$

C.9 Parallel viscosity

Old form of $(\nabla \pi_a^{\parallel})_{\parallel}$ was

$$(\nabla \pi_a^{\parallel})_{\parallel} = \begin{cases} 0, & a \neq 1 \\ -\frac{4}{3} b_x B^{\frac{3}{2}} \frac{\partial}{h_x \partial x} \left(\frac{c_{\eta_{ax}}^{(CL)} c_{\eta_{ax}}^{(Luciani)}}{B^2} \eta_{ax}^{(CL)} b_x \frac{\partial (B^{\frac{1}{2}} (V_{a\parallel} + 0 \cdot \frac{B}{B_x} V_{ax}^{(dia)} + 0 \cdot \frac{B}{B_x} V_x^{(E)}))}{h_x \partial x} \right), & a = 1 \\ -0 \cdot B^{\frac{3}{2}} b_x \frac{\partial}{h_x \partial x} \left(\frac{b_x}{\nu_{ax} B^2} \frac{\partial (B^{\frac{1}{2}} (q_{a\parallel}^{(0)} + \frac{B}{B_x} q_{ax}^{(dia)}))}{h_x \partial x} \right) \end{cases}$$

Contribution from anomalous viscosity was contained in l.h.s.

Because of $\eta_a^{(AN)} \ll \tilde{\eta}_{ax}^{(CL)}$, $\eta_{ax} \approx \tilde{\eta}_{ax}^{(CL)}$ and taking into account $B_x h_x h_y = \text{const}$

$$-\frac{4}{3} b_x B^{\frac{3}{2}} \frac{\partial}{h_x \partial x} \left(\frac{c_{\eta_{ax}}^{(CL)} c_{\eta_{ax}}^{(Luciani)}}{B^2} \eta_{ax}^{(CL)} b_x \frac{\partial (B^{\frac{1}{2}} V_{a\parallel})}{h_x \partial x} \right) = \frac{\partial}{h_z \sqrt{g} \partial x} \left(\frac{h_z \sqrt{g}}{h_x} \left(\frac{4}{3} \eta_{ax} \frac{\partial \ln h_z}{h_x \partial x} V_{\parallel a} - \frac{4}{3} \eta_{ax} \frac{\partial V_{\parallel a}}{h_x \partial x} \right) \right) - \frac{\partial}{h_z \sqrt{g} \partial x} \left(\frac{h_z \sqrt{g}}{h_x} \frac{4}{3} \eta_{ax} \frac{\partial \ln (h_z B^{\frac{1}{2}})}{B^{\frac{1}{2}} h_x \partial x} \right) B^{\frac{1}{2}} V_{\parallel a}, \quad a = 1 \quad (\text{C.254})$$

New form of $(\nabla \pi_a^{\parallel})_{\parallel}$

$$(\nabla \pi_a^{\parallel})_{\parallel} = \begin{cases} 0, & a \neq 1 \\ \frac{\partial}{h_z \sqrt{g} \partial x} \left(\frac{h_z \sqrt{g}}{h_x} \left(\frac{4}{3} \eta_{ax} \frac{\partial \ln h_z}{h_x \partial x} V_{\parallel a} - \frac{4}{3} \eta_{ax} \frac{\partial V_{\parallel a}}{h_x \partial x} \right) \right) - \frac{\partial}{h_z \sqrt{g} \partial x} \left(\frac{h_z \sqrt{g}}{h_x} \frac{4}{3} \eta_{ax} \frac{\partial \ln (h_z B^{\frac{1}{2}})}{B^{\frac{1}{2}} h_x \partial x} \right) B^{\frac{1}{2}} V_{\parallel a}, & a = 1 \end{cases}$$

C.10 Scaling of residuals

$$\text{Continuity equation residual for species } a: \frac{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} |r_{n_a,i,j}|}{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sqrt{g_{i,j}} n_{a,i,j}} \quad (\text{C.255})$$

$$\text{Parallel momentum equation residual for species } a: \frac{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} |r_{u_a,i,j}|}{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sqrt{g_{i,j}} n_{a,i,j} \sqrt{m_a T_i}} \quad (\text{C.256})$$

$$\text{Electron energy equation residual: } \frac{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} |r_{h_e,i,j}|}{\left(\frac{3}{2} + BoRis\right) \sum_{a=0}^{ns-1} \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sqrt{g_{i,j}} n_{a,i,j} (z_a T_e + T_i)} \quad (\text{C.257})$$

$$\text{Ion energy equation residual: } \frac{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} |r_{h_i,i,j}|}{\left(\frac{3}{2} + BoRis\right) \sum_{a=0}^{ns-1} \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sqrt{g_{i,j}} n_{a,i,j} (z_a T_e + T_i)} \quad (\text{C.258})$$

$$\text{Electric potential equation residual: } \frac{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} |r_{p_o,i,j}|}{\sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sqrt{g_{i,j}} e n_{e,i,j}} \quad (\text{C.259})$$

Appendix D

Boundary conditions in B2.5

D.1 Introduction

The boundary conditions are implemented in the B2.5 code by means of the boundary sources, that is in a similar way as the volume sources due to the atomic processes, but they are defined only in the boundary cells of the computational grid [26]. It means that in a case of the boundary conditions at the material surfaces the boundary cells have a role of combined electrostatic sheath and quasineutral collisionless magnetic presheath, which cannot be modelled with the B2.5 fluid equations. According to the existing fluid models of the magnetized boundary plasma, the role of the magnetic presheath, which is located between the collisional plasma presheath and electrostatic sheath, is to deflect the charged particles from their average motion along the magnetic field lines to the direction perpendicular to the wall. The average ion velocity along the magnetic field direction at the entrance of the magnetic presheath from the collisional presheath side is at least the local ion sound velocity, that is $c_{sMPSE} = \sqrt{\frac{k_B(ZT_{eMPSE} + T_{iMPSE})}{m_i}}$ in a two-component plasma, or $u_{\parallel MPSE} \geq c_{sMPSE}$ [51]. The component of the average ion velocity at the entrance of the electrostatic sheath, which is perpendicular to the wall, is also greater than or equal to the local ion sound velocity, i.e. $u_{SE} \geq c_{sSE}$ [52, 51]. Since both the magnetic presheath and electrostatic sheath are collisionless regions and assuming that the dissipative effects due to some turbulent mechanisms are negligible in these regions, particle, momentum and energy fluxes are conserved in these regions. The B2.5 equations for the bulk plasma are approximately valid in the collisional presheath, assuming that the charged particle velocity distributions do not differ significantly from Maxwellian or drifting-Maxwellian distributions. (However, it should be emphasized that according to the two-scale kinetic analysis of the boundary plasma near the solid surface, pure Maxwellian or drifting-Maxwellian ion velocity distributions cannot satisfy the generalized Bohm criterion [52, 53, 54]). It means that they provide the values for the charged particle densities, drift velocities, ion and electron temperatures at the magnetic presheath edge (*MPSE*), which can be used for calculating the fluxes at the magnetic presheath edge. The value of the poloidal component of the ion drift velocity at the magnetic presheath edge obtained from the bulk transport equations should be corrected, because of the effects related to the stronger local electric field in the magnetic presheath. This electric field causes a strong acceleration of the ions along the magnetic field lines, similarly to a non-magnetic plasma presheath or when the magnetic field is perpendicular to the surface, and an additional poloidal $\mathbf{E} \times \mathbf{B}$ drift due to the radial component of the electric field. In some cases the $\mathbf{E} \times \mathbf{B}$ drift velocity can be comparable with the local ion sound velocity [32, 18, 55]. The correction of the poloidal component of the ion drift velocity due to the $\mathbf{B} \times \nabla p$ drift is usually negligible, since it contributes mostly to the ion flow parallel to the surface [34]. However, more detailed investigations of the importance of this drift should be performed in the future studies, which should include also kinetic models and numerical simulations (e.g. particle and/or kinetic simulations) of the magnetized boundary plasma.

The boundary sources in the B2.5 code are defined by the boundary cell averages or volume integrals of the negative divergences of the incoming fluxes into the boundary cells. The boundary condition for the continuity equation for

each atomic species is given by:

$$S_p = - \int_{B.V.} \nabla \cdot \mathbf{\Gamma} dV = - \int_{B.S.} \mathbf{\Gamma} \cdot d\mathbf{A}, \quad (\text{D.1})$$

which means that in the discretized form of this equation the value of the particle flux, $\Gamma_j = n \langle v_j \rangle$, at the magnetic presheath edge as a function of the macroscopic plasma parameters should be specified. The boundary condition for the momentum equation is:

$$\mathbf{S}_m = - \int_{B.V.} \nabla \cdot \mathbf{M} dV = - \int_{B.S.} \mathbf{M} \cdot d\mathbf{A}. \quad (\text{D.2})$$

In this case the appropriate components of the momentum tensor, $M_{jk} = mn \langle v_j v_k \rangle$, at the magnetic presheath edge should be specified in the discretized equations. Similarly, the energy or heat boundary sources are given by:

$$S_q = - \int_{B.V.} \nabla \cdot \mathbf{Q} dV = - \int_{B.S.} \mathbf{Q} \cdot d\mathbf{A}, \quad (\text{D.3})$$

where the appropriate components of the electron and/or ion heat flux, $Q_j = \frac{1}{2} mn \langle v^2 v_j \rangle$, at the magnetic presheath edge as functions of the macroscopic plasma parameters should be specified in the discretized equations. The boundary charge source is given by:

$$S_{ch} = - \int_{B.V.} \nabla \cdot \mathbf{j} dV = - \int_{B.S.} \mathbf{j} \cdot d\mathbf{A}, \quad (\text{D.4})$$

where the electric current density, $\mathbf{j} = e (\sum_{\alpha} Z_{\alpha} \mathbf{\Gamma}_{\alpha} - \mathbf{\Gamma}_e)$, at the magnetic presheath edge should be specified in the discretized equations.

D.2 Analytical Expressions for the Standard ("Old") Boundary Sources in the B2.5 Code

The boundary sources are expressed as linear and/or weakly nonlinear functions of the macroscopic plasma parameters ($n_{\alpha}, n_e, u_{\alpha x}, T_i, T_e, \Phi$), calculated in the cells neighbouring the boundary cells. Specific dependencies of the boundary sources on these parameters are determined by the choice of the coefficients $C_{pj\alpha}$, $C_{mj\alpha}$, $C_{hij\alpha}$, C_{hej} and C_{chj} . These coefficients are free input parameters, which depend on the theoretical model used for describing the boundary conditions. They must be written by the user in the input file `b2ah.dat` (section 3.2). If the `b2ah.dat` file already exists and the boundary conditions for the new simulation are not significantly different, it is practical to write any minor changes to the boundary conditions in the input file `b2mn.dat` (section 3.5). (If the existing `b2ah.dat` file is corrected, it is not necessary to do that.) When the theoretical model predicts more complicated nonlinear dependence of the boundary fluxes on the plasma parameters, the effects of these nonlinearities can be only approximately taken into consideration in the calculation, e.g. by some appropriate choice of the numerical values of the input parameters. More details about the numerical implementation of the boundary sources can be found in the `b2stbc.F` source code.

D.2.1 Definitions of Physical Parameters Used in the Analytical Expressions for the Standard Boundary Sources

All following quantities are calculated locally at the appropriate boundary – "SOUTH" (S), "NORTH" (N), "WEST" (W), and "EAST" (E) –, i.e. in the boundary cells:

$$S_0 = V/h_{y1}, \quad h_{y1} = h_y \cos(\widehat{xy}) \quad (\text{D.5})$$

for the "SOUTH" and "NORTH" boundaries, where V is the volume of the boundary cell, h_y is the y -diameter (radial length) of the boundary cell and $\cos(\widehat{xy})$ is the cosine of the complementary angle between the x -direction and y -direction on the boundary cell;

$$S_0 = V/h_x \quad (\text{D.6})$$

for the "WEST" and "EAST" boundaries, where V is the volume of the boundary cell and h_x is the x -diameter (poloidal length) of the boundary cell;

$$S_1 = |S_1^\pm| = \left| \frac{B_y}{B} A \right| \quad (\text{D.7})$$

for the "SOUTH" and "NORTH" boundaries, where $S_1^\pm$ is the product of the magnetic field pitch and the surface area on the cell face between the boundary cell and its neighbouring (previous) cell $((ix, -1)$ and $(ix, 0)$ for the "SOUTH" and (ix, ny) and $(ix, ny - 1)$ for the "NORTH");

$$S_1 = |S_1^\pm| = \left| \frac{B_x}{B} A \right| \quad (\text{D.8})$$

for the "WEST" and "EAST" boundaries, where $S_1^\pm$ is the product of the magnetic field pitch and the surface area on the cell face between the boundary cell and its neighbouring (previous) cell $((-1, iy)$ and $(0, iy)$ for the "WEST" and (nx, iy) and $(nx - 1, iy)$ for the "EAST");

$$c_{s\alpha} = \sqrt{\frac{k_B (Z_\alpha T_e + T_i)}{A_\alpha m_p}} = \sqrt{\frac{k_B (Z_\alpha T_e + T_i)}{m_\alpha}} \quad (\text{D.9})$$

is the isothermal ion sound velocity for the atomic species α ;

$$c_{st} = \sqrt{\frac{\sum_{Z_\alpha \neq 0} n_\alpha k_B (Z_\alpha T_e + T_i)}{\sum_{Z_\alpha \neq 0} n_\alpha A_\alpha m_p}} = \sqrt{\frac{\sum_{Z_\alpha \neq 0} n_\alpha k_B (Z_\alpha T_e + T_i)}{\sum_{Z_\alpha \neq 0} n_\alpha m_\alpha}} \quad (\text{D.10})$$

is the isothermal ion sound velocity for a multi-species plasma.

$$c_{sD\alpha} = \max(c_{s\alpha}, 0.1c_{s\alpha} \pm (B_z/B_x) u_{Dx}) \quad (\text{D.11})$$

is the conditionally corrected ion drift velocity of the atomic species α due to the poloidal $\mathbf{E} \times \mathbf{B}$ drift, where "+" applies to the "WEST" and "-" to the "EAST" boundaries; the approximate correction is used only if $0.1c_{s\alpha} \pm (B_z/B_x) u_{Dx} > c_{s\alpha}$;

$$c_{stD} = \max(c_{st}, 0.1c_{st} \pm (B_z/B_x) u_{Dx}) \quad (\text{D.12})$$

is the conditionally corrected ion drift velocity of the multi-species plasma due to the poloidal $\mathbf{E} \times \mathbf{B}$ drift, where "+" applies to the "WEST" and "-" to the "EAST" boundaries; the approximate correction is used only if $0.1c_{st} \pm (B_z/B_x) u_{Dx} > c_{st}$;

$$\rho_\alpha = A_\alpha m_p n_\alpha = m_\alpha n_\alpha \quad (\text{D.13})$$

is the mass density of atomic species α ;

$$n_i = \sum_\alpha n_\alpha \quad (\text{D.14})$$

is the total atomic species (ion) particle density;

$$t_\alpha = \frac{n_\alpha}{n_i} \quad (\text{D.15})$$

is the relative concentration (particle density) of the atomic species α ;

$$\mathbf{j}_i = \pm e \sum_\alpha Z_\alpha \Gamma_\alpha \quad (\text{D.16})$$

is the positive ion electric current density in the boundary cells, where "+" applies to the "NORTH" and "EAST" boundaries and "-" to the "SOUTH" and "WEST" boundaries;

$$v_{th,e} = \sqrt{\frac{k_B T_e}{m_e}} \quad (\text{D.17})$$

is the electron thermal velocity;

$$J_e = \frac{1}{\sqrt{2\pi}} e n_e v_{th,e} S_1 \exp\left(-\frac{e\Phi}{k_B T_e}\right) \quad (\text{D.18})$$

is the current carried by the electrons; the electron and ion temperatures are expressed in energy units and

$$\mathbf{Q}_{h\alpha D} = \kappa_{\times\alpha} (\mathbf{b} \times \nabla T_\alpha) S_0 \quad (\text{D.19})$$

is the diamagnetic component of the heat current for species α , where $\mathbf{b} = \frac{\mathbf{B}}{B}$.

D.2.2 Standard Boundary Sources

Analytical formulae for the standard boundary sources are presented in this subsection. Their numerical implementation is included in the `b2stbc.F` source code. The `B2.5` code architecture is flexible enough so that it is possible to change the expressions for the boundary sources without changing the other parts of the source code, i.e. only the numerical values of the boundary sources are used in further calculations.

”SOUTH” Boundary (Cells (ix,-1))

Particle Sources

$$S_{p\alpha} = S_{p0\alpha} + S_{p1\alpha} n_\alpha \quad (\text{D.20})$$

$$S_{p0\alpha} = C_{p0\alpha} S_0 \quad (\text{D.21})$$

$$S_{p1\alpha} = C_{p1\alpha} S_0 + C_{p2\alpha} c_{s\alpha} S_0 + C_{p3\alpha} c_{s\alpha} S_1 + C_{p4\alpha} c_{st} S_0 + C_{p5\alpha} c_{st} S_1 \quad (\text{D.22})$$

Parallel momentum Sources

$$S_{m\alpha} = S_{m0\alpha} + S_{m1\alpha} u_{\alpha x} + S_{m2\alpha} \rho_\alpha + S_{m3\alpha} \rho_\alpha u_{\alpha x} \quad (\text{D.23})$$

$$S_{m0\alpha} = C_{m0\alpha} S_0 \quad (\text{D.24})$$

$$S_{m1\alpha} = 0 \quad (\text{D.25})$$

$$S_{m2\alpha} = C_{m1\alpha} (-c_{s\alpha}) S_1^\pm + C_{m2\alpha} (-c_{st}) S_1^\pm + C_{m5\alpha} \frac{\max(0, \Gamma_{\alpha y})}{n_\alpha} \quad (\text{D.26})$$

$$S_{m3\alpha} = C_{m3\alpha} S_0 + C_{m4\alpha} S_1 + C_{m6\alpha} \frac{\max(0, -\Gamma_{\alpha y})}{n_\alpha} \quad (\text{D.27})$$

All Atom/Ion Heat Source

$$S_{hi} = S_{hi0} + S_{hi1}T_i + S_{hi2}n_i + S_{hi3}n_iT_i \quad (D.28)$$

$$S_{hi0} = \sum_{\alpha} [C_{hi0\alpha}t_{\alpha}S_0 + C_{hi6\alpha} \max(0, \Gamma_{\alpha y})] + Q_{hiDy} \quad (D.29)$$

$$S_{hi1} = \sum_{\alpha} C_{hi1\alpha}t_{\alpha}S_0 + \frac{Q_{hiDy}}{T_i} \quad (D.30)$$

$$S_{hi2} = 0 \quad (D.31)$$

$$S_{hi3} = \sum_{\alpha} \left[C_{hi2\alpha}t_{\alpha}c_{s\alpha}S_0 + C_{hi3\alpha}t_{\alpha}c_{s\alpha}S_1 + C_{hi4\alpha}t_{\alpha}c_{st}S_0 + \right. \\ \left. C_{hi5\alpha}t_{\alpha}c_{st}S_1 + C_{hi7\alpha} \frac{\max(0, -\Gamma_{\alpha y})}{n_i} \right] \quad (D.32)$$

Electron Heat Source

$$S_{he} = S_{he0} + S_{he1}T_e + S_{he2}n_e + S_{he3}n_eT_e \quad (D.33)$$

$$S_{he0} = C_{he0}S_0 + C_{he6} \max(0, \Gamma_{ey}) + Q_{heDy} \quad (D.34)$$

$$S_{he1} = C_{he1}S_0 + \frac{Q_{heDy}}{T_e} \quad (D.35)$$

$$S_{he2} = 0 \quad (D.36)$$

$$S_{he3} = C_{he2}v_{th,e}S_0 + C_{he3}v_{th,e}S_1 + C_{he4}c_{st}S_0 + \\ C_{he5}c_{st}S_1 + C_{he7} \frac{\max(0, -\Gamma_{ey})}{n_e} \quad (D.37)$$

Electric Charge Source

$$S_{ch} = S_{ch0} + S_{ch1}\Phi + S_{ch2}n_e + S_{ch3}n_e\Phi \quad (D.38)$$

$$S_{ch0} = C_{ch7}J_{ey} - C_{ch6}J_{iy} + \max(C_{ch6}J_{iy}, C_{ch7}J_{ey}) \frac{e\Phi}{T_e} \quad (D.39)$$

$$S_{ch1} = -\frac{\max(C_{ch6}J_{iy}, C_{ch7}J_{ey})e}{T_e} \quad (D.40)$$

$$S_{ch2} = C_{ch0}ev_{th,e}S_0 + C_{ch2}ev_{th,e}S_1 + C_{ch4}ec_{st}S_1 \quad (D.41)$$

$$S_{ch3} = C_{ch1} \frac{e^2v_{th,e}S_0}{T_e} + C_{ch3} \frac{e^2v_{th,e}S_1}{T_e} + C_{ch5} \frac{e^2c_{st}S_1}{T_e} \quad (D.42)$$

”NORTH” Boundary (ix,ny)

Particle Sources

$$S_{p\alpha} = S_{p0\alpha} + S_{p1\alpha}n_{\alpha} \quad (D.43)$$

$$S_{p0\alpha} = C_{p0\alpha}S_0 \quad (D.44)$$

$$S_{p1\alpha} = C_{p1\alpha}S_0 + C_{p2\alpha}c_{s\alpha}S_0 + C_{p3\alpha}c_{s\alpha}S_1 + C_{p4\alpha}c_{st}S_0 + C_{p5\alpha}c_{st}S_1 \quad (D.45)$$

Parallel momentum Sources

$$S_{m\alpha} = S_{m0\alpha} + S_{m1\alpha}u_{\alpha x} + S_{m2\alpha}\rho_{\alpha} + S_{m3\alpha}\rho_{\alpha}u_{\alpha x} \quad (D.46)$$

$$S_{m0\alpha} = C_{m0\alpha}S_0 \quad (D.47)$$

$$S_{m1\alpha} = 0 \quad (D.48)$$

$$S_{m2\alpha} = C_{m1\alpha}c_{s\alpha}S_1^{\pm} + C_{m2\alpha}c_{st}S_1^{\pm} + C_{m5\alpha}\frac{\max(0, -\Gamma_{\alpha y})}{n_{\alpha}} \quad (D.49)$$

$$S_{m3\alpha} = C_{m3\alpha}S_0 + C_{m4\alpha}S_1 + C_{m6\alpha}\frac{\max(0, \Gamma_{\alpha y})}{n_{\alpha}} \quad (D.50)$$

All Atom/Ion Heat Source

$$S_{hi} = S_{hi0} + S_{hi1}T_i + S_{hi2}n_i + S_{hi3}n_iT_i \quad (D.51)$$

$$S_{hi0} = \sum_{\alpha} [C_{hi0\alpha}t_{\alpha}S_0 + C_{hi6\alpha}\max(0, -\Gamma_{\alpha y})] - Q_{hiDy} \quad (D.52)$$

$$S_{hi1} = \sum_{\alpha} C_{hi1\alpha}t_{\alpha}S_0 - \frac{Q_{hiDy}}{T_i} \quad (D.53)$$

$$S_{hi2} = 0 \quad (D.54)$$

$$S_{hi3} = \sum_{\alpha} \left[C_{hi2\alpha}t_{\alpha}c_{s\alpha}S_0 + C_{hi3\alpha}t_{\alpha}c_{s\alpha}S_1 + C_{hi4\alpha}t_{\alpha}c_{st}S_0 + C_{hi5\alpha}t_{\alpha}c_{st}S_1 + C_{hi7\alpha}\frac{\max(0, \Gamma_{\alpha y})}{n_i} \right] \quad (D.55)$$

Electron Heat Source

$$S_{he} = S_{he0} + S_{he1}T_e + S_{he2}n_e + S_{he3}n_eT_e \quad (D.56)$$

$$S_{he0} = C_{he0}S_0 + C_{he6} \max(0, -\Gamma_{ey}) - Q_{heDy} \quad (D.57)$$

$$S_{he1} = C_{he1}S_0 - \frac{Q_{heDy}}{T_e} \quad (D.58)$$

$$S_{he2} = 0 \quad (D.59)$$

$$S_{he3} = C_{he2}v_{th,e}S_0 + C_{he3}v_{th,e}S_1 + C_{he4}c_{st}S_0 + C_{he5}c_{st}S_1 + C_{he7} \frac{\max(0, \Gamma_{ey})}{n_e} \quad (D.60)$$

Electric Charge Source

$$S_{ch} = S_{ch0} + S_{ch1}\Phi + S_{ch2}n_e + S_{ch3}n_e\Phi \quad (D.61)$$

$$S_{ch0} = C_{ch7}J_{ey} - C_{ch6}J_{iy} + \max(C_{ch6}J_{iy}, C_{ch7}J_{ey}) \frac{e\Phi}{T_e} \quad (D.62)$$

$$S_{ch1} = -\frac{\max(C_{ch6}J_{iy}, C_{ch7}J_{ey})e}{T_e} \quad (D.63)$$

$$S_{ch2} = C_{ch0}ev_{th,e}S_0 + C_{ch2}ev_{th,e}S_1 + C_{ch4}ec_{st}S_1 \quad (D.64)$$

$$S_{ch3} = C_{ch1} \frac{e^2v_{th,e}S_0}{T_e} + C_{ch3} \frac{e^2v_{th,e}S_1}{T_e} + C_{ch5} \frac{e^2c_{st}S_1}{T_e} \quad (D.65)$$

”WEST” Boundary (Cells (-1,iy), Divertor Plate)

Particle Sources

$$S_{p\alpha} = S_{p0\alpha} + S_{p1\alpha}n_\alpha \quad (D.66)$$

$$S_{p0\alpha} = C_{p0\alpha}S_0 + C_{p6\alpha} \max(0, u_{\alpha x}S_1^\pm) \quad (D.67)$$

$$S_{p1\alpha} = C_{p1\alpha}S_0 + C_{p2\alpha}c_{sD\alpha}S_0 + C_{p3\alpha}c_{sD\alpha}S_1 + C_{p4\alpha}c_{stD}S_0 + C_{p5\alpha}c_{stD}S_1 + C_{p7\alpha} \max(0, -u_{\alpha x}S_1^\pm) \quad (D.68)$$

Parallel momentum Sources

$$S_{m\alpha} = S_{m0\alpha} + S_{m1\alpha}u_{\alpha x} + S_{m2\alpha}\rho_{\alpha} + S_{m3\alpha}\rho_{\alpha}u_{\alpha x} \quad (D.69)$$

$$S_{m0\alpha} = C_{m0\alpha}S_0 \quad (D.70)$$

$$S_{m1\alpha} = 0 \quad (D.71)$$

$$S_{m2\alpha} = C_{m1\alpha}(-c_{sD\alpha})S_1^{\pm} + C_{m2\alpha}(-c_{stD})S_1^{\pm} + C_{m5\alpha}\frac{\max(0, \Gamma_{\alpha x})}{n_{\alpha}} \quad (D.72)$$

$$S_{m3\alpha} = C_{m3\alpha}S_0 + C_{m4\alpha}S_1 + C_{m6\alpha}\frac{\max(0, -\Gamma_{\alpha x})}{n_{\alpha}} \quad (D.73)$$

All Atom/Ion Heat Source

$$S_{hi} = S_{hi0} + S_{hi1}T_i + S_{hi2}n_i + S_{hi3}n_iT_i \quad (D.74)$$

$$S_{hi0} = \sum_{\alpha} [C_{hi0\alpha}t_{\alpha}S_0 + C_{hi6\alpha}\max(0, \Gamma_{\alpha x})] + Q_{hiDx} \quad (D.75)$$

$$S_{hi1} = \sum_{\alpha} C_{hi1\alpha}t_{\alpha}S_0 + \frac{Q_{hiDx}}{T_i} \quad (D.76)$$

$$S_{hi2} = 0 \quad (D.77)$$

$$S_{hi3} = \sum_{\alpha} \left[C_{hi2\alpha}t_{\alpha}c_{sD\alpha}S_0 + C_{hi3\alpha}t_{\alpha}c_{sD\alpha}S_1 + C_{hi4\alpha}t_{\alpha}c_{stD}S_0 + \right. \\ \left. C_{hi5\alpha}t_{\alpha}c_{stD}S_1 + C_{hi7\alpha}\frac{\max(0, -\Gamma_{\alpha x})}{n_i} \right] \quad (D.78)$$

Electron Heat Source

$$S_{he} = S_{he0} + S_{he1}T_e + S_{he2}n_e + S_{he3}n_eT_e \quad (D.79)$$

$$S_{he0} = C_{he0}S_0 + C_{he6}\max(0, \Gamma_{ex}) + Q_{heDx} \quad (D.80)$$

$$S_{he1} = C_{he1}S_0 + \frac{Q_{heDx}}{T_e} \quad (D.81)$$

$$S_{he2} = 0 \quad (D.82)$$

$$S_{he3} = C_{he2}v_{th,e}S_0 + C_{he3}v_{th,e}S_1 + C_{he4}c_{stD}S_0 + \\ C_{he5}c_{stD}S_1 + C_{he7}\frac{\max(0, -\Gamma_{ex})}{n_e} \quad (D.83)$$

Electric Charge Source

$$S_{ch} = S_{ch0} + S_{ch1}\Phi + S_{ch2}n_e + S_{ch3}n_e\Phi \quad (D.84)$$

$$S_{ch0} = C_{ch7}J_{ex} - C_{ch6}J_{ix} + \max(C_{ch6}J_{ix}, C_{ch7}J_{ex}) \frac{e\Phi}{T_e} \quad (D.85)$$

$$S_{ch1} = -\frac{\max(C_{ch6}J_{ix}, C_{ch7}J_{ex})e}{T_e} \quad (D.86)$$

$$S_{ch2} = C_{ch0}ev_{th,e}S_0 + C_{ch2}ev_{th,e}S_1 + C_{ch4}ec_{stD}S_1 \quad (D.87)$$

$$S_{ch3} = C_{ch1}\frac{e^2v_{th,e}S_0}{T_e} + C_{ch3}\frac{e^2v_{th,e}S_1}{T_e} + C_{ch5}\frac{e^2c_{stD}S_1}{T_e} \quad (D.88)$$

”EAST” Boundary (Cells (nx,iy), Divertor Plate)

Particle Sources

$$S_{p\alpha} = S_{p0\alpha} + S_{p1\alpha}n_\alpha \quad (D.89)$$

$$S_{p0\alpha} = C_{p0\alpha}S_0 + C_{p6\alpha}\max(0, -u_{\alpha x}S_1^\pm) \quad (D.90)$$

$$S_{p1\alpha} = C_{p1\alpha}S_0 + C_{p2\alpha}c_{sD\alpha}S_0 + C_{p3\alpha}c_{sD\alpha}S_1 + C_{p4\alpha}c_{stD}S_0 + C_{p5\alpha}c_{stD}S_1 + C_{p7\alpha}\max(0, u_{\alpha x}S_1^\pm) \quad (D.91)$$

Parallel momentum Sources

$$S_{m\alpha} = S_{m0\alpha} + S_{m1\alpha}u_{\alpha x} + S_{m2\alpha}\rho_\alpha + S_{m3\alpha}\rho_\alpha u_{\alpha x} \quad (D.92)$$

$$S_{m0\alpha} = C_{m0\alpha}S_0 \quad (D.93)$$

$$S_{m1\alpha} = 0 \quad (D.94)$$

$$S_{m2\alpha} = C_{m1\alpha}c_{sD\alpha}S_1^\pm + C_{m2\alpha}c_{stD}S_1^\pm + C_{m5\alpha}\frac{\max(0, -\Gamma_{\alpha x})}{n_\alpha} \quad (D.95)$$

$$S_{m3\alpha} = C_{m3\alpha}S_0 + C_{m4\alpha}S_1 + C_{m6\alpha}\frac{\max(0, \Gamma_{\alpha x})}{n_\alpha} \quad (D.96)$$

All Atom/Ion Heat Source

$$S_{hi} = S_{hi0} + S_{hi1}T_i + S_{hi2}n_i + S_{hi3}n_iT_i \quad (D.97)$$

$$S_{hi0} = \sum_{\alpha} [C_{hi0\alpha}t_{\alpha}S_0 + C_{hi6\alpha}\max(0, -\Gamma_{\alpha x})] - Q_{hiDx} \quad (D.98)$$

$$S_{hi1} = \sum_{\alpha} C_{hi1\alpha}t_{\alpha}S_0 - \frac{Q_{hiDx}}{T_i} \quad (D.99)$$

$$S_{hi2} = 0 \quad (D.100)$$

$$S_{hi3} = \sum_{\alpha} \left[C_{hi2\alpha}t_{\alpha}c_{sD\alpha}S_0 + C_{hi3\alpha}t_{\alpha}c_{sD\alpha}S_1 + C_{hi4\alpha}t_{\alpha}c_{stD}S_0 + C_{hi5\alpha}t_{\alpha}c_{stD}S_1 + C_{hi7\alpha}\frac{\max(0, \Gamma_{\alpha x})}{n_i} \right] \quad (D.101)$$

Electron Heat Source

$$S_{he} = S_{he0} + S_{he1}T_e + S_{he2}n_e + S_{he3}n_eT_e \quad (D.102)$$

$$S_{he0} = C_{he0}S_0 + C_{he6}\max(0, -\Gamma_{ex}) - Q_{heDx} \quad (D.103)$$

$$S_{he1} = C_{he1}S_0 - \frac{Q_{heDx}}{T_e} \quad (D.104)$$

$$S_{he2} = 0 \quad (D.105)$$

$$S_{he3} = C_{he2}v_{th,e}S_0 + C_{he3}v_{th,e}S_1 + C_{he4}c_{stD}S_0 + C_{he5}c_{stD}S_1 + C_{he7}\frac{\max(0, \Gamma_{ex})}{n_e} \quad (D.106)$$

Electric Charge Source

$$S_{ch} = S_{ch0} + S_{ch1}\Phi + S_{ch2}n_e + S_{ch3}n_e\Phi \quad (D.107)$$

$$S_{ch0} = C_{ch7}J_{ex} - C_{ch6}J_{ix} + \max(C_{ch6}J_{ix}, C_{ch7}J_{ex})\frac{e\Phi}{T_e} \quad (D.108)$$

$$S_{ch1} = -\frac{\max(C_{ch6}J_{ix}, C_{ch7}J_{ex})e}{T_e} \quad (D.109)$$

$$S_{ch2} = C_{ch0}ev_{th,e}S_0 + C_{ch2}ev_{th,e}S_1 + C_{ch4}ec_{stD}S_1 \quad (D.110)$$

$$S_{ch3} = C_{ch1}\frac{e^2v_{th,e}S_0}{T_e} + C_{ch3}\frac{e^2v_{th,e}S_1}{T_e} + C_{ch5}\frac{e^2c_{stD}S_1}{T_e} \quad (D.111)$$

Appendix E

Quantities stored in b2time.nc

The following scalar time dependent quantities are stored in `b2time.nc` and can be plotted by

```
2dt tesepm
```

to plot the midplane separatrix electron temperature as a function of time, or

```
2d nesepm tesepm
```

to plot the midplane separatrix electron temperature versus the midplane separatrix electron density. The frequency at which (and whether) these quantities are written out is controlled by the '`b2mndr.b2time`' switch.

If '`b2mndr.stim`' is set to a negative value, data from the current **SOLPS-ITER** run will be appended to the existing `b2time.nc` file, otherwise the file is created anew.

In the case of double-null runs, the divertor and separatrix quantities are given in pairs. For the midplane quantities, the first value is for the outer midplane, and the second value for the inner midplane. Divertor quantities show first the bottom then the top divertor.

For double-null (DN) and lower single-null (LSN) topologies, the Western edge refers to the inboard divertor, while it points to the outboard divertor for upper single-null (USN) topologies. Conversely, the Eastern edge is the outboard divertor for DN and LSN cases, but the inboard divertor for USN cases.

In the case of linear runs, the midplane refers, by default, to the midpoint of the domain. The divertor throat quantities are not defined.

In all cases, the point of reference of the midplane quantities can be adjusted by changing the value of `jxa` (and `jxi` for double-null runs) by assigning it an explicit value in the `b2mn.dat` file.

quantity	description	units
ntstep	timestep index	
timesa	time stamp	s
fnixip	integrated poloidal main ion particle flux, Western edge	s^{-1}
feexip	integrated poloidal electron energy flux, Western edge	W
feixip	integrated poloidal ion energy flux, Western edge	W
fetxip	integrated poloidal total energy flux, Western edge	W
fchxip	integrated poloidal current, Western edge	A
fnixap	integrated poloidal main ion particle flux, Eastern edge	s^{-1}
feexap	integrated poloidal electron energy flux, Eastern edge	W
feixap	integrated poloidal ion energy flux, Eastern edge	W
fetxap	integrated poloidal total energy flux, Eastern edge	W
fchxap	integrated poloidal current, Eastern edge	A
<i>continued on next page</i>		

continued from previous page		
quantity	description	units
nesepi	separatrix electron density, Western edge	m^{-3}
tesepi	separatrix electron temperature, Western edge	eV
tisepi	separatrix ion temperature, Western edge	eV
tpsepi	separatrix target temperature, Western edge	K
posepi	separatrix potential, Western edge	V
nesepm	separatrix electron density, outer midplane	m^{-3}
tesepm	separatrix electron temperature, outer midplane	eV
tisepm	separatrix ion temperature, outer midplane	eV
posepm	separatrix potential, outer midplane	V
dnsepm	main ion density diffusion coefficient, outer midplane	$m^2.s^{-1}$
dpsepm	main neutral pressure diffusion coefficient, outer midplane	$m^2.s^{-1}$
kesepm	electron thermal diffusivity, outer midplane	$m^2.s^{-1}$
kisepm	ion thermal diffusivity, outer midplane	$m^2.s^{-1}$
vxsepm	main ion poloidal pinch velocity, outer midplane	$m.s^{-1}$
vysepm	main ion radial pinch velocity, outer midplane	$m.s^{-1}$
vssepm	main ion particle viscosity, outer midplane	$m^2.s^{-1}$
nesepa	separatrix electron density, Eastern edge	m^{-3}
tesepa	separatrix electron temperature, Eastern edge	eV
tisepa	separatrix ion temperature, Eastern edge	eV
tpsepa	separatrix target temperature, Eastern edge	K
posepa	separatrix potential, Eastern edge	V
nemxip	maximum electron density, Western edge	m^{-3}
temxip	maximum electron temperature, Western edge	eV
timxip	maximum ion temperature, Western edge	eV
tpmxip	maximum target temperature, Western edge	K
pomxip	maximum potential, Western edge	V
nemxap	maximum electron density, Eastern edge	m^{-3}
temxap	maximum electron temperature, Eastern edge	eV
timxap	maximum ion temperature, Eastern edge	eV
tpmxap	maximum target temperature, Eastern edge	K
pomxap	maximum potential, Eastern edge	V
fniyip	integrated radial main ion particle flux, main chamber	s^{-1}
feeyip	integrated radial electron energy flux, main chamber	W
feiyip	integrated radial ion energy flux, main chamber	W
fetyip	integrated radial total energy flux, main chamber	W
fchyip	integrated radial current, main chamber	A
fniyap	integrated radial main ion particle flux, divertor region	s^{-1}
feeyap	integrated radial electron energy flux, divertor region	W
feiyap	integrated radial ion energy flux, divertor region	W
fetyap	integrated radial total energy flux, divertor region	W
fchyap	integrated radial current, divertor region	A
pwmxip	maximum total power flux, Western edge	$W.m^{-2}$
pwmxap	maximum total power flux, Eastern edge	$W.m^{-2}$
tmne	total number of particles	
tmte	total electron energy	eV
tmti	total ion energy	eV
tmhacore	H_α emissivity, core	$photons.m^{-2}.sr^{-1}?$
tmhasol	H_α emissivity, SOL above the X-point	$photons.m^{-2}.sr^{-1}?$
tmhadiv	H_α emissivity, divertor region	$photons.m^{-2}.sr^{-1}?$
fnisip	poloidal main ion particle flux, into Western separatrix throat	s^{-1}
continued on next page		

<i>continued from previous page</i>		
quantity	description	units
feesip	poloidal electron energy flux, into Western separatrix throat	W
feisip	poloidal ion energy flux, into Western separatrix throat	W
fetsip	poloidal total energy flux, into Western separatrix throat	W
fchsip	poloidal current, into Western separatrix throat	A
fnisap	poloidal main ion particle flux, into Eastern separatrix throat	s^{-1}
feesap	poloidal electron energy flux, into Eastern separatrix throat	W
feisap	poloidal ion energy flux, into Eastern separatrix throat	W
fetsap	poloidal total energy flux, into Eastern separatrix throat	W
fchsap	poloidal current, into Eastern separatrix throat	A
fnisipp	poloidal main ion particle flux, core x-pt flux region	s^{-1}
feesipp	poloidal electron energy flux, core x-pt flux region	W
feisipp	poloidal ion energy flux, core x-pt flux region	W
fetsipp	poloidal total energy flux, core x-pt flux region	W
fchsipp	poloidal current, core x-pt flux region	A
fnisapp	poloidal main ion particle flux, x-pt private flux region	s^{-1}
feesapp	poloidal electron energy flux, x-pt private flux region	W
feisapp	poloidal ion energy flux, x-pt private flux region	W
fetsapp	poloidal total energy flux, x-pt private flux region	W
fchsapp	poloidal current, x-pt private flux region	A

Table E.1: Scalar time dependent quantities stored in b2time.nc

The following are vector time dependent quantities stored in `b2time.nc`. These are quantities along lines of constant ix at the Western target (l) (inner for LSN, outer for USN, and lower inner for DN configurations), inner midplane (i), outer midplane (a) and Eastern target (r) (outer for LSN, inner for USN, and lower outer for DN configurations). The positions of the inner and outer midplanes can be selected by `b2mwti-jxi` and `b2mwti-jxa`, respectively, in `b2mn.dat`.

Note that, when processed with `2d_profiles`, the fluxes are renormalised by the contact area to the neighbouring cells, and therefore their units must all be multiplied by m^{-2} . When using `2d_profiles_extended`, additional files are created using the name template `f?3d?P.last10` where the area used is the surface area perpendicular to the magnetic field direction.

quantity	description	units
fn3dl	poloidal particle flux of main ion, Western edge	$s^{-1}.m^{-2}$
fl3dl	poloidal electron flux, Western edge	$s^{-1}.m^{-2}$
fo3dl	poloidal total ion flux, Western edge	$s^{-1}.m^{-2}$
fe3dl	poloidal electron energy flux, Western edge	$W.m^{-2}$
fi3dl	poloidal ion energy flux, Western edge	$W.m^{-2}$
ft3dl	poloidal total energy flux, Western edge	$W.m^{-2}$
fc3dl	poloidal current, Western edge	$A.m^{-2}$
ne3dl	electron density, Western edge	m^{-3}
te3dl	electron temperature, Western edge	eV
ti3dl	ion temperature, Western edge	eV
tp3dl	target temperature, Western edge	K
po3dl	potential, Western edge	V
an3dl	atom density of main neutral, Western edge	m^{-3}
mn3dl	molecule density of 1 st molecule, Western edge	m^{-3}
ne3di	electron density, inboard midplane	m^{-3}
<i>continued on next page</i>		

continued from previous page		
quantity	description	units
te3di	electron temperature, inboard midplane	eV
ti3di	ion temperature, inboard midplane	eV
po3di	potential, inboard midplane	V
an3di	atom density of main neutral, inboard midplane	m^{-3}
mn3di	molecule density of 1 st molecule, inboard midplane	m^{-3}
dn3di	particle density diffusivity, inboard midplane	$m^2.s^{-1}$
dp3di	particle pressure diffusivity, inboard midplane	$m^2.s^{-1}$
ke3di	electron heat diffusivity, inboard midplane	$m^2.s^{-1}$
ki3di	ion heat diffusivity, inboard midplane	$m^2.s^{-1}$
vs3di	main ion viscosity, inboard midplane	$m^2.s^{-1}$
vx3di	main ion poloidal pinch velocity, inboard midplane	$m.s^{-1}$
vy3di	main ion radial pinch velocity, inboard midplane	$m.s^{-1}$
ne3da	electron density, outboard midplane	m^{-3}
te3da	electron temperature, outboard midplane	eV
ti3da	ion temperature, outboard midplane	eV
po3da	potential, outboard midplane	V
an3da	atom density of main neutral, outboard midplane	m^{-3}
mn3da	molecule density of 1 st molecule, outboard midplane	m^{-3}
dn3da	particle density diffusivity, outboard midplane	$m^2.s^{-1}$
dp3da	particle pressure diffusivity, outboard midplane	$m^2.s^{-1}$
ke3da	electron heat diffusivity, outboard midplane	$m^2.s^{-1}$
ki3da	ion heat diffusivity, outboard midplane	$m^2.s^{-1}$
vs3da	main ion viscosity, outboard midplane	$m^2.s^{-1}$
vx3da	main ion poloidal pinch velocity, outboard midplane	$m.s^{-1}$
vy3da	main ion radial pinch velocity, outboard midplane	$m.s^{-1}$
ne3dr	electron density, Eastern edge	m^{-3}
te3dr	electron temperature, Eastern edge	eV
ti3dr	ion temperature, Eastern edge	eV
tp3dr	target temperature, Eastern edge	K
po3dr	potential, Eastern edge	V
an3dr	atom density of main neutral, Eastern edge	m^{-3}
mn3dr	molecule density of 1 st molecule, Eastern edge	m^{-3}
fn3dr	poloidal particle flux of main ion, Eastern edge	$s^{-1}.m^{-2}$
fe3dr	poloidal electron energy flux, Eastern edge	$W.m^{-2}$
fi3dr	poloidal ion energy flux, Eastern edge	$W.m^{-2}$
ft3dr	poloidal total energy flux, Eastern edge	$W.m^{-2}$
fc3dr	poloidal current, Eastern edge	$A.m^{-2}$
fi3dr	poloidal electron flux, Eastern edge	$s^{-1}.m^{-2}$
fo3dr	poloidal total ion flux, Eastern edge	$s^{-1}.m^{-2}$

Table E.2: Vector time dependent quantities stored in b2time.nc. Note that the conversion to "per unit area" for fluxes is done in the script "2d_profiles"; the actual data stored in b2time.nc for fluxes are the fluxes per cell. There is a similar conversion from J to eV for temperatures.

In the case of double-null geometries, additional quantities are created related to the inboard top ($t_{\perp}$) and outboard top ($t_{\top}$) divertors. These are listed below in Table E.3.

quantity	description	units
ne3dtl	electron density, inboard top divertor	m^{-3}
te3dtl	electron temperature, inboard top divertor	eV
ti3dtl	ion temperature, inboard top divertor	eV
tp3dtl	target temperature, inboard top divertor	K
po3dtl	potential, inboard top divertor	V
an3dtl	atom density of main neutral, inboard top divertor	m^{-3}
mn3dtl	molecule density of 1 st molecule, inboard top divertor	m^{-3}
fn3dtl	poloidal particle flux of main ion, inboard top divertor	s^{-1}
fe3dtl	poloidal electron energy flux, inboard top divertor	W
fi3dtl	poloidal ion energy flux, inboard top divertor	W
ft3dtl	poloidal total energy flux, inboard top divertor	W
fc3dtl	poloidal current, inboard top divertor	A
fl3dtl	poloidal electron flux, inboard top divertor	s^{-1}
fo3dtl	poloidal total ion flux, inboard top divertor	s^{-1}
ne3dtr	electron density, outboard top divertor	m^{-3}
te3dtr	electron temperature, outboard top divertor	eV
ti3dtr	ion temperature, outboard top divertor	eV
tp3dtr	target temperature, outboard top divertor	K
po3dtr	potential, outboard top divertor	V
an3dtr	atom density of main neutral, outboard top divertor	m^{-3}
mn3dtr	molecule density of 1 st molecule, outboard top divertor	m^{-3}
fn3dtr	poloidal particle flux of main ion, outboard top divertor	s^{-1}
fe3dtr	poloidal electron energy flux, outboard top divertor	W
fi3dtr	poloidal ion energy flux, outboard top divertor	W
ft3dtr	poloidal total energy flux, outboard top divertor	W
fc3dtr	poloidal current, outboard top divertor	A
fl3dtr	poloidal electron flux, outboard top divertor	s^{-1}
fo3dtr	poloidal total ion flux, outboard top divertor	s^{-1}

Table E.3: Double-null vector time dependent quantities stored in b2time.nc

Additional multi-dimensional writes to `b2time.nc` can be made using the “b2mwti_2dwrite” switch. If set to “1”, then a few 2D arrays (n_e , T_e , T_i) are output with each write to `b2time.nc`. If set to “2”, then additional quantities such as fluxes are added, see below Table E.4.

quantity	description	units
If <code>b2mwti_2dwrite</code> ≥ 1		
ne2d	electron density	m^{-3}
te2d	electron temperature	eV
ti2d	ion temperature	eV
and if <code>b2mwti_2dwrite</code> = 2		
po2d	electric potential	V
kin2d	parallel kinetic energy	J
rsahi2d	ionization energy source/sink	W
rsana2d	ionization rate	s^{-1}
rrahi2d	recombination energy source/sink	W
rrana2d	recombination rate	s^{-1}
rcxhi2d	charge exchange energy source/sink	W
continued on next page		

<i>continued from previous page</i>		
quantity	description	units
rcxna2d	charge exchange rate	s^{-1}
rgrad2d	Line radiation rate	W
rqahe2d	Electron cooling rate	W
fhe2d	Electron heat flux	W
fhi2d	Ion heat flux	W
fch2d	Electric current	A
fna2d	Particle flux	s^{-1}

Table E.4: Multi-dimensional time-dependent quantities stored in `b2time.nc`

When using the averaging scheme for `Eirene` coupled runs (see Section 5.4), obtained by activating the '`b2mndt_av`' switch, the `b2time.nc` file will also contain some averaged data, listed below in table E.5. These quantities can be visualized by means of the `2dt_av` script.

quantity	description	units
nastep	timestep index	
batchsa	time stamp for averaged batch	s
nesepi_av	averaged separatrix electron density, Western edge	m^{-3}
tesepi_av	averaged separatrix electron temperature, Western edge	eV
tisepi_av	averaged separatrix ion temperature, Western edge	eV
posepi_av	averaged separatrix potential, Western edge	V
nesepm_av	averaged separatrix electron density, outer midplane	m^{-3}
tesepm_av	averaged separatrix electron temperature, outer midplane	eV
tisepm_av	averaged separatrix ion temperature, outer midplane	eV
posepm_av	averaged separatrix potential, outer midplane	V
nesepa_av	averaged separatrix electron density, Eastern edge	m^{-3}
tesepa_av	averaged separatrix electron temperature, Eastern edge	eV
tisepa_av	averaged separatrix ion temperature, Eastern edge	eV
posepa_av	averaged separatrix potential, Eastern edge	V
nemxip_av	averaged maximum electron density, Western edge	m^{-3}
temxip_av	averaged maximum electron temperature, Western edge	eV
timxip_av	averaged maximum ion temperature, Western edge	eV
pomxip_av	averaged maximum potential, Western edge	V
nemxap_av	averaged maximum electron density, Eastern edge	m^{-3}
temxap_av	averaged maximum electron temperature, Eastern edge	eV
timxap_av	averaged maximum ion temperature, Eastern edge	eV
pomxap_av	averaged maximum potential, Eastern edge	V
nesepi_std	averaged separatrix electron density, Western edge	m^{-3}
tesepi_std	averaged separatrix electron temperature, Western edge	eV
tisepi_std	averaged separatrix ion temperature, Western edge	eV
posepi_std	averaged separatrix potential, Western edge	V
nesepm_std	variance for separatrix electron density, outer midplane	m^{-3}
tesepm_std	variance for separatrix electron temperature, outer midplane	eV
tisepm_std	variance for separatrix ion temperature, outer midplane	eV
posepm_std	variance for separatrix potential, outer midplane	V
nesepa_std	variance for separatrix electron density, Eastern edge	m^{-3}
tesepa_std	variance for separatrix electron temperature, Eastern edge	eV
tisepa_std	variance for separatrix ion temperature, Eastern edge	eV
<i>continued on next page</i>		

<i>continued from previous page</i>		
quantity	description	units
posepa_std	variance for separatrix potential, Eastern edge	V
nemxip_std	variance for maximum electron density, Western edge	m^{-3}
temxip_std	variance for maximum electron temperature, Western edge	eV
timxip_std	variance for maximum ion temperature, Western edge	eV
pomxip_std	variance for maximum potential, Western edge	V
nemxap_std	variance for maximum electron density, Eastern edge	m^{-3}
temxap_std	variance for maximum electron temperature, Eastern edge	eV
timxap_std	variance for maximum ion temperature, Eastern edge	eV
pomxap_std	variance for maximum potential, Eastern edge	V

Table E.5: Scalar batch-averaged time dependent quantities stored in b2time.nc

Appendix F

Quantities stored in b2tallies.nc

These are the tallies stored in `b2tallies.nc`. The frequency at which (and whether) these quantities are written out is controlled by the `'tallies_netcdf'` switch.

If `'b2mndr_stim'` is set to a negative value, data from the current **SOLPS-ITER** run will be appended to the existing `b2tallies.nc` file, otherwise the file is created anew.

The meaning of the dimensions are:

vreg volume region tally (see Appendix H and section 3.5.4). Here, however, region 0 stands for the whole plasma.

xreg *x*-directed region tally (see Appendix H)

yreg *y*-directed region tally (see Appendix H)

ns species index

time time index

quantity	dimensions	units	description
b2bremreg	vreg, time	<i>W</i>	Bremsstrahlung radiation rate
b2divua	vreg, time	<i>W</i>	Heating term from ∇u_a
b2divue	vreg, time	<i>W</i>	Heating term from ∇u_e
b2exba	vreg, time	<i>W</i>	Heating term to ions from $\mathbf{E} \times \mathbf{B}$
b2exbe	vreg, time	<i>W</i>	Heating term to electrons from $\mathbf{E} \times \mathbf{B}$
b2fraa	vreg, time	<i>W</i>	Heating from inter-species friction
b2joule	vreg, time	<i>W</i>	Joule heating
b2qie	vreg, time	<i>W</i>	Electron/ion exchange term
b2radreg	vreg, time	<i>W</i>	Line radiation rate
b2she	vreg, time	<i>W</i>	Incoming electron heat sources
b2she0	vreg, time	<i>W</i>	Electron heat sources from <code>b2sihs.F</code>
b2shi	vreg, time	<i>W</i>	Incoming ion heat sources
b2shi0	vreg, time	<i>W</i>	Ion heat sources from <code>b2sihs.F</code>
b2stbc_sch_reg	vreg, time	<i>A</i>	Boundary charge source
b2stbc_she_reg	vreg, time	<i>W</i>	Boundary electron energy source
b2stbc_shi_reg	vreg, time	<i>W</i>	Boundary ion energy source
b2stbc_smo_reg	vreg, ns, time	<i>kg.m.s⁻¹</i>	Boundary parallel momentum source
b2stbc_sna_reg	vreg, ns, time	<i>s⁻¹</i>	Boundary particle source
b2stbc_sne_reg	vreg, time	<i>s⁻¹</i>	Boundary electron source
b2stbr_sch_reg	vreg, time	<i>A</i>	Recycling charge source
b2stbr_she_reg	vreg, time	<i>W</i>	Recycling electron energy source
<i>continued on next page</i>			

continued from previous page			
quantity	dimensions	units	description
b2stbr_shi_reg	vreg, time	W	Recycling ion energy source
b2stbr_smo_reg	vreg, ns, time	$kg.m.s^{-1}$	Recycling parallel momentum source
b2stbr_sna_reg	vreg, ns, time	s^{-1}	Recycling particle source
b2stbr_sne_reg	vreg, time	s^{-1}	Recycling electron source
b2stbm_sch_reg	vreg, time	A	Additional charge source
b2stbm_she_reg	vreg, time	W	Additional electron energy source
b2stbm_shi_reg	vreg, time	W	Additional ion energy source
b2stbm_smo_reg	vreg, ns, time	$kg.m.s^{-1}$	Additional parallel momentum source
b2stbm_sna_reg	vreg, ns, time	s^{-1}	Additional particle source
b2stbm_sne_reg	vreg, time	s^{-1}	Additional electron source
b2sext_sch_reg	vreg, time	A	External charge source
b2sext_she_reg	vreg, time	W	External electron energy source
b2sext_shi_reg	vreg, time	W	External ion energy source
b2sext_smo_reg	vreg, ns, time	$kg.m.s^{-1}$	External parallel momentum source
b2sext_sna_reg	vreg, ns, time	s^{-1}	External particle source
b2sext_sne_reg	vreg, time	s^{-1}	External electron source
b2str	vreg, time	W	Strange term
b2visa	vreg, time	W	Ion viscous heating
b2wrong1	vreg, time	W	Obsolete
b2wrong2	vreg, time	W	Obsolete
b2wrong3	vreg, time	W	Obsolete
fchxreg	xreg, time	A	Poloidal current
fchyreg	yreg, time	A	Radial current
fhexreg	xreg, time	W	Poloidal electron thermal energy flux
fheyreg	yreg, time	W	Radial electron thermal energy flux
fhixreg	xreg, time	W	Poloidal ion thermal energy flux
fhiyreg	yreg, time	W	Radial ion thermal energy flux
fhmxxreg	xreg, ns, time	W	Poloidal parallel kinetic energy flux
fhmxyreg	yreg, ns, time	W	Radial parallel kinetic energy flux
fhpxxreg	xreg, ns, time	W	Poloidal ionization energy flux
fhpxyreg	yreg, ns, time	W	Radial ionization energy flux
fhjxxreg	xreg, time	W	Poloidal electrostatic energy flux
fhjyreg	yreg, time	W	Radial electrostatic energy flux
fhtxxreg	xreg, time	W	Poloidal total energy flux
fhtyreg	yreg, time	W	Radial total energy flux
fnaxreg	xreg, ns, time	s^{-1}	Poloidal particle flux
fnayreg	yreg, ns, time	s^{-1}	Radial particle flux
nareg	vreg, ns, time	m^{-3}	Average ion density $\frac{\int n_a dV}{\int dV}$
ne2reg	vreg, time	m^{-3}	Average weighted density $\frac{\int Z_a^2 n_a dV}{\int dV}$
nereg	vreg, time	m^{-3}	Average electron density $\frac{\int n_e dV}{\int dV}$
nireg	vreg, time	m^{-3}	Average total ion density $\frac{\int n_i dV}{\int dV}$
poreg	vreg, time	eV	Average potential $\frac{\int \Phi dV}{\int dV}$
qconvexreg	xreg, time	W	Poloidally convected electron energy
qconveyreg	yreg, time	W	Radially convected electron energy
qconvixreg	xreg, ns, time	W	Poloidally convected ion energy
qconviyreg	yreg, ns, time	W	Radially convected ion energy
rcxhireg	vreg, ns, time	W	Charge-exchange ion energy source
rcxmreg	vreg, ns, time	$kg.m.s^{-1}$	Charge-exchange ion parallel momentum source
continued on next page			

<i>continued from previous page</i>			
quantity	dimensions	units	description
rcxnareg	vreg, ns, time	s^{-1}	Charge-exchange particle source
rdneureg	vreg, time	W	Total radiation from Eirene neutrals
resco_reg	vreg, ns, time		Continuity equation residuals
reshe_reg	vreg, time	W	Electron heat equation residuals
reshi_reg	vreg, time	W	Ion heat equation residuals
resmo_reg	vreg, ns, time	$kg.m.s^{-1}$	Parallel momentum equations residuals
resmt_reg	vreg, time	$kg.m.s^{-1}$	Total parallel momentum equation residuals
respo_reg	vreg, time	eV	Potential equation residuals
rqahereg	vreg, ns, time	W	Electron cooling rate
rqbrmreg	vreg, ns, time	W	Bremsstrahlung
rqradreg	vreg, ns, time	W	Line radiation
rrahireg	vreg, ns, time	W	Recombination ion heat source
rramoreg	vreg, ns, time	$kg.m.s^{-1}$	Recombination parallel momentum source
rranareg	vreg, ns, time	s^{-1}	Recombination particle source
rsahireg	vreg, ns, time	W	Ionisation ion heat source
rsamoreg	vreg, ns, time	$kg.m.s^{-1}$	Ionisation parallel momentum source
rsanareg	vreg, ns, time	s^{-1}	Ionisation particle source
tereg	vreg, time	eV	Average electron temperature $\frac{\int T_e n_e dV}{\int n_e dV}$
tireg	vreg, time	eV	Average ion temperature $\frac{\int T_i n_i dV}{\int n_i dV}$
volreg	vreg, time	m^3	Total volume $\int dV$

Table F.1: Quantities stored in b2tallies.nc

Appendix G

Quantities printed by 'b2wdat_iout' = '4'

The switch setting 'b2wdat_iout' = '4' allows for a number of output files, developed in Saint Petersburg by Elizaveta Kaveeva (with some technical help from Sergey Voskoboynikov and Ilya Senichenkov), in order to obtain explicitly each term from the equations listed in Appendix C exactly as they are computed in the code.

For this purpose the output of each quantity is produced exactly from the subroutine where this quantity is computed, e.g. the velocity u_a (ua) is printed from the subroutine b2npmo. Each quantity is printed into a separate ASCII file as a 2D table. All these files are stored in the output/ folder inside the run directory. The general naming convention is the following (except for geometry related quantities and a few others):

<subroutine_name>_<quantity>[_<XXX>].dat

Here, <subroutine_name> is the name of subroutine (or sometimes the name of the Fortran source file containing this subroutine) from which the output is produced, while <quantity> is the quantity name, which usually coincides with the name of the corresponding variable inside the subroutine, if such a variable exists. Otherwise some association is used. The species index is denoted here as XXX (three digits), and is used for species-dependent quantities (such as na or ua), and is omitted from common quantities (such as te, ti or po). For example, the file containing the density of the species with index is = 1 would be

b2npco_na_001.dat

The content of the file looks like the following:

	-1	0	1	2	3
36	2.525043E+19	2.517455E+19	1.625484E+19	1.444400E+19	1.407206E+19
35	2.681619E+19	2.673561E+19	1.804678E+19	1.575019E+19	1.516221E+19
34	3.360574E+19	3.350655E+19	2.279144E+19	1.984601E+19	1.907786E+19
33	4.343278E+19	4.330891E+19	2.978478E+19	2.597130E+19	2.498425E+19
32	5.664300E+19	5.649203E+19	3.951054E+19	3.444186E+19	3.309993E+19
31	7.393565E+19	7.375144E+19	5.240743E+19	4.567851E+19	4.384763E+19
30	9.590684E+19	9.568531E+19	6.914344E+19	6.032204E+19	5.781704E+19
29	1.230569E+20	1.227959E+20	9.057685E+19	7.919435E+19	7.575477E+19
28	1.556381E+20	1.553424E+20	1.176512E+20	1.032794E+20	9.859909E+19
27	1.932463E+20	1.929148E+20	1.497150E+20	1.323182E+20	1.263528E+20
26	2.335330E+20	2.331599E+20	1.844608E+20	1.643623E+20	1.576109E+20
25	2.721367E+20	2.716981E+20	2.163190E+20	1.948967E+20	1.888368E+20
24	2.992329E+20	2.987071E+20	2.372826E+20	2.165452E+20	2.127225E+20
23	3.069312E+20	3.063232E+20	2.417565E+20	2.224695E+20	2.209700E+20

22	2.972805E+20	2.966824E+20	2.323948E+20	2.140750E+20	2.134266E+20
21	2.785257E+20	2.779398E+20	2.158066E+20	1.981130E+20	1.971063E+20
20	2.592405E+20	2.586764E+20	1.986355E+20	1.815245E+20	1.801132E+20
19	2.447657E+20	2.442036E+20	1.861651E+20	1.694604E+20	1.678487E+20
18	2.378737E+20	2.373074E+20	1.802841E+20	1.638746E+20	1.621619E+20
17	2.345108E+20	2.339292E+20	1.774551E+20	1.611951E+20	1.594105E+20
16	2.302662E+20	2.297536E+20	1.742077E+20	1.580295E+20	1.561263E+20
15	2.263556E+20	2.258411E+20	1.711906E+20	1.550573E+20	1.530871E+20
14	2.219169E+20	2.214601E+20	1.678177E+20	1.517517E+20	1.497154E+20
13	2.171240E+20	2.166303E+20	1.639932E+20	1.481086E+20	1.459853E+20
12	2.121178E+20	2.116556E+20	1.601201E+20	1.444459E+20	1.422191E+20
11	2.064163E+20	2.059699E+20	1.556076E+20	1.401716E+20	1.378837E+20
10	2.005827E+20	2.000971E+20	1.507593E+20	1.356803E+20	1.333992E+20
9	1.944365E+20	1.940038E+20	1.458187E+20	1.311082E+20	1.288094E+20
8	1.872176E+20	1.868137E+20	1.399629E+20	1.256621E+20	1.233961E+20
7	1.797522E+20	1.793348E+20	1.339149E+20	1.200777E+20	1.178179E+20
6	1.717291E+20	1.713409E+20	1.275705E+20	1.142528E+20	1.119222E+20
5	1.627565E+20	1.623631E+20	1.203216E+20	1.076693E+20	1.053815E+20
4	1.534325E+20	1.530565E+20	1.128591E+20	1.008665E+20	9.870904E+19
3	1.433210E+20	1.429840E+20	1.049081E+20	9.352627E+19	9.151503E+19
2	1.324803E+20	1.321455E+20	9.642263E+19	8.568567E+19	8.379455E+19
1	1.208139E+20	1.205130E+20	8.754242E+19	7.755114E+19	7.573841E+19
0	1.083148E+20	1.080124E+20	7.790863E+19	6.878814E+19	6.712576E+19
-1	1.015473E+20	1.012638E+20	7.310223E+19	6.428731E+19	6.265369E+19

Here the first row (truncated in the example above) gives the poloidal cell index i_x , the first column (to the right) contains the radial cell index i_y , the rest are the values of the 2D array in each cell.

The price for the possibility to output each quantity exactly from the subroutine where this quantity is calculated is the production of output files each time the corresponding subroutine is called. This gives an additional load to the filesystem due to the increased number of file writing operations and slows the computation significantly. Therefore it is NOT recommended to perform long-lasting runs with 'b2wdat_iout' set to '4'. Instead, after a normal run with 'b2wdat_iout' = '0' finishes, it is suggested to set 'b2wdat_iout' = '4' and to continue the run for a small number of time steps, say, between 1 and 10. Then the whole set of output files may be read by any suitable tool (e.g. MATLAB) for further analysis, plotting, comparing, etc.

The table below provides a listing of the various files that may be produced, the variables they contain and the equations to which they relate.

Note that some quantities (like density, velocity, temperature, potentials, magnetic field, metric coefficients, etc.) are computed at cell centers, while others (like particle, energy, and momentum fluxes, currents, and quantities which contribute to them) are computed at cell faces.

quantity	units	file(s)	equation
<i>Main quantities</i>			
n_a	m^{-3}	b2npco_na[XXX].dat or b2npc9_na[XXX].dat or b2npc11_na[XXX].dat	(C.1)
u_a	m/s	b2npmo_ua[XXX].dat	(C.56), (C.57)
T_e	J	b2npht_te.dat or b2nph9_te.dat	(C.98)
T_i	J	b2npht_ti.dat or b2nph9_ti.dat	(C.145)
T_e	eV	te_eV.dat	(C.98)
continued on next page			

<i>continued from previous page</i>			
quantity	units	file(s)	equation
T_i	eV	ti_eV.dat	(C.145)
φ	V	b2nppo_po.dat or b2npp7_po.dat	(C.181)
n_e	m^{-3}	ne.dat	(C.27)
<i>Geometry and related quantities</i>			
$\sqrt{g}$	m^3	vol.dat	
h_x	m	hx.dat	
h_y	m	hy.dat	
h_{y1}	m	hy1.dat	
h_z	m	hz.dat	
B	T	bb.dat	
B_x	T	bbx.dat	
B_z	T	bbz.dat	
R_0	m	crx0.dat	
Z_0	m	crz0.dat	
R_1	m	crx1.dat	
Z_1	m	crz1.dat	
R_2	m	crx2.dat	
Z_2	m	crz2.dat	
R_3	m	crx3.dat	
Z_3	m	crz3.dat	
<i>Continuity equation and related quantities</i>			
$S_a^n \sqrt{g}$	s^{-1}	b2npco_sna[XXX].dat or b2npc9_sna[XXX].dat or b2npc11_sna[XXX].dat	(C.1)
$\left(S_{I_a}^n + S_{I_{a+1}}^n\right) \sqrt{g}$	s^{-1}	b2stel_sna_ion[XXX].dat	(C.31), (C.32)
$\left(S_{R_a}^n + S_{R_{a-1}}^n\right) \sqrt{g}$	s^{-1}	b2stel_sna_rec[XXX].dat	(C.34), (C.35)
$\left(S_{CX_a}^n + S_{CX_{a-1}}^n\right) \sqrt{g}$	s^{-1}	b2stcx_sna_[XXX].dat	(C.37)
$S_a^{\text{Eirene}} \sqrt{g}$	s^{-1}	b2stbr_sna_eir[XXX].dat	
$S_a^{BC} \sqrt{g}$	s^{-1}	b2stbc_phys_sna[XXX].dat	
$-\sqrt{g} \frac{\partial n_a}{\partial t}$	s^{-1}	b2srdt_snadt[XXX].dat	(C.1)
$\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax}$	s^{-1}	b2npc9_fnax[XXX].dat or b2npc11_fnax[XXX].dat	(C.2)
$\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ay}$	s^{-1}	b2npc9_fnay[XXX].dat or b2npc11_fnay[XXX].dat	(C.3)
$\frac{\sqrt{g}}{h_x} b_x V_{ a} n_a$	s^{-1}	b2tfnb_bxuanax[XXX].dat	(C.2), (C.7)
$\frac{\sqrt{g}}{h_x} V_{ax}^{(E)} n_a$	s^{-1}	b2tfnb_vaecrbnax[XXX].dat	(C.13)
$\frac{\sqrt{g}}{h_x} V_{ay}^{(E)} n_a$	s^{-1}	b2tfnb_vaecrbnay[XXX].dat	(C.13)
$\frac{\sqrt{g}}{h_x} V_a^{\text{corr-dpc}} n_a$	s^{-1}	b2tfnb_dpccornax[XXX].dat	(C.29)
$\frac{\sqrt{g}}{h_x} V_{ay}^{(AN)} n_a$	s^{-1}	b2tfnb_cv1bnay[XXX].dat	(C.48)
$-D_{n,a} \frac{\sqrt{g}}{h_x} \frac{1}{h_x} \frac{\partial n_a}{\partial x}$	s^{-1}	b2tfnb_dPat_mdf_gradnax[XXX].dat	(C.50)
<i>continued on next page</i>			

continued from previous page			
quantity	units	file(s)	equation
$-D_{n,a} \frac{\sqrt{g}}{h_y} \frac{1}{h_y} \frac{\partial n_a}{\partial y}$	s^{-1}	b2tfnb_dPat_mdf_gradnay [XXX] .dat	(C.50)
$-\tilde{D}_{p,ax} \frac{\sqrt{g}}{h_x} \frac{1}{h_x} \frac{\partial p_a}{\partial x}$	s^{-1}	b2tfnb_dgradpbx [XXX] .dat	(C.53)
$-\tilde{D}_{p,ay} \frac{\sqrt{g}}{h_y} \frac{1}{h_y} \frac{\partial p_a}{\partial y}$	s^{-1}	b2tfnb_dgradpby [XXX] .dat	(C.54)
$\frac{\sqrt{g}}{h_x} \tilde{\Gamma}_{ax}^{(PSch)} \text{ }^1$	s^{-1}	b2tfnb_fnbPSchx [XXX] .dat	(C.49)
$\frac{\sqrt{g}}{h_y} \tilde{\Gamma}_{ay}^{(PSch)} \text{ }^2$	s^{-1}	b2tfnb_fnbPSchy [XXX] .dat	(C.49)
$D_{n,a} \frac{\sqrt{g}}{h_x^2}$	m^3/s	b2trno_cdnax [XXX] .dat	(C.50)
$D_{n,a} \frac{\sqrt{g}}{h_y^2}$	m^3/s	b2trno_cdnay [XXX] .dat	(C.50)
$\tilde{D}_{ax}^p \frac{\sqrt{g}}{h_x^2}$	$m \cdot s/kg$	b2trno_cdpax [XXX] .dat	(C.53)
$\tilde{D}_{ay}^p \frac{\sqrt{g}}{h_y^2}$	$m \cdot s/kg$	b2trno_cdpay [XXX] .dat	(C.54)
$D_{n,a}^{AN}$	m^2/s	b2tqna_dna0 [XXX] .dat	(C.50)
D_a^p	m^2/s	b2tqna_dpa0 [XXX] .dat	(C.55)
$V_{ax}^{(E)}$	m/s	b2tfnb_vbecrbx001.dat	(C.44)
$V_{ay}^{(E)}$	m/s	b2tfnb_vbecrby001.dat	(C.45)
$W_x^{(dia)}$	m/s	b2tfnb_wbdiax [XXX] .dat	(C.25)
$W_y^{(dia)}$	m/s	b2tfnb_wbdiax [XXX] .dat	(C.25)
$\tilde{V}_{ax}^{(dia)}$	m/s	b2tfnb_vbdiax [XXX] .dat	(C.46)
$\tilde{V}_{ay}^{(dia)}$	m/s	b2tfnb_vbdiax [XXX] .dat	(C.47)
$\frac{\sqrt{g}}{h_x} \tilde{V}_{ax}^{(dia)} n_a$	s^{-1}	b2tfnb_vadianax [XXX] .dat	(C.18)
$\frac{\sqrt{g}}{h_x} \tilde{V}_{ay}^{(dia)} n_a$	s^{-1}	b2tfnb_vadianay [XXX] .dat	(C.18)
$E_{k,a}$	J	b2tfnb_kbnrgy [XXX] .dat	(C.24)
$V_{ay}^{(AN)}$	m/s	b2tqna_vla0y [XXX] .dat	(C.48)
Γ_{ax}	s^{-1}	b2tfnb_fnbx [XXX] .dat	(C.7)
Γ_{ay}	s^{-1}	b2tfnb_fnby [XXX] .dat	(C.8)
residuals	s^{-1}	b2npco_resco [XXX] .dat or b2npc9_resco [XXX] .dat or b2npc11_resco [XXX] .dat	
Momentum balance equation and related quantities			
$V_{ a}$	m/s	b2npmo_ua [XXX] .dat	(C.56), (C.57)
$S_a^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smb [XXX] .dat	(C.58)
$-m_a \frac{\partial n_a}{\partial t} V_{ a} h_z \sqrt{g}$	$N \cdot m$	b2srdt_smodt [XXX] .dat	(C.56), (C.57)
$\frac{h_z \sqrt{g}}{h_x} \Gamma_{ax}^m$	$N \cdot m$	b2npmo_fmox [XXX] .dat	(C.85),
$\frac{h_z \sqrt{g}}{h_y} \Gamma_{ay}^m$	$N \cdot m$	b2npmo_fmoy [XXX] .dat	(C.86),
$-\frac{b_x}{h_x} \frac{\partial n_a T_i}{\partial x} h_z \sqrt{g}$	$N \cdot m$	b2sigp_smogpi [XXX] .dat	(C.56), (C.57)
continued on next page			

¹Mind the sign! It is opposite in the output file and the flux balance.

²Mind the sign! It is opposite in the output file and the flux balance.

continued from previous page			
quantity	units	file(s)	equation
$-Z_a e n_a \frac{b_x}{h_x} \frac{\partial \Phi}{\partial x} h_z \sqrt{g}$	$N \cdot m$	b2sigp_smogpo[XXX].dat	(C.56)
$S_{CX_a}^m h_z \sqrt{g}$	$N \cdot m$	b2stcx_smq[XXX].dat	(C.70)
$S_{CF}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smocf[XXX].dat	(C.62), (C.64)
$S_{I_a}^m h_z \sqrt{g}$	$N \cdot m$	b2stel_smq_ion[XXX].dat	(C.58), (C.64)
$S_{R_a}^m h_z \sqrt{g}$	$N \cdot m$	b2stel_smq_rec[XXX].dat	(C.58), (C.64)
$S_{Therm,ia}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smotfia[XXX].dat	(C.79)
$S_{Therm,ea}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smotfea[XXX].dat	(C.77)
$S_{fr,ea}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smofrea[XXX].dat	(C.78)
$S_{fr,ia}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smofria[XXX].dat	(C.80)
$S_{AN_a}^m h_z \sqrt{g}$	$N \cdot m$	b2npmo_smoan[XXX].dat	(C.63)
$S_{BC_a}^m h_z \sqrt{g}$	$N \cdot m$	b2stbc_phys_smo[XXX].dat	
$\left(\nabla \cdot \frac{\leftrightarrow V_{\parallel}}{\pi_a} \right)_{\parallel}^{res} h_z \sqrt{g}$	$N \cdot m$	b2npmo_smovv001.dat	(C.59), (C.60)
$\left(\nabla \cdot \frac{\leftrightarrow q_{\parallel}}{\pi_a} \right)_{\parallel} h_z \sqrt{g}$	$N \cdot m$	b2npmo_smovh001.dat	(C.59), (C.61)
$\tilde{\eta}_{ax}^{(CL)} \frac{h_z \sqrt{g}}{h_x^2}$	$kg \cdot m/s$	b2trcl_luciani_fllim_cvsahzx[XXX].dat	(C.90)
$\tilde{\eta}_{ax}^{(CL)} \frac{h_z \sqrt{g}}{h_x^2}$	$kg \cdot m/s$	b2trcl_luciani_cvsahzx[XXX].dat	(C.93)
$\eta_a^{(AN)} \frac{h_z \sqrt{g}}{h_x^2}$	$kg \cdot m/s$	b2trno_cvsahzx[XXX].dat	(C.88)
$\eta_a^{(AN)} \frac{h_z \sqrt{g}}{h_x^2}$	$kg \cdot m/s$	b2trno_cvsahzy[XXX].dat	(C.88)
$\eta_{ax}^{CL} \frac{h_z \sqrt{g}}{h_x^2}$	$kg \cdot m/s$	b2trcl_cvsahzx[XXX].dat	(C.94)
τ_a	s	b2tqca_taua[XXX].dat	(C.95)
$\frac{h_z \sqrt{g}}{h_x} \Gamma_{ax}^{Corr}$	m^2/s	b2tfnb_fnb_fcorx[XXX].dat	(C.9)
$\frac{h_z \sqrt{g}}{h_x} \Gamma_{ay}^{Corr}$	m^2/s	b2tfnb_fnb_fcory[XXX].dat	(C.10)
$-D_a^n \frac{1}{h_x} \frac{\partial n_a}{\partial x} \frac{h_z \sqrt{g}}{h_x}$	m/s	b2tfnb_dPat_2diagradnax[XXX].dat	(C.9)
ν_a^{visc}	m^2/s	b2tqna_vsa0[XXX].dat	(C.88)
$-\frac{h_z \sqrt{g}}{h_x} \eta_{ax} \frac{\partial V_{\parallel a}}{h_x \partial x}$	kg/s	b2urmo_etaPat_graduax[XXX].dat	(C.85)
$-\frac{h_z \sqrt{g}}{h_y} \eta_{ay} \frac{\partial V_{\parallel a}}{h_y \partial y}$	$N \cdot m$	b2urmo_etaPat_graduay[XXX].dat	(C.86)
$\frac{h_z \sqrt{g}}{h_x} \left(\frac{4}{3} \tilde{\eta}_{ax}^{(CL)} \frac{\partial \ln h_z}{h_x \partial x} \right)$	$kg \cdot m/s$	b2npmo_flubvx001.dat	(C.85)
residuals	$N \cdot m$	b2npmo_resmo[XXX].dat	
Electron heat balance equation and related quantities			
T_e	J	b2npht_te.dat or b2nph9_te.dat	(C.98)
$\frac{\sqrt{g}}{h_x} \tilde{q}_{ex}$	W	b2nph9_fhex.dat	(C.105)
$\frac{\sqrt{g}}{h_y} \tilde{q}_{ey}$	W	b2nph9_fhey.dat	(C.136)
$S_e^{heat} \sqrt{g}$	W	b2npht_she.dat or b2nph9_she.dat	(C.99)
$Q_R \sqrt{g}$	W	b2stel_she_rad.dat	(C.102)
continued on next page			

continued from previous page			
quantity	units	file(s)	equation
$Q_{\Delta}\sqrt{g}$	W	b2npht_shei.dat or b2nph9_shei.dat	(C.100)
$Q_e^{\text{EIRENE}}\sqrt{g}$	W	b2stbr_she_eir.dat	(C.99)
$Q_{Fei}\sqrt{g}$	W	b2sihs_shefr.dat	(C.104)
$\frac{n_e T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} b_x V_{e\parallel} \right) \sqrt{g}$	W	b2sihs_shedu.dat	(C.98)
$n_e T_e \left(\nabla \cdot \vec{V}^E \right) \sqrt{g}$	W	b2sihs_shedd.dat	(C.98)
$Q_e^{BC}\sqrt{g}$	W	b2stbc_phys_she.dat	
$-\frac{j_y^{(ST)}}{en_e} \frac{\partial n_e T_e}{h_y \partial y} \sqrt{g}$	W	b2sihs_shest.dat	(C.98)
$-\frac{3}{2} \frac{\sqrt{g}}{\partial t} \sqrt{g}$	W	b2srdt_shedt.dat	(C.98)
$\frac{3}{2} \frac{\sqrt{g}}{h_x} \Gamma_{ex} T_e$	W	b2tfhe_qe_32x.dat	(C.109)
$\frac{3}{2} \frac{\sqrt{g}}{h_y} \Gamma_{ey} T_e$	W	b2tfhe_qe_32y.dat	(C.139)
$\frac{\sqrt{g}}{h_x} q_{ex}^{P.Sch\ 3}$	W	b2tfhe_fhePSchx.dat	(C.107)
$\frac{\sqrt{g}}{h_y} q_{ey}^{P.Sch\ 4}$	W	b2tfhe_fhePSchy.dat	(C.137)
$-\frac{\sqrt{g}}{h_x} \kappa_{ex} \frac{1}{h_x} \frac{\partial T_e}{\partial x}$	W	b2tfhe_qe_ke_gTx.dat	(C.108)
$-\frac{\sqrt{g}}{h_y} \kappa_{ey} \frac{1}{h_y} \frac{\partial T_e}{\partial y}$	W	b2tfhe_qe_ke_gTy.dat	(C.138)
$-\tilde{c}_e^{(2)} \frac{j_x^{(\parallel)}}{e} T_e^5$	W	b2tfhe_qe_fchpTex.dat	(C.106)
$\alpha_{ey} \hat{E}_y T_e$	W	b2tfhe_qe_alphaTehy.dat	(C.141)
$\Gamma_{ax}^{(he)} \frac{\sqrt{g}}{h_x}$	s^{-1}	b2tfnb_fnb_hex.dat	(C.110)
$\Gamma_{ay}^{(he)} \frac{\sqrt{g}}{h_y}$	s^{-1}	b2tfnb_fnb_hey.dat	(C.140)
$\kappa_{ex} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2tfhe_chcex_Pat.dat	(C.116)
$\kappa_{ey} \frac{\sqrt{g}}{h_y^2}$	s^{-1}	b2tfhe_chcey_Pat.dat	(C.142)
$\kappa_e^{(AN)} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trno_chcex.dat	(C.117)
$\kappa_e^{(AN)} \frac{\sqrt{g}}{h_y^2}$	s^{-1}	b2trno_chcey.dat	(C.117)
$\chi_e^{(AN)}$	m^2/s	b2tqna_hce0.dat	(C.117)
$\tilde{\kappa}_{ex}^{CL} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trcl_luciani_fllim_chcex.dat	(C.56)
$\tilde{\kappa}_{ex}^{CL} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trcl_luciani_chcex.dat	(C.119)
$\kappa_{ex}^{CL} \frac{\sqrt{g}}{h_x^2} b_x^2$	s^{-1}	b2trcl_chcex.dat	(C.121)
τ_e	s	b2tqce_tau_e.dat	(C.123)
continued on next page			

³Mind the sign! It is opposite in the output file and the heat balance.

⁴Mind the sign! It is opposite in the output file and the heat balance.

⁵If the switch b2tfhe_alphaTeh = 1 (and b2tfhe_fchpTe = 0), then this output item contains $\alpha_{ex} \hat{E}_x T_e$

continued from previous page			
quantity	units	file(s)	equation
$V_{e\parallel}$	m/s	b2sifrtf_ue.dat	(C.124)
$\sigma_{\parallel} \frac{\sqrt{g}}{h_x^2}$	$\frac{1}{m \cdot S}$	b2trcl_luciani_csigx.dat	(C.125), (C.127)
$\sigma_{CL\parallel} \frac{\sqrt{g}}{h_x^2}$	$\frac{1}{m \cdot S}$	b2trcl_csigx.dat	(C.129)
$\alpha_{ex} \frac{\sqrt{g}}{h_x^2}$	$\frac{m}{V \cdot s}$	b2trcl_luciani_fllim_calfx.dat	(C.130)
$\tilde{\alpha}_{CLx} \frac{\sqrt{g}}{h_x^2}$	$\frac{m}{V \cdot s}$	b2trcl_lucian_calfx.dat	(C.133), (C.135)
$\alpha_{CLx} \frac{\sqrt{g}}{h_x^2}$	$\frac{m}{V \cdot s}$	b2trcl_calfx.dat	(C.131)
$\alpha^{(ST)}$	??	b2trno_chcey_stoch.dat	(C.143)
$-\frac{5}{2} \frac{\sqrt{g}}{h_x} n_e T_e^2 \frac{B_z}{eh_y} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right)$	W	b2tfhe__qediax.dat	(C.108)
$\frac{5}{2} \frac{\sqrt{g}}{h_y} n_e T_e^2 \frac{B_z}{eh_x} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right)$	W	b2tfhe__qediax.dat	(C.138)
$r_{Q_a}^{\text{brm}} \sqrt{g}$	W	b2stel_rqbrm[XXX]	(C.103)
$r_{Q_a}^{\text{line}} \sqrt{g}$	W	b2stel_rqrad[XXX]	(C.103)
residuals	W	b2npht_reshe.dat or b2nph9_reshe.dat	
Ion heat balance equation and related quantities			
T_i	W	b2npht_shi.dat or b2nph9_shi.dat	(C.145)
$S_i^{\text{heat}} \sqrt{g}$	W	b2npht_shi.dat or b2nph9_shi.dat	(C.1)
$-\frac{3}{2} \sqrt{g} \frac{\partial n_i T_i}{\partial t}$	W	b2srdt_shidt.dat	(C.145)
$\frac{\sqrt{g}}{h_x} \tilde{q}_{ix}$	W	b2nph9_fhix.dat	(C.153)
$\frac{\sqrt{g}}{h_y} \tilde{q}_{iy}$	W	b2nph9_fhiy.dat	(C.174)
$\sum_{a=0}^{ns-1} n_a T_i \left(\nabla \cdot \vec{V}^E \right) \sqrt{g}$	W	b2sihs__shidd.dat	(C.146)
$-\sum_{a=0}^{ns-1} n_a T_i \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} V_{a\parallel} b_x \right) \sqrt{g}$	W	b2sihs__shidu.dat	(C.145), (C.146)
$\sum_{\text{fluid species}} \eta_{ax} \left(\frac{\partial V_{a\parallel}}{h_x \partial x} \right)^2 \sqrt{g}$	W	b2sihs__shivc.dat	(C.146)
$\sum_{\text{fluid species}} \eta_a^{(AN)} \left(\frac{\partial V_{a\parallel}}{h_y \partial y} \right)^2 \sqrt{g}$	W	b2sihs__shiva.dat	(C.146)
$Q_{Fab} \sqrt{g}$	W	b2sihs__shifr.dat	(C.147), (C.148)
$Q_I^{(i)} \sqrt{g}$	W	b2stel_shi_ion.dat	(C.149)
$Q_R^{(i)} \sqrt{g}$	W	b2stel_shi_rec.dat	(C.146), (C.151)
$Q_i^{\text{EIRENE}} \sqrt{g}$	W	b2stel_shi_eir.dat	(C.146)
$-\frac{3}{2} \frac{\sqrt{g}}{h_x} \Gamma_{ix} T_i$	W	b2tfhi__qi_32x.dat	(C.156)
$-\frac{3}{2} \frac{\sqrt{g}}{h_y} \Gamma_{iy} T_i$	W	b2tfhi__qi_32y.dat	(C.156)
continued on next page			

continued from previous page			
quantity	units	file(s)	equation
$\frac{\sqrt{g}}{h_x} q_{ix}^{P.Sch}$	W	b2tfhi__fhiPSchx.dat	(C.154)
$\frac{\sqrt{g}}{h_y} q_{iy}^{P.Sch}$	W	b2tfhi__fhiPSchy.dat	(C.175)
$-\frac{\sqrt{g}}{h_x} \kappa_{ix} \frac{1}{h_x} \frac{\partial T_i}{\partial x}$	W	b2tfhi__qi_ki_gTx.dat	(C.153)
$-\frac{\sqrt{g}}{h_y} \kappa_{iy} \frac{1}{h_y} \frac{\partial T_i}{\partial y}$	W	b2tfhi__qi_ki_gTy.dat	(C.174)
$\chi_a^{(AN)}$	m^2/s	b2tqna_hcib[XXX].dat	(C.162), (C.163)
$\kappa_{ix} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2tfhi__chcix_Pat.dat	(C.159)
$\kappa_{iy} \frac{\sqrt{g}}{h_y^2}$	s^{-1}	b2tfhi__chciy_Pat.dat	(C.178),
$\kappa_{ix}^{(AN)} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trno_chcix.dat	(C.159)
$\kappa_{iy}^{(AN)} \frac{\sqrt{g}}{h_y^2}$	s^{-1}	b2trno_chciy.dat	(C.159)
$\tilde{\kappa}_{ix}^{(CL)} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trcl_luciani_fllim_chci.dat	(C.164)
$\tilde{\kappa}_{ix}^{(CL)} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trcl_luciani_chcix.dat	(C.167), (C.172)
$\kappa_{ix}^{(CL)} \frac{\sqrt{g}}{h_x^2}$	s^{-1}	b2trcl_chcix.dat	(C.171)
$\frac{5}{2} \frac{\sqrt{g}}{h_x} T_i \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a \tilde{V}_{ax}^{(dia)}$	W	b2tfhi__qidiax.dat	(C.155)
$\frac{5}{2} \frac{\sqrt{g}}{h_y} T_i \sum_{\substack{a=0 \\ z_a \neq 0}}^{ns-1} n_a \tilde{V}_{ay}^{(dia)}$	W	b2tfhi__qidiax.dat	(C.176)
residuals	W	b2npht_reshi.dat or b2nph9_reshi.dat	
Current continuity equation and related quantities			
φ	V	b2nppo_po.dat or b2npp7_po.dat	(C.181)
$\frac{\sqrt{g}}{h_x} j_x$	A	b2nppo_fchx.dat or b2npp7_fchx.dat	(C.181), (C.182)
$\frac{\sqrt{g}}{h_y} j_y$	A	b2nppo_fchy.dat or b2npp7_fchy.dat	(C.181), (C.191)
$\frac{\sqrt{g}}{h_x} j_x^{()}$	A	b2tfhe__fch_px.dat	(C.125), (C.182)
$\frac{\sqrt{g}}{h_x} \tilde{j}_x^{(dia)}$	A	b2tfhe__fchdiax.dat	(C.184)
$\frac{\sqrt{g}}{h_y} \tilde{j}_y^{(dia)}$	A	b2tfhe__fchdiay.dat	(C.193)
$\frac{\sqrt{g}}{h_x} \tilde{j}_x^{(in)}$	A	b2tfhe__fchinertx.dat	(C.185)
$\frac{\sqrt{g}}{h_y} \tilde{j}_y^{(in)}$	A	b2tfhe__fchinerty.dat	(C.194)
continued on next page			

continued from previous page			
quantity	units	file(s)	equation
$\frac{\sqrt{g} \tilde{j}_x^{(vis\parallel)}}{h_x}$	A	b2tfhe__fchvisparx.dat	(C.186)
$\frac{\sqrt{g} \tilde{j}_y^{(vis\parallel)}}{h_y}$	A	b2tfhe__fchvispary.dat	(C.195)
$\frac{\sqrt{g} \tilde{j}_x^{(s)}}{h_x}$	A	b2tfhe__fchinx.dat	(C.182), (C.188)
$\frac{\sqrt{g} \tilde{j}_y^{(s)}}{h_y}$	A	b2tfhe__fchiny.dat	(C.191), (C.199)
$\sigma_{aa0} b_z^2 \frac{\sqrt{g}}{h_x^2}$	S	b2tral_csiginx01_00.dat	(C.189)
$\frac{\sqrt{g} \tilde{j}_x^{(AN)}}{h_x}$	A	b2tfhe__fchanmlx.dat	(C.183)
$\frac{\sqrt{g} \tilde{j}_y^{(AN)}}{h_y}$	A	b2tfhe__fchanmly.dat	(C.192)
σ_{ANy}	S/m ²	b2tqna_sig0.dat	(C.203)
$\frac{\sqrt{g} \tilde{j}_y^{(vis\perp)}}{h_y}$	A	b2tfhe__fchvispery.dat	(C.196)
$\frac{\sqrt{g} \tilde{j}_y^{(visq)}}{h_y}$	A	b2tfhe__fchvisqy.dat	(C.198)
$\frac{\sqrt{g} \tilde{j}_y^{(ST)}}{h_y}$	A	b2tfhe__fchstochy.dat	(C.200)
$\frac{\sqrt{g}}{h_y^2} \sigma^{(ST)}$	S	b2trno_csigy_stoch.dat	(C.200)
residuals	A	b2nppo_respo.dat or b2npp7_respo.dat	(C.181)

Table G.1: Files produced by means of 'b2wdat_iout = 4', the variable they contain, and the equation to which they are related

It may be convenient to build a total energy flux $\tilde{q}^{\text{ENERGY}}$ (cf. Eq. C.231) and many other quantities which contribute to it from the quantities listed in the table G.1. Note that since the energy flux is computed at cell faces, it is necessary to compute it from the quantities also computed at cell faces.

$$q_x^{\text{ENERGY}} = q_{ex}^{\text{ENERGY}} + q_{ix}^{\text{ENERGY}} \quad (\text{G.1})$$

$$q_y^{\text{ENERGY}} = q_{ey}^{\text{ENERGY}} + q_{iy}^{\text{ENERGY}} \quad (\text{G.2})$$

$$q_{ex}^{\text{ENERGY}} = \tilde{q}_{ex} + q_{ex}^{P.Sch} + q_{ex}^{dia} + \frac{2}{3} (q_{ex}^{ExB} + q_{ex}^{u\parallel} + q_{ex}^{uDPC}) \quad (\text{G.3})$$

$$q_{ix}^{\text{ENERGY}} = \tilde{q}_{ix} + q_{ix}^{P.Sch} + q_{ix}^{dia} + \frac{2}{3} (q_{ix}^{ExB} + q_{ix}^{u\parallel} + q_{ix}^{uDPC}) + q_{ix}^j + q_{ix}^{conv} \quad (\text{G.4})$$

$$q_{ey}^{\text{ENERGY}} = \tilde{q}_{ey} + q_{ey}^{P.Sch} + q_{ey}^{dia} + \frac{2}{3} q_{ey}^{ExB} \quad (\text{G.5})$$

$$q_{iy}^{\text{ENERGY}} = \tilde{q}_{iy} + q_{iy}^{P.Sch} + q_{iy}^{dia} + \frac{2}{3} q_{iy}^{ExB} + q_{iy}^{conv} + q_{iy}^j \quad (\text{G.6})$$

$$\tilde{q}_{ex}^{dia} = -\frac{5}{2}T_e^{(\text{face})} \sum_a \frac{z_a^2 T_e^{(\text{face})}}{T_i^{(\text{face})}} \tilde{V}_{ax}^{(dia)} n_a \quad (\text{G.7})$$

$$\tilde{q}_{ey}^{dia} = -\frac{5}{2}T_e^{(\text{face})} \sum_a \frac{z_a^2 T_e^{(\text{face})}}{T_i^{(\text{face})}} \tilde{V}_{ay}^{(dia)} n_a \quad (\text{G.8})$$

$$\tilde{q}_{ix}^{dia} = \frac{5}{2}T_i^{(\text{face})} \sum_a \tilde{V}_{ax}^{(dia)} n_a \quad (\text{G.9})$$

$$\tilde{q}_{iy}^{dia} = \frac{5}{2}T_i^{(\text{face})} \sum_a \tilde{V}_{ay}^{(dia)} n_a \quad (\text{G.10})$$

Here and below the superscript '(face)' means that the temperature (or other quantity) should be computed at the cell face as a half-sum its neighboring cell-centered values. Since $\frac{\sqrt{g}}{h_x} \tilde{V}_{ax}^{(dia)} n_a$ and $\frac{\sqrt{g}}{h_y} \tilde{V}_{ay}^{(dia)} n_a$ are the particle fluxes through the cell faces, $\frac{\sqrt{g}}{h_x}$ and $\frac{\sqrt{g}}{h_y}$ are the cell face areas, and these quantities are computed at cell faces, we need to compute temperatures at cell faces too to get heat flux densities correctly computed at cell faces. A similar approach is applied below in all equations within this Appendix.

$$q_{ex}^{ExB} = \frac{3}{2}T_e^{(\text{face})} \sum_a z_a n_a V_{ax}^E \quad (\text{G.11})$$

$$q_{ey}^{ExB} = \frac{3}{2}T_e^{(\text{face})} \sum_a z_a n_a V_{ay}^E \quad (\text{G.12})$$

$$q_{ix}^{ExB} = \frac{3}{2}T_i^{(\text{face})} \sum_a n_a V_{ax}^E \quad (\text{G.13})$$

$$q_{iy}^{ExB} = \frac{3}{2}T_i^{(\text{face})} \sum_a n_a V_{ay}^E \quad (\text{G.14})$$

$$q_{ex}^{u_{\parallel}} = \frac{3}{2}T_e^{(\text{face})} \left(\sum_a z_a b_x n_a V_{\parallel a} - \frac{j_x^{(\parallel)}}{e} \right) \quad (\text{G.15})$$

$$q_{ix}^{u_{\parallel}} = \frac{3}{2}T_i^{(\text{face})} \sum_a b_x n_a V_{\parallel a} \quad (\text{G.16})$$

$$q_{ex}^{uDPC} = \frac{3}{2}T_e^{(\text{face})} \sum_a z_a n_a V_a^{\text{corr.dpc}} \quad (\text{G.17})$$

$$q_{ix}^{uDPC} = \frac{3}{2}T_i^{(\text{face})} \sum_a n_a V_a^{\text{corr.dpc}} \quad (\text{G.18})$$

$$q_{ix}^{conv} = \sum_a \frac{m_a \left(V_{\parallel a}^{(\text{face})} \right)^2}{2} \left(\tilde{\Gamma}_{ax} + \tilde{\Gamma}_{ax}^{(PSch)} + \tilde{V}_{ax}^{(dia)} n_a \right) \quad (\text{G.19})$$

$$q_{iy}^{conv} = \sum_a \frac{m_a \left(V_{\parallel a}^{(\text{face})} \right)^2}{2} \left(\tilde{\Gamma}_{ay} + \tilde{\Gamma}_{ay}^{(PSch)} + \tilde{V}_{ay}^{(dia)} n_a \right) \quad (\text{G.20})$$

$$q_{ix}^j = \frac{T_i^{(\text{face})}}{e} \sum_a \left(j_x^{(AN)} + j_x^{(in)} + \tilde{j}_x^{(vis\parallel)} + \tilde{j}_x^{(vis\perp)} + \tilde{j}_x^{(visq)} \right) \quad (\text{G.21})$$

$$q_{iy}^j = \frac{T_i^{(\text{face})}}{e} \sum_a \left(j_y^{(AN)} + j_y^{(in)} + \tilde{j}_y^{(vis\parallel)} + \tilde{j}_y^{(vis\perp)} + \tilde{j}_y^{(visq)} \right) \quad (\text{G.22})$$

Additionally one can consider a flux of energy in the form of potential energy of ionization. This energy is spent by electrons at the place of ionization and released back at the place of recombination (or is transferred to the targets thus contributing to the target heat loads). The associated energy flux is

$$q_x^{ioniz} = \sum_a \left(\tilde{\Gamma}_{ax} + \tilde{\Gamma}_{ax}^{(PSch)} + \tilde{V}_{ax}^{(dia)} n_a \right) E_a^{pot} \quad (\text{G.23})$$

$$q_y^{ioniz} = \sum_a \left(\tilde{\Gamma}_{ax} + \tilde{\Gamma}_{ax}^{(PSch)} + \tilde{V}_{ax}^{(dia)} n_a \right) E_a^{pot} \quad (\text{G.24})$$

with

$$E_a^{pot} = \sum_{i=1}^{z_a} I_a^{ioniz} \quad (\text{G.25})$$

the sum of the ionization potentials I_a^{ioniz} of all ions with a charge state below the considered ion (inclusive) within their isonuclear sequence.

Appendix H

Region definitions

Three types of regions are defined in `b2agfs.F`: volume, x -directed and y -directed. The number of volume regions should also be stored in the `NREG` environment variable (or through the presence of an `NREG` ASCII file containing only the number of regions) as it is used for a variety of analysis scripts. Typical configurations for 1-D, limiter, SN, DN (Connected and Disconnected) and periodic stellarator island cases are shown below:

H.1 Regions for 1-D cases

For this geometry, NREG should be set to 1.

Volume regions:

```
+-----+
|               |
|               1               |
|               |
+-----+
```

region 1 === plasma

X-directed regions:

```
+-----+
|               |
|               2               |
|               1               |
+-----+
```

region 1 === Western edge

region 2 === Eastern edge

Y-directed regions:

```
+-----2-----+
|               |
|               |
|               |
+-----1-----+
```

region 1 === core boundary

region 2 === SOL boundary

In cases where the problem of interest has only one target and a plasma entry surface, the target should be located to the right of the domain and the plasma should enter from the left.

An informative review of how the code geometry and physics model can be adapted to linear devices is given in [\[56\]](#).

H.2 Regions for limiter cases

For this geometry, NREG should be set to 2.

Volume regions:

```

+-----+
|               |
|               2 |
|               |
+/------\+
||         ||
||               1 ||
||         ||
++-----++

```

```

region 1 === core
region 2 === SOL

```

X-directed regions:

```

+-----+
|               |
|               2 |
|               |
+/------\+
||         ||
||3         ||
||         ||
++-----++

```

```

region 1 === left target
region 2 === right target
region 3 === connection between both sides of core

```

Y-directed regions:

```

+-----3-----+
|               |
|               |
|               |
+/------2-----\+
||         ||
||         ||
||         ||
++-----1-----++

```

```

region 1 === core boundary
region 2 === separatrix
region 3 === SOL boundary

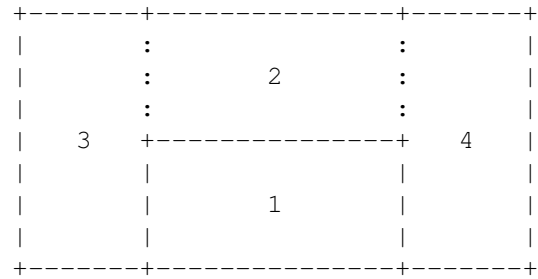
```

H.3 Regions for SN

For this geometry, NREG should be set to 4 (default).

Volume regions:

See also section 3.5.4 for a specific example.



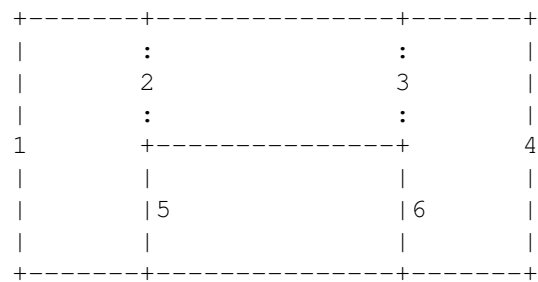
region 1 === core

region 2 === SOL

region 3 === Western divertor (inboard for LSN, outboard for USN)

region 4 === Eastern divertor (outboard for LSN, inboard for USN)

X-directed regions:



region 1 === Western target (inboard for LSN, outboard for USN)

region 2 === entrance to Western divertor

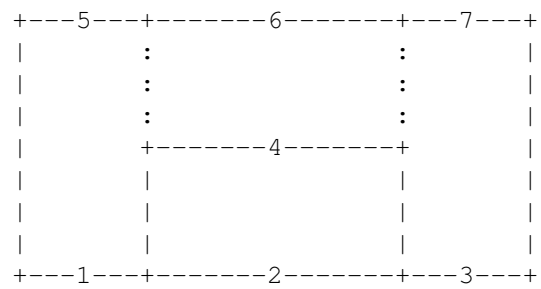
region 3 === entrance to Eastern divertor

region 4 === Eastern target (outboard for LSN, inboard for USN)

region 5 === connection between both sides of core

region 6 === connection between both sides of divertor

Y-directed regions:



region 1 == Western divertor private flux wall boundary (inboard for LSN, outboard for USN)
region 2 == core boundary
region 3 == Eastern divertor private flux wall boundary (outboard for LSN, inboard for USN)
region 4 == separatrix
region 5 == Western divertor to main chamber wall boundary (inboard for LSN, outboard for USN)
region 6 == SOL to main chamber wall boundary
region 7 == Eastern divertor to main chamber wall boundary (outboard for LSN, inboard for USN)

H.4 Regions for CDN

For this geometry, NREG should be set to 8.

Volume regions:

	:	:		:	:	
	:	2	:	:	6	:
	:	:		:	:	
	3	+	4		7	+
		1			5	
+	+	+	+	+	+	+

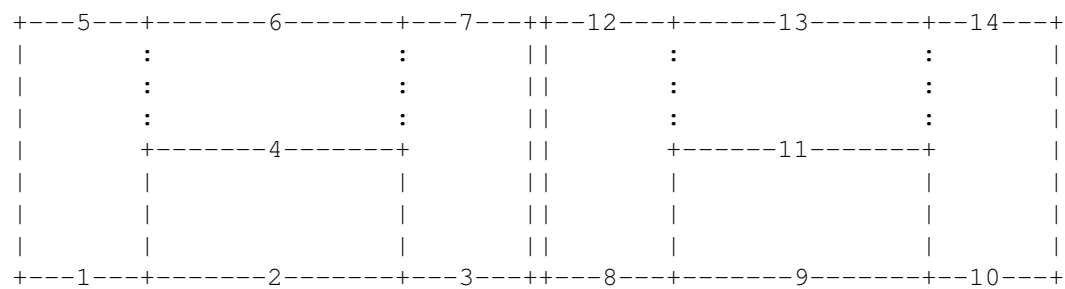
region 1 === left core
region 2 === left SOL
region 3 === left inboard divertor
region 4 === left outboard divertor
region 5 === right core
region 6 === right SOL
region 7 === right inboard divertor
region 8 === right outboard divertor

X-directed regions:

	:	:		:	:	
	2	3		6	7	
	:	:		:	:	
1	+	45	+	8		
	9	10		11	12	
+	+	+	+	+	+	+

region 1 === left inboard divertor target
region 2 === entrance to left inboard divertor
region 3 === entrance to left outboard divertor
region 4 === left outboard target
region 5 === right inboard target
region 6 === entrance to right inboard divertor
region 7 === entrance to right outboard divertor
region 8 === right outboard target
region 9 === connection between left inboard and right outboard core
region 10 === connection between left outboard and right inboard private flux region
region 11 === connection between right inboard and left outboard core
region 12 === connection between right outboard and left inboard private flux region

Y-directed regions:



region 1 === left inboard divertor private flux wall boundary
region 2 === left core boundary
region 3 === left outboard divertor private flux wall boundary
region 4 === left separatrix
region 5 === left inboard divertor main chamber wall boundary
region 6 === left SOL main chamber wall boundary
region 7 === left outboard divertor main chamber wall boundary
region 8 === right inboard divertor private flux wall boundary
region 9 === right core boundary
region 10 === right outboard divertor private flux wall boundary
region 11 === right separatrix
region 12 === right inboard divertor main chamber wall boundary
region 13 === right SOL main chamber wall boundary
region 14 === right outboard divertor main chamber wall boundary

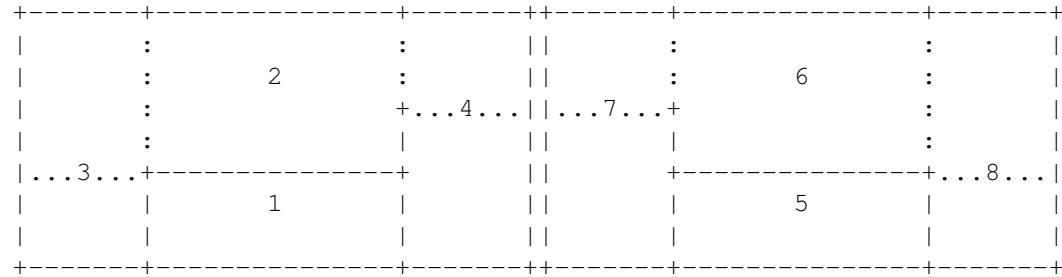
H.5 Regions for DDN

The case presented below corresponds to one where the active (or innermost) X-point is located at the bottom of the plasma (i.e. a DND case). For the opposite DNU configuration, the only change is the relative radial positions of the inner divertor boundaries between SOL-connected and PFR-connected halves and the position of the x -directed region 13 moves to being between x -directed regions 9 and 2 instead of being between x -directed regions 11 and 6.

For both cases, NREG should be set to 8.

Volume regions:

See also section 3.5.4 for a specific example.

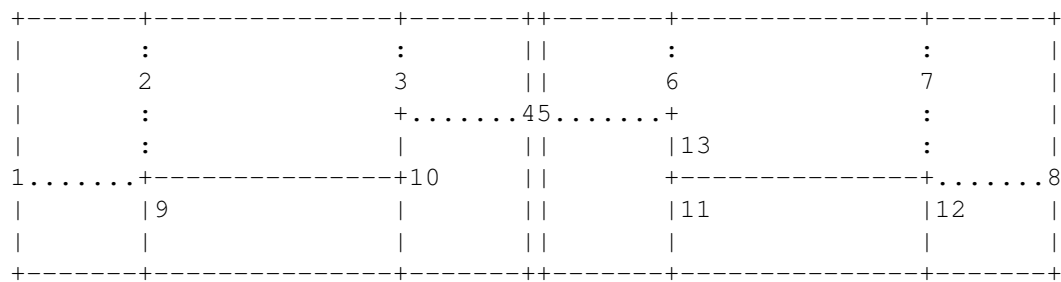


```

region 1 === left core
region 2 === left SOL
region 3 === left inboard divertor
region 4 === left outboard divertor
region 5 === right core
region 6 === right SOL
region 7 === right inboard divertor
region 8 === right outboard divertor

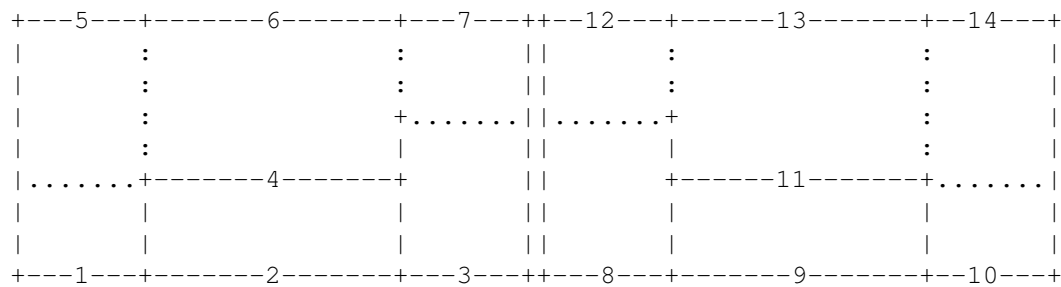
```

X-directed regions:



region 1 === left inboard divertor target
 region 2 === entrance to left inboard divertor
 region 3 === entrance to left outboard divertor
 region 4 === left outboard target
 region 5 === right inboard target
 region 6 === entrance to right inboard divertor
 region 7 === entrance to right outboard divertor
 region 8 === right outboard target
 region 9 === connection between left inboard and right outboard core
 region 10 === connection between left outboard and right inboard private flux region
 region 11 === connection between right inboard and left outboard core
 region 12 === connection between right outboard and left inboard private flux region
 region 13 === connection between right and left SOL

Y-directed regions:



region 1 === left inboard divertor private flux wall boundary
 region 2 === left core boundary
 region 3 === left outboard divertor private flux wall boundary
 region 4 === left separatrix
 region 5 === left inboard divertor main chamber wall boundary
 region 6 === left SOL main chamber wall boundary
 region 7 === left outboard divertor main chamber wall boundary
 region 8 === right inboard divertor private flux wall boundary
 region 9 === right core boundary
 region 10 === right outboard divertor private flux wall boundary
 region 11 === right separatrix
 region 12 === right inboard divertor main chamber wall boundary
 region 13 === right SOL main chamber wall boundary
 region 14 === right outboard divertor main chamber wall boundary

H.6 Regions for periodic BC stellarator island case

The reader is referred to [57] for a fuller description of this geometry. Note that the side edges of the core (volume region 1) are periodicity boundaries.

For this geometry, NREG should be set to 5.

Volume regions:

```

++-----++
||              5              ||
+\\-----/+
|              2              |
+-----+-----+-----+
|      |      |      |      |
|   3   |      1      |   4   |
|      |      |      |      |
+-----+-----+-----+

```

```

region 1 === core
region 2 === SOL
region 3 === inboard divertor
region 4 === outboard divertor
region 5 === island

```

X-directed regions:

```

++-----++
||7              ||
+\\-----/+
|      2          3      |
1-----+-----4
|      |      |      |
|      |5      |6      |
|      |      |      |
+-----+-----+

```

```

region 1 === inboard target
region 2 === entrance to inboard divertor
region 3 === entrance to outboard divertor
region 4 === outboard target
region 5 === connection between both sides of core
region 6 === connection between both sides of private flux region
region 7 === connection between both sides of island

```


Y-directed regions:

```

++-----6-----++
||                               ||
+\\-----8-----/+
|                               |
+---5---+-----4-----+---7---+
|       |               |       |
|       |               |       |
|       |               |       |
+---1---+-----2-----+---3---+

```

region 1 === inboard divertor private flux wall boundary
 region 2 === core boundary
 region 3 === outboard divertor private flux wall boundary
 region 4 === separatrix
 region 5 === entrance to inboard private flux region
 region 6 === island center
 region 7 === entrance to outboard private flux region
 region 8 === island boundary

Appendix I

b2plot manual

I.1 Introduction

b2plot is designed to take data from B2, B2.5 or B2.5-Eirene and plot it under interactive control. It uses NCAR and so the user must have access to the NCAR environment. Some, but far from all, functionalities have been ported so that **b2plot** may also work in the PLPLOT environment. Preprocessing before compilation with `-DPLPLOT` option is then necessary (see section J.3 for more details). Data from the B2SXDR, and, if available, the B2SDIAG (for **SOLPS 4.x** runs), `b2fstate`, `b2fplasma`, `b2frates`, `b2fgmtry`, `b2mn.dat` (for **SOLPS 5.x** and **SOLPS-ITER** runs) and `fort.44` and `fort.46` (when running with Eirene) files are read in, and then commands are read in from `stdin`.

The basic idea in using **b2plot** is to specify the data quantity, some transformations, and then the plot command.

Note that b2plot deals with cell-face quantities by interpolating such quantities to cell-centres so that, within b2plot, all quantities are cell-centred quantities.

b2plot is stack based: seven different stacks are used. The most important stack is the "matrix" stack which is used to store quantities on the computational grid. These quantities can be $nx \times ny \times ns$ where nx and ny are the size of the grid, and ns ranges over the species. An "integer" stack is used to supply arguments for some of the commands. A "real" stack is also used to supply arguments for some commands and, importantly, for storing values to be used in doing arithmetic on the "matrix" stack. The fourth stack stores "string" values, and is also used for supplying data for some commands. The fifth stack is the "Eirene" stack, which contains quantities related to the neutrals, and computed on the Eirene triangular grid (only for **SOLPS-ITER** runs). The Eirene stack data is of size $ntrii \times ns$, where $ntrii$ is the number of triangles in the Eirene computational grid and ns ranges over the relevant species (atoms, molecules, or test ions). The sixth stack is a "wall" stack, which contains quantities that can be defined only on the edges of the domain. The wall data is of size $nbnd \times ns$, where $nbnd$ is the number of boundary cells. The seventh stack is the "target" stack, which contains data relevant to the target deposition and heat transfer models (only for **SOLPS 5.x** and **SOLPS-ITER** runs). The target data is of size $nbnd \times ndepth$, where $ndepth$ is the number of depth discretization points of the target elements in the direction normal to the vessel surface. As input lines are parsed, each input entity is identified as being a command, a real value, an integer value or a string value. If insufficient data is stored on the appropriate stacks for command execution, an error message will be returned and no other action will be taken. The real, integer or string values are placed on the "real", "integer" or "string" stacks, respectively. Integers and reals are distinguished by the latter having a decimal point, or having an `e` or `E` indicating exponential notation. The "integer", "real" and "string" stacks have a capacity of 1000 values each, and the strings can be up to 256 characters long. If a stack overflow occurs, an error is returned and no action is taken. The "matrix", "wall" and "target" stacks have a capacity of 10 data sets each. If a stack overflow occurs, the data set at the bottom of the stack is dropped, and the command can proceed. Stack overflows can be avoided by using the appropriate **drop** (see I.4.4) commands once a stack quantity is no longer needed.

I.2 Commands

Commands are distinguished from strings by the latter being surrounded by single quote marks. Several types of commands are distinguished (sometimes somewhat arbitrarily): available data quantities (plasma, neutrals, wall, or target), available commands acting on stack quantities on or the plots, setting real plot parameters (like maxima and minima), string valued commands, and storing constants onto the stacks.

The available plasma data quantities are given in table I.1. These are quantities like the electron temperature, or the particle densities and typically store the results on the "matrix" stack. Some of these commands also take arguments when calculating derived data.

quantity	description (<i>units</i>)
ahal	Atomic H_α emissivity ($photons.m^{-3}.s^{-1}$)
alf	Thermo-electric coefficient ($\Omega^{-1}.m^{-1}$)
alph	Ratio of pressure gradient to critical ballooning value
b	Magnetic field components (T)
b2br	Bremsstrahlung losses (W)
b2eater	Eirene atomic species index
b2espcr	Nuclear sequence index
b2im	Momentum ionization/recombination losses (s^{-1}) (SOLPS 4.x runs)
b2in	Particle ionization/recombination losses (s^{-1}) (SOLPS 4.x runs)
b2io	Ionization/recombination losses (W) (SOLPS 4.x runs)
b2ne	Electron energy neutral losses (W) (SOLPS 4.x runs)
b2ni	Ion energy neutral losses (W) (SOLPS 4.x runs)
b2nm	Momentum neutral losses (s^{-1}) (SOLPS 4.x runs)
b2nn	Particle neutral losses (s^{-1}) (SOLPS 4.x runs)
b2ra	Line radiation losses (W)
bb	Total magnetic field (T)
bc	Boundary condition number
bcch	Boundary charge source $b2stbc\_sch$ (A)
bche	Boundary electron heat source $b2stbc\_she$ (W)
bchi	Boundary ion heat source $b2stbc\_shi$ (W)
bcmo	Boundary parallel momentum source $b2stbc\_smo$ (N)
bcna	Boundary particle source $b2stbc\_sna$ ($particles.s^{-1}$)
bix	Bottom neighbour ix coordinate
biy	Bottom neighbour iy coordinate
brch	Recycling charge source $b2stbr\_sch$ (A) (including any Eirene contributions)
brhe	Recycling electron heat source $b2stbr\_she$ (W) (including any Eirene contributions)
brhi	Recycling ion heat source $b2stbr\_shi$ (W) (including any Eirene contributions)
brmo	Recycling parallel momentum source $b2stbr\_smo$ (N) (including any Eirene contributions)
brna	Recycling particle source $b2stbr\_sna$ ($particles.s^{-1}$) (including any Eirene contributions)
bvis	Braams viscosity piece ($m.kg^{-1}.s^{-1}$)
bx, bpol	Poloidal magnetic field (T)
by, brad	Radial magnetic field (T)
bz, btor	Toroidal magnetic field (T)
cd	Poloidal component of diamagnetic current ($m.s^{-1}$)
ceqp	Equipartition term multiplier
chhichii	Chang-Hinton χ_i ($m^2.s^{-1}$)
chineo	Neoclassical heat conductivity χ ($m^2.s^{-1}$)
coni	Coronal equilibrium densities
conn	Connection length (m)
cora	Coronal equilibrium radiation rate (per homonuclear sequence) (W)
cpa	Pressure correction (Pa)
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
cpo	Electric potential correction (V)
cr	Radial coordinate (m)
crx	Corner radial coordinate (m)
cry	Corner vertical coordinate (m)
cte	Electron temperature correction (eV)
cti	Ion temperature correction (eV)
cua	Parallel velocity correction ($m.s^{-1}$)
cut	Total momentum correction ($kg.m.s^{-1}$)
cvsx	Parallel classical viscosity ($m.kg^{-1}.s^{-1}$)
cvsy	Radial classical viscosity ($m.kg^{-1}.s^{-1}$)
cvv	CORVV (SOLPS 4.x runs)
cz	Vertical coordinate (m)
d	Particle density diffusion coefficient ($m^2.s^{-1}$)
dab2	Atomic density (m^{-3})
dd	Radial component of diamagnetic current ($m.s^{-1}$)
deltab	Ratio of collision frequency to gyro-frequency
deltal	Ratio of Larmor radius to minimum gradient length
deltapar	Ratio of radial to parallel dynamics
deltaw	Ratio of banana width to minimum gradient length
deltawn	Ratio of banana width to density gradient length
deltawt	Ratio of banana width to temperature gradient length
dfhe	Divergence of the electron energy flux (W)
dfhi	Divergence of the ion energy flux (W)
dfhj	Divergence of the electrostatic energy flux (A)
dfhm	Divergence of the parallel kinetic energy flux (W)
dfhp	Divergence of the potential energy flux (W)
dfmo	Divergence of the momentum flux ($N.s$)
dfna	Divergence of the particle flux (s^{-1})
dhet, dfhet	Divergence of the electron total energy flux (W)
dhit, dfhit	Divergence of the electron total energy flux (W)
dib2	Test ion density (m^{-3})
dmb2	Molecular density (m^{-3})
dnneo	Neoclassical particle diffusivity D_n ($m^2.s^{-1}$)
dp	Particle pressure diffusion coefficient ($m^2.s^{-1}$)
dvua	Ion heat source $b2sihs\_divua$ (W)
dvue	Electron heat source $b2sihs\_divue$ (W)
dx	Cell poloidal length dx (m) (SOLPS 4.x runs)
dy	Cell radial width dy (m) (SOLPS 4.x runs)
ebal	Energy balance (W)
ebld	Energy balance (-divergence of fluxes) (W)
ebls	Energy balance (sources) (W)
edissml	Power to atoms due to molecule dissociation (W)
ehx	ehx (V)
ehy	ehy (V)
exba	$\mathbf{E} \times \mathbf{B}$ ion heating term $b2sihs\_exba$ (W)
exbe	$\mathbf{E} \times \mathbf{B}$ electron heating term $b2sihs\_exbe$ (W)
exch	External charge source ext_sch (A)
exhe	External electron heat source ext_she (W)
exhi	External ion heat source ext_shi (W)
exmo	External parallel momentum source ext_smo (N)
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
exna	External particle source <i>ext_sna</i> (<i>particles.s⁻¹</i>)
fac_exb	$\mathbf{E} \times \mathbf{B}$ drift multiplier <i>fac_ExB</i>
fac_vis	Viscosity drift terms multiplier <i>fac_vis</i>
facdrift	Diamagnetic drift terms multiplier <i>facdrift</i>
fax_ext	External species poloidal flux (<i>s⁻¹</i>)
fay_ext	External species radial flux (<i>s⁻¹</i>)
fchx	Poloidal current (<i>A</i>)
fchx32	Convective poloidal current (<i>A</i>) (Does not include drift contributions)
fchx52	Conductive poloidal current (<i>A</i>) (Does not include drift contributions)
fchxn	Poloidal component of inertial current <i>fchin</i> (<i>A</i>)
fchxvspa	Poloidal component of parallel viscosity current <i>fchvispar</i> (<i>A</i>)
fchxvspe	Poloidal component of perpendicular viscosity current <i>fchvisper</i> (<i>A</i>)
fchp	Parallel current (<i>A</i>)
fchy	Radial current (<i>A</i>)
fchy32	Convective radial current (<i>A</i>) (Does not include drift contributions)
fchy52	Conductive radial current (<i>A</i>) (Does not include drift contributions)
fchyn	Radial component of inertial current <i>fchin</i> (<i>A</i>)
fchyvspa	Radial component of parallel viscosity current <i>fchvispar</i> (<i>A</i>)
fchyvspe	Radial component of perpendicular viscosity current <i>fchvisper</i> (<i>A</i>)
fetx	Total poloidal internal energy flux (<i>W</i>)
fety	Total radial internal energy flux (<i>W</i>)
ffbz	Toroidal magnetic flux function <i>ffbz</i> (<i>Wb/m</i>)
fhctx	Poloidal total electron energy flux (<i>W</i>)
fhety	Radial total electron energy flux (<i>W</i>)
fhcx	Poloidal electron energy flux (<i>W</i>)
fhcy	Radial electron energy flux (<i>W</i>)
fhctx	Poloidal total ion energy flux (<i>W</i>)
fhity	Radial total ion energy flux (<i>W</i>)
fhix	Poloidal ion energy flux (<i>W</i>)
fhix	Radial ion energy flux (<i>W</i>)
fhjx	Poloidal electrostatic energy flux (<i>W</i>)
fhjy	Radial electrostatic energy flux (<i>W</i>)
fhmx	Poloidal parallel kinetic energy flux (<i>W</i>)
fhmy	Radial parallel kinetic energy flux (<i>W</i>)
fhpx	Poloidal potential energy flux (<i>W</i>)
fhpy	Radial potential energy flux (<i>W</i>)
fhctx	Poloidal total energy flux (<i>W</i>)
fhity	Perpendicular total energy flux (<i>W</i>)
flal	Thermo-electric classical flux limit factor
flalc	Thermo-electric coefficient limit factor
fle	Electron heat flux limit factor
fli	Ion heat flux limit factor
fke	Electron heat diffusivity classical flux limit factor
fki	Ion heat diffusivity classical flux limit factor
fluca	Thermo-electric Luciani flux limit factor
fluce	Electron heat diffusivity Luciani flux limit factor
fluci	Ion heat diffusivity Luciani flux limit factor
flucs	Electric conductivity Luciani flux limit factor
flucv	Viscosity Luciani flux limit factor
flvisc	Viscosity flux limit factor
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
fmx	Poloidal momentum flux (N)
fmoy	Radial momentum flux (N)
fmx	Poloidal mass flux ($kg.s^{-1}$)
fmy	Radial mass flux ($kg.s^{-1}$)
fnax	Poloidal particle flux (s^{-1})
fnax32	Convective poloidal particle flux (s^{-1})
fnax52	Conductive poloidal particle flux (s^{-1})
fnaxnodr	Poloidal particle flux without drifts (s^{-1})
fnay	Radial particle flux (s^{-1})
fnay32	Convective radial particle flux (s^{-1})
fnay52	Conductive radial particle flux (s^{-1})
fnaynodr	Radial particle flux without drifts (s^{-1})
fnex	Poloidal electron flux (s^{-1})
fnex32	Convective poloidal electron flux (s^{-1}) (Does not include drift contributions)
fnex52	Conductive poloidal electron flux (s^{-1}) (Does not include drift contributions)
fnex	Radial electron flux (s^{-1})
fnex32	Convective radial electron flux (s^{-1}) (Does not include drift contributions)
fnex52	Conductive radial electron flux (s^{-1}) (Does not include drift contributions)
fnix32	Convective poloidal total ion flux (s^{-1}) (Does not include drift contributions)
fnix52	Conductive poloidal total ion flux (s^{-1}) (Does not include drift contributions)
fnix32	Convective radial total ion flux (s^{-1}) (Does not include drift contributions)
fnix52	Conductive radial total ion flux (s^{-1}) (Does not include drift contributions)
fraa	Friction heating term $b2sihs\_fraa$ (W)
gdcl	Guard cells
hx	Poloidal cell length (m)
hy	Radial cell width (m)
hy1	Shear-corrected radial cell width (m)
iix	Poloidal cell index ix
iyy	Radial cell index iy
ionrad	Power loss due to molecular ion radiation (W)
iseq	Species index is is equal to
isge	Species index is is greater than or equal to
isgt	Species index is is greater than
isle	Species index is is less than or equal to
islt	Species index is is less than
isne	Species index is is not equal to
ix	Poloidal cell index ix (normalized)
ixeq	Poloidal cell index ix is equal to
ixge	Poloidal cell index ix is greater than or equal to
ixgt	Poloidal cell index ix is greater than
ixle	Poloidal cell index ix is less than or equal to
ixlt	Poloidal cell index ix is less than
ixne	Poloidal cell index ix is not equal to
iy	Radial cell index iy (normalized)
iyeq	Radial cell index iy is equal to
iyge	Radial cell index iy is greater than or equal to
iygt	Radial cell index iy is greater than
iyle	Radial cell index iy is less than or equal to
iylt	Radial cell index iy is less than
iyne	Radial cell index iy is not equal to
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
joul	Joule heating term <i>b2sihs.joule</i> (<i>W</i>)
ke	Parallel kinetic energy (<i>J</i>)
kye	Electron heat diffusivity χ_e ($m^2.s^{-1}$)
kyi	Ion heat diffusivity χ_i ($m^2.s^{-1}$)
kyi0	Total ion heat diffusivity ($m^2.s^{-1}$)
lix	Left neighbour poloidal index <i>leftix</i>
liy	Left neighbour radial index <i>leftiy</i>
lnlam	Coulomb logarithm
lnn	Density gradient length (<i>m</i>)
ltt	Temperature gradient length (<i>m</i>)
mach	Mach number
mach4	Mach number (SOLPS 4.x calculation)
mch2	Normalized parallel velocity
mhal	Molecular H_α emissivity ($photons.m^{-3}.s^{-1}$)
mi	Ion mass (<i>kg</i>)
mi_ext	External species mass (<i>kg</i>)
molrad	Power loss due to molecule radiation (<i>W</i>)
na	Ion density (m^{-3}) (SOLPS 5.x and SOLPS-ITER runs)
na0	Ion density <i>na</i> at previous time step (m^{-3})
na_ext	External species density (m^{-3})
ne	Electron density (m^{-3})
ne2	ne2 (m^{-3})
ne_ext	Electron density associated with external species (m^{-3})
neutrad	Power loss due to atom radiation (<i>W</i>)
nffhix	Neutral flux limits for poloidal ion energy flux
nffhiy	Neutral flux limits for radial ion energy flux
nffnax	Neutral flux limits for poloidal particle flux
nffnay	Neutral flux limits for radial particle flux
nh	Total atomic hydrogen density (m^{-3})
ni	Total density (m^{-3}) (SOLPS 5.x and SOLPS-ITER runs)
ni	Ion density (m^{-3}) (SOLPS 4.x runs)
ni0	Density <i>ni</i> at previous time step (m^{-3})
niso	Isolating region number
nreg	Region number
nua	Total collision frequency (s^{-1})
nuab	Two species collision frequency (s^{-1})
nustar	Normalized collisionality
omega	Gyro-frequency ($rad.s^{-1}$)
p*	Dynamic pressure (<i>Pa</i>)
pbal	Particle balance
pefa	Poloidal atomic energy flux ($W.m^{-2}$)
pefm	Poloidal molecular energy flux ($W.m^{-2}$)
pfla	Poloidal atomic flux ($s^{-1}.m^{-2}$)
pflm	Poloidal molecular flux ($s^{-1}.m^{-2}$)
phi	Plasma potential $\varphi(V)$ (SOLPS 4.x runs)
pit, pitch	Pitch
po	Electric potential $\varphi(V)$
po0	Electric potential <i>po</i> at previous time step (<i>V</i>)
pot	Potential energy (cumulative) (<i>eV</i>)
pot*	Ionization energy [shifted down one] (<i>eV</i>)
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
poti	Ionization energy (eV)
pr	Static pressure (Pa)
psi	Poloidal magnetic flux function Ψ (Wb)
pz	Plasma pressure (Pa)
q	Safety factor
qc	Cosine of the angle between flux line direction and left cell face
qz0	Sine of the angle away from orthogonality of the grid
qz1	Cosine of the angle away from orthogonality of the grid
rc	Recycling condition number
rco	Continuity residuals
rcxi	Ion energy neutral losses ($rcxhi$) (W)
rcxm	Ion momentum neutral losses ($rcxmo$) (N)
rcxn	Particle neutral losses ($rcxna$) (s^{-1})
rdi	RESDI (SOLPS 4.x runs)
rdi1	RESDI (SOLPS 4.x runs)
ree	Electron energy equation residuals (W)
refa	Radial atomic energy flux ($W.m^{-2}$)
refm	Radial molecular energy flux ($W.m^{-2}$)
rei	Ion energy equation residuals (W)
rfla	Radial atomic flux ($s^{-1}.m^{-2}$)
rflm	Radial molecular flux ($s^{-1}.m^{-2}$)
rhof	Full-B Larmor radius (m)
rix	Right neighbour poloidal index <i>rightix</i>
riy	Right neighbour radial index <i>rightiy</i>
rlcl	Real cells
rmo	Momentum equation residuals ($kg.m.s^{-1}$)
rmt	Total momentum equation residuals ($kg.m.s^{-1}$)
ro	Mass density ($kg.m^{-3}$)
rpo	Potential equation residuals (V)
rqa	Electron cooling rate ($rqahe$) (W)
rrai	Recombination ion energy losses ($rrahi$) (W)
rram	Recombination ion momentum losses ($rramo$) (N)
rran	Recombination ion particle losses ($rrana$) (s^{-1})
rsai	Ionization ion energy losses ($rsahi$) (W)
rsam	Ionization ion momentum losses ($rsamo$) (N)
rsan	Ionization ion particle losses ($rsana$) (s^{-1})
rud	RESUD (SOLPS 4.x runs)
rudc	RESUDC (SOLPS 4.x runs)
rvd	RESVD (SOLPS 4.x runs)
s	Magnetic shear
sch	Charge sources (C)
she	Electron energy sources (W) (excluding equipartition)
shi	Ion energy sources (W) (excluding equipartition)
sig	Anomalous conductivity ($\Omega^{-1}.m^{-1}$)
smaf	Parallel momentum source from friction $b2npmo\_smaf$ (N)
smag	Parallel momentum source from pressure gradient $b2npmo\_smag$ (N)
smav	Parallel momentum source from additional viscosity $b2npmo\_smav$ (N)
smfr	Friction force (N)
smo	Momentum sources (N)
smp	Self-pressure force (N) (SOLPS 4.x runs)
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
smpr	Thermal force (proportional to $exhp$, not valid when $b2sigp\_style = 2$) (N)
smpt	Thermal force (proportional to T_i and T_e gradients, not valid when $b2sigp\_style = 2$) (N)
smq	Total momentum sources (N)
smth	Thermal force (N)
sna	Particle sources ($particles.s^{-1}$)
srcml	Molecule dissociation rate ($molecules.s^{-1}$)
str	Heating from strange term $b2sihs\_str$ (W)
sx	Poloidal area of contact for cell (m^2)
sxperp	Poloidal area perpendicular to x -direction (m^2)
sy	Radial area of contact for cell (m^2)
sz	Poloidal cross-sectional area for cell (m^2)
tab2	Neutral atom temperature (eV)
ta_ext	External species temperature (eV)
te	Electron temperature (eV)
te0	Electron temperature te at previous time step (eV)
ti	Ion temperature (eV)
ti0	Ion temperature ti at previous time step (eV)
tib2	Test ion temperature (eV)
tix	Top neighbour poloidal index $topix$
tiy	Top neighbour radial index $topiy$
tmb2	Neutral molecule temperature (eV)
ua	Parallel velocity (ua) ($m.s^{-1}$) (SOLPS 5.x and SOLPS-ITER runs)
ua0	Parallel velocity ua at previous time step ($m.s^{-1}$)
ua_ext	External species parallel velocity ($m.s^{-1}$)
ud	Poloidal component of drift velocity ($m.s^{-1}$)
ue	Parallel electron velocity (ue) ($m.s^{-1}$)
up	Parallel velocity (up) ($m.s^{-1}$) (SOLPS 4.x runs)
up	Poloidal velocity (up) ($m.s^{-1}$) (SOLPS 5.x and SOLPS-ITER runs)
uu	Poloidal velocity (uu) ($m.s^{-1}$) (SOLPS 4.x runs)
vadia.x	Poloidal component of divergent part of ion diamagnetic drift velocity ($m.s^{-1}$)
vadia.y	Radial component of divergent part of ion diamagnetic drift velocity ($m.s^{-1}$)
valfvén	Alfvén velocity $v_{Alfvén}$ ($m.s^{-1}$)
vcon	v_{con} ($m.s^{-1}$) (SOLPS 4.x runs)
vd	Radial component of drift velocity ($m.s^{-1}$)
vedia.x	Poloidal component of divergent part of electron diamagnetic drift velocity ($m.s^{-1}$)
vedia.y	Radial component of divergent part of electron diamagnetic drift velocity ($m.s^{-1}$)
veexb.x	Poloidal component of electron $\mathbf{E} \times \mathbf{B}$ drift velocity ($m.s^{-1}$)
veexb.y	Radial component of electron $\mathbf{E} \times \mathbf{B}$ drift velocity ($m.s^{-1}$)
vexb.x	Poloidal component of ion $\mathbf{E} \times \mathbf{B}$ drift velocity ($m.s^{-1}$)
vexb.y	Radial component of ion $\mathbf{E} \times \mathbf{B}$ drift velocity ($m.s^{-1}$)
visa	Viscous heating term $b2sihs\_visa$ (W)
visc	Viscosity ($m.kg^{-1}.s^{-1}$)
vlax	Poloidal pinch velocity ($m.s^{-1}$)
vlay	Radial pinch velocity ($m.s^{-1}$)
vol	Cell volume (m^3)
vnneo	Neoclassical particle pinch v_n ($m.s^{-1}$)
vth	Thermal velocity v_{th} ($m.s^{-1}$)
vthe	Electron thermal velocity $v_{th,e}$ ($m.s^{-1}$)
vthi	Ion thermal velocity $v_{th,i}$ ($m.s^{-1}$)
vv	Radial velocity (vv) ($m.s^{-1}$)
continued on next page	

continued from previous page	
quantity	description (<i>units</i>)
vv0	Radial velocity vv at previous time step ($m.s^{-1}$)
vwneo	Neoclassical heat pinch v_w ($m.s^{-1}$)
wadia.x	Poloidal component of total diamagnetic drift velocity ($m.s^{-1}$)
wadia.y	Radial component of total diamagnetic drift velocity ($m.s^{-1}$)
wb	Width of the banana orbit (m)
wce	Electron cyclotron frequency ω_{ce} ($rad.s^{-1}$)
wci	Ion cyclotron frequency ω_{ci} ($rad.s^{-1}$)
wpe	Plasma frequency ω_{pe} ($rad.s^{-1}$)
ww	Toroidal velocity (ww) ($m.s^{-1}$)
xreg	x -directed flux region number
xymap	Index of wall segment $xymap$
yreg	y -directed flux region number
za_ext	External species charge (amu)
za2_ext	External species squared charge (amu)
zeff	Z effective (amu)
zi, za	Ion charge (amu)
zn	Nuclear charge (amu)
zn_ext	External species nuclear charge (amu)

Table I.1: Available quantities

Note that the **nreg**, **niso**, **xreg**, and **yreg** commands require an integer argument $\langle ireg \rangle$. If $\langle ireg \rangle = 0$, the arrays will contain the region number of each cell. If $\langle ireg \rangle$ is greater than 0, the cells in region with index $\langle ireg \rangle$ will contain 1.0, and the rest will contain zero. If $\langle ireg \rangle$ is less than 0, the cells in region with index $-\langle ireg \rangle$ will contain zero, and the rest will contain 1.0.

The source arrays **sna**, **smo**, **smq**, **she**, **shi**, and **sch** can take an optional integer argument $\langle isel \rangle$, varying from 0 to 3 (0 to 1 for **sna**). Then the data stack contains the linearized source data according to the description found in Appendix A.3. In the absence of this integer argument, the data stack will contain the total source terms. The same treatment applies to **smaf**, **smag**, and **smav**.

For the corner coordinates arrays **crx** and **cry**, an integer argument must be provided to determine the corner index according to the following:

- 0** Bottom left
- 1** Bottom right
- 2** Top left
- 3** Top right

where the topology is that of the computational domain. This means that for a cell with indices (ix, iy) , the left face is shared with $(ix - 1, iy)$, the right face with $(ix + 1, iy)$, the top face with $(ix, iy + 1)$, and the bottom face with $(ix, iy - 1)$.

The arrays **ixlt**, **ixle**, **ixeq**, **ixne**, **ixgt**, **ixge**, **iylt**, **iyle**, **iyeq**, **iyne**, **iygt**, **iyge**, **islt**, **isle**, **iseq**, **isne**, **isgt**, and **isge** also all require an integer argument $\langle i \rangle$, with which the logical comparison is done. If the comparison test is true, the arrays will contain 1.0, otherwise they will contain zero.

Please note that the sign convention for the parallel ion (**ua**) and electron (**ue**) velocity depends on the sign of the poloidal magnetic field. The poloidal field is positive if it points in the direction of increasing poloidal index ix . The parallel velocities are positive if they point in the *same direction as the poloidal magnetic field*. Note further that this

sign convention *does not apply* to the parallel current **fchp**, for which the standard sign convention applies, i.e. it is positive if pointing in the direction of increasing poloidal index ix .

The commands acting on the matrix stack are given in table I.2. These involve some manipulation or use of the data on the "matrix" stack. The stack requirements of each command are shown in the table for the integer ('i'), real ('r'), string ('s'), matrix ('m'), wall ('w') and target ('t') stacks. Whenever these stack requirements are not met, an error is returned and no action is taken.

command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
abs	0	0	0	1	0	0	0	I.4.2	Absolute value
arox	0	0	0	1	0	0	0	I.4.8	Use as array for arrow X -component
aroy	0	0	0	1	0	0	0	I.4.8	Use as array for arrow Y -component
ave	1	0	0	0	0	0	0	I.4.10	Perform average
bala	0	0	0	1	0	0	0	I.4.11	Compute balance
chs	0	0	0	1	0	0	0	I.4.2	Change sign
dase	0	0	0	1	0	0	0	I.4.10	Set as <i>dab2</i> array
ddx	0	0	0	1	0	0	0	I.4.2	Differentiate with respect to x
ddy	0	0	0	1	0	0	0	I.4.2	Differentiate with respect to y
drop	0	0	0	1	0	0	0	I.4.4	Drop array on top of stack
dup	0	0	0	1	0	0	0	I.4.4	Duplicate array on top of stack
eirc	0	0	0	0	0	0	0	I.4.10	Copy Eirene data onto B2.5 data
f.lx	1	0	0	1	0	0	0	I.4.1	Plot against $\log(x)$
f.ly	1	0	0	1	0	0	0	I.4.1	Plot against $\log(y)$
f.x	1	0	0	1	0	0	0	I.4.1	Plot against x
f.y	1	0	0	1	0	0	0	I.4.1	Plot against y
fpsi	0	0	0	0	0	0	0	I.4.6	Plot data against poloidal magnetic flux (Ψ) value
intx	0	0	0	1	0	0	0	I.4.2	Integrate along x
inty	0	0	0	1	0	0	0	I.4.2	Integrate along y
line	1	0	0	0	0	0	0	I.4.1	Choose x - or y -directed graphs
list	0	0	0	1	0	0	0	I.4.4	List the contents of the matrix stack
ln	0	0	0	1	0	0	0	I.4.2	Take natural logarithm of array
log	0	0	0	1	0	0	0	I.4.2	Take $\log_{10}$ of array
m*	0	0	0	2	0	0	0	I.4.3	Matrix multiplication
m+	0	0	0	2	0	0	0	I.4.3	Matrix addition
m-	0	0	0	2	0	0	0	I.4.3	Matrix subtraction
m/	0	0	0	2	0	0	0	I.4.3	Matrix division
mdims	0	0	0	0	0	0	0	I.4.4	Prints the dimensions and contents of the matrix stack
meq	0	0	0	2	0	0	0	I.4.3	Matrix equal to
mge	0	0	0	2	0	0	0	I.4.3	Matrix greater than or equal to
mgt	0	0	0	2	0	0	0	I.4.3	Matrix greater than
mle	0	0	0	2	0	0	0	I.4.3	Matrix less than or equal to
mlt	0	0	0	2	0	0	0	I.4.3	Matrix less than
mmax	0	0	0	2	0	0	0	I.4.3	Maximum of two matrices
mmid	0	0	0	0	0	0	0	I.4.6	Map data to midplane
mmin	0	0	0	2	0	0	0	I.4.3	Minimum of two matrices
mne	0	0	0	2	0	0	0	I.4.3	Matrix not equal to
mpsi	0	0	0	0	0	0	0	I.4.6	Plot data against normalised poloidal magnetic flux (ψ) value
nese	0	0	0	0	0	0	0	I.4.10	Overwrite onto electron density array
nspe, nspec	3	0	0	0	0	0	0	I.4.10	New spectrometer
over	0	0	0	2	0	0	0	I.4.4	Copy over second top element of matrix stack
pne	0	0	0	1	0	0	0	I.4.2	Divide by electron density
pos	0	0	0	1	0	0	0	I.4.3	Is positive?

continued on next page

continued from previous page

command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
psx	0	0	0	1	0	0	0	I.4.2	Divide by poloidal contact area
psxperp	0	0	0	1	0	0	0	I.4.2	Divide by area perpendicular to flux tube
psy	0	0	0	1	0	0	0	I.4.2	Divide by radial contact area
pvol	0	0	0	1	0	0	0	I.4.2	Divide by cell volume
rigz	0	0	0	0	0	0	0	I.4.2	Toggles use of ignorable cells (SOLPS 4.x runs)
rm*	0	1	0	1	0	0	0	I.4.3	Matrix multiplication by real
rm**	0	1	0	1	0	0	0	I.4.3	Matrix taken to a real power
rm+	0	1	0	1	0	0	0	I.4.3	Real added to matrix
rm-	0	1	0	1	0	0	0	I.4.3	Real subtracted from matrix
rm/	0	1	0	1	0	0	0	I.4.3	Matrix divided by real
rmma, rmmax	0	1	0	1	0	0	0	I.4.3	Maximum between real and matrix
rmmi, rmmin	0	1	0	1	0	0	0	I.4.3	Minimum between real and matrix
rpsi	0	0	0	0	0	0	0	I.4.6	Plot data against ρ_{pol} ($= \sqrt{\psi}$) value
spec	0	0	0	0	0	0	0	I.4.10	Old spectrometer
sqr	0	0	0	1	0	0	0	I.4.2	Square root of matrix
stoe	0	0	0	1	0	0	0	I.4.4	Copy top of matrix stack to Eirene stack
stow	0	0	0	1	0	0	0	I.4.4	Copy top of matrix stack to wall stack
sumx	0	0	0	1	0	0	0	I.4.2	Sum over x
sumy	0	0	0	1	0	0	0	I.4.2	Sum over y
sumz	2	0	0	1	0	0	0	I.4.2	Sum over species
swap	0	0	0	2	0	0	0	I.4.4	Swap top two matrices on stack
t_#p	1	0	0	0	0	0	0	I.4.1	Number of trajectories
tese	0	0	0	1	0	0	0	I.4.10	Copy top of stack onto electron temperature
traj	7+*	0	0	1	0	0	0	I.4.1	Trajectories
uuse	0	0	0	1	0	0	0	I.4.1	Set X -component of trajectory flows
vvse	0	0	0	1	0	0	0	I.4.1	Set Y -component of trajectory flows
wtos	0	0	0	0	0	1	0	I.4.4	Copy wall stack array onto matrix stack
xsel	1	0	0	1	0	0	0	I.4.2	Select X value
xtow	1	0	0	0	0	0	0	I.4.4	Copy X -row of matrix onto wall stack
ysel	1	0	0	1	0	0	0	I.4.2	Select Y value
ytow	1	0	0	0	0	0	0	I.4.4	Copy Y -column of matrix onto wall stack
zrng	2	0	0	1	0	0	0	I.4.2	Select species range
zsel	1	0	0	1	0	0	0	I.4.2	Select species index

Table I.2: Commands acting on the "matrix" stack together with their stack requirements.

The commands acting on the "matrix" stack plotting are given in table I.3. These modify some behavior or appearance of the "matrix" stack plot. Their arguments requirements are indicated in the table.

command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
a4l	0	0	0	0	0	0	0	I.4.5	A4 landscape
a4p	0	0	0	0	0	0	0	I.4.5	A4 portrait
acol	1	0	0	0	0	0	0	I.4.8	Arrow colour
arow	0	1	0	0	0	0	0	I.4.8	Set arrow length scale
arww	0	1	0	0	0	0	0	I.4.8	Set arrow line width

continued on next page

continued from previous page									
command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
aspc	0	0	0	0	0	0	0	I.4.6	Toggle aspect ratio
bcol	1	0	0	0	0	0	0	I.4.8	Colour of label bar title
bold	0	0	0	0	0	0	0	I.4.7	Toggles bold graph frame
both	0	0	0	0	0	0	0	I.4.6	Computational and physical domain plots
bw	0	0	0	0	0	0	0	I.4.7	Black-and-white plots only
clvd	0	3	0	0	0	0	0	I.4.9	Customize contour levels (delta)
clvl	0	0	0	0	0	0	0	I.4.9	Turn off contour levels
clvn	1	2	0	0	0	0	0	I.4.9	Customize contour levels (number)
clvp	2	0	0	1	0	0	0	I.4.9	Customize contour levels (proportion)
clvs	1	*	0	0	0	0	0	I.4.9	Customize contour levels (specified)
col	0	0	0	0	0	0	0	I.4.7	Colour plots
comp	0	0	0	0	0	0	0	I.4.6	Computational domain plots only
cont	0	0	0	1	0	0	0	I.4.1	Contour plot
dcol	1	0	0	0	0	0	0	I.4.8	Colour for matrix data line plots
eclv	0	0	0	0	0	0	0	I.4.9	Empty contour levels
ecol	1	0	0	0	0	0	0	I.4.8	Experimental data colour
expl	0	0	0	0	0	0	0	I.4.8	Toggle line or scatter experimental data plot
exps	1	0	0	0	0	0	0	I.4.8	Symbol for experimental scatter plot
ewid	0	1	0	0	0	0	0	I.4.8	Experimental data line width
fclv	0	0	0	0	0	0	0	I.4.9	Fill contour levels
fcol	1	0	0	0	0	0	0	I.4.8	Color for graph frame, axes, ticks and labels
filo	1	0	0	0	0	0	0	I.4.9	Fill option
font	1	0	0	0	0	0	0	I.4.7	Set font
gcol	1	0	0	0	0	0	0	I.4.8	Colour of graph title
glob	0	0	0	0	0	0	0	I.4.7	Copy <i>extralabel</i> to the global label field
glsz	0	1	0	0	0	0	0	I.4.7	Global header font size
grid	0	0	0	0	0	0	0	I.4.8	Add grid to plot
hcol	1	0	0	0	0	0	0	I.4.8	Colour of global header
i*	2	0	0	0	0	0	0	I.4.3	Integer multiplication
i**	2	0	0	0	0	0	0	I.4.3	Integer to the power
i+	2	0	0	0	0	0	0	I.4.3	Integer addition
i-	2	0	0	0	0	0	0	I.4.3	Integer subtraction
i/	2	0	0	0	0	0	0	I.4.3	Integer division
idrop	1	0	0	0	0	0	0	I.4.4	Drop top element from the integer stack
idup	1	0	0	0	0	0	0	I.4.4	Duplicate top element from the integer stack
imp	1	0	0	0	0	0	0	I.4.10	Set <i>jxi</i> (inner midplane coordinate)
iover	2	0	0	0	0	0	0	I.4.4	Copy over second element from the integer stack to top
iprt	1	0	0	0	0	0	0	I.4.4	Outputs the integer stack
iswap	2	0	0	0	0	0	0	I.4.4	Swap top two elements from the integer stack
itoe	1	0	0	0	0	0	0	I.4.4	Copy integer onto Eirene stack
itom	1	0	0	0	0	0	0	I.4.4	Copy integer onto matrix stack
itor	1	0	0	0	0	0	0	I.4.4	Copy integer onto real stack
itostr	1	0	0	0	0	0	0	I.4.4	Write integer onto string stack
itot	1	0	0	0	0	0	0	I.4.4	Copy integer onto target stack
itow	1	0	0	0	0	0	0	I.4.4	Copy integer onto wall stack
lacl	0	0	0	0	0	0	0	I.4.9	Labels along contour plots
lbof	0	0	0	0	0	0	0	I.4.9	Labels off
lbon	0	0	0	0	0	0	0	I.4.9	Labels on
lbsa	0	0	0	0	0	0	0	I.4.9	Toggle label lines

continued on next page

continued from previous page									
command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
lbsz	0	1	0	0	0	0	0	I.4.7	Label size
lfsz	1	0	0	0	0	0	0	I.4.7	Line label font size
lhor	0	0	0	0	0	0	0	I.4.9	Horizontal labels
linf	0	0	0	0	0	0	0	I.4.6	Linear functions plot
lint	0	0	0	0	0	0	0	I.4.1	Linear time for trajectories plot
linx	0	0	0	0	0	0	0	I.4.6	Linear X scale
logf	0	0	0	0	0	0	0	I.4.6	Logarithmic functions plot
logt	0	0	0	0	0	0	0	I.4.1	Logarithmic time for trajectories plot
logx	0	0	0	0	0	0	0	I.4.6	Logarithmic X scale
lwid	0	1	0	0	0	0	0	I.4.7	Line width
mcol	1	0	0	0	0	0	0	I.4.8	Mesh (and triangular grid) colour
mesh	0	0	0	0	0	0	0	I.4.1	Plot mesh
mvs	1	0	0	0	0	0	0	I.4.10	Movie skip
ndec	1	0	0	0	0	0	0	I.4.6	Number of logarithmic decades
new	0	0	0	0	0	0	0	I.4.7	New page
neo_ic	1	0	0	0	0	0	0	I.4.10	Option for neoclassical calculation
nice	0	0	0	0	0	0	0	I.4.7	Toggle whether nice axes will be used
noax	0	0	0	0	0	0	0	I.4.7	No axes
nocp	0	0	0	0	0	0	0	I.4.9	No contour plot
ocol	1	0	0	0	0	0	0	I.4.8	Outline colour
omp	1	0	0	0	0	0	0	I.4.10	Set jxa (outer midplane coordinate)
outl	0	0	0	0	0	0	0	I.4.8	Plot outline
owid	0	1	0	0	0	0	0	I.4.8	Outline line width
page	2	0	0	0	0	0	0	I.4.5	Number of graphs per page
part	0	1	0	0	0	0	0	I.4.11	Partition colour map
phys	0	0	0	0	0	0	0	I.4.6	Physical domain plots only
r*	0	2	0	0	0	0	0	I.4.3	Real multiplication
r**	0	2	0	0	0	0	0	I.4.3	Real to the power
r+	0	2	0	0	0	0	0	I.4.3	Real addition
r-	0	2	0	0	0	0	0	I.4.3	Real subtraction
r/	0	2	0	0	0	0	0	I.4.3	Real division
rdrop	0	1	0	0	0	0	0	I.4.4	Drop top element from the real stack
rdup	0	1	0	0	0	0	0	I.4.4	Duplicate top element from the real stack
reso	0	0	0	0	0	0	0	I.4.10	Reset output
rigz	0	0	0	0	0	0	0	I.4.2	Toggle use of ignore regions
rover	0	2	0	0	0	0	0	I.4.4	Copy over second element from the real stack to top
rpri	0	1	0	0	0	0	0	I.4.4	Outputs the real stack
rswap	0	2	0	0	0	0	0	I.4.4	Swap top two elements from the real stack
rtom	0	1	0	0	0	0	0	I.4.4	Copy real onto Eirene stack
rtom	0	1	0	0	0	0	0	I.4.4	Copy real onto matrix stack
rtostr	0	1	0	0	0	0	0	I.4.4	Write real onto string stack
rtot	0	1	0	0	0	0	0	I.4.4	Copy real onto target stack
rtow	0	1	0	0	0	0	0	I.4.4	Copy real onto wall stack
same	0	0	0	0	0	0	0	I.4.7	Next graph on same axes
scol	1	0	0	0	0	0	0	I.4.8	Separatrix colour
sep	0	0	0	0	0	0	0	I.4.8	Plot separatrices
sepw	0	1	0	0	0	0	0	I.4.8	Separatrix line width
shco	0	0	0	0	0	0	0	I.4.11	
spln	0	0	0	1	0	0	0	I.4.9	Spline plot

continued on next page

continued from previous page									
command	stack requirements							section	description
	'i'	'r'	's'	'm'	'e'	'w'	't'		
sq	0	0	0	0	0	0	0	I.4.5	Square aspect ratio plot
stdl	0	0	0	0	0	0	0	I.4.7	Set standard label as <i>extralabel</i>
surf	0	0	0	1	0	0	0	I.4.1	Surface plot
swid	1	0	0	0	0	0	0	I.4.10	Set WINID
t_r+	0	0	0	0	0	0	0	I.4.11	Set ratio to <i>.true.</i>
t_r-	0	0	0	0	0	0	0	I.4.11	Set ratio to <i>.false.</i>
target	0	0	0	0	0	0	1	I.4.8	Add target data to plot
tcoll	1	0	0	0	0	0	0	I.4.8	Colour for target data line plots
trgoff	0	0	0	0	0	0	0	I.4.8	Turn off target data in plot
trig	0	0	0	0	0	0	0	I.4.8	Add triangular grid to plot
trimesh	0	0	0	0	0	0	0	I.4.1	Plot triangular grid
vcol	1	0	0	0	0	0	0	I.4.8	Vessel colour
vesl	0	0	0	0	0	0	0	I.4.8	Plot vessel
vwid	0	1	0	0	0	0	0	I.4.8	Vessel line width
wall	0	0	0	0	0	1	0	I.4.8	Add wall data to plot
wcol	1	0	0	0	0	0	0	I.4.8	Colour for wall data line plots
writ , write	0	0	0	0	0	0	0	I.4.10	Write output file
xcol	1	0	0	0	0	0	0	I.4.8	Colour of extra label
xpar	0	0	0	0	0	0	0	I.4.6	Set X coordinate to parallel length of fieldline
xpol	0	0	0	0	0	0	0	I.4.6	Set X coordinate to poloidal length
xyrz	0	0	0	0	0	0	0	I.4.10	Output plot ranges and boundaries
yref	1	0	0	0	0	0	0	I.4.10	Set <i>yref</i> (radial reference coordinate)
ytra	1	0	0	0	0	0	0	I.4.8	Transform Y coordinate for comparison to experiment
zcol	1	0	0	0	0	0	0	I.4.8	Colour of zero axes
zero	0	0	0	0	0	0	0	I.4.8	Add zero axes to plot
zwid	0	1	0	0	0	0	0	I.4.8	Zero axes line width

Table I.3: Commands acting on the "matrix" stack plotting together with their stack requirements.

The maxima and minima for plotting can be set by changing the real variables listed in table I.4. These commands take one value from the "real" stack and store it in an associated global variable.

name	use (<i>units</i>)
a0	R radius of the ellipse in <i>chord.F</i> (m)
b0	R radius of the ellipse in <i>chord.F</i> (m)
emax	Maximum function value for plotting Eirene stack data
emin	Minimum function value for plotting Eirene stack data
fmax	Maximum function value for plotting matrix data
fmin	Minimum function value for plotting matrix data
omax	Maximum ordinate for 2-D plots
omin	Minimum ordinate for 2-D plots
r0	R position of the ellipse in <i>chord.F</i> (m)
rmax	Maximum R for plotting in the physical domain (m)
rmin	Minimum R for plotting in the physical domain (m)
rshi	Relative R shift to be applied to chords (m)
smax	Maximum ordinate for wall plots
continued on next page	

continued from previous page	
name	use (<i>units</i>)
smin	Minimum ordinate for wall plots
t_pt	Total flight time of particles in <code>caltraj</code> (<i>s</i>)
trgfmax	Maximum function value for plotting in the target domain
trgfmin	Minimum function value plotting in the target domain
trgomax	Maximum ordinate for target domain 2-D plots
trgomin	Minimum ordinate for target domain 2-D plots
trgrmax	Maximum <i>R</i> for plotting in the target domain (<i>m</i>)
trgrmin	Minimum <i>R</i> for plotting in the target domain (<i>m</i>)
trgxmax	Maximum <i>x</i> for plotting in the target domain
trgxmin	Minimum <i>x</i> for plotting in the target domain
trgymax	Maximum <i>y</i> for plotting in the target domain
trgymin	Minimum <i>y</i> for plotting in the target domain
trgzmax	Maximum <i>Z</i> for plotting in the target domain (<i>m</i>)
trgzmin	Minimum <i>Z</i> for plotting in the target domain (<i>m</i>)
tt0	Function value inside the ellipse in <code>chord.F</code>
wmax	Maximum function value for plotting in wall domain
wmin	Minimum function value for plotting in wall domain
xmax	Maximum <i>x</i> for plotting in the computational domain
xmin	Minimum <i>x</i> for plotting in the computational domain
xref	Reference position in <i>x</i> : 0.0=left, 0.5=mid, 1.0=right
ymax	Maximum <i>y</i> for plotting in the computational domain
ymin	Minimum <i>y</i> for plotting in the computational domain
z0	<i>Z</i> position of the ellipse in <code>chord.F</code> (<i>m</i>)
zmax	Maximum <i>Z</i> for plotting in the physical domain (<i>m</i>)
zmin	Minimum <i>Z</i> for plotting in the physical domain (<i>m</i>)
zshi	Relative <i>Z</i> shift to be applied to chords (<i>m</i>)

Table I.4: Commands assigning real variables from stack for use in plotting

Values for the text variables mentioned in tables I.5 can also be set. These text variables often form the argument for a command. Typically these commands take one value off the "string" stack.

name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
adas	0	0	1	0	0	0	0	I.5.24	Access ADAS data
amju , amjuel	0	0	4	0	0	0	0	I.5.23	Access AMJUEL data
cam	0	0	1	0	0	0	0	I.5.12	Camera plot
chbx	0	0	1	0	0	0	0	I.5.8	Boxcar average of chords
chmp	0	0	1	0	0	0	0	I.5.11	Chord contributions to the mesh element
chor	0	0	1	0	0	0	0	I.5.3	Filename containing chords (for integration)
chpl	0	0	1	0	0	0	0	I.5.5	Filename containing chords (to plot)
chvl	0	0	1	0	0	0	0	I.5.9	Value at positions
cltb	0	0	1	0	0	0	0	I.5.2	Filename for a colour table
cmd	0	0	1	0	0	0	0	I.5.6	Filename from which to take commands
cmdcdf	0	0	1	0	0	0	0	I.5.7	Loop command file on CDF data
ecam	0	0	1	0	1	0	0	I.5.16	Camera plot for Eirene stack data
echbx	0	0	1	0	1	0	0	I.5.13	Boxcar average of chords for Eirene stack data

continued on next page

continued from previous page									
name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
echmp	0	0	1	0	1	0	0	I.5.15	Chord contributions to the mesh element from Eirene stack
echor	0	0	1	0	1	0	0	I.5.4	Filename containing chords (for integration of Eirene stack)
echvl	0	0	1	0	1	0	0	I.5.14	Value at positions of Eirene stack data
expf	0	0	1	0	0	0	0	I.4.8	Experimental data file
extl	0	0	1	0	0	0	0	I.5.1	Extra label
flux	0	0	1	0	0	0	0	I.5.10	Call WOSlib flux routine
glbl	0	0	1	1	0	0	0	I.4.7	Set graph label
movi	0	0	1	0	0	0	0	I.5.17	Movie
movie , nmovie	2	0	1	0	0	0	0	I.5.18	New movie option
mprt	0	0	1	1	0	0	0	I.4.10	Output top of matrix stack to a file
mtab	4	0	1	1	0	0	0	I.4.10	Output part of the matrix on top of the stack to a file
omovie	2	0	2	0	0	0	0	I.5.19	Old movie option
oson	0	0	1	0	0	0	0	I.5.21	Output geometry in Sonnet format
outp	2	0	1	1	0	0	0	I.4.10	Prints value of specific element
pdev	0	0	1	0	0	0	0	I.4.10	PLPLOT device name
saw	2	0	1	0	0	0	0	I.4.4	Read external data onto the matrix stack
saws	3	0	1	0	0	0	0	I.4.4	Read external fluid species data onto the matrix stack
sawx	3	0	1	0	0	0	0	I.4.4	Read external additional species data onto the matrix stack
scat	0	0	2	0	0	0	0	I.4.4	Concatenate two strings
sdrop	0	0	1	0	0	0	0	I.4.4	Drop top of string stack
sdup	0	0	1	0	0	0	0	I.4.4	Duplicate top of string stack
slbl	1	0	1	1	0	0	0	I.4.7	Set species label
sonn	0	0	1	0	0	0	0	I.5.20	Read a new Sonnet geometry
sover	0	0	2	0	0	0	0	I.4.4	Copy over the second element of the string stack
sprt	0	0	1	0	0	0	0	I.4.4	Outputs the string stack
sswap	0	0	2	0	0	0	0	I.4.4	Swap the top two elements of the string stack
tlbl	0	0	1	0	0	0	1	I.4.7	Set target graph label
trgblbl	0	0	1	0	0	0	0	I.4.7	Set target bar label
trgprt	0	0	1	0	0	0	1	I.4.10	Output top of target stack to a file
trgtab	4	0	1	0	0	0	1	I.4.10	Output part of the top of the target stack to a file
uedgegrd	0	1	1	0	0	0	0	I.5.22	Output geometry in UEDGE format
wallblbl	0	0	1	0	0	0	0	I.4.7	Set wall bar label
wlbl	0	0	1	0	0	1	0	I.4.7	Set wall graph label
wprt	0	0	1	0	0	1	0	I.4.10	Output top of wall stack to a file
wslbl	1	0	1	0	0	1	0	I.4.7	Set wall species label
wtab	2	0	1	0	0	1	0	I.4.10	Output part of the top of the wall stack to a file

Table I.5: Commands taking character strings as parameters. Note that the string should be delimited by single quotes.

In table I.6, we give the list of quantities that are available for use on the "Eirene" stack, containing values defined on the triangular grid defined for the Eirene computations. These quantities are only available for coupled SOLPS-ITER runs.

name	description (<i>units</i>)
edena	Atoms energy density ($eV.m^{-3}$)
edeni	Test ions energy density ($eV.m^{-3}$)
edenm	Molecules energy density ($eV.m^{-3}$)
continued on next page	

continued from previous page	
name	description (<i>units</i>)
massa	Atom mass (<i>kg</i>)
massi	Molecular ion mass (<i>kg</i>)
massm	Molecule mass (<i>kg</i>)
pdena	Atoms particle density (m^{-3})
pdeni	Test ions particle density (m^{-3})
pdenm	Molecules particle density (m^{-3})
pu _x	<i>x</i> -component of poloidal unit vector at triangle center
pu _y	<i>y</i> -component of poloidal unit vector at triangle center
pv _x	<i>x</i> -component of radial unit vector at triangle center
pv _y	<i>y</i> -component of radial unit vector at triangle center
tindex	Triangle index
tr	Triangle centroid <i>R</i> coordinate (<i>m</i>)
triarea	Triangle area (m^2)
triavol	Triangle volume (m^3)
tz	Triangle centroid <i>z</i> coordinate (<i>m</i>)
vxa	Atoms <i>x</i> -directed velocity ($m.s^{-1}$)
vxdena	Atoms <i>x</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vxdeni	Test ions <i>x</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vxdenm	Molecules <i>x</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vxi	Test ions <i>x</i> -directed velocity ($m.s^{-1}$)
vxm	Molecules <i>x</i> -directed velocity ($m.s^{-1}$)
vya	Atoms <i>y</i> -directed velocity ($m.s^{-1}$)
vydena	Atoms <i>y</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vydeni	Test ions <i>y</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vydenm	Molecules <i>y</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vyi	Test ions <i>y</i> -directed velocity ($m.s^{-1}$)
vym	Molecules <i>y</i> -directed velocity ($m.s^{-1}$)
vza	Atoms <i>z</i> -directed velocity ($m.s^{-1}$)
vzdena	Atoms <i>z</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vzdeni	Test ions <i>z</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vzdenm	Molecules <i>z</i> -directed momentum density ($kg.m^{-2}.s^{-1}$)
vzi	Test ions <i>z</i> -directed velocity ($m.s^{-1}$)
vzm	Molecules <i>z</i> -directed velocity ($m.s^{-1}$)

Table I.6: Quantities that can be stored in the "Eirene" stack.

Note that the neutral pressure is readily obtainable as $\frac{2}{3}$ of the energy density.

In table I.7, we give the list of commands that act on the "Eirene" stack or affect graphs of those data.

name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
eabs	0	0	0	0	1	0	0	I.4.2	Absolute value
earow	0	1	0	0	0	0	0	I.4.8	Set arrow length scale for Eirene data plots
earox	0	0	0	0	1	0	0	I.4.8	Use as Eirene array for arrow <i>x</i> -component
earoy	0	0	0	0	1	0	0	I.4.8	Use as Eirene array for arrow <i>y</i> -component
eavg	0	8	0	0	1	0	0	I.4.10	Perform a volume average of Eirene data over a rectangular region
echs	0	0	0	0	1	0	0	I.4.2	Change sign of Eirene data
eclvd	0	3	0	0	0	0	0	I.4.9	Customize Eirene contour levels (delta)

continued on next page

continued from previous page									
name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
eclvn	1	2	0	0	0	0	0	I.4.9	Customize Eirene contour levels (number)
eclvp	2	0	0	0	1	0	0	I.4.9	Customize Eirene contour levels (proportion)
eclvs	1	*	0	0	0	0	0	I.4.9	Customize Eirene contour levels (specified)
edims	0	0	0	0	0	0	0	I.4.4	Prints the dimensions and contents of the Eirene stack
edrop	0	0	0	0	1	0	0	I.4.4	Drop top of Eirene stack
edup	0	0	0	0	1	0	0	I.4.4	Duplicate top of Eirene stack
eexp	0	0	0	0	1	0	0	I.4.2	Take exponential of Eirene data
elbl	0	0	1	0	0	0	0	I.4.7	Provide a label for the Eirene data graph
elist	0	0	0	0	1	0	0	I.4.4	Lists the contents of the Eirene stack
eln	0	0	0	0	1	0	0	I.4.2	Take natural logarithm of Eirene data
elog	0	0	0	0	1	0	0	I.4.2	Take $\log_{10}$ of Eirene data
em*	0	0	0	0	2	0	0	I.4.3	Eirene stack multiplication
em+	0	0	0	0	2	0	0	I.4.3	Eirene stack addition
em-	0	0	0	0	2	0	0	I.4.3	Eirene stack subtraction
em/	0	0	0	0	2	0	0	I.4.3	Eirene stack division
emeq	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is equal to
emge	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is greater than or equal to
emgt	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is greater than
emle	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is less than or equal to
emlt	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is less than
emmax	0	0	0	0	2	0	0	I.4.3	Maximum of top two elements on the Eirene stack
emmin	0	0	0	0	2	0	0	I.4.3	Minimum of top two elements on the Eirene stack
emne	0	1	0	0	1	0	0	I.4.3	Eirene matrix element is not equal to
eoutp	1	0	1	0	1	0	0	I.4.10	Output the value of a specific Eirene triangle
eover	0	0	0	0	2	0	0	I.4.4	Copy over the second element of the Eirene stack
epos	0	0	0	0	1	0	0	I.4.3	Is Eirene data element positive?
eprt	0	0	1	0	1	0	0	I.4.10	Prints the Eirene data onto a file
esel	1	0	0	0	1	0	0	I.4.2	Select Eirene triangle
esqrt	0	0	0	0	1	0	0	I.4.2	Take square root of Eirene data
esumz	2	0	0	0	1	0	0	I.4.2	Sum over species
esurf	0	0	0	0	1	0	0	I.4.1	Plot Eirene data on the triangular grid
eswap	0	0	0	0	2	0	0	I.4.4	Swap top two elements of Eirene stack
etal	0	0	1	0	0	0	0	I.4.10	Read Eirene tally output file onto the Eirene stack
etos	0	0	0	0	1	0	0	I.4.4	Copy Eirene data onto matrix stack
evac	0	1	0	0	1	0	0	I.4.10	Fill vacuum region of Eirene stack with a value
wzrng	2	0	0	0	1	0	0	I.4.2	Select species range in Eirene data
ezsel	1	0	0	0	1	0	0	I.4.2	Select species in Eirene data
rem*	0	1	0	0	1	0	0	I.4.3	Eirene data * real
rem**	0	1	0	0	1	0	0	I.4.3	Eirene data ** real
rem+	0	1	0	0	1	0	0	I.4.3	Eirene data + real
rem-	0	1	0	0	1	0	0	I.4.3	Eirene data - real
rem/	0	1	0	0	1	0	0	I.4.3	Eirene data / real
remma	0	1	0	0	1	0	0	I.4.3	Maximum of Eirene data and real
remmi	0	1	0	0	1	0	0	I.4.3	Minimum of Eirene data and real
psta	0	0	0	0	1	0	0	I.4.2	Divide Eirene data per triangle area
pstv	0	0	0	0	1	0	0	I.4.2	Divide Eirene data per triangle volume
sumt	0	0	0	0	1	0	0	I.4.2	Sum over triangular grid
continued on next page									

continued from previous page									
name	stack requirements						section	use	
	'i'	'r'	's'	'm'	'e'	'w'	't'		

Table I.7: Commands related to the "Eirene" stack and triangular grid data plots.

In table I.8, we give the list of quantities that are available for use on the "wall" stack, containing values defined at the boundary surface at the edge of the computational domain.

name	description (<i>units</i>)
backscac	Backscattering energy fraction
backscat	Backscattering fraction
bbdyemis	Blackbody emissivity
c_cp_b1	Coating heat capacity b_1 factor ($J.kg^{-1}.K^{-1}$)
c_cp_b2	Coating heat capacity b_2 factor ($J.kg^{-1}.K^{-2}$)
c_cp_b3	Coating heat capacity b_3 factor (either ($J.kg^{-1}.K$) [old format] or ($J.kg^{-1}.K^{-3}$) [new format])
c_cp_b4	Coating heat capacity b_4 factor ($J.kg^{-1}.K^{-4}$)
ckappa_a	Coating factor a in $\kappa = 1/(aT + b)$ ($m.W^{-1}$)
ckappaa1	Coating factor a_1 in κ polynomial fit ($W.m^{-1}.K^{-1}$)
ckappaa2	Coating factor a_2 in κ polynomial fit ($W.m^{-1}.K^{-2}$)
ckappaa3	Coating factor a_3 in κ polynomial fit ($W.m^{-1}.K^{-3}$)
ckappaa4	Coating factor a_4 in κ polynomial fit ($W.m^{-1}.K^{-4}$)
ckappaa5	Coating factor a_5 in κ polynomial fit ($W.m^{-1}.K^{-5}$)
ckappa_b	Coating factor b in $\kappa = 1/(aT + b)$ ($m.K.W^{-1}$)
ckapppa1	Coating factor a_1 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-1}$)
ckapppa2	Coating factor a_2 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-2}$)
ckapppa3	Coating factor a_3 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-3}$)
ckapppa4	Coating factor a_4 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-4}$)
ckapppa5	Coating factor a_5 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-5}$)
coathick	Wall element coating thickness (m)
cooltemp	Coolant temperature on back side of wall element (K)
cp_b1	Heat capacity b_1 factor ($J.kg^{-1}.K^{-1}$)
cp_b2	Heat capacity b_2 factor ($J.kg^{-1}.K^{-2}$)
cp_b3	Heat capacity b_3 factor (either ($J.kg^{-1}.K$) [old format] or ($J.kg^{-1}.K^{-3}$) [new format])
cp_b4	Heat capacity b_4 factor ($J.kg^{-1}.K^{-4}$)
ctgdns	Wall element coating material density ($kg.m^{-3}$)
depo_ml	Monolayer deposition (<i>monolayers</i>)
deposi	Gross deposition (<i>particles</i>)
east	Wall elements corresponding to the East boundary
ei	Incident energy (eV)
erel	Chemical sputtering factor <i>erel</i>
ero_ml	Monolayer erosion (<i>monolayers</i>)
erosion	Gross erosion (<i>particles</i>)
ibnd	Wall element number
imapx	Wall element <i>ix</i> position
imapy	Wall element <i>iy</i> position
isweq	Species index <i>is</i> equal to
iswge	Species index <i>is</i> greater than or equal to
iswgt	Species index <i>is</i> greater than
continued on next page	

continued from previous page	
name	description (<i>units</i>)
iswle	Species index <i>is</i> less than or equal to
iswlt	Species index <i>is</i> less than
iswne	Species index <i>is</i> not equal to
iweq	Wall elements whose index is equal to
iwge	Wall elements whose index is greater than or equal to
iwgt	Wall elements whose index is greater than
iwle	Wall elements whose index is less than or equal to
iwlt	Wall elements whose index is less than
iwne	Wall elements whose index is not equal to
kappa_a	Factor a in $\kappa = 1/(aT + b)$ ($m.W^{-1}$)
kappa_a1	Factor a_1 in κ polynomial fit ($W.m^{-1}.K^{-1}$)
kappa_a2	Factor a_2 in κ polynomial fit ($W.m^{-1}.K^{-2}$)
kappa_a3	Factor a_3 in κ polynomial fit ($W.m^{-1}.K^{-3}$)
kappa_a4	Factor a_4 in κ polynomial fit ($W.m^{-1}.K^{-4}$)
kappa_a5	Factor a_5 in κ polynomial fit ($W.m^{-1}.K^{-5}$)
kappa_b	Factor b in $\kappa = 1/(aT + b)$ ($m.K.W^{-1}$)
kappapa1	Factor a_1 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-1}$)
kappapa2	Factor a_2 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-2}$)
kappapa3	Factor a_3 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-3}$)
kappapa4	Factor a_4 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-4}$)
kappapa5	Factor a_5 in $\kappa_{\perp}$ polynomial fit ($W.m^{-1}.K^{-5}$)
lcp	Layer heat capacity ($J.kg^{-1}.K^{-1}$)
ldepth	Layer depth (m)
lemis	Layer blackbody emissivity
lkappa	Layer heat conductivity ($W.m^{-1}.K^{-1}$)
lrho	Layer mass density ($kg.m^{-3}$)
lthick	Layer thickness (m)
north	Wall elements corresponding to the North boundary
nseg	Wall segment number
plathick	Wall element thickness (m)
qevap	Energy lost by vaporisation ($W.m^{-2}$)
qrad	Energy lost by blackbody radiation ($W.m^{-2}$)
r1	Start R position of wall element (m)
r2	End R position of wall element (m)
south	Wall elements corresponding to the South boundary
sput	Physical sputtering yield
sput2	Physical sputtering energy yield
sputchem	Chemical sputtering yield
sputfrac	Layer fraction
sputres	Radiation enhanced sputtering yield
stckcoef	Sticking coefficient
sublheat	Heat of sublimation ($J.kg^{-1}$)
tgdns	Wall element material density ($kg.m^{-3}$)
thermev	Thermal evaporation flux ($m^{-2}.s^{-1}$)
time	Wall element time (s)
vapprs	Vapor pressure (Pa)
vapprs_a	Vapor pressure A factor (K)
vapprs_b	Vapor pressure B factor
vaprtmp	Vaporisation temperature of wall material (K)
wallarea	Wall area (m^2)
continued on next page	

continued from previous page	
name	description (<i>units</i>)
walldist	Distance along wall (<i>m</i>)
wallflux	Wall particle flux (s^{-1})
wallhflux	Wall heat flux ($W.m^{-2}$)
wallngth	Wall element length (<i>m</i>)
walltemp	Wall surface temperature (<i>K</i>)
west	Wall elements corresponding to the West boundary
z1	Start <i>Z</i> position of wall element (<i>m</i>)
z2	End <i>Z</i> position of wall element (<i>m</i>)

Table I.8: Quantities that can be stored in the "wall" stack.

Note that the **nseg** command requires an integer argument $\langle iseg \rangle$. If $\langle iseg \rangle = 0$, the arrays will contain the wall segment number of each element. If $\langle iseg \rangle$ is larger than 0, the elements in segment with index $\langle iseg \rangle$ will contain 1.0, and the rest will contain zero. If $\langle iseg \rangle$ is less than 0, the elements in segment with index $-\langle iseg \rangle$ will contain zero, and the rest will contain 1.0.

The arrays **iwlt**, **iwle**, **iwel**, **iwne**, **iwge**, **iwgt**, **iswlt**, **iswle**, **iswel**, **iswne**, **iswge**, and **iswgt** all require an integer argument $\langle i \rangle$, with which the logical comparison is done. If the comparison test is true, the arrays will contain 1.0, otherwise they will contain zero.

In table I.9, we give the list of commands that act on the "wall" stack or affect wall graphs.

name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
ddw	0	0	0	0	0	1	0	I.4.2	Take derivative along wall length
intw	0	0	0	0	0	1	0	I.4.2	Integrate over walls (per segment)
intw*	0	0	0	0	0	1	0	I.4.2	Integrate over walls (all segments)
linw	0	0	0	0	0	1	0	I.4.8	Use linear scale for wall plot
linxw	0	0	0	0	0	1	0	I.4.8	Use linear abscissa for wall plot
logw	0	0	0	0	0	0	0	I.4.8	Use logarithmic scale for wall plot
logxw	0	0	0	0	0	0	0	I.4.8	Use logarithmic abscissa for wall plot
outpw	1	0	1	0	0	1	0	I.4.10	Output the value of a specific wall element
plwl	0	0	0	0	0	1	0	I.4.1	Plot in wall domain
pswa	0	0	0	0	0	1	0	I.4.2	Divide per wall element area
pwne	0	0	0	0	0	1	0	I.4.2	Divide by electron density
rwm*	0	1	0	0	0	1	0	I.4.3	Wall data * real
rwm**	0	1	0	0	0	1	0	I.4.3	Wall data ** real
rwm+	0	1	0	0	0	1	0	I.4.3	Wall data + real
rwm-	0	1	0	0	0	1	0	I.4.3	Wall data - real
rwm/	0	1	0	0	0	1	0	I.4.3	Wall data / real
rwmma	0	1	0	0	0	1	0	I.4.3	Maximum of wall data and real
rwmmi	0	1	0	0	0	1	0	I.4.3	Minimum of wall data and real
sumw	0	0	0	0	0	1	0	I.4.2	Sum over walls
wabs	0	0	0	0	0	1	0	I.4.2	Absolute value
wall	0	0	0	0	0	1	0	I.4.8	Add wall data to contour plot
wall_ank	1	0	0	0	0	0	0	I.4.11	Add ANK wall data to contour plot
wallf	0	0	1	0	0	0	0	I.4.8	Add experimental data to wall plot
wchs	0	0	0	0	0	1	0	I.4.2	Change sign of wall data
wclvd	0	3	0	0	0	0	0	I.4.9	Customize wall contour levels (delta)
continued on next page									

continued from previous page									
name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
wclvn	1	2	0	0	0	0	0	I.4.9	Customize wall contour levels (number)
wclvp	2	0	0	0	0	1	0	I.4.9	Customize wall contour levels (proportion)
wclvs	1	*	0	0	0	0	0	I.4.9	Customize wall contour levels (specified)
wdrop	0	0	0	0	0	1	0	I.4.4	Drop top of wall stack
wdup	0	0	0	0	0	1	0	I.4.4	Duplicate top of wall stack
wexp	0	0	0	0	0	1	0	I.4.2	Take exponential of wall data
wlist	0	0	0	0	0	1	0	I.4.4	Lists the contents of the wall stack
wln	0	0	0	0	0	1	0	I.4.2	Take natural logarithm of wall data
wlog	0	0	0	0	0	1	0	I.4.2	Take $\log_{10}$ of wall data
wm*	0	0	0	0	0	2	0	I.4.3	Wall stack multiplication
wm+	0	0	0	0	0	2	0	I.4.3	Wall stack addition
wm-	0	0	0	0	0	2	0	I.4.3	Wall stack subtraction
wm/	0	0	0	0	0	2	0	I.4.3	Wall stack division
wmeq	0	1	0	0	0	1	0	I.4.3	Wall matrix element is equal to
wmge	0	1	0	0	0	1	0	I.4.3	Wall matrix element is greater than or equal to
wmgt	0	1	0	0	0	1	0	I.4.3	Wall matrix element is greater than
wmle	0	1	0	0	0	1	0	I.4.3	Wall matrix element is less than or equal to
wmlt	0	1	0	0	0	1	0	I.4.3	Wall matrix element is less than
wmmax	0	0	0	0	0	2	0	I.4.3	Maximum of top two elements on the wall stack
wmmin	0	0	0	0	0	2	0	I.4.3	Minimum of top two elements on the wall stack
wmne	0	1	0	0	0	1	0	I.4.3	Wall matrix element is not equal to
woff	0	0	0	0	0	0	0	I.4.8	Turn off wall plots
wover	0	0	0	0	0	2	0	I.4.4	Copy over the second element of the wall stack
wpos	0	0	0	0	0	1	0	I.4.3	Is wall data element positive?
wprt	0	0	1	0	0	1	0	I.4.10	Prints the wall data onto a file
wsaw	1	0	1	0	0	0	0	I.4.4	Read external data onto the wall stack
wsel	1	0	0	0	0	1	0	I.4.2	Select wall element
wsqrt	0	0	0	0	0	1	0	I.4.2	Take square root of wall data
wsumz	2	0	0	0	0	1	0	I.4.2	Sum over species
wswap	0	0	0	0	0	2	0	I.4.4	Swap top two elements of wall stack
wtos	0	0	0	0	0	1	0	I.4.4	Copy wall data onto matrix stack
wtot	0	0	0	0	0	1	0	I.4.4	Copy wall data onto target stack
wwid	0	1	0	0	0	0	0	I.4.8	Wall linewidth
wzrng	2	0	0	0	0	1	0	I.4.2	Select species range in wall data
wzsel	1	0	0	0	0	1	0	I.4.2	Select species in wall data

Table I.9: Commands related to the "wall" stack and wall plots.

In table I.10, we give the list of quantities that are available for use on the "target" stack, containing values defined inside the target elements, if the appropriate **SOLPS 5.x** models have been activated. Note that by default, the "target" stack contains the target temperature (in K) upon **b2plot** start-up.

name	description (<i>units</i>)
cp	Target specific heat capacity ($J.kg^{-1}.K^{-1}$)
dkappadt	Derivative of target heat conductivity ($W.m^{-1}.K^{-2}$)
itrgeq	Is the target element index equal to?
itrgge	Is the target element index greater than or equal to?
continued on next page	

continued from previous page	
name	description (<i>units</i>)
itrsgt	Is the target element index greater than?
itrgle	Is the target element index less than or equal to?
itrslt	Is the target element index less than?
itrgne	Is the target element index not equal to?
kappa	Target heat conductivity (from polynomial fit) ($W.m^{-1}.K^{-1}$)
kappa_ab	Target heat conductivity (from $1/(aT + b)$ fit) ($W.m^{-1}.K^{-1}$)
kappap	Perpendicular target heat conductivity ($W.m^{-1}.K^{-1}$)
kappapdt	Perpendicular target heat conductivity derivative ($W.m^{-1}.K^{-2}$)
trgdepth	Target element depth (m)
trgeast	Target elements pertaining to the East boundary
trgibnd	Target element index
trgiz	Target depth index
trgl	ANK target loading (obsolete, use <i>100 wlld</i> instead)
trgnorth	Target elements pertaining to the North boundary
trgr1	Radial coordinate of first side of target element (m)
trgr2	Radial coordinate of second side of target element (m)
trgseg	Target wall segment index
trgsouth	Target elements pertaining to the South boundary
trgtemp	Target temperature (K) (default)
trgvol	Target element volume (m^3)
trgwest	Target elements pertaining to the West boundary
trgwidth	Target element width (m)
trgz1	Z coordinate of first side of target element (m)
trgz2	Z coordinate of second side of target element (m)
ztrgeq	Is the target element depth index equal to?
ztrgge	Is the target element depth index greater than or equal to?
ztrsgt	Is the target element depth index greater than?
ztrgle	Is the target element depth index less than or equal to?
ztrslt	Is the target element depth index less than?
ztrgne	Is the target element depth index not equal to?

Table I.10: Available target quantities for plotting.

The arrays **itrslt**, **itrgle**, **itrgeq**, **itrgne**, **itrgge**, **itrsgt**, **ztrslt**, **ztrgle**, **ztrgeq**, **ztrgne**, **ztrgge**, and **ztrsgt** all require an integer argument $\langle i \rangle$, with which the logical comparison is done. If the comparison test is true, the arrays will contain 1.0, otherwise they will contain zero.

The **trgseg** command requires an integer argument $\langle iseg \rangle$. If $\langle iseg \rangle = 0$, the arrays will contain the wall segment number of each element. If $\langle iseg \rangle$ is larger than 0, the elements in segment with index $\langle iseg \rangle$ will contain 1.0, and the rest will contain zero. If $\langle iseg \rangle$ is less than 0, the elements in segment with index $-\langle iseg \rangle$ will contain zero, and the others will contain 1.0.

In table I.11, we give the list of commands that act on the "target" stack or affect target graphs, with their respective stack requirements.

name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
intrtgw	0	0	0	0	0	0	1	I.4.2	Integrate along walls (all segments)
intrtgw*	0	0	0	0	0	0	1	I.4.2	Integrate along walls (per segment)

continued on next page

continued from previous page									
name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
inttrgz	0	0	0	0	0	0	1	I.4.2	Integrate in depth
lintrg	0	0	0	0	0	0	0	I.4.6	Linear target plot
linxtrg	0	0	0	0	0	0	0	I.4.6	Linear abscissae target plot
logtrg	0	0	0	0	0	0	0	I.4.6	Logarithmic target plot
logxtrg	0	0	0	0	0	0	0	I.4.6	Logarithmic abscissae target plot
outptrg	2	0	1	0	0	0	1	I.4.10	Outputs an element of target data
rtrgm*	0	1	0	0	0	0	1	I.4.3	Multiply the target data with a real
rtrgm**	0	1	0	0	0	0	1	I.4.3	Take the target data to a real power
rtrgm+	0	1	0	0	0	0	1	I.4.3	Add a real to the target data
rtrgm-	0	1	0	0	0	0	1	I.4.3	Subtract a real from the target data
rtrgm/	0	1	0	0	0	0	1	I.4.3	Divide the target data by a real
rtrgmmax	0	1	0	0	0	0	1	I.4.3	Take the maximum of the target data and a real
rtrgmmin	0	1	0	0	0	0	1	I.4.3	Take the minimum of the target data and a real
sumtrgw	0	0	0	0	0	0	1	I.4.2	Sum along wall coordinate
sumtrgz	0	0	0	0	0	0	1	I.4.2	Sum over target depth
tclvd	0	3	0	0	0	0	0	I.4.9	Customize target contour levels (delta)
tclvn	1	2	0	0	0	0	0	I.4.9	Customize target contour levels (number)
tclvp	2	0	0	0	0	0	1	I.4.9	Customize target contour levels (proportion)
tclvs	1	*	0	0	0	0	0	I.4.9	Customize target contour levels (specified)
trg.w	1	0	0	0	0	0	1	I.4.1	Line target plot against wall element
trg.z	1	0	0	0	0	0	1	I.4.1	Line target plot against target depth
trgabs	0	0	0	0	0	0	1	I.4.2	Absolute value of target data
trgchs	0	0	0	0	0	0	1	I.4.2	Change sign of target data
trgddp	0	0	0	0	0	0	1	I.4.2	Take derivative of target data along perpendicular direction
trgddw	0	0	0	0	0	0	1	I.4.2	Take derivative of target data along wall coordinate
trgddz	0	0	0	0	0	0	1	I.4.2	Take derivative of target data along target depth
trgdrop	0	0	0	0	0	0	1	I.4.4	Drop the top element from the target stack
trgdup	0	0	0	0	0	0	1	I.4.4	Duplicate the top element from the target stack
trgeq	0	1	0	0	0	0	1	I.4.3	Target data is equal to
trgexp	0	0	0	0	0	0	1	I.4.2	Take exponential of target data
trgge	0	1	0	0	0	0	1	I.4.3	Target data is greater than or equal to
trggt	0	1	0	0	0	0	1	I.4.3	Target data is greater than
trgle	0	1	0	0	0	0	1	I.4.3	Target data is less than or equal to
trglist	0	0	0	0	0	0	1	I.4.4	List the contents of the target stack
trgln	0	0	0	0	0	0	1	I.4.2	Take natural logarithm of target data
trglog	0	0	0	0	0	0	1	I.4.2	Take $\log_{10}$ of target data
trgl<	0	1	0	0	0	0	1	I.4.3	Target data is less than
trgm*	0	0	0	0	0	0	2	I.4.3	Target stack multiplication
trgm+	0	0	0	0	0	0	2	I.4.3	Target stack addition
trgm-	0	0	0	0	0	0	2	I.4.3	Target stack subtraction
trgm/	0	0	0	0	0	0	2	I.4.3	Target stack division
trgmmax	0	0	0	0	0	0	2	I.4.3	Maximum of top two elements of target stack
trgmmin	0	0	0	0	0	0	2	I.4.3	Minimum of top two elements of target stack
trgne	0	1	0	0	0	0	1	I.4.3	Target data is not equal to
trgover	0	0	0	0	0	0	2	I.4.4	Copy over second element on top of target stack
trgpos	0	0	0	0	0	0	1	I.4.3	Is target data positive?
trgsqrt	0	0	0	0	0	0	1	I.4.2	Take square root of target data
trgsurf	0	0	0	0	0	0	1	I.4.1	Create target data contour plot
trgswap	0	0	0	0	0	0	2	I.4.4	Swap top two elements of target stack
continued on next page									

continued from previous page									
name	stack requirements							section	use
	'i'	'r'	's'	'm'	'e'	'w'	't'		
trgtow	0	0	0	0	0	0	1	I.4.4	Copy target data onto wall stack
trgwsel	1	0	0	0	0	0	1	I.4.2	Select wall element in target data
trgzsel	1	0	0	0	0	0	1	I.4.2	Select depth element in target data

Table I.11: Commands related to the "target" stack and target plots.

Additionally, some real and integer constants used by the code can be read into the "real" and "integer" stacks of **b2plot** for further manipulation. These are given in the table I.12.

name	description (<i>units</i>)
avogr	Avogadro's constant N_A (mol^{-1})
c	Speed of light c (m/s)
eps0	Electric permittivity of vacuum ϵ_0 (F)
ev	Electron-Volt eV (J)
ismain	Index of the main plasma species
ixref	Index of reference poloidal location
iyref	Index of reference radial location
jsep	Radial index of outermost closed flux surface (just inside the separatrix)
jxa	Outer midplane index
jxi	Inner midplane index
kbolt	Boltzmann's constant k_B ($J.K^{-1}$)
me	Electron mass m_e (kg)
mp	Proton mass m_p (kg)
mu0	Magnetic permittivity of vacuum μ_0 ($F.m^{-1}$)
natm	Number of atomic species followed by Eirene
nbnd	Number of boundary cells
ndepth	Size of target arrays in depth direction
nion	Number of molecular ion species followed by Eirene
nmol	Number of molecular species followed by Eirene
ns	Number of species
nspecies	Number of homonuclear sequences
nx	Mesh size in poloidal direction
nxtl	Position index of top inner divertor in DN cases
nxtr	Position index of top outer divertor in DN cases
ny	Mesh size in radial direction
pi	π
qe	Electron charge q_e (C)

Table I.12: Available **b2plot** constants.

In table I.13 some additional commands are given.

name	use
coro	Print coronal rates
debu , debug	Set debugging information level
continued on next page	

continued from previous page	
name	use
eof	Indicate end of file
f30	Create a <code>fort.30</code> file
f31	Create a <code>fort.31</code> file
help	Print available commands
tranprof	Create <code>b2.transport.inputfile</code> file
user	Available for user-defined functions
wlld	Produces ANK wall loading files
quit , exit, end	Terminate program

Table I.13: Additional commands.

The **coro** command, with integer argument $\langle is \rangle$ and real argument $\langle T \rangle$ prints to `stdout` the coronal radiation rates for the homonuclear sequence of species $\langle is \rangle$ at temperature $\langle T \rangle$ (in eV).

The **debu** command, with integer argument $\langle level \rangle$, sets up the level of debugging information to $\langle level \rangle$ (default 0). The higher the value of $\langle level \rangle$, the more output is produced. **debug** is an alias for **debu**.

The **wlld** command requires an integer argument $\langle ig \rangle$. Let $\langle ig \rangle = JKLM$. If $M > 0$, then the file `radsrc.b2e` is produced, containing the radiation sources (in MW/m^3). If $L > 0$, then the file `wlld.dat` is produced, which contains the power loading from the plasma onto the Eirene walls (this description does not include splitting of the targets by the grid and is obsolete). If $K > 0$, then a set of files is produced containing the power loading on the targets, provided the necessary preparatory data exists in the DG model. These files are:

ld_tg_*.dat provides the power loading at the targets, including contributions from plasma particles, neutrals, and radiation.

fp_tg_*.dat provides the particle fluxes at the targets

peak_power provides the location and value of the peak power at the lower targets

peak_power_u provides the location and value of the peak power at the upper targets

peak_power_w gives the width of the power profiles

where the asterisk (*) stands for *i* (inner target), *iu* (inner upper target), *ou* (outer upper target) or *o* (outer target), respectively. If $J > 0$, then the codes produces a set of `wlld_*.dat`, where the asterisk (*) is the name of each "plot zone" defined in the DG model. Each such file contains the power loading onto the set of walls defined in that plot zone. If $LM = 99$ and $J > 0$, additional diagnostics are produced that indicate the radiation split by isonuclear sequence. If $LM = 98$ and $J > 0$, additional diagnostics are produced that indicate the radiation coming from the core additional sources.

If the user wants to use the heat transfer model for the monoblock and/or the reflector plates, then he must define in the DG model the inner/outer target, the reflector plates and dome plot zones using the label 'trg[i-o]', '[i-o]rp' and 'dome' respectively. The `wlld_*.dat` produced should then be postprocessed by `b2ymb` (see 3.18), `b2yrp` (3.19) and `b2ydm` (3.20).

I.3 Common errors

Amongst the most common errors are

- not distinguishing between real and integer values — real values have a decimal point in them or are written in exponential notation, integers consist only of numerals (preceded by a sign, if necessary);

- not preceding commands by their arguments — the whole program is structured to use a prefix notation (think HP calculator!);
- not delimiting character strings by single quotes.

I.4 Available Functions and Operators in b2plot

Section I.4.1 describes the commands used to produce a plot. Sections I.4.2 describes commands that manipulate quantities on the "matrix" stack, and I.4.3 describes the commands that can be used to do arithmetic. Stack manipulating commands are described in section I.4.4. Graph appearance is covered in sections I.4.5 (page format), I.4.6 (graph format) and I.4.7 (graph appearance). Overlays to contour plots are covered in section I.4.8 and changing contour parameters in section I.4.9. There is a grab bag of miscellaneous commands in section I.4.10.

I.4.1 Data plotting commands

These commands are typically used to produce a plot.

See section I.4.7 for a list of commands that can be used to change the graph appearance (text font, label size, overlays), and section I.4.6 for commands changing the graph format (physical or computational domain, aspect ratio, linear or logarithmic).

See also the **chor** (I.5.3), **chpl** (I.5.5), **chbx** (I.5.8), **chvl** (I.5.9), **flux** (I.5.10), **chmp** (I.5.11), **cam** (I.5.12), **ecam** (I.5.16), **echbx** (I.5.13), **echmp** (I.5.15), **echor** (I.5.4), **echvl** (I.5.14) and **movi** (I.5.17) commands.

A simple example is:

```
echo a4l 2 1 page 2.0 lbsz stdl glob vesl sep logf 3 ndec phys ni cont dab2 surf | b2plot
```

which produced figure I.1.

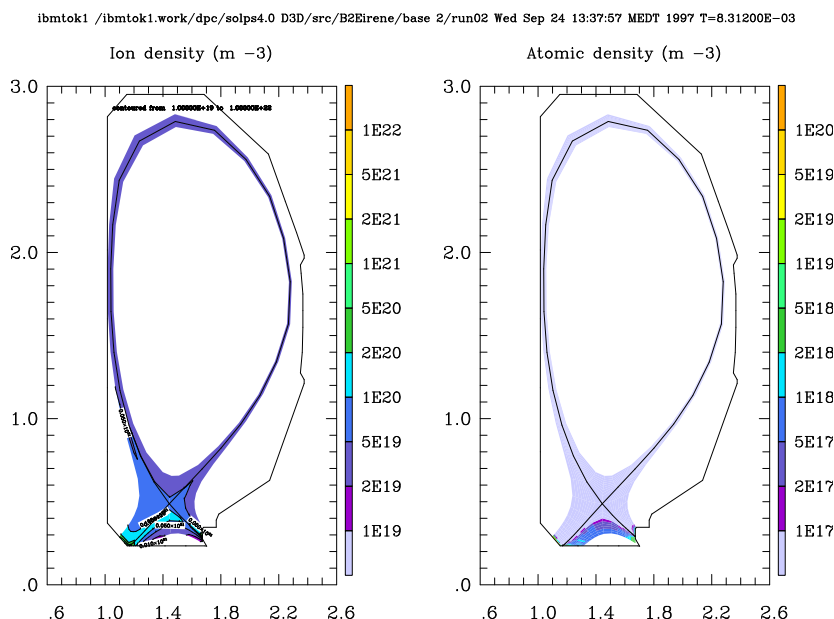


Figure I.1: Ion and atomic densities for DIII-D.

cont

Plots contour plots of the current data. See figure I.1. See section I.4.9 for a list of commands that can be used to change the appearance of contour plots. Section I.4.5 covers global changes to the plotting (square, landscape or portrait, number of graphs per page). The use of **surf** (see I.4.1) is preferred over that of **cont**.

<pos,I> f.x

Plots the 2-D graph of the matrix data at $iy = \langle \text{pos} \rangle$ versus x . Vertical lines will be added to the plot to delimit any eventual unconnected domain boundaries. Use of **xpol** (default, see I.4.6) or **xpar** (see I.4.6) commands allows one to toggle between plotting against poloidal or parallel distance along the field line.

<pos,I> f.y

Plots the 2-D graph of the matrix data at $ix = \langle \text{pos} \rangle$ versus y . Use of the **mmid** (see I.4.6), **fpsi** (see I.4.6), **mpsi** (see I.4.6) or **rpsi** (see I.4.6) commands allows for plotting versus the outer midplane (or another position chosen by the user) radial coordinate, full poloidal magnetic flux, normalized flux, or ρ_{pol} , respectively.

<n1,I> f.lx

Plots the 2-D graph of the matrix data at $iy = \langle \text{pos} \rangle$ versus $\log(x)$. Equivalent to a combination of **logx** (see I.4.6) and **f.x** above, but remembers this old setting of linear or logarithmic abscissa scale.

<n1,I> f.ly

Plots the 2-D graph of the matrix data at $ix = \langle \text{pos} \rangle$ versus $\log(y)$. Equivalent to a combination of **logx** (see I.4.6) and **f.y** above, but remembers this old setting of linear or logarithmic abscissa scale.

plwl

Plots in the "wall" domain (applicable to wall data only). This creates 2-D plots, with vertical lines separating the various wall segments.

<pos,I> trg.w

Plots the 2-D graph of target data at $iz = \langle \text{pos} \rangle$ versus wall coordinate. Vertical lines will be added to the plot to delimit the various wall segments.

<pos,I> trg.z

Plots the 2-D graph of target data at element $i = \langle \text{pos} \rangle$ versus z .

<n1,I> line

Chooses between plotting against x -coordinate (default, $\langle n1 \rangle = 1$) or y -coordinate ($\langle n1 \rangle = 2$). Not sure this works.

mesh

Plots the mesh. The colour of the mesh can be chosen with the **mcol** (see [I.4.8](#)) command.

trigmesh

Plots the triangular grid. The colour of the mesh can be chosen with the **mcol** (see [I.4.8](#)) command.

surf

Produces a "patch" plot where each cell is plotted in the colour associated with its value. This can be useful when the data is noisy and contour plots take excessive time. See figure [I.1](#). Recommended.

esurf

Produces a "patch" plot where each triangle from the Eirene grid is plotted in the colour associated with its value.

trgsurf

Produces a "patch" plot of the target domain. Note that the side in contact with the plasma is at the bottom of the target computational domain.

traj

Plots trajectories. The following command

```
echo phys a4p vesl sep outl 70 1 70 36 1000 0 1 1 traj | b2plot
```

produced the results shown in figure [I.2](#).

The command takes a variable number of integer arguments, the minimum number being 8.

The first 4 numbers specify the bottom left and top right coordinates of a rectangle in which the test particles are to be started (ix_{start} iy_{start} ix_{end} iy_{end}) (0's will be replaced by the appropriate limit). The next number specifies the amount of test particles to be started (if positive the particles are started weighted by volume, if negative then the weighting function is taken off the matrix stack). This is then followed by a flag (1 for a movie, otherwise normal trajectories), and then the species indices come next. The last argument specifies the number of species. The use of **uuse** and **vvse** allows one to replace the particle velocities matrices used by default to compute the trajectories with data from the matrix stack.

The following example uses 2000 particles, weighted by particle number, over the entire domain, for the 6 charge states of carbon.

```
stdl glob
a4p
phys
-1.2 zmin 1.2 zmax
1.0 lbsz
'rgb.pal' cltb
vesl sep outl
2000 t_#p
1e-3 t_pt
```

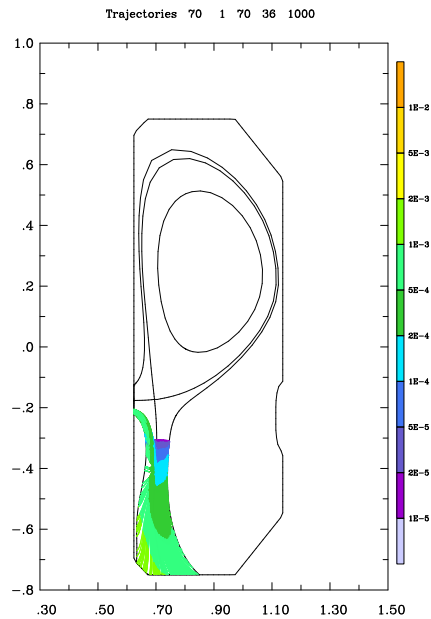


Figure I.2: Trajectory plot.

```
na vol m*
0 0 0 0 -2000 1 2 3 4 5 6 7 6 traj
drop
```

From the code documentation:

```
C      This routine is used to plot trajectories of particles using the
C      velocity fields uu and vv. To call this routine
C      one must specify at least 8 parameters: ixmin, iymmin, ixmax, iymax,
C      # of particles, 1 or 0 for movie or lines, species indices (at least
C      one), and last but not least the total number of species indices one
C      has given in the step before. A valid call could be:
C      0 0 95 35 100 0 0 1 traj
C      This should plot the trajectories of 100 particles which start
C      in any of the cells between cell (0,0) (southwest corner) and
C      cell (95,35) (northeast corner) as lines for hydrogen (0=H).
C      Note that the cell and species indices start at zero!
C      The number of time steps (plots) is given by the "t_#p" command.
C      The total flight time of the particles is given by the "t_pt" command.
C      The velocity fields uu and vv can be overwritten using the
C      "uuse" and "vvse" commands. Their defaults values are ua and fna/na/sy
C      respectively. If the number of particles is positive, then the
C      particles are born randomly, with the probability of being born in a
C      cell being proportional to the cell volume. This probability weighting
C      can be changed by calling "traj" with a negative number of particles.
C      Then the weighting function is whatever field is on top of the matrix
C      stack. For example, this set of commands does the following:
C      fnax na m/ psxperp 0 zsel uuse      ! utraj set to poloidal neutral flow velocity
C      fnay na m/ psy 0 zsel vvse          ! vvtraj set to radial neutral flow velocity
C      1000 t_#p                            ! 1000 time steps
C      1e-4 t_pt                            ! followed for a total time of 1.e-4 s.
C      na 0 zsel                            ! weighted with neutral density
```

```

C      0 0 0 0 -1000 0 0 1 traj      ! over the entire physical domain (0 0 0 0)
C      ! 1000 particles using non-default weighting function (-1000)
C      ! trajectories plotted (0)
C      ! for species 0 (=H neutrals) (0)
C      ! total of species considered is 1 (1)
C      =====

```

The trajectories are coloured according to the length of time the particle has traveled along it (illustrated by the label bar to the right of the graph). This label bar can be set to be logarithmic (default) with the **logt** command, or linear with the **lint** command.

I.4.2 Data manipulating functions

abs

Take the absolute value of the matrix data values.

eabs

Take the absolute value of the Eirene data values.

wabs

Take the absolute value of the wall data values.

trgabs

Take the absolute value of the target data values.

chs

Change sign of matrix data.

echs

Change sign of Eirene data.

trgchs

Change sign of target data.

wchs

Change sign of wall data.

ddw

Differentiates the wall data with respect to coordinate along wall perimeter.

ddx

Differentiates the matrix data with respect to poloidal coordinate x .

ddy

Differentiates the matrix data with respect to radial coordinate y .

trgddp

Differentiates the target data with respect to perpendicular coordinate.

trgddw

Differentiates the target data with respect to coordinate along wall perimeter.

trgddz

Differentiates the target data with respect to depth coordinate z .

inttrgw

Integrates data from the target stack along wall direction, summing over all wall segments.

inttrgw*

Integrates data from the target stack along wall direction, each segment being done separately.

inttrgz

Integrates data from the target stack in depth direction.

intw

Integrates data from the wall stack, summing over all wall segments.

intw*

Integrates data from the wall stack, each segment being done separately.

intx

Integrates data from the matrix stack along the x -direction, taking connectivities into account.

inty

Integrates data from the matrix stack along the y -direction.

pne

Divides the matrix data by the electron density n_e .

pwn

Divides the wall data by the local electron density n_e .

pswa

Divides the wall data by the wall element area.

psx

Divides the matrix data by s_x — the contact cell area between cells in the x -direction.

psxperp

Divides the matrix data by s_x — the cell area perpendicular to the x -direction.

psy

Divides the matrix data by s_y — the cell area perpendicular to the y -direction.

pvol

Divides the matrix data by the volume of the cells.

psta

Divides the Eirene data by the triangular element area.

pstv

Divides the Eirene data by the triangular element volume.

sumt

Sums the Eirene data over all triangles, and then copies the sum to all elements.

sumtrgw

Sums the target data along the wall coordinate, and then copies the sum to all elements.

sumtrgz

Sums the target data depth-wise, and then copies the sum to all z -positions.

sumw

Sums the wall data, and then copies the sum to all elements.

sumx

Sums the matrix data over the entire x -direction, and then copies the sum to all x -positions.

sumy

Sums the matrix data over the entire y -direction, and then copies the sum to all y -positions.

<is_{start},I> <is_{end},I> sumz

Sums the matrix data over the z -direction from $\langle is_{start} \rangle$ to $\langle is_{end} \rangle$, and then copies the sum to the first z position. If $\langle is_{start} \rangle = \langle is_{end} \rangle = 0$, then the sum is done over all species.

<is_{start},I> <is_{end},I> esumz

Sums the Eirene data over the z -direction from $\langle is_{start} \rangle$ to $\langle is_{end} \rangle$, and then copies the sum to the first z position. If $\langle is_{start} \rangle = \langle is_{end} \rangle = 0$, then the sum is done over all species.

<is_{start},I> <is_{end},I> wsumz

Sums the wall data over the z -direction from $\langle is_{start} \rangle$ to $\langle is_{end} \rangle$, and then copies the sum to the first z position. If $\langle is_{start} \rangle = \langle is_{end} \rangle = 0$, then the sum is done over all species.

<is_{start},I> <is_{end},I> zrng

Selects the subset over species extended from species index $\langle is_{start} \rangle$ to species index $\langle is_{end} \rangle$ from the matrix data.

<is_{start},I> <is_{end},I> ezrng

Selects the subset over species extended from species index $\langle is_{start} \rangle$ to species index $\langle is_{end} \rangle$ from the Eirene data.

<is_{start},I> <is_{end},I> wzrng

Selects the subset over species extended from species index $\langle is_{start} \rangle$ to species index $\langle is_{end} \rangle$ from the wall data.

<i,I> esel

Propagates the Eirene data value from element $\langle i \rangle$ to all other elements.

<i,I> wsel

Propagates the wall data value from element $\langle i \rangle$ to all other elements.

<ix,I> xsel

Propagates the matrix data value in the specified position $\langle ix \rangle$ to all other x values.

<iy,I> ysel

Propagates the matrix data value in the specified position $\langle iy \rangle$ to all other y values.

<is,I> zsel

Propagates the matrix data value in the specified position $\langle is \rangle$ to all other z (species) values.

<i,I> trgwsel

Propagates the target data value from the specified element $\langle i \rangle$ to all other elements.

<is,I> ezsel

Propagates the Eirene data value in the specified position $\langle is \rangle$ to all other z (species) values.

<iz,I> trgtsel

Propagates the target data value at the specified depth $\langle iz \rangle$ to all other z values.

<is,I> wzsel

Propagates the wall data value in the specified position $\langle is \rangle$ to all other z (species) values.

rigz

Toggles the inclusion of ignorable cells in the plots. The ignorable cells are the guard cells at the boundaries and cells from isolated regions.

log

Computes logarithm base 10 of a matrix.

ln

Computes natural logarithm of a matrix.

sqrt

Takes square root of a matrix.

exp

Computes exponential of a matrix.

elog

Computes logarithm base 10 of Eirene data.

eln

Computes natural logarithm of Eirene data.

esqrt

Takes square root of Eirene data.

eexp

Computes exponential of Eirene data.

trglog

Computes logarithm base 10 of target data.

trgln

Computes natural logarithm of target data.

trgsqrt

Takes square root of target data.

trgexp

Computes exponential of target data.

wlog

Computes logarithm base 10 of wall data.

wln

Computes natural logarithm of wall data.

wsqrt

Takes square root of wall data.

wexp

Computes exponential of wall data.

I.4.3 Arithmetic commands**m+**

Adds the two arrays on the top of the matrix stack.

m-

Subtracts the array on the top of the matrix stack from the array second from the top.

m*

Multiplies the two arrays on the top of the matrix stack.

m/

Divides the array second from the top of the matrix stack by the array on the top. Division by a zero element will yield zero.

mmin

Minimum of two top matrices on the stack.

mmax

Minimum of two top matrices on the stack.

em+

Adds the two arrays on the top of the Eirene stack.

em-

Subtracts the array on the top of the Eirene stack from the array second from the top.

em*

Multiplies the two arrays on the top of the Eirene stack.

em/

Divides the array second from the top of the Eirene stack by the array on the top. Division by a zero element will yield zero.

emmin

Minimum of two top arrays on the Eirene stack.

emmax

Maximum of two top arrays on the Eirene stack.

<r1,R> <r2,R> r+

Adds the two real values $\langle r1 \rangle$ and $\langle r2 \rangle$ on the top of the real stack.

<r1,R> <r2,R> r-

Subtracts the real value on the top of the real stack $\langle r2 \rangle$ from the real value second from the top $\langle r1 \rangle$.

<r1,R> <r2,R> r*

Multiplies the two real values $\langle r1 \rangle$ and $\langle r2 \rangle$ on the top of the real stack.

<r1,R> <r2,R> r/

Divides the real value second from the top of the real stack $\langle r1 \rangle$ by the real value on the top of the real stack $\langle r2 \rangle$. Division by 0 will yield an error and do nothing.

<r1,R> <r2,R> r**

Raises the real value second from the top of the real stack $\langle r1 \rangle$ to the power of the real value on the top $\langle r2 \rangle$. 0^0 will yield an error and do nothing.

<number,R> rm+

Adds <number> to the array on the top of the matrix stack.

<number,R> rm-

Subtracts <number> from the array on the top of the matrix stack.

<number,R> rm*

Multiplies the array on the top of the matrix stack by <number>.

<number,R> rm/

Divides the array on the top of the matrix stack by <number>. Division by zero will yield an error and do nothing.

<number,R> rm**

Raises each element of the array on the top of the matrix stack to the <number>-th power. If <number>=0, the result will be 1.0 for all cells.

<number,R> rmmi

Minimum of a matrix and <number>.

<number,R> rmma

Maximum of a matrix and <number>.

<number,R> rem+

Adds <number> to the array on the top of the Eirene stack.

<number,R

>

rem-

Subtracts <number> from the array on the top of the Eirene stack.

<number,R> rem*

Multiplies the array on the top of the Eirene stack by <number>.

<number,R> rem/

Divides the array on the top of the Eirene stack by <number>. Division by zero will yield an error and do nothing.

<number,R> rem**

Raises each element of the array on the top of the Eirene stack to the <number>-th power. If <number>=0, the result will be 1.0 for all cells.

<number,R> remmi

Minimum of an Eirene array and <number>.

<number,R> remma

Maximum of an Eirene array and <number>.

<number,R> rtrgm+

Adds <number> to the array on the top of the target stack.

<number,R> rtrgm-

Subtracts <number> from the array on the top of the target stack.

<number,R> rtrgm*

Multiplies the array on the top of the target stack by <number>.

<number,R> rtrgm/

Divides the array on the top of the target stack by <number>. Division by zero will yield an error and do nothing.

<number,R> rtrgm**

Raises each element of the array on the top of the target stack to the <number>-th power. If <number>=0, the result will be 1.0 for all elements.

<number,R> rtrgm mi

Minimum of target data and <number>.

<number,R> rtrgm ma

Maximum of target data and <number>.

<number,R> rwm+

Adds ⟨number⟩ to the array on the top of the wall stack.

<number,R> rwm-

Subtracts ⟨number⟩ from the array on the top of the wall stack.

<number,R> rwm*

Multiplies the array on the top of the wall stack by ⟨number⟩.

<number,R> rwm/

Divides the array on the top of the wall stack by ⟨number⟩. Division by zero will yield an error and do nothing.

<number,R> rwm**

Raises each element of the array on the top of the wall stack to the ⟨number⟩-th power. If ⟨number⟩=0.0, the result will be 1.0 for all elements.

<number,R> rwmmin

Minimum of wall data and ⟨number⟩.

<number,R> rwmmax

Maximum of wall data and ⟨number⟩.

<number,R> meq

If a matrix element is equal to ⟨number⟩, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> mne

If a matrix element is not equal to ⟨number⟩, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> mge

If a matrix element is greater than or equal to ⟨number⟩, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> mgt

If a matrix element is greater than ⟨number⟩, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> mle

If a matrix element is less than or equal to $\langle \text{number} \rangle$, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> mlt

If a matrix element is less than $\langle \text{number} \rangle$, sets this matrix element to 1.0. Otherwise, sets it to zero.

pos

If a matrix element is positive, sets this matrix element to 1.0. Otherwise, sets it to zero.

<number,R> emeq

If an Eirene array element is equal to $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

<number,R> emne

If an Eirene array element is not equal to $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

<number,R> emge

If an Eirene array element is greater than or equal to $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

<number,R> emgt

If an Eirene array element is greater than $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

<number,R> emle

If an Eirene array element is less than or equal to $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

<number,R> emlt

If an Eirene array element is less than $\langle \text{number} \rangle$, sets this element to 1.0. Otherwise, sets it to zero.

epos

If an Eirene array element is positive, sets this element to 1.0. Otherwise, sets it to zero.

<i1,I> <i2,I> i+

Adds the two integer values $\langle i1 \rangle$ and $\langle i2 \rangle$ on the top of the integer stack.

<i1,I> <i2,I> i-

Subtracts the integer value on the top of the integer stack $\langle i2 \rangle$ from the integer value second from the top $\langle i1 \rangle$.

<i1,I> <i2,I> i*

Multiplies the two integer values $\langle i1 \rangle$ and $\langle i2 \rangle$ on the top of the integer stack.

<i1,I> <i2,I> i/

Divides the integer value second from the top of the integer stack $\langle i1 \rangle$ by the integer value on top of the integer stack $\langle i2 \rangle$. This is integer division! Division by 0 will yield an error and do nothing.

<i1,I> <i2,I> i**

Raises the integer value second from the top of the integer stack $\langle i1 \rangle$ to the power of the integer value on the top $\langle i2 \rangle$. 0^0 will yield an error and do nothing.

<number,R> trgeq

If a target element is equal to $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

<number,R> trgne

If a target element is not equal to $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

<number,R> trgge

If a target element is greater than or equal to $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

<number,R> trggt

If a target element is greater than $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

<number,R> trgle

If a target element is less than or equal to $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

<number,R> trgl

If a target element is less than $\langle \text{number} \rangle$, sets this target element to 1.0. Otherwise, sets it to zero.

trgmmmin

Minimum of two top matrices on the target stack.

trgmmax

Maximum of two top matrices on the target stack.

trgm+

Adds the two arrays on the top of the target stack.

trgm-

Subtracts the array on the top of the target stack from the array second from the top.

trgm*

Multiplies the two arrays on the top of the target stack.

trgm/

Divides the array second from the top of the target stack by the array on the top. Division by a zero element will yield zero.

trgpos

If a target element is positive, sets this target element to 1.0. Otherwise, sets it to zero.

wm+

Adds the two arrays on the top of the wall stack.

wm-

Subtracts the array on the top of the wall stack from the array second from the top.

wm*

Multiplies the two arrays on the top of the wall stack.

wm/

Divides the array second from the top of the wall stack by the array on the top. Division by a zero element will yield zero.

wmmin

Minimum of two top data sets on the wall stack.

wmmax

Minimum of two top data sets on the wall stack.

<number,R> wmeq

If a wall element is equal to $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

<number,R> wmne

If a wall element is not equal to $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

<number,R> wmge

If a wall element is greater than or equal to $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

<number,R> wmggt

If a wall element is greater than $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

<number,R> wmle

If a wall element is less than or equal to $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

<number,R> wmlt

If a wall element is less than $\langle \text{number} \rangle$, sets this wall element to 1.0. Otherwise, sets it to zero.

wpos

If a wall element is positive, sets this wall element to 1.0. Otherwise, sets it to zero.

I.4.4 Stack manipulating commands

drop

Deletes the array on the top of the matrix stack.

dup

Duplicates the array on the top of the matrix stack.

swap

Swaps the top two arrays on the matrix stack.

over

Copies the array second from the top to the top of the matrix stack.

list

Lists the contents of the matrix stack by printing the titles of the stored matrices to `stdout`.

mdims

Prints the dimensions and contents (i.e. data set titles) of the matrix stack to `stdout`.

rdrop

Deletes the real on the top of the real stack.

rdup

Duplicates the real on the top of the real stack.

rswap

Swaps the top two reals on the real stack.

rover

Copies the real second from the top to the top of the real stack.

rtostr

Writes the real on top of the real stack to the top of the string stack.

rtom

Copies the real on top of the real stack to the top of the matrix stack.

rtoe

Copies the real on top of the real stack to the top of the Eirene stack.

rtow

Copies the real on top of the real stack to the top of the wall stack.

rtot

Copies the real on top of the real stack to the top of the target stack.

rpri

Outputs the contents of the real stack to `stdout`, and to `b2plot.write` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/b2plot.write` (in the **SOLPS 5.x** environment) if **writ** is active.

idrop

Deletes the integer on the top of the integer stack.

idup

Duplicates the integer on the top of the integer stack.

iswap

Swaps the top two integers on the integer stack.

iover

Copies the integer second from the top to the top of the integer stack.

itor

Copies the integer on top of the integer stack to the top of the real stack.

itostr

Writes the integer on top of the integer stack to the top of the string stack.

itom

Copies the integer on top of the integer stack to the top of the matrix stack.

itoe

Copies the integer on top of the integer stack to the top of the Eirene stack.

itow

Copies the integer on top of the integer stack to the top of the wall stack.

itot

Copies the integer on top of the integer stack to the top of the target stack.

iprt

Outputs the contents of the integer stack to `stdout`, and to `b2plot.write` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/b2plot.write` (in the **SOLPS 5.x** environment) if **writ** is active.

scat

Concatenates the top two strings of the string stack.

sdrop

Deletes the string on the top of the string stack.

sdup

Duplicates the string on the top of the string stack.

sswap

Swaps the top two strings on the string stack.

sover

Copies the string second from the top to the top of the string stack.

sprt

Outputs the contents of the string stack to `stdout`, and to `b2plot.write` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/b2plot.write` (in the **SOLPS 5.x** environment) if **writ** is active.

<nx,I> <ny,I> <filename,S> saw

Reads the matrix of size $(\langle nx \rangle + 2) \times (\langle ny \rangle + 2)$ from file `<filename>` into the matrix stack.

<nx,I> <ny,I> <nz,I> <filename,S> saws

Reads the matrix of size $(\langle nx \rangle + 2) \times (\langle ny \rangle + 2) \times \langle nz \rangle$ from file `<filename>` into the matrix stack. Expects *nz* to match *ns* and that the data refers to the fluid plasma species.

<nx,I> <ny,I> <nz,I> <filename,S> sawx

Reads the matrix of size $(\langle nx \rangle + 2) \times (\langle ny \rangle + 2) \times \langle nz \rangle$ from file $\langle filename \rangle$ into the matrix stack. Expects nz to match *ns_ext* and that the data refers to the additional species followed by the external code to which **SOLPS** is coupled.

<nwall,I> <filename,S> wsaw

Reads the array of size $\langle nwall \rangle$ from file $\langle filename \rangle$ into the wall stack.

stoe

Copies the array on top of the matrix stack to the Eirene stack. Only the matrix elements from internal cells will be copied over, onto the corresponding triangular elements.

stow

Copies the array on top of the matrix stack to the wall data stack. Only the matrix elements from boundary cells will be copied over, onto the corresponding wall elements.

edrop

Deletes the array on the top of the Eirene stack.

edup

Duplicates the array on the top of the Eirene stack.

eswap

Swaps the top two arrays on the Eirene stack.

ecover

Copies the array second from the top to the top of the Eirene stack.

elist

Lists the contents of the Eirene stack by printing the titles of the stored arrays to `stdout`.

edims

Prints the dimensions and contents (i.e. data set titles) of the Eirene stack to `stdout`.

etos

Copies the array on top of the Eirene stack to the matrix stack. The average value of the Eirene elements from internal cells will be copied over, onto the corresponding matrix elements.

trgdrop

Deletes the array on the top of the target stack.

trgdup

Duplicates the array on the top of the target stack.

trgswap

Swaps the top two arrays on the target stack.

trgover

Copies the array second from the top to the top of the target stack.

trglist

Lists the contents of the target stack by printing the titles of the stored matrices to `stdout`.

trgtow

Copies the matrix on top of the target data stack onto the wall data stack. Only the top surface target data values will be copied onto the corresponding wall elements.

wdrop

Deletes the array on the top of the wall stack.

wdup

Duplicates the array on the top of the wall stack.

wswap

Swaps the top two arrays on the wall stack.

wover

Copies the array second from the top to the top of the wall stack.

wlist

Lists the contents of the wall stack by printing the titles of the stored matrices to `stdout`.

wtos

Copies the array on top of the wall data stack onto the matrix stack. Each wall element value is copied onto the corresponding boundary cell in the matrix stack. Inner cells of the matrix stack will contain zero.

wtot

Copies the array on top of the wall data stack onto the target stack. Each wall element value is copied onto the corresponding column of cells in the target stack.

<ix,I> xtow

Copies the data from the ix -th row of the array on top of the matrix stack to the wall data stack.

<iy,I> ytow

Copies the data from the iy -th column of the array on top of the matrix stack to the wall data stack.

I.4.5 Page format parameters

a4l

Sets the plot area for A4 landscape. Use `"ctrans -d ps.mono.land — a4page — lpr"`.

a4p

Sets the plot area for A4 portrait. Use `"ctrans -d ps.mono — a4page — lpr"`.

sq

Sets the plot area to be square. This ensures that the x and y scaling factors are the same.

<nx,I> <ny,I> page

Plots $\langle nx \rangle$ graphs in the x -direction and $\langle ny \rangle$ in the y -direction on the same page.

I.4.6 Graph format commands

both

Plots on both the physical and computational meshes.

comp

Plots on computational mesh only.

phys

Plots on the physical mesh only.

aspc

Toggles whether aspect ratio should be preserved for physical and computational domain plots (default `.true.`). If `.false.`, the plot region limits will be those defined from the *xmin xmax ymin ymax*, etc... values.

linf

Sets linear scaling for the function values (default).

logf

Sets logarithmic scaling for the function values.

lintrg

Sets linear scaling for the target data values (default).

logtrg

Sets logarithmic scaling for the target data values.

linxtrg

Sets linear abscissae for the target data values (default).

logxtrg

Sets logarithmic abscissae scaling for the target data values.

linx

Sets linear scaling for the abscissae values (default).

logx

Sets logarithmic scaling for the abscissae values.

xpar

Plots 2-D plots against parallel x distance.

xpol

Plots 2-D plots against poloidal x distance (default).

<n,I> ndec

Sets number of decades to $\langle n \rangle$ for logarithmic plots (default 10). If $n = 0$, restores the default value.

mmid

Toggles mapping to midplane coordinate for 2-D plots. The midplane is assumed to be located at $ix = jxa$. See commands **omp** (section I.4.10) and **jxa** (section I.4.10) for setting and knowing its value. Special cases: For double-null cases, HFS data is mapped to the inner midplane $ix = jxi$, and for private flux data, it is mapped to the distance below the X-point.

fpsi

Toggles plotting 2-D radial plots against the full poloidal flux coordinate Ψ . If turned on, then turns off **mpsi** and **rpsi**.

mpsi

Toggles plotting 2-D radial plots against the normalised poloidal flux coordinate ψ . If turned on, then turns off **fpsi** and **rpsi**.

rpsi

Toggles plotting 2-D radial plots against the square root of normalised poloidal flux coordinate ($\rho_{pol} = \sqrt{\Psi}$). If turned on, then turns off **fpsi** and **mpsi**.

I.4.7 Graph appearance commands

bw

Want black and white for some plots.

col

Want colour for some plots.

bold

Toggles having the graph frames, axes, ticks and labels be in bold type.

<number,I> font

Changes the current font type to <number>

- -25 TIMES ROMAN
- -26 TIMES BOLD
- -21 HELVETICA
- -22 HELVETICA BOLD
- -29 COURIER
- -30 COURIER BOLD

There are even more available fonts — refer to NCAR manpage.

In general, it is possible to change fonts within a text string to be printed on the graph, for the commands below taking a string as argument. For example, the command

```
'D-\Ga\R (photons m^{-3} s^{-1})' extl
```

would give an extra graph label containing "D- α (photons $m^{-3} s^{-1}$)".

To switch font properties inside the string, use a double backslash followed by one of the letters:

R Switch to Roman characters

G Switch to Greek characters

d Switch to default characters

g Switch to Greek font

r Switch to Roman font

n Switch to normal level

h Resize font 130%

t Resize font 70%

m Switch to math font

Exponents can be requested with a string such as '^ {exp}', subscripts with '_ {sub}', environment variables with '\$ {var}', or the result of shell commands with 'cmd'.

stdl

Sets standard label as *extralabel*:

```
'\${HOST} \${PWD} `date` T=\${TIME}'
```

<label, S> glbl

Sets the current graph label to <label>.

<label, S> elbl

Sets the current Eirene data graph label to <label>.

<number,I> <label, S> slbl

Sets the species label of species index $\langle \text{number} \rangle$ to $\langle \text{label} \rangle$ for the current graph.

<label, S> tlbl

Sets the current target graph label to $\langle \text{label} \rangle$.

<label, S> trghlbl

Sets the current target bar label to $\langle \text{label} \rangle$.

<label, S> wlbl

Sets the current wall graph label to $\langle \text{label} \rangle$.

<number,I> <label, S> wslbl

Sets the species label of species index $\langle \text{number} \rangle$ to $\langle \text{label} \rangle$ for the current wall graph.

<label, S> walllbl

Sets the current wall bar label to $\langle \text{label} \rangle$.

glob

Copies the contents of *extralabel* into the global header for all graphs.

<factor,R> glsz

Sets font size for global header to $\langle \text{factor} \rangle$ times default size (default = 1.0).

<factor,R> lbsz

Sets label size to $\langle \text{factor} \rangle$ times default size (default = 1.0). Does not apply to the global header.

<number,I> lfsz

Sets line label font size to $\langle \text{number} \rangle$ (default = 0). If $\langle \text{number} \rangle$ is between 0 and 3, then the size represents 1., 1.5, 2., and 3. times an 8-plotter address unit width. Otherwise, it represents the character width in plotter address units.

nice

Toggles whether graphs will use nice axes with rounded values (`.true.` by default).

noax

Does not plot axes.

<factor,R> lwid

Changes line width to $\langle \text{factor} \rangle$ times default width.

same

The next graphs occur on the same set of axes. Terminated with **new**.

new

Starts graphs on a new set of axes.

I.4.8 Overlays to contour and line plots**zero**

Toggles the plotting of dotted axes lines at zero.

<factor,R> zwid

Changes width of dotted axes lines at zero to $\langle \text{factor} \rangle$ times default width.

<ncol,I> zcol

Plots axes line at zero with colour $\langle \text{ncol} \rangle$ (default 1). The colours available by default are:

- 0 White (background)
- 1 Black (default)
- 2 Lavender
- 3 Dark Violet
- 4 Slate Blue
- 5 Royal Blue
- 6 Deep Sky Blue
- 7 Aqua
- 8 Green
- 9 Celeste
- 10 Chartreuse
- 11 Green Yellow
- 12 Yellow

- 13 Golden
- 14 Orange
- 15 Orange Red
- 16 Red
- 17 Orchid

These values can be overridden by use of the **cltb** command.

<ncol,I> dcol

Colour to be used for matrix data line plots (default 1). If **dcol** is set to a negative value, the data plot lines will use a succession of colours starting from |<dcol>| with increasing index. See **zcol** above for a list of available colours.

<ncol,I> tcol

Colour to be used for target data line plots (default 1). If **tcol** is set to a negative value, the data plot lines will use a succession of colours starting from |tcol| with increasing index. See **zcol** above for a list of available colours.

<ncol,I> wcol

Colour to be used for wall data line plots (default 1). If **wcol** is set to a negative value, the data plot lines will use a succession of colours starting from |wcol| with increasing index. See **zcol** above for a list of available colours.

<ncol,I> bcol

Colour to be used for printing label bar title (default 1). See **zcol** above for a list of available colours.

<ncol,I> fcol

Colour to be used for printing the graph frame, axes, ticks and labels (default 1). See **zcol** above for a list of available colours.

<ncol,I> gcol

Colour to be used for printing graph title (default 1). See **zcol** above for a list of available colours.

<ncol,I> hcol

Colour to be used for printing global header (default 1). See **zcol** above for a list of available colours.

<ncol,I> xcol

Colour to be used for printing extra label (default 1). See **zcol** above for a list of available colours.

<length,R> **arow**

Produces flow chart of arrays *up* and *vv* using a maximum arrow length of $\langle \text{length} \rangle$ meters. The use of *up* and *vv* can be superseded by the the commands **arox** and **aroy**, respectively, which will then use data from the matrix stack for computing the arrow directions instead.

<length,R> **earow**

Produces flow chart of Eirene array *vxa* and *vya* using a maximum arrow length of $\langle \text{length} \rangle$ meters. The use of *vxa* and *vya* can be superseded by the the commands **earox** and **earoy**, respectively, which will then use data from the Eirene stack for computing the arrow directions instead.

<widthfactor,R> **arww**

Changes default arrow width by $\langle \text{widthfactor} \rangle$ (default 1.0).

<ncol,I> **acol**

Plots overlaid arrows with colour $\langle \text{ncol} \rangle$ (default 1). See **zcol** above for a list of available colours.

<data file,S> **expf**

File from which to take experimental data to be plotted with code data.

expl

Toggles between plotting the experimental data as a line (default) or as a scatter plot.

<expsymbol,I> **exps**

Choice of the symbol to be used for the experimental data scatter plots (default 5). The numbering convention is the same as used in the `libgr` library from Eirene.

<data file,S> **wallf**

File from which to take experimental data to be plotted with wall data.

<ncol,I> **ecol**

Plots experimental data overlay with colour $\langle \text{ncol} \rangle$ (default 1). If **ecol** is set to a negative value, the data plot lines will use a succession of colours starting from $|\text{ecol}|$ with increasing index. See **zcol** above for a list of available colours.

<widthfactor,R> **ewid**

Changes the experimental data overlay line width by $\langle \text{widthfactor} \rangle$ (default 1.0).

<n,I> ytra

Choose transformation function ($\langle n \rangle_{ne.0}$) for radial coordinate to compare with experimental data, according to `n.s.r.z.a_inner` and `n.s.r.z.a_outer` files in the `$(SOLPSTOP)/data.local` or `$(SOLPSTOP)/data` directories.

sep

Toggles drawing of the separatrices.

<ncol,I> scol

Plots separatrices with colour $\langle ncol \rangle$ (default 1). Also applies to the vertical separators drawn for 2-D plots. See **zcol** above for a list of available colours.

<widthfactor,R> sepw

Changes the default width of the separatrices line by $\langle widthfactor \rangle$ (default 1.0). Also applies to the vertical separators drawn for 2-D plots.

vesl

Toggles drawing of the outline of vacuum vessel. The data for this option is obtained from the file `mesh.extra` if it exists. If the `mesh.extra` file is not found, does nothing.

<ncol,I> vcol

Plots overlaid vacuum vessel outline with colour $\langle ncol \rangle$ (default 1). See **zcol** above for a list of available colours.

<widthfactor,R> vwid

Changes the default width of the vessel outline by $\langle widthfactor \rangle$ (default 1.0).

grid

Toggles drawing of overlaid mesh.

trig

Toggles drawing of overlaid triangular grid.

<ncol,I> mcol

Sets colour of overlaid mesh and triangular grid to $\langle ncol \rangle$ (default 1). See **zcol** above for a list of available colours.

outl

Toggles plotting of plasma domain outline. When target elements overlays are also plotted, then the target elements outline is activated as well.

<ncol,I> ocol

Sets colour of overlaid plasma outline to $\langle \text{ncol} \rangle$ (default 1). See **zcol** above for a list of available colours.

<widthfactor,R> owid

Changes the default width of the overlaid plasma outline by $\langle \text{widthfactor} \rangle$ (default 1.0).

wall

Toggles plotting of additional wall data.

woff

Turns off plotting of additional wall data (default).

<widthfactor,R> wwid

Changes the default width of the wall data overlay line by $\langle \text{widthfactor} \rangle$ (default 10.0).

logw

Selects logarithmic scale for wall plots.

linw

Selects linear scale for wall plots (default).

logxw

Selects logarithmic abscissae for wall plots.

linxw

Selects linear abscissae for wall plots (default).

target

Turns on plotting of additional target data.

trgoff

Turns off plotting of additional target data (default).

I.4.9 Changing contour parameters

nocp

Does not plot contour plot.

<level_{min},R> <level_{max},R> <delta,R> clvd

Starts contour levels at <level_{min}> increasing by <delta> up to <level_{max}>.

<level_{min},R> <level_{max},R> <n,I> clvn

Produces <n> contour levels going from <level_{min}> to <level_{max}>.

<n,I> <level₁,R> ... <level_n,R> clvs

Sets <n> contour levels to the values specified by <label_i>.

0 clvs

can be used to go back to the default contour labelling scheme after use of **clvd**, **clvn**, **clvp** or **clvs**.

<npplvm,I> <npplvp,I> clvp

Sets up contour levels according to their proportion inside the plasma domain. The levels are arranged so that the integral of the absolute value of the data over the region between each two neighbouring level lines is the same. If <npplvm>.lt.0, then this integral is equal (approximately) to $\frac{1}{\|<npplvp>\|}$ of the integral over the whole volume. If <npplvm>.ge.0, then the regions of positive and negative data values are treated separately, with $\|<npplvp>\|$ data intervals for the positive and <npplvm> for the negative data range. In this case, if <npplvp>.ge.0 then 0 is added to the level list, otherwise, the negative and positive values closest to zero fall in the same zone.

<level_{min},R> <level_{max},R> <delta,R> eclvd

Starts contour levels at <level_{min}> increasing by <delta> up to <level_{max}> for Eirene data contour plot.

<level_{min},R> <level_{max},R> <n,I> eclvn

Produces <n> contour levels going from <level_{min}> to <level_{max}> for Eirene data contour plot.

<n,I> <level₁,R> ... <level_n,R> eclvs

Sets <n> contour levels to the values specified by <label_i> for Eirene data contour plots. As above, the command **0 eclvs** can be used to revert back to default behaviour.

<npplvm,I> <npplvp,I> eclvp

Sets up contour levels according to their proportion inside the Eirene domain, as per the **clvp** logic above.

<level_{min},R> <level_{max},R> <delta,R> tclvd

Starts contour levels at <level_{min}> increasing by <delta> up to <level_{max}> for target data contour plot.

<level_{min},R> <level_{max},R> <n,I> tclvn

Produces <n> contour levels going from <level_{min}> to <level_{max}> for target data contour plot.

<n,I> <level₁,R> ... <level_n,R> tclvs

Sets <n> contour levels to the values specified by <label_i> for target data contour plots. As above, the command 0 **tclvs** can be used to revert back to default behaviour.

<npplvm,I> <npplvp,I> tclvp

Sets up contour levels according to their proportion inside the target domain, as per the **clvp** logic above.

<level_{min},R> <level_{max},R> <delta,R> wclvd

Selects threshold levels at <level_{min}> increasing by <delta> up to <level_{max}>, for use in the wall data colour scheme.

<level_{min},R> <level_{max},R> <n,I> wclvn

Produces <n> threshold levels starting from <level_{min}> to <level_{max}> for the wall data colour scheme.

<n,I> <level₁,R> ... <level_n,R> wclvs

Sets <n> threshold levels to the values specified by <label_i> for the wall data colour scheme. As above, the command 0 **wclvs** can be used to revert back to default behaviour.

<npplvm,I> <npplvp,I> wclvp

Sets threshold levels for the wall data colour scheme, as per the **cvlp** logic above.

clvl

Switches off contour level lines to be drawn into filled contour plot.

eclv

Switches to empty contour levels (for output on BW-Laser).

felv

Switches to filled contour levels (default).

lacl

Switches to labels along contour lines.

lhor

Switches to horizontal labels (default).

lbon

Draws niveau labels into contour plot. (default).

lbof

Does not draw labels into contour plot.

lbsa

Toggles labeling of lines for 2-D plots with multiple curves (default . TRUE .).

spln

Uses spline interpolation for plot (very slow !!!).

<fill_opt,I> filo

Sets the fill option to <fill_opt>. If it equals 0, then colours are selected from the available colour range and adjacent levels might have the same value, otherwise modulo selection of colours is enabled and colour cycling might occur.

I.4.10 miscellaneous**neo_ic**

The command takes a single integer argument and uses this to choose which transport contributions to include from NEOART.

The choices are:

- 0** NEOART: classical,
- 1** NEOART: banana plateau,
- 2** NEOART: Pfirsch-Schlueter,
- 3** NEOART: banana plateau + Pfirsch-Schlueter, and

4 NEOART: all.

The available neoclassical quantities are:

dnneo Neoclassical particle diffusivity D_n ($m^2.s^{-1}$),

chineo Neoclassical heat conductivity χ ($m^2.s^{-1}$),

vnneo Neoclassical particle pinch v_n ($m.s^{-1}$), and

vwneo Neoclassical heat pinch v_w ($m.s^{-1}$).

Related quantities are:

lnn Density gradient length (m),

lth Temperature gradient length (m),

nuab Two species collision frequency (s^{-1}),

nua Total collision frequency (s^{-1}),

nustar Normalized collisionality,

vth Thermal velocity v_{th} ($m.s^{-1}$),

rhol Full-B Larmor radius (m),

deltal Ratio of Larmor radius to minimum gradient length,

wb Width of the banana orbit (m),

deltaw Ratio of banana width to minimum gradient length

deltawn Ratio of banana width to density gradient length

deltawt Ratio of banana width to temperature gradient length

omega Gyro-frequency ($rad.s^{-1}$)

deltab Ratio of collision frequency to gyro-frequency, and

deltapar Ratio of perpendicular to parallel dynamics.

dase

Overwrites the *dab2* array with the top of the matrix stack.

eirc

Copies *Eirene* data onto *B2.5* data. Will only be performed once. Specifically, the densities of neutrals species (when a match between *B2.5* and *Eirene* species can be found) computed from *Eirene* (*dab2*) are added to the fluid neutral densities from *B2.5* (*na*). Same for the mass density field *ro*. If the atomic particle fluxes *pfla* and *rfla* are available, then the fluxes and velocities for the neutral species are also updated, meaning the *fnax*, *fnay*, *ua*, *up*, and *vv* fields, along with the default values of the *uutraj*, *vvtraj*, *arrowx*, and *arrowy* arrays.

<filename,S> etal

Copies data from the *Eirene* tally output file *<filename>* to the *Eirene* data stack. Such files are written from the volume tally output in block 11 of the *Eirene* input file. This function assumes the tally file contains 2-D data on the *Eirene* triangular grid.

<value,R> evac

Assigns real value $\langle \text{value} \rangle$ to all triangles from the top of the Eirene data stack that do not map to a B2 . 5 grid cell.

writ

Writes out the labels and values for all simple 2-D plots to `b2plot.write` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/b2plot.write` (in the **SOLPS 5.x** environment). **write** is an alias for **writ**.

<filename,S> mpri

Writes out the top of the matrix stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`.

<filename,S> epri

Writes out the top of the Eirene stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`.

<x1,R> <y1,R> <x2,R> <y2,R> <x3,R> <y3,R> <x4,R> <y4,R> eavg

Writes out the volume average of the data on top of the Eirene stack over the rectangular region delimited by the four doublets of coordinates $[(x1,y1);(x2,y2);(x3,y3);(x4,y4)]$.

<ix1,I> <ix2,I> <iy1,I> <iy2,I> <filename,S> mtab

Writes out a table extracted from the matrix on the top of the stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`. 0 0 pairs for the *ix* or *iy* values will output the entire *ix* or *iy* range, respectively.

<filename,S> wpri

Writes out the top of the wall stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`.

<iw1,I> <iw2,I> <filename,S> wtab

Writes out a table extracted from the array on the top of the wall stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`. A 0 0 pair for the *iw* value will output the entire range.

<filename,S> trgpri

Writes out the top of the target stack to $\langle \text{filename} \rangle$ (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/⟨filename⟩`.

<iw1,I> <iw2,I> <depth1,I> <depth2,I> <filename,S> trgtab

Writes out a table extracted from the matrix on the top of the target stack to `<filename>` (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) `b2pl.exe.dir/<filename>`. 0 0 pairs for the *iw* or *depth* values will output the entire wall element or depth ranges, respectively.

<ix,I> <iy,I> <filename,S> outp

Writes out data from element (`<ix>`,`<iy>`) from the top of the matrix stack to `<filename>` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/<filename>` (in the **SOLPS 5.x** environment).

<i,I> <filename,S> eoutp

Writes out data from triangle element `<i>` from the top of the matrix stack to `<filename>` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/<filename>` (in the **SOLPS 5.x** environment).

<i,I> <iz,I> <filename,S> outptrg

Writes out data from element (`<i>`,`<iz>`) from the top of the target stack to `<filename>` (in the **SOLPS 4.x** environment) or `b2pl.exe.dir/<filename>` (in the **SOLPS 5.x** environment).

<i,I> <filename,S> outpw

Writes out data from element `<i>` from the top of the wall stack to `<filename>` (in the **SOLPS 4.x** environment) or (in the **SOLPS 5.x** environment) to `b2pl.exe.dir/<filename>`.

f30

Produces the `fort.30` file containing the geometry for use within Eirene.

f31

Produces the `fort.31` file containing the plasma state for use within Eirene.

tranprof

Outputs the transport profile information as a `b2.transport.parameters` file.

reso

Resets plotting output.

spec

Generates the spectral line emissivities from the Summers' data stored in `$SOLPSTOP/modules/B2.5/Database/Summers` (which can be superseded by the contents of `$SOLPSTOP/data.local/Summers`). One pre-selected diagnostic line per ionization stage is provided. Data is only provided for Helium, Boron, Carbon, Oxygen and Neon.

<is,I> <lambda,I> <delta,I> nspe

Generates the spectral line emissivities according to the ADAS database [6], stored in `$SOLPSTOP/modules/adas` or `$SOLPSTOP/data.local/adas`, for species index $\langle is \rangle$, centered around wavelength $\langle \lambda \rangle$ (in Å), with width $\langle \delta \rangle$ (in Å). **nspec** is an alias for **nspe**.

<n,I> mvsk

Sets the movie skip factor to $\langle n \rangle$ (default 1).

nese

Overwrites the n_e array with the top of the matrix stack.

tese

Overwrites the T_e array with the top of the matrix stack.

<ix,I> imp

Overwrites the inner midplane location index j_{xi} with $\langle ix \rangle$. See description of 'b2mwti_jxi' switch (see Appendix A.4) for the computation of its default value.

<ix,I> omp

Overwrites the outer midplane location index j_{xa} with $\langle ix \rangle$. See description of 'b2mwti_jxa' switch (see Appendix A.4) for the computation of its default value.

<iy,I> yref

Overwrites the radial reference location index iy_{ref} with $\langle iy \rangle$. See description of 'set_transport_iyref' switch (see Appendix A.4) for the computation of its default value (usually the separatrix location).

<igm,I> <iig,I> trgl

ANK target loading. Not yet written.

<n,I> ave

Averages over $\langle n \rangle$ chords for **chor** and **chvl**.

<device name,S> pdev

Sets value of PLPLOT device name to $\langle \text{device name} \rangle$.

<winid,I> swid

Sets value of WIN_ID environment variable (part of TCL/TK interface) to <winid>.

xyrz

Outputs *xmin*, *xmax*, *ymin*, *ymax*, *rmin*, *rmax*, *zmin*, and *zmax* to stdout.

<ig,I> user

Optional user routine. The choice of user routine is governed by <ig>. Currently coded functions are:

ig = -20 Produces a file `diagno.dat` with coordinates of the grid cell centers and densities, temperatures for all species

ig = -19 Produces a file `brooks.dat` with coordinates of the grid cell centers and the plasma parameters (density, temperatures, drift velocity) in these points

ig = -18 Produces input files for a DIVIMP run. Requires one field on the matrix stack and one filename on the string stack to indicate which field is to be produced and where it is to be written

ig = -17 Prints detailed particle balance for the cells around the lower x-point in file `x-point.dat`

ig = -16 Produces a file `krieger.dat` with the plasma densities, temperatures, fluid velocities in the cell centers

ig = -15 Produces data files for the neutral fluxes on the specified surfaces. File names are `ndt_*.dat` (not yet available)

ig = -14 Produces data files for the cell-face-integrated fluxes of ions on every target. File names are `fi_tg_*.dat`

ig = -13 Produces a file `radsources.dat` with coordinates of the grid cell centers and the radiation intensity (total and components) in these points - for bolometry optimisation

ig = -12 Produces a file with coordinates of the grid cell centers (`wuerz-g.dat`) and the plasma parameters (densities, temperatures, fluid velocities) (`wuerz-d.dat`) in the Eastern divertor

ig = -11 Creates tables with coordinates of the grid cell corners along the separatrices and along the magnetic surfaces, together with the wall geometry in specified zones for separate plotting, in files `gm_sep.dat`, `gm_sep_i.dat`, `gm_sep_o.dat`, `gm_SOL.dat`, `gm_SOL_i.dat`, `gm_SOL_o.dat`, `gm_PFR.dat`, `gm_PFR_u.dat`, and `gm_core.dat`, according to magnetic topology

ig = -10 Creates table of fluxes and temperatures at the grid edges in file `gridedge.dat`

ig = -9 Prints out the total number of nuclei of each kind in the system in file `nuclei.tot`

ig = -8 Creates files `pr.dat` (plasma pressure), `prn.dat` (neutral pressure), `prt.dat` (total pressure) and `sm.dat` (neutral momentum source)

ig = -7 Prints out the neutral data read from `fort.44` for debugging

ig = -6 Produces a file `zaveraev.dat` with coordinates of the grid cell centers and the plasma parameters (density, temperatures, neutral density) in these points

ig = -5 Writes the CII sources in the plasma for DIVIMP in file `CII_source.dvi`

ig = -4 Produces a map of model radiation emissivity on a rectangular grid for 3D calculation of the wall loading in files `fpri.dat`, `fpro.dat`, `fprp.dat`, `plenter.dat`, `plenter.grd`. Requires 11 numbers on the real stack for providing the emissivity profile shape and overall magnitude (see code in `b2pplent.F` routine for details)

ig = -3 Prints out the particle flux integrals onto the targets in file `particle.flx`

ig = -2 Produces a file `horton.dat` with coordinates of the grid cell centers and the plasma parameters (density, temperatures, neutral density, and so on) in these points

ig = -1 Produces a file `abramov.dat` with coordinates of the grid cell centers and the plasma parameters (density, temperatures, neutral density) in these points

0 ≤ ig ≤ ny Produces a file `oleg.dat` with plasma parameters (density, temperature, parallel flow velocity, parallel heat flux) against the distance along the magnetic field along flux surface index `<ig>`

ny < ig < 1000 Produces two files with estimated helium concentration (`fraction.he`) and neutral pressure (`pressure.ntr`) in the PFR for summary plots

ig ≥ 1000 Adds a line to the file `../sum_tmp/spec_pwr.dat` with the power entering both divertors, power radiated along the divertor legs, power radiated near the targets, areas of the separatrix branches and mean specific power radiated along the divertor legs per square meter. `<ig>` -1000 cells are treated as "close to the targets"

I.4.11 Still need to do

bala Computes departure from pressure balance along field line

shco currently not used (meant for TCL/TK interface)

t.r+ set *ratio* to `.true.` (default)

t.r- set *ratio* to `.false.` Not sure this does anything.

part partition colour map

wall_ank Add ANK wall data to contour plot

eof, exit, quit, end

Use of any of **eof**, **exit**, **end** or **quit** will terminate the **b2plot** run.

I.5 Commands taking a string parameter

I.5.1 <extra label> extl

Specifies an additional label to appear on the graphs.

Options to the 'extl' string

In your configuration file you can specify additional 'text' which is added to the title of a plot. There are several T_EX-like commands available:

- `\R` Switch to the "Roman" part of the PWRITX database
- `\G` Switch to the "Greek" part of the PWRITX database
- `\d` switches to default typeface
- `\g` switches to Greek typeface
- `\r` switches to standard roman typeface

- `\h` changes font size to huge
- `\t` changes font size to tiny
- `^` switches to superscript, e.g. word^{super}
- `_` switches to subscript, e.g. word_{sub}
- `\n` returns to normal level
- `\h` increases font size by 130 percent
- `\t` decreases font size by 70 percent

It is however not possible to iterate super-/subscript with any of the other options like Roman or Greek font type! This can be achieved by using `^` and `_`, closing super- or subscript with `\n`:

`^\glt\n\r` text expands to λ^r text

Environment variables can also be included by using, for example `${PWD}`. `PWD` could be any environment variable except `TIME`, `LABEL` or `SONNETFILE` (either lower or uppercase) where the values stored in the `B2SXDR` file are used (for **SOLPS 4.x** runs).

Commands to be executed externally by a shell and returning a value can be entered between ```s, e.g.

```
`date`
```

See section [I.4.7](#) for the label set by **stdl** for an example using environment variables as well as shell expansions.

I.5.2 <palette file> **cltb**

Reads in the specifications for an alternative palette. The data format is RGB triplets for each colour.

If the file does not exist in the current working directory, the directories `$SOLPSTOP/data.local/palettes`, `$SOLPSTOP/scripts.local/palettes` and `$SOLPSTOP/scripts/palettes` are searched in order.

Many different palettes are available and the user can easily create its own. For instance, the 'blue_to_red' color maps are useful for data fields that have a zero in the middle (such as Mach numbers or velocities). The integer number at the end of the filename indicates how many colours the palette contains. Figure [I.3](#) was produced using the commands below, and shows examples of the 'jet', 'blue_to_red', and 'rainbow' palettes.

```
3 1 page a4l
sep outl comp
2.0 lbsz 3.0 sepw
0. 192. 17 clvn te
'jet_16.pal' cltb 'jet palette' extl
surf

ti
'blue_to_red_16.pal' cltb 'blue_to_red palette' extl
surf

mach
-1.2 1.2 17 clvn
'rainbow_16' cltb 'rainbow palette' extl
surf
```

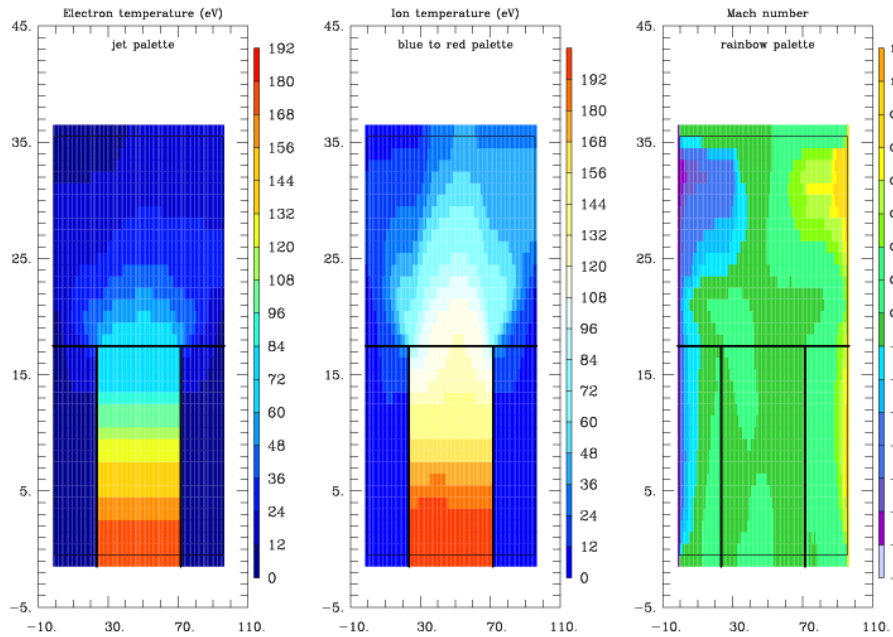


Figure I.3: Demonstration of various color palettes available in **b2plot**

I.5.3 <chord file> chor

Reads in chords from the specified file and calculates the line integrals for the currently specified data. The data format in the file consists of a label (in single quotes) followed by an integer specifying the number of points to be sampled along the line. Following this line are lines consisting of pairs of X , Y , and Z coordinates and a numeric value. The numeric value is used as the ordinate when plotting the chord integrals.

If the file does not exist in the current working directory, the directories `$SOLPSTOP/data.local/chords` and `$SOLPSTOP/data/chords` are searched in order.

I.5.4 <chord file> echor

Same as **chor** above, but does the line integral over the triangular grid.

I.5.5 <chord file> chpl

The chords specified by the chord file are plotted in the physical plane. Data format for the file is as for **chor** (see I.5.3).

Currently the following chord file examples (from AUG diagnostics) are provided (in the `$SOLPSTOP/doc/solps/chords` directory):

- BLS, figure I.4(a), use:

```
spec 'BLS.chords' chor
```

- BOL, figure I.4(b), use:

```
b2ra pvol abs 0 0 sumz 'BOL.chords' chor
```
- DCN, figure I.4(c), use:

```
ne 'DCN.chords' chor
```
- DIV, figure I.4(d), use:

```
spec 'DIV.chords' chor
```
- HAL, figure I.5(a), I.5(b), I.5(c), I.5(d) and I.5(e):

```
ahal mhal m+ 'HAL.chords' chor
```

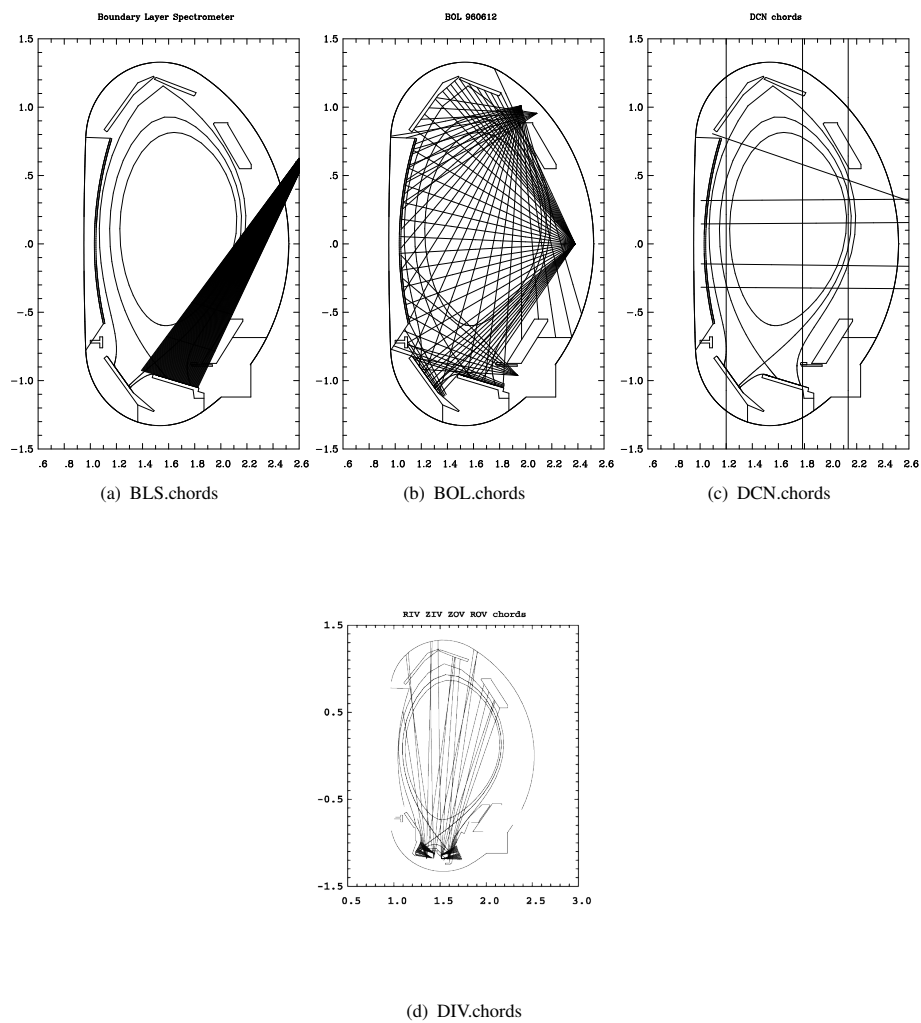


Figure I.4: BLS, BOL, DCN and DIV chords.

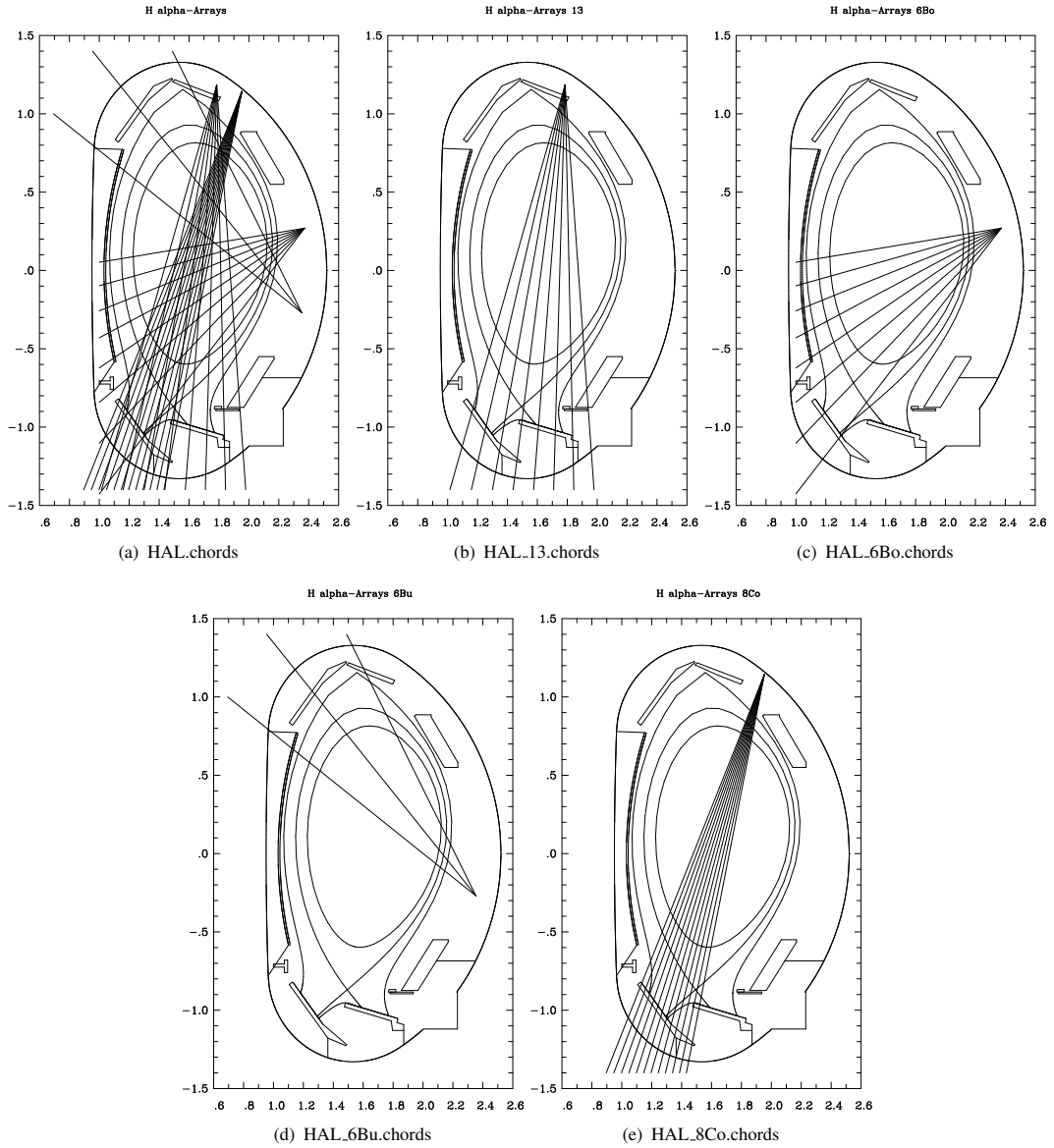


Figure I.5: HAL chords.

I.5.6 <command file> cmd

Further commands are taken from the specified file. If the file does not exist in the current working directory, the directories `$SOLPSTOP/scripts.local/commands` and `$SOLPSTOP/scripts/commands` are searched in order. The command file may contain comment lines, starting with `'*'`, `'!'` or `'#'`.

I.5.7 <CDF command file> cmdcdf

CDF commands used repetitively (for movie frames) are taken from the specified file. If the file does not exist in the current working directory, the directories `$SOLPSTOP/scripts.local/commands` and `$SOLPSTOP/scripts/commands` are searched in order.

I.5.8 <chord file> chbx

Chord integral averaged over three chords. Equivalent to using the **3 ave** command.

I.5.9 <position file> chvl

The value is plotted at the specified positions.

I.5.10 <flux info> flux

Needs to be compiled with `-DWOS`. String should contain shot number, time within the shot, and the diagnostic name. Then causes the equilibrium for this shot to overlay subsequent plots.

I.5.11 <chord file> chmp

Plots a contour plot of the chord contribution from the mesh elements.

I.5.12 <chord file> cam

Camera plot. The chord file needs to contain some additional information on the first line.

I.5.13 <chord file> echbx

Chord integral of Eirene data averaged over three chords. Equivalent to using the **3 ave** command.

I.5.14 <position file> echvl

The Eirene stack data value is plotted at the specified positions.

I.5.15 <chord file> echmp

Plots a contour plot of the chord contribution from the triangles.

I.5.16 <chord file> ecam

Camera plot of Eirene data. The chord file needs to contain some additional information on the first line.

I.5.17 <CDF file name> movi

Used for specifying the CDF file if a movie is to be made.

I.5.18 <istart,I> <iend,I> <varname,S> movie

Creates a movie for timeslice <istart> to time slice <iend> of variable <varname>. The list of available quantities in given in Appendix E. The time interval between frames can be controlled with the **mvsk** command. **nmovie** is an alias for **movie**.

I.5.19 <istart,I> <iend,I> <filename,S> <varname,S> omovie

Creates a movie for timeslice <istart> to time slice <iend> of variable <varname> read from file <filename>.

I.5.20 <sonnetfile> sonn

Reads in a new Sonnet geometry file.

I.5.21 <output_sonnet file> oson

Outputs the geometry in Sonnet ASCII format.

I.5.22 <z_{shift},R> <UEDGE grid file> uedgegrd

Outputs the geometry in UEDGE ASCII format. The grid can be shifted in the vertical direction by an amount of <z_{shift}> meters.

I.5.23 <filename> <H123> <reac> <crc> amju

Computes a reaction rate as per the Eirene specification. Consult the Eirene manual for details. **amjuel** is an alias for **amju**.

I.5.24 <reac> adas

Computes a reaction rate as per the ADAS specification. The <reac> string should be something of the form 'acd89'. The matrix obtained contains then all the rates from these files, for all species in the run, extracted from the files stored in \$SOLPSTOP/modules/adas or \$SOLPSTOP/data.local/adas.

I.6 Environment variables

B2PLOT_DEV is used to specify plotting to any combination of "x11", "ps" or "cgm" (default is "ps"). Under tcsh, use

```
setenv B2PLOT_DEV 'x11 ps'
```

SOLPSTOP used to specify the top of the SOLPS tree.

Environment variables can also be used in the **extl** command (see I.5.1). If not explicitly set by the user, they inherit their value from the setup.csh.* files in the \$SOLPSTOP directory.

I.7 Examples

Some simple (and not so simple) examples of plots that can be obtained with **b2plot** commands are shown in the next section.

I.7.1 Mesh plot

Here (figure I.6) we have two plots of the computational and physical grids, captioned with the commands used to obtain them.

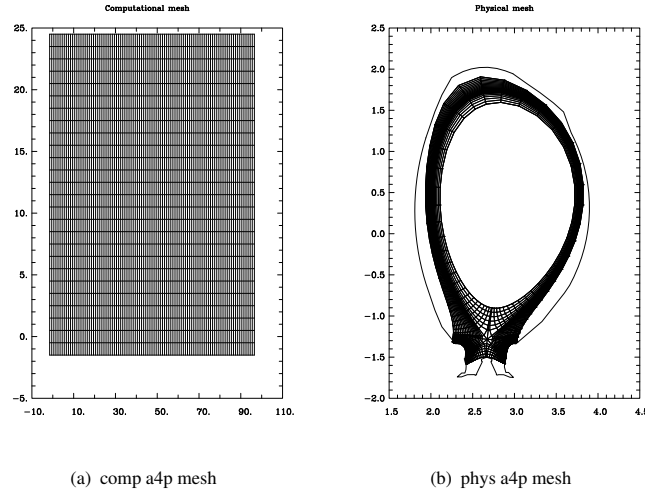


Figure I.6: Mesh plots.

I.7.2 Plotting plasma solution data

Data from the plasma solution, stored on the "matrix" stack, can be visualized as per figure I.7.

Note that here the physical domain plot (**phys** figure I.7(b)) has been modified to have more readable labels (**2.0 lbsz**), a logarithmic scale (**logf**) covering 3 decades (**3 ndec**), includes the magnetic separatrix (**sep**), the plasma outline (**outl**), the vacuum vessel (**vesl**), and has an extended vertical range (**-2.2 zmin 2.2 zmax**). Note however that the plotting program always attempts to maintain "nice" axis bounds and conserve the graph aspect ration (unless **aspc** is toggled).

Other examples:

In figure I.8(b), we obtain the ratio of ion to electron temperature, while figure I.8(a) is a region plot (like the ones in Fig. 3.1, obtained with **sep vesl outl 0 nreg surf**, and Fig. 3.2, obtained with **sep 0 nreg surf**).

I.7.3 Chord plots

In figure I.9 we have an example of chord plots as applied to the KL2 diagnostic on JET.

The first **chpl** plot (I.9(a)) gives the physical position of the chords, as obtained from the JET/KL2.chords file in the \$SOLPSTOP/data/chords directory. The second **chpl** plot (I.9(b)), in landscape mode **a4l**, shows the emissivity of a C⁺ line (species 3) at $\lambda = 6581 \pm 1 \text{ \AA}$ (**3 6581 1 nspec**). The **extl** instruction gives the extra label on the second title line.

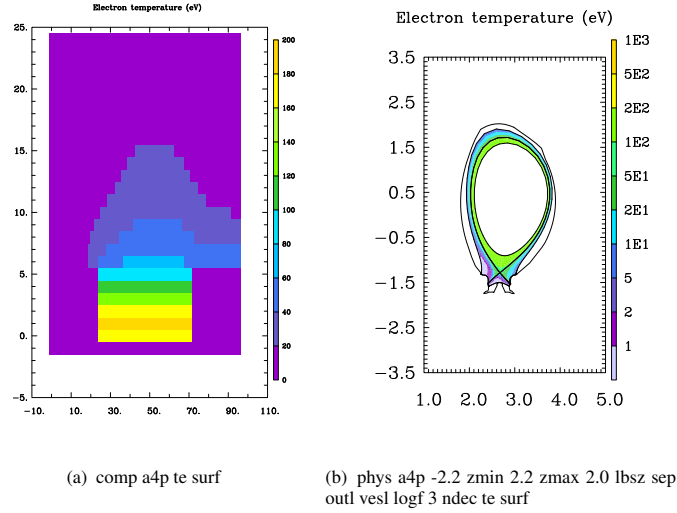


Figure I.7: Electron temperature plots.

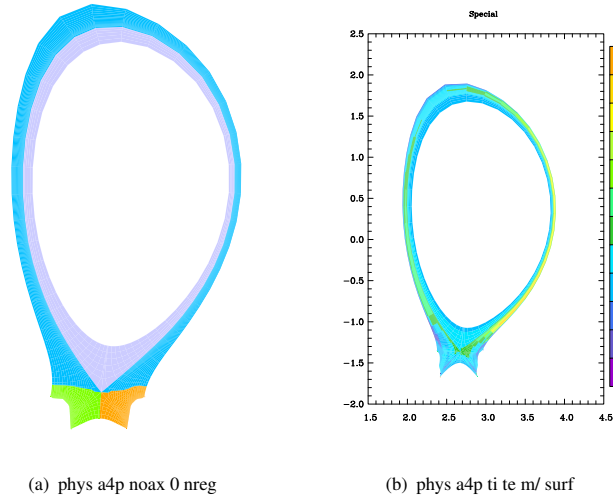


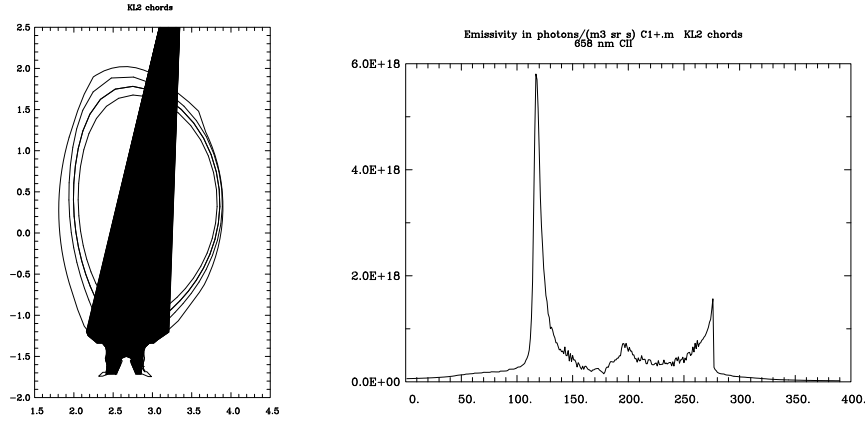
Figure I.8: Other solution data plots.

I.7.4 Wall plots

In figure I.10, we show examples of wall plots.

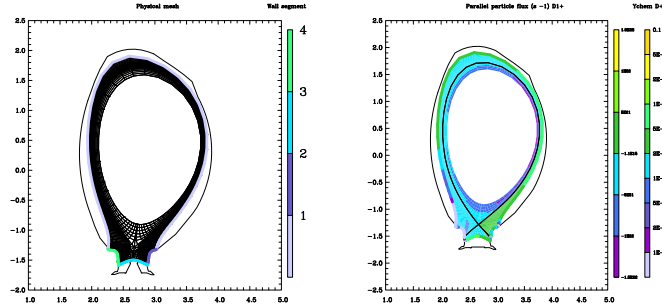
The wall segment indices are illustrated in figure I.10(a), obtained with the **0 nseg** command. We impose the contour levels to be at the values 1 through 4 with the **4 1. 2. 3. 4. wclvs** command. The thickness of the drawn wall line is given by **10.0 wwid**. **wall** asks for the wall data overlay on top of the requested **mesh** plot.

Another more complex example is given in figure I.10(b). It combines a **surf** plot of the D^+ poloidal particle flux (**fnax 1 zsel**, as D^+ has species index 1), with a **wall** plot of the chemical sputtering yield for D^+ (**sputchem 1 wzsel**). The latter is done on a logarithmic scale (**logw**), covering 3 decades (**3 ndec**). The label for the colour bar for the wall data is shortened with **'Ychem D+' wallblbl**. The wall colour bar is the rightmost one. Not choosing the **zsel** and **wzsel** options would have given a series of plots, one per species, with their individual poloidal particle flux and chemical sputtering yield. If only the **wzsel** option is used, one obtains a series of **surf** plots with all the same "wall" data overlay. If only the **zsel** option is used, then the **wall** overlay will come from the first species, namely D^0 in this



(a) a4p vesi outl sep 'JET/KL2.chords' chpl (b) a4l phys 3 6581 1 nspec '658 nm CII' extl 'JET/KL2.chords' chor

Figure I.9: Chord plots applied to JET.



(a) phys 0 nseg 4 1. 2. 3. 4. wclvs 10.0 wwld (b) vesi sep phys wall 10.0 wwld sputchem 1
wall mesh wzsel 3 ndec logw fnax 1 zsel 'Ychem D+'
wallblbl surf

Figure I.10: Wall plots.

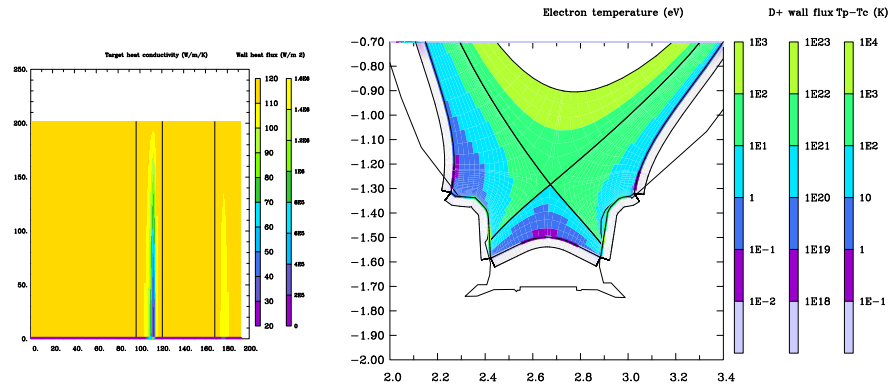
case. It is therefore recommended to always make sure that the species dimensions of the "matrix" and "wall" data match!

I.7.5 Target plots

In figure I.11, we show examples of target plots.

The first plot (figure I.11(a)) shows the target heat conductivity (**kappa**) in the target computational domain (**comp** and **trgsurf**). The vertical separators (**sep**) indicate the segment boundaries. The overlay line at $y=0$, i.e. at the target surface in contact with the plasma is the **wall** data, here the incoming plasma heat flux (**wallhflx**).

The second plot (figure I.11(b)) is a triple plot, containing a plasma quantity from the "matrix" stack, (here **te**), with **wall** and **target** overlays. The "matrix" (**logf**), "wall" (**logw**) and "target" (**trglog**) data are all plotted on a logarithmic scale. The target data is obtained with the command



(a) comp sep 10.0 wwid wallhflx wall kappa (b) a4l outl vesl 2.2 rmin 3.2 rmax -1.0 zmax -1.75 zmin sep 8.0 wwid logw te
trgsurf logf 5 ndec cooltemp wtot trgm- phys 'Tp-Tc (K)' trgbibl wallflux 1 wzsel 'D+
wall flux' wallblbl wall target logtrg surf

Figure I.11: Target plots.

```
cooltemp wtot trgm-
```

which takes the "wall" quantity **cooltemp**, the temperature at the cooled end of the target plate, copies it over to the "target" stack (**wtot**), and subtracts it from the preceding "target" stack value (here the target temperature, which is loaded onto the stack by default at startup). So we obtain the amount by which the target has been heated by the plasma, and label it with '**Tp-Tc (K)**' **trgbibl**. The wall overlay data is the D^+ particle flux to the wall (**wallflux 1 wzsel**), labeled with '**D+ wall flux**' **wallblbl**. The colour bars, from left to right, correspond to the "matrix", "wall" and "target" data arrays.

I.7.6 Radiation plots

The following was used to produce a plot (figure I.12) of the emissivity arising out of line radiation for C and Ne.

```
! b2plot < b2rad.plot ; ctrans -d ps.land.color gmeta | a4page > ! tmp.ps
a4l 2 1 page lbof logf phys
b2ra pvol abs 2 8 sumz 1.0e4 fmin 1.0e8 fmax 'Carbon' extl surf
b2ra pvol abs 9 19 sumz 1.0e4 fmin 1.0e8 fmax 'Neon' extl surf
```

The first line is a comment indicating how **b2plot** should be run (the file containing the above lines was called **b2rad.plot**), and how the PostScript file should be generated.

The second line sets up an A4 landscape format with two graphs next to each other, sets labelling of contour levels off, logarithmic scaling and plotting on the physical domain only.

The third and fourth lines plot the line radiation data summed over species indices 2 to 8 and 9 to 19, respectively, assuming a D+C+Ne case, setting the data minimum to 10^4 and maximum to 10^8 , labelling the graphs "Carbon" and "Neon" respectively.

Please note that **b2ra** only contains the line radiation emanating from the fluid species followed by B2.5. If you wish to also include the radiation emanating from the neutral kinetic species computed by **Eirene**, you need to add contributions from **neutrad**, **molrad**, and **ionrad**, which correspond to the radiation due to atoms, molecules, and molecular ions, respectively. Please be mindful however that **neutrad** has dimensions *natmi* (**molrad** *nmoli* and **ionrad** *nioni*) while **b2ra** has dimensions *ns*. Moreover, if you are intending to compute the total radiation rate, you should also include the bremsstrahlung contribution from **b2br**. You would then have as your set of commands, now using a D+He+C example:

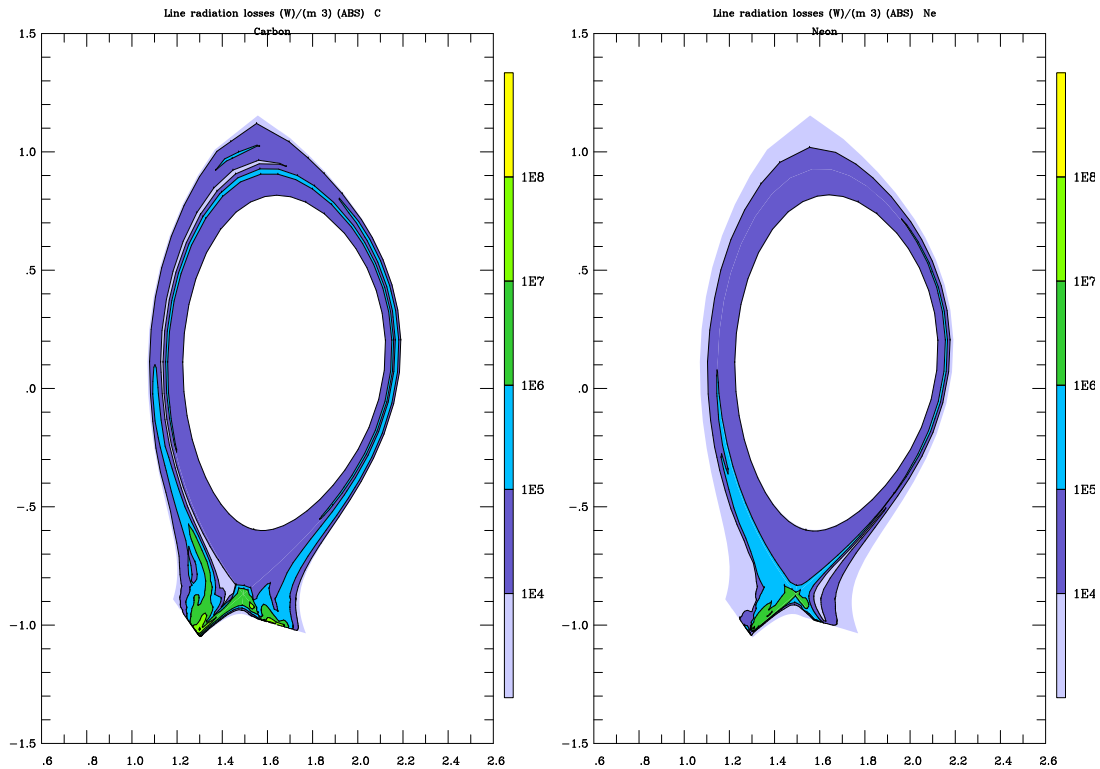


Figure I.12: C and Ne radiation.

```
a4l 3 1 page lbof logf phys sep vesl outl 1.5 lbsz 5 ndec
b2ra b2br m+ abs 0 1 sumz
neutrad 0 zsel m+ molrad 0 zsel m+ ionrad 0 zsel m+ pvol 0 ' ' slbl
'Total radiation loss (W/m^{3})' glbl 'Hydrogenic species' extl surf
b2ra b2br m+ abs 2 4 sumz
neutrad 1 zsel m+ pvol 0 ' ' slbl
'Total radiation loss (W/m^{3})' glbl 'Helium species' extl surf
b2ra b2br m+ abs 5 11 sumz
neutrad 2 zsel m+ pvol 0 ' ' slbl
'Total radiation loss (W/m^{3})' glbl 'Carbon species' extl surf
```

I.7.7 Fluxes and values plot

The following was used to calculate and plot integrated values of fluxes and some local quantities along various lines. See figure I.14.

```
'${HOST} ${PWD} `date` T=${TIME}' extl glob
a4p 2 2 page
phys
feex sumy 19 f.x drop
feey sumx 57 f.y drop
feix sumy 19 f.x drop
feiy sumx 57 f.y drop
fnix 1 1 sumz sumy 19 f.x drop
fnix 2 7 sumz sumy 19 f.x drop
fnix 8 17 sumz sumy 19 f.x drop
```

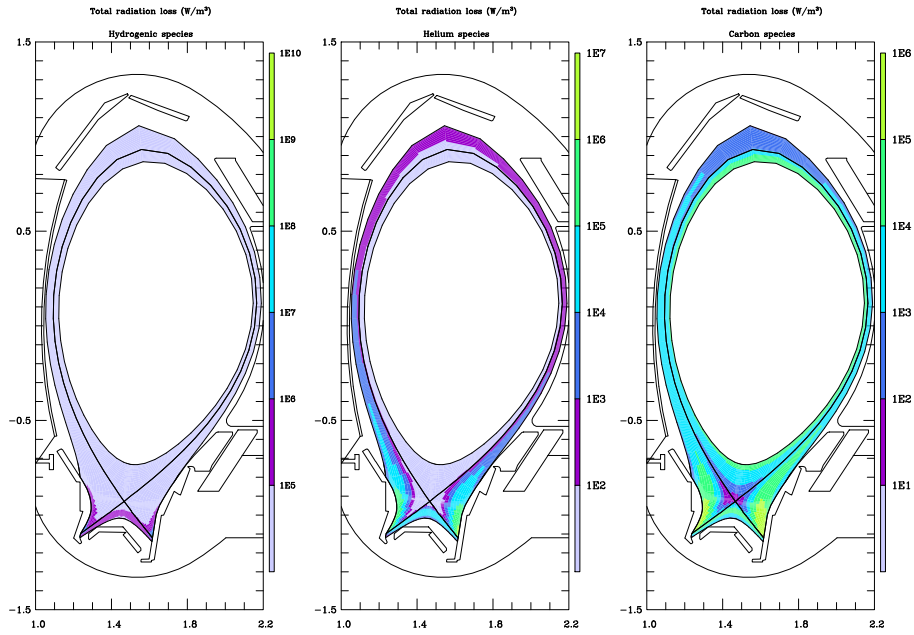


Figure I.13: Total radiation for a D+He+C example case.

```
fni 18 19 sumz sumy 19 f.x drop
fni 1 1 sumz sumx 57 f.y drop
fni 2 7 sumz sumx 57 f.y drop
fni 8 17 sumz sumx 57 f.y drop
fni 18 19 sumz sumx 57 f.y drop
phys
ni 1 1 sumz 0.0 fmin 0.0 fmax 57 f.y drop
ni 2 7 sumz 0.0 fmin 0.0 fmax 57 f.y drop
ni 8 17 sumz 0.0 fmin 0.0 fmax 57 f.y drop
ni 18 19 sumz 0.0 fmin 0.0 fmax 57 f.y drop
te 0.0 fmin 0.0 fmax 57 f.y drop
ti 0.0 fmin 0.0 fmax 57 f.y drop
zeff 0.0 fmin 0.0 fmax 57 f.y drop
ne 0.0 fmin 0.0 fmax 57 f.y drop
```

This example plots a series of line plots in a 2×2 format on an A4 portrait page. Liberal use is made of **sumx** and **sumy** to produce appropriate sums, and of **sumz** to sum over the same species. **f.x** and **f.y** are used to plot the line plots. Setting both **fmin** and **fmax** to 0.0 means each plot is scaled separately — otherwise the scaling is done on all of the data.

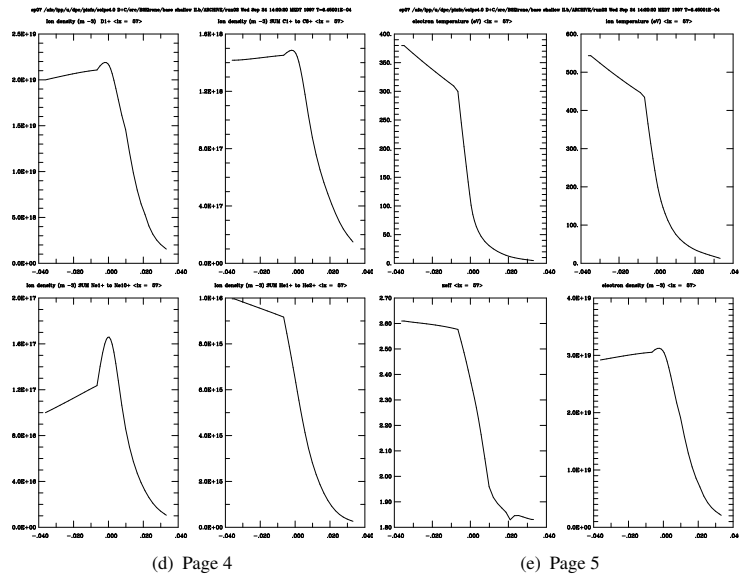
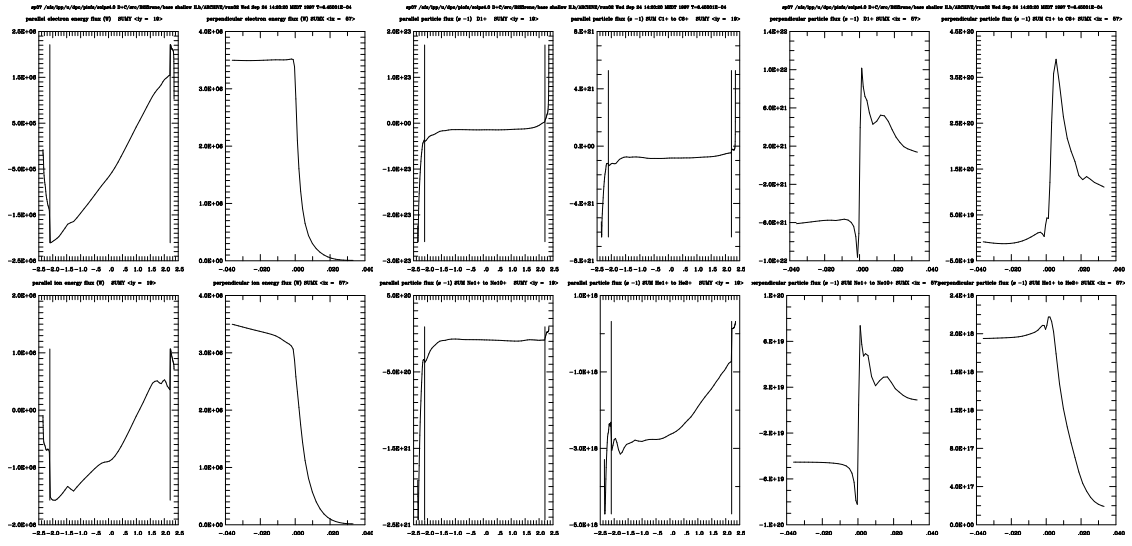


Figure I.14: Line plots of fluxes and other quantities.

Appendix J

Miscellaneous

J.1 DIMENSIONS.F

The file `$(SOLPSTOP)/modules/B2.5/src/include/DIMENSIONS.F` and its local version `$(SOLPSTOP)/modules/B2.5/src/include.local/DIMENSIONS.F` contain key dimensions that determine the sizes of arrays that can be defined by namelists in `b2.*.parameters` files (for compatibility with certain compilers).

Upon the first creation of the local version of the `DIMENSIONS.F` file, one must make the `Makefile` aware of its presence by doing the following:

```
stop
gmake listobj
gmake tags
gmake depend
gmake
```

so that the local file is used. The last `gmake` command may of course be replaced by `gmake solps(_debug)`, `gmake solps_mpi(_debug)`, `gmake all(_debug)` or `gmake all_mpi(_debug)` as needed.

The quantities defined in `DIMENSIONS.F` are:

DEF_ISOEXTRA the maximum number of extra isolating cells

DEF_MAXCHORDS Maximum number of chords to be considered (optional, internal default value is 10000)

DEF_NADD the maximum number of additional vacuum cells outside the gridded domain in `Eirene`

DEF_NATM the maximum number of neutral atomic species to be handled by `Eirene`

DEF_NBC the maximum number of boundary conditions provided in `b2.boundary.parameters` (optional, internal default value is 20)

DEF_NCHEN the maximum number of energy bins for the diagnostic chords in `Eirene`

DEF_NCUT the maximum number of "cuts" (2 for single null cases, 4 for double nulls)

DEF_NFL the maximum number of charged fluids passed from `B2.5` to `Eirene`

DEF_NGSTAL the maximum number of grids on which spatially resolved surface tallies are computed

DEF_NION the maximum number of molecular ion species to be handled by `Eirene`

DEF_NISO the maximum number of isolated regions within the `B2.5` computational domain (optional, internal default value is 10)

DEF_NLIM the maximum number of additional surfaces to be handled by DivGeo

DEF_NMOL the maximum number of neutral molecular species to be handled by Eirene

DEF_NPHID the maximum number of toroidal standard surfaces

DEF_NPLS the maximum number of background plasma species to be handled by Eirene

DEF_NPRNL the maximum number of particles that can be saved in the time dependent arrays

DEF_NREAC the maximum number of reaction cards that can be created by Uinp

DEF_NRDS the maximum number of molecular dissociation reaction cards that can be created by Uinp

DEF_NREC the maximum number of recombination reaction cards that can be created by Uinp

DEF_NREL the maximum number of elastic collisions reaction cards that can be created by Uinp

DEF_NSD the maximum number of plasma species handled in B2 . 5

DEF_NSPCSFRFG the maximum number of groups of Eirene surfaces for which the data can be transferred from B2 . 5 (DG specification "Surface special")

DEF_NSRFS the maximum number of surfaces in a stratum

DEF_NSTRA the maximum number of Eirene strata

DEF_NSTS the maximum number of non-default standard surfaces to be handled by Eirene

DEF_NTOR the maximum number of toroidal periodicity boundaries

DEF_NXD the maximum number of cells in the x (parallel or poloidal) direction not including two guard cells

DEF_NYD the maximum number of cells in the radial direction not including two guard cells

HITS_PER_CHORD Maximum number of chord points per cell (optional, internal default value is 50)

The current default settings of DIMENSIONS . F, which allow for a 128×36 DN geometry case including species H, D, T, He, Be and C, are:

```

! Common dimensions
!
! version : 01.12.98 21:42
!
#ifdef DIMENSIONS_DEFINED
#define DIMENSIONS_DEFINED

!
! parameters that are common to Eirene and B2
!

! NOTE: DEF_NXD should not include the additional cells to handle the cuts
#define DEF_NXD 128
#define DEF_NCUT 4
#define DEF_ISOEXTRA 0
#define DEF_NYD 36
#define DEF_NPHID 1
#define DEF_NSTRA (5*DEF_NCUT/2+2)*DEF_NATM+1
#define DEF_NFL DEF_NSD
#define DEF_NPLS (DEF_NSD-DEF_NATM)+(DEF_NATM+DEF_NMOL)*(DEF_NATM+DEF_NMOL)
#define DEF_NATM 6
#define DEF_NMOL 3
#define DEF_NION 3
#define DEF_NLIM 300
#define DEF_NSTS 50
#define DEF_NTOR 60
!*** Max. number of groups of Eirene surfaces for which the data can
!*** be transferred from B2 (DG specification "Surface special")
#define DEF_NSPCSFRFG 40

```

```

! new! [2002.04.22]
#define DEF_NADD 4
#define DEF_NPRNL 50000
! new! [2002.06.14]
#define DEF_NSRFS 4

!
! parameters that are unique to B2
!

#define DEF_NSD 42
#define DEF_NBC 6*(DEF_NCUR/2+1)
#define DEF_NISO 10

!
! parameters that are unique to Eirene
!

#define DEF_NGSTAL 1

!
! parameters needed by uinp
!

#define DEF_NREAC 60
#define DEF_NREI 70
#define DEF_NREC DEF_NATM
#define DEF_NREL (DEF_NATM+DEF_NMOL)*(DEF_NATM+DEF_NMOL)

#endif

```

J.2 Making movies

J.3 Compiling B2.5

(See also section 2.2.1, page 33.)

The following is the Makefile provided with the Master in the sbb directory:

```

# Test whether necessary environment variables are defined; if not, exit
ifndef HOSTNAME
    $(error HOSTNAME not defined)
endif
ifndef COMPILER
    $(error COMPILER not defined)
endif

SRCB2   = ${PWD}
SRCDIR  = ${SRCB2}/src
DOCDIR  = ${SRCDIR}/documentation
PYTHON  = python
INCLUDE =
TAGSLIST =

MAKES = ${SRCB2}/Makefile
DEFINES = ${B25_DEFINES} ${SOLPS_CPP}
# Include global SOLPS compiler settings
ifndef SOLPS_CPP
NODENAME = $(shell echo `hostname`)
ifeq ($(shell [ -e ${SOLPSTOP}/SETUP/config.${HOSTNAME}.${COMPILER} ] && echo yes || echo no),
    yes)
    include ${SOLPSTOP}/SETUP/config.${HOSTNAME}.${COMPILER}

```

```

    MAKES += ${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER} ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}
else
    $(warning ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER} not found.)
endif
else
    MAKES += ${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER} ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}
endif
ifeq ($(shell [ -e ${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}.local ] && echo yes || echo no ),yes)
    MAKES += ${SOLPSTOP}/SETUP/setup.csh.${HOST_NAME}.${COMPILER}.local
endif
ifeq ($(shell [ -e ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}.local ] && echo yes || echo no ),yes)
    include ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}.local
    MAKES += ${SOLPSTOP}/SETUP/config.${HOST_NAME}.${COMPILER}.local
endif

ifdef USE_EIRENE
    ifndef SOLPSTOP
        $(error SOLPSTOP not defined, but trying to compile with Eirene)
    endif
endif
ifdef SOLPS_MPI
    ifndef USE_MPI
        USE_MPI = -DUSE_MPI
    endif
endif
ifdef SOLPS_OPENMP
    ifndef USE_OPENMP
        USE_OPENMP = -D_OPENMP
    endif
endif

# Default prefix for OBJDIR: standalone
PREF_OBJDIR = standalone
ifdef USE_EIRENE
    PREF_OBJDIR = couple_SOLPS-ITER
endif

# Extensions for object directories when various options are used
ifdef USE_MPI
    EXT_MPI = .mpi
endif
ifdef USE_OPENMP
    EXT_OPENMP = .openmp
endif
ifdef USE_IMPGYRO
    EXT_IMPGYRO = .ig
endif
ifdef SOLPS_DEBUG
    EXT_DEBUG = .debug
    IMAS_AMNS_DEBUG = yes
else
    IMAS_AMNS_DEBUG = no
endif

# Directory where objectcode/binaries will be created
OBJDIR = ${SRCB2}/builds/${PREF_OBJDIR}.${HOST_NAME}.${COMPILER}${EXT_OPENMP}${EXT_MPI}${EXT_IMPGYRO}${EXT_DEBUG}

# If compiling with Eirene, look in default place for Eirene sources/lib
ifdef USE_EIRENE
    SRCEIR = ${SOLPSTOP}/modules/Eirene/src
    EIRDIR = ${SOLPSTOP}/modules/Eirene/builds/couple_SOLPS-ITER.${HOST_NAME}.${COMPILER}${EXT_MPI}${EXT_IMPGYRO}${EXT_DEBUG}
endif

```

```

endif
ifdef SOLPSTOP
NCSDIR = ${SOLPSTOP}/scripts/nc2text_simple
NCODIR = ${SOLPSTOP}/scripts/${HOST_NAME}.${COMPILER}
ifdef LD.NETCDF
    NC2TXT = $(shell echo `which nc2text`)
endif
endif

ifeq ($(shell [ -e ${OBJDIR}/LISTOBJ ] && echo yes || echo no ),yes)
    include ${OBJDIR}/LISTOBJ
endif
    include ${SRCB2}/config/compile
    MAKES += ${SRCB2}/config/compile ${SRCB2}/config/config.${HOST_NAME}.${COMPILER}
ifeq ($(shell [ -e ${SRCB2}/config/config.${HOST_NAME}.${COMPILER}.local ] && echo yes || echo no
),yes)
    include ${SRCB2}/config/config.${HOST_NAME}.${COMPILER}.local
    MAKES += ${SRCB2}/config/config.${HOST_NAME}.${COMPILER}.local
endif

# Add external includes first
ifdef NCDIR
INCLUDE += -I${NCDIR}/include
endif

ifdef USE.IMPGYRO
INCLUDE += ${MPI.CPP}
else
ifdef USE.MPI
INCLUDE += ${MPI.CPP}
else
INCLUDE += -I${SRCDIR}/mpi_dummy
endif
endif

ifdef MDSPLUS_DIR
ifndef LD.MDSPLUS
LD.MDSPLUS ?= -L${MDSPLUS_DIR}/lib -lMdsLib_client
endif
INCLUDE += -I${MDSPLUS_DIR}/include
ifndef SOLPS_CPP
DEFINES += -DMDSPLUS
endif
endif

# If compiling with Paraview Catalyst
ifdef LD.CATALYST
SRCCAT = ${SRCDIR}/catalyst
INCLUDE += $(shell paraview-config --include)
endif

# Local includes
BASEDIR = ${OBJDIR}
INCLLOCAL = ${SRCDIR}/include.local
MODLOCAL = ${SRCDIR}/modules.local
SRCLOCAL = ${SRCB2}/src.local
SOLPS4 = ${SOLPSTOP}/modules/solps4-5/src/solps4_B2
EIR4 = ${SOLPSTOP}/modules/solps4-5/src/Eirene_modules
ifeq ($(shell [ -d ${MODLOCAL} ] && echo yes || echo no ),yes)
INCLUDE += -I${MODLOCAL}
TAGSLIST += ${MODLOCAL}/*.F
endif
ifeq ($(shell [ -d ${SRCLOCAL} ] && echo yes || echo no ),yes)
INCLUDE += -I${SRCLOCAL}
TAGSLIST += ${SRCLOCAL}/*.F
endif
ifeq ($(shell [ -d ${INCLLOCAL} ] && echo yes || echo no ),yes)

```

```

INCLUDE += -I${INCLOCAL}
TAGSLIST += ${INCLOCAL}/*. *
endif
INCLUDE += -I${SRCDIR}/common -I${SRCDIR}/include
SOLPS4INCLUDE = -I${SOLPSTOP}/modules/solps4-5/src/B2_include
TAGSLIST += ${SRCDIR}/include/*. * ${SRCDIR}/common/*. * ${SRCDIR}/common/COUPLE/*.F ${SRCDIR}/*. *.
F ${SRCDIR}/*. *.F90 ${DOCDIR}/*.xml

ifdef USE_MPI
DEFINES += ${USE_MPI}
else
SOLPS4INCLUDE += -I${SOLPSTOP}/modules/solps4-5/src/Eirene_commons
endif
ifdef USE_OPENMP
DEFINES += ${USE_OPENMP}
endif
ifdef USE_IMPGYRO
DEFINES += ${USE_IMPGYRO}
endif
ifdef PERFMON
DEFINES += ${PERFMON}
endif
ifdef USE_EIRENE
DEFINES += ${USE_EIRENE}
endif
ifdef SOLPS_DEBUG
DEFINES += -DDBG
endif

# Switches to disable individual OpenMP regions, for debugging
# uncomment to activate
#
# It is not necessary to define all these flags to completely disable the
# OpenMP parallelization, in order to that, just compile without OpenMP
# compiler options (ifort -qopenmp or similar)

ifdef USE_OPENMP
#DEFINES += -DNO_OPENMP_B2XPFE
DEFINES += -DNO_OPENMP_B2SIFRTF
#DEFINES += -DNO_OPENMP_B2SIHS
#DEFINES += -DNO_OPENMP_B2SPCX
#DEFINES += -DNO_OPENMP_B2SPEL
#DEFINES += -DNO_OPENMP_B2SQCX
#DEFINES += -DNO_OPENMP_B2SQEL
#DEFINES += -DNO_OPENMP_B2SRDT
#DEFINES += -DNO_OPENMP_B2SRST
#DEFINES += -DNO_OPENMP_B2STCX
#DEFINES += -DNO_OPENMP_B2STEL
#DEFINES += -DNO_OPENMP_MYBLAS
#DEFINES += -DNO_OPENMP_SFILL
#DEFINES += -DNO_OPENMP_B2NEWS_LOOP1
DEFINES += -DNO_OPENMP_B2NEWS_LOOP2
#DEFINES += -DNO_OPENMP_B2NEWS_LOOP3
#DEFINES += -DNO_OPENMP_B2NEWS_LOOP4
#DEFINES += -DNO_OPENMP_B2NEWS_UNDERSCORE_LOOP1
DEFINES += -DNO_OPENMP_B2NEWS_UNDERSCORE_LOOP2
#DEFINES += -DNO_OPENMP_B2NEWS_UNDERSCORE_LOOP3
#DEFINES += -DNO_OPENMP_B2NEWS_UNDERSCORE_LOOP4
DEFINES += -DNO_OPENMP_B2NPMO
endif

VHEAD =
ifeq ($(shell [ -d ${MODLOCAL} ] && echo yes || echo no ),yes)
VHEAD+=${MODLOCAL}:
endif
ifeq ($(shell [ -d ${SRCLOCAL} ] && echo yes || echo no ),yes)
VHEAD+=${SRCLOCAL}:

```

```

endif
# Required spacing variables for substitution
space :=
space +=
empty :=
#VPATH=$(subst $(space),$(empty),${VHEAD}${SRCDIR}/modules:${SRCDIR}/b2aux:${SRCDIR}/convert:${SRCDIR}/documentation:${SRCDIR}/driver:${SRCDIR}/equations:${SRCDIR}/input:${SRCDIR}/output:${SRCDIR}/postprocessing:${SRCDIR}/preprocessing:${SRCDIR}/solvers:${SRCDIR}/sources:${SRCDIR}/transport:${SRCDIR}/utility:${SRCDIR}/b2plot:${SRCDIR}/user)
VPATH=${VHEAD}${SRCDIR}/modules:${SRCDIR}/b2aux:${SRCDIR}/convert:${SRCDIR}/documentation:${SRCDIR}/driver:${SRCDIR}/equations:${SRCDIR}/input:${SRCDIR}/output:${SRCDIR}/postprocessing:${SRCDIR}/preprocessing:${SRCDIR}/solvers:${SRCDIR}/sources:${SRCDIR}/transport:${SRCDIR}/utility:${SRCDIR}/b2plot:${SRCDIR}/user
FPATH := ${VPATH}
VPATH += ${SRCDIR}/test
VPATH += ${SRCDIR}/ids
VPATH += ${SRCDIR}/ids/archive
FFPATH += ${SRCDIR}/ids
FFPATH += ${SRCDIR}/modules

MODLIST =
MODLISTF =
MODLISTF90 =
ifeq ($(shell [ -d ${MODLOCAL} ] && echo yes || echo no ),yes)
MODLIST += ${MODLOCAL}/*.F
MODLISTF += ${MODLOCAL}/*.F
endif
MODLIST += ${SRCDIR}/*/b2mod_*.F ${SRCDIR}/*/b2mod_*.F90 ${SRCDIR}/ids/*.F90
MODLISTF += ${SRCDIR}/*/b2mod_*.F
MODLISTF90 += ${SRCDIR}/*/b2mod_*.F90 ${SRCDIR}/ids/*.F90

ifeq ($(shell [ -d ${SOLPS4} ] && echo yes || echo no ),yes)
S4LIST = ${SOLPS4}/*.F
endif
ifeq ($(shell [ -d ${EIR4} ] && echo yes || echo no ),yes)
ifdef USEEIRENE
E4LIST = ${EIR4}/precision.F ${EIR4}/parmmod.F ${EIR4}/braeir.F ${EIR4}/ccoupl.F ${EIR4}/clgin.F
        ${EIR4}/eirdiag.F ${EIR4}/ceirsrt.F
else
E4LIST = ${EIR4}/*.F
endif
endif

ifdef LD_CATALYST
ALLOBS = ${OBS:%.o=${OBJDIR}/%.o} ${OBJDIR}/cxxAdaptor.o
else
ALLOBS = ${OBS:%.o=${OBJDIR}/%.o}
endif

PROG_GE = b2pl.exe
PROG_GR = b2yg.exe b2yi.exe b2ym.exe b2yn.exe b2yp.exe b2yq.exe b2yr.exe
PROG_MN = b2mn.exe b2mnastra.exe
PROG_AM = b2ar.exe
PROG_XD = b2xd.exe
PROG_OE = b2ag.exe b2co.exe b2fu.exe b2ts.exe b2uf.exe b2ye.exe b2yt.exe b2ymb.exe b2yrp.exe
        b2ydm.exe calc_atomic_data.exe
PROG_OT = b2ah.exe b2ai.exe b2yi_gnuplot.exe b2yh.exe b2yv.exe b2fgmtry_mod.exe
        b2plasmastate_inspect.exe
PROG_90 = check_b2_output.exe
PROG_OP = b2op.exe
PROG_OQ = b2mn_opt.exe
PROG_MD = b2md.exe b2rd.exe
PROG_ID = b2_ual_write.exe b2_ual_write_b2mod.exe
PROG_TT = test_shrink_label.exe
PROG_NC = nc2text_simple.exe

EXCLUDELIST = ${patsubst %.exe, %\\., ${PROG_GE} ${PROG_GR} ${PROG_MN} ${PROG_AM} ${PROG_XD} ${

```

```

PROG.OE} ${PROG.OT} ${PROG_90} ${PROG_MD} ${PROG_OP} ${PROG_OQ} ${PROG_ID} ${PROG.TT}}
EXELIST = ${patsubst %.exe, %.o, ${PROG_GE} ${PROG_GR} ${PROG_MN} ${PROG_AM} ${PROG_XD} ${PROG_OE}
} ${PROG_OT} ${PROG_MD} ${PROG_OP} ${PROG_OQ}}
EX90LIST = ${patsubst %.exe, %.o, ${PROG_90} ${PROG_ID}}

GEEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_GE}}
GREXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_GR}}
XDEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_XD}}
MNEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_MN}}
AMEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_AM}}
OEEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_OE}}
OTEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_OT}}
O9EXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_90}}
OPEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_OP}}
OQEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_OQ}}
MDEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_MD}}
IDEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG_ID}}
TTEXE = ${patsubst %.exe, ${OBJDIR}/%.exe, ${PROG.TT}}
NCEXE = ${patsubst %.exe, ${NCODIR}/%.exe, ${PROG_NC}}

.PHONY: DEFAULT NOPLOT ALL VERSION mods clean depend listobj tags echo local force test
nc2text-simple nc2text

DEFAULT: VERSION ${MNEXE} ${AMEXE} ${OEEXE} ${OTEXE} ${O9EXE}
ALL: VERSION ${MNEXE} ${AMEXE} ${OEEXE} ${OTEXE} ${O9EXE} ${XDEXE}
NOPLOT: VERSION ${MNEXE} ${AMEXE} ${OEEXE} ${OTEXE} ${O9EXE}
ifdef NCARG_ROOT
ifeq ($(strip ${GLL_HOME}),)
$(warning B2.5 graphical post-processing programs may not work because GLLHOME is not defined.)
endif
DEFAULT: ${GEEXE} ${GREXE}
ALL: ${GEEXE} ${GREXE}
else
$(warning B2.5 graphical post-processing programs will not be compiled because NCARG_ROOT is not
defined.)
endif
ifdef MDSPLUS_DIR
DEFAULT: ${MDEXE}
ALL: ${MDEXE}
NOPLOT: ${MDEXE}
endif
ifdef PAR_OPT
DEFAULT: ${OPEXE} ${OQEXE}
ALL: ${OPEXE} ${OQEXE}
NOPLOT: ${OPEXE} ${OQEXE}
endif
ifdef IMAS_VERSION
DEFAULT: ${IDEXE}
ALL: ${IDEXE}
ids: ${IDEXE}
NOPLOT: ${IDEXE}
endif
ifdef SOLPS_CPP
ifdef LD_NETCDF
DEFAULT: ${NCEXE} nc2text
ALL: ${NCEXE} nc2text
NOPLOT: ${NCEXE} nc2text
endif
endif
MAIN: VERSION ${MNEXE}

ifdef USE_EIRENE
VPATH+=${SRCEIR}/modules:${SRCEIR}/interfaces/couple_SOLPS-ITER
MODLIST+=${SRCEIR}/modules/*.f ${SRCEIR}/modules/*.f90 ${SRCEIR}/interfaces/couple_SOLPS-ITER/
eirmod_*.f ${SRCEIR}/interfaces/couple_SOLPS-ITER/eirmod_*.F90
MODLISTF+=${SRCEIR}/modules/*.f ${SRCEIR}/interfaces/couple_SOLPS-ITER/eirmod_*.f
MODLISTF90+=${SRCEIR}/modules/*.f90 ${SRCEIR}/interfaces/couple_SOLPS-ITER/eirmod_*.F90

```

```

MNEXTA=${EIRDIR}/libeirene.a ${EIRDIR}/libgr.dummy.a ${EIRDIR}/ioflush.o
else
# MNEXTA=${EIRDIR}/eirmod_balanced_strategy.o ${EIRDIR}/eirmod_braeir.o ${EIRDIR}/eirmod_brascl.
o ${EIRDIR}/eirmod_braspoi.o ${EIRDIR}/eirmod_cadgeo.o ${EIRDIR}/eirmod_cai.o ${EIRDIR}/
eirmod_calstr_buffered.o ${EIRDIR}/eirmod_ccona.o ${EIRDIR}/eirmod_ccoupl.o ${EIRDIR}/
eirmod_ccrm.o ${EIRDIR}/eirmod_cestim.o ${EIRDIR}/eirmod_cfplk.o ${EIRDIR}/eirmod_cgeom.o ${
EIRDIR}/eirmod_cgrid.o ${EIRDIR}/eirmod_cgrptl.o ${EIRDIR}/eirmod_cinit.o ${EIRDIR}/
eirmod_clast.o ${EIRDIR}/eirmod_clgin.o ${EIRDIR}/eirmod_clogau.o ${EIRDIR}/eirmod_comnnl.o $
${EIRDIR}/eirmod_comprt.o ${EIRDIR}/eirmod_comsig.o ${EIRDIR}/eirmod_comsou.o ${EIRDIR}/
eirmod_comspl.o ${EIRDIR}/eirmod_comusr.o ${EIRDIR}/eirmod_comxs.o ${EIRDIR}/eirmod_coutau.o
${EIRDIR}/eirmod_cpes.o ${EIRDIR}/eirmod_cpl3d.o ${EIRDIR}/eirmod_cplmsk.o ${EIRDIR}/
eirmod_cplot.o ${EIRDIR}/eirmod_cpolyg.o ${EIRDIR}/eirmod_crand.o ${EIRDIR}/eirmod_crech.o ${
EIRDIR}/eirmod_cref.o ${EIRDIR}/eirmod_crefmod.o ${EIRDIR}/eirmod_csdvi.o ${EIRDIR}/
eirmod_csdvi_bkg.o ${EIRDIR}/eirmod_csdvi_cop.o ${EIRDIR}/eirmod_cspei.o ${EIRDIR}/
eirmod_cspez.o ${EIRDIR}/eirmod_cstep.o ${EIRDIR}/eirmod_ctetra.o ${EIRDIR}/eirmod_ccontext.o ${
EIRDIR}/eirmod_ctrcei.o ${EIRDIR}/eirmod_ctrig.o ${EIRDIR}/eirmod_ctorsurf.o ${EIRDIR}/
eirmod_cupd.o ${EIRDIR}/eirmod_cvarusr.o ${EIRDIR}/eirmod_czt1.o ${EIRDIR}/eirmod_eibra.o ${
EIRDIR}/eirmod_eirdiag.o ${EIRDIR}/eirmod_infcop.o ${EIRDIR}/eirmod_module_avltree.o ${EIRDIR}
/eirmod_mpi.o ${EIRDIR}/eirmod_octree.o ${EIRDIR}/eirmod_parmmod.o ${EIRDIR}/
eirmod_precision.o ${EIRDIR}/eirmod_solps.o
# EXCLUDELIST += ${patsubst ${OBJDIR}/%.o, %.o, ${MNEXTA} }
endif
ifdef LD_CATALYST
VPATH += :${SRCDIR}/catalyst
FFPATH += :${SRCDIR}/catalyst
endif

IDSMODS = ${PROG_ID}:%.exe=${OBJDIR}/%.${MOD}
MODULES = ${patsubst %.f90,%.o,${patsubst %.F90,%.o,${patsubst %.f,%.o,${patsubst %.F,%.o,${shell
basename -a ${MODLIST} } } } } }
MODOBJS = ${MODULES}:%.o=${OBJDIR}/%.o
MODMODS = $(filter-out ${IDSMODS},${MODOBJS}:%.o=%.${MOD})
SOLPS4OBJS = ${patsubst ${SOLPS4}/%.F,${OBJDIR}/%.o,${shell echo ${S4LIST} } }
EIR4MODS = ${patsubst ${EIR4}/%.F,${OBJDIR}/%.o,${shell echo ${E4LIST} } }
EIR4OBJS = ${patsubst ${EIR4}/%.F,${OBJDIR}/%.o,${shell echo ${E4LIST} } }

ifeq (${MOD},o)
LIBOBJS = $(filter-out ${MODOBJS},${ALLOBJS})
else
LIBOBJS = ${ALLOBJS}
endif

mods : ${MODMODS}

${DOCDIR}/b2cdci.F: ${DOCDIR}/b2input.xml ${DOCDIR}/b2cdci.py
-cd ${DOCDIR}; ${PYTHON} b2cdci.py || echo "! Error building b2cdci.F from b2input.xml" >
${DOCDIR}/b2cdci.F

${DOCDIR}/b2cdcn.F: ${DOCDIR}/b2input.xml ${DOCDIR}/b2cdcn.py
-cd ${DOCDIR}; ${PYTHON} b2cdcn.py || echo "! Error building b2cdcn.F from b2input.xml" >
${DOCDIR}/b2cdcn.F

ifdef USE_EIRENE
${OBJDIR}/libgr.dummy.a:
ln -sf ${EIRDIR}/libgr.dummy.a ${OBJDIR}

${OBJDIR}/libeirene.a:
ln -sf ${EIRDIR}/libeirene.a ${OBJDIR}

${OBJDIR}/ioflush.o:
ln -sf ${EIRDIR}/ioflush.o ${OBJDIR}

ifneq (${MOD},o)
${OBJDIR}/eirmod_extrab25.${MOD}:
ln -sf ${EIRDIR}/eirmod_extrab25.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_balanced_strategy.${MOD}:

```



```

ln -sf ${EIRDIR}/eirmod_balanced_strategy.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_braeir.${MOD}:
ln -sf ${EIRDIR}/eirmod_braeir.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_brascl.${MOD}:
ln -sf ${EIRDIR}/eirmod_brascl.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_braspoi.${MOD}:
ln -sf ${EIRDIR}/eirmod_braspoi.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cadgeo.${MOD}:
ln -sf ${EIRDIR}/eirmod_cadgeo.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cai.${MOD}:
ln -sf ${EIRDIR}/eirmod_cai.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_calstr_buffered.${MOD}:
ln -sf ${EIRDIR}/eirmod_calstr_buffered.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ccona.${MOD}:
ln -sf ${EIRDIR}/eirmod_ccona.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ccoupl.${MOD}:
ln -sf ${EIRDIR}/eirmod_ccoupl.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ccrm.${MOD}:
ln -sf ${EIRDIR}/eirmod_ccrm.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cestim.${MOD}:
ln -sf ${EIRDIR}/eirmod_cestim.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cfplk.${MOD}:
ln -sf ${EIRDIR}/eirmod_cfplk.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cgeom.${MOD}:
ln -sf ${EIRDIR}/eirmod_cgeom.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cgrid.${MOD}:
ln -sf ${EIRDIR}/eirmod_cgrid.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cgrptl.${MOD}:
ln -sf ${EIRDIR}/eirmod_cgrptl.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cinit.${MOD}:
ln -sf ${EIRDIR}/eirmod_cinit.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_clast.${MOD}:
ln -sf ${EIRDIR}/eirmod_clast.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_clgin.${MOD}:
ln -sf ${EIRDIR}/eirmod_clgin.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_clogau.${MOD}:
ln -sf ${EIRDIR}/eirmod_clogau.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_clmsur.${MOD}:
ln -sf ${EIRDIR}/eirmod_clmsur.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comnnl.${MOD}:
ln -sf ${EIRDIR}/eirmod_comnnl.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comprt.${MOD}:
ln -sf ${EIRDIR}/eirmod_comprt.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comsig.${MOD}:
ln -sf ${EIRDIR}/eirmod_comsig.${MOD} ${OBJDIR}

```

```

${OBJDIR}/eirmod_comsou.${MOD}:
ln -sf ${EIRDIR}/eirmod_comsou.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comspl.${MOD}:
ln -sf ${EIRDIR}/eirmod_comspl.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comusr.${MOD}:
ln -sf ${EIRDIR}/eirmod_comusr.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_comxs.${MOD}:
ln -sf ${EIRDIR}/eirmod_comxs.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_coutau.${MOD}:
ln -sf ${EIRDIR}/eirmod_coutau.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cpes.${MOD}:
ln -sf ${EIRDIR}/eirmod_cpes.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cp13d.${MOD}:
ln -sf ${EIRDIR}/eirmod_cp13d.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cplmsk.${MOD}:
ln -sf ${EIRDIR}/eirmod_cplmsk.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cplot.${MOD}:
ln -sf ${EIRDIR}/eirmod_cplot.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cpolyg.${MOD}:
ln -sf ${EIRDIR}/eirmod_cpolyg.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_crand.${MOD}:
ln -sf ${EIRDIR}/eirmod_crand.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_crech.${MOD}:
ln -sf ${EIRDIR}/eirmod_crech.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cref.${MOD}:
ln -sf ${EIRDIR}/eirmod_cref.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_crefmod.${MOD}:
ln -sf ${EIRDIR}/eirmod_crefmod.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_csdvi.${MOD}:
ln -sf ${EIRDIR}/eirmod_csdvi.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_csdvi_bgk.${MOD}:
ln -sf ${EIRDIR}/eirmod_csdvi_bgk.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_csdvi_cop.${MOD}:
ln -sf ${EIRDIR}/eirmod_csdvi_cop.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cspei.${MOD}:
ln -sf ${EIRDIR}/eirmod_cspei.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cspez.${MOD}:
ln -sf ${EIRDIR}/eirmod_cspez.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cstep.${MOD}:
ln -sf ${EIRDIR}/eirmod_cstep.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ctetra.${MOD}:
ln -sf ${EIRDIR}/eirmod_ctetra.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ctxt.${MOD}:
ln -sf ${EIRDIR}/eirmod_ctxt.${MOD} ${OBJDIR}

```

```

${OBJDIR}/eirmod_ctrcei.${MOD}:
    ln -sf ${EIRDIR}/eirmod_ctrcei.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ctrig.${MOD}:
    ln -sf ${EIRDIR}/eirmod_ctrig.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_ctorsf.${MOD}:
    ln -sf ${EIRDIR}/eirmod_ctorsf.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cupd.${MOD}:
    ln -sf ${EIRDIR}/eirmod_cupd.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_cvarusr.${MOD}:
    ln -sf ${EIRDIR}/eirmod_cvarusr.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_czt1.${MOD}:
    ln -sf ${EIRDIR}/eirmod_czt1.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_eirbra.${MOD}:
    ln -sf ${EIRDIR}/eirmod_eirbra.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_eirdiag.${MOD}:
    ln -sf ${EIRDIR}/eirmod_eirdiag.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_infcop.${MOD}: ${OBJDIR}/eirmod_cplot.${MOD}
    ln -sf ${EIRDIR}/eirmod_infcop.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_module_avltree.${MOD}:
    ln -sf ${EIRDIR}/eirmod_module_avltree.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_mpi.${MOD}:
    ln -sf ${EIRDIR}/eirmod_mpi.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_octree.${MOD}:
    ln -sf ${EIRDIR}/eirmod_octree.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_parmmod.${MOD}:
    ln -sf ${EIRDIR}/eirmod_parmmod.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_precision.${MOD}:
    ln -sf ${EIRDIR}/eirmod_precision.${MOD} ${OBJDIR}

${OBJDIR}/eirmod_solps.${MOD}:
    ln -sf ${EIRDIR}/eirmod_solps.${MOD} ${OBJDIR}
endif

${OBJDIR}/eirmod_extrab25.o:
    ln -sf ${EIRDIR}/eirmod_extrab25.o ${OBJDIR}

${OBJDIR}/eirmod_balanced_strategy.o:
    ln -sf ${EIRDIR}/eirmod_balanced_strategy.o ${OBJDIR}

${OBJDIR}/eirmod_braeir.o:
    ln -sf ${EIRDIR}/eirmod_braeir.o ${OBJDIR}

${OBJDIR}/eirmod_brasc1.o:
    ln -sf ${EIRDIR}/eirmod_brasc1.o ${OBJDIR}

${OBJDIR}/eirmod_braspoi.o:
    ln -sf ${EIRDIR}/eirmod_braspoi.o ${OBJDIR}

${OBJDIR}/eirmod_cadgeo.o:
    ln -sf ${EIRDIR}/eirmod_cadgeo.o ${OBJDIR}

${OBJDIR}/eirmod_cai.o:
    ln -sf ${EIRDIR}/eirmod_cai.o ${OBJDIR}

```

```

${OBJDIR}/eirmod_calstr_buffered.o:
    ln -sf ${EIRDIR}/eirmod_calstr_buffered.o ${OBJDIR}

${OBJDIR}/eirmod_ccona.o:
    ln -sf ${EIRDIR}/eirmod_ccona.o ${OBJDIR}

${OBJDIR}/eirmod_cestim.o:
    ln -sf ${EIRDIR}/eirmod_cestim.o ${OBJDIR}

${OBJDIR}/eirmod_ccoupl.o:
    ln -sf ${EIRDIR}/eirmod_ccoupl.o ${OBJDIR}

${OBJDIR}/eirmod_ccrm.o:
    ln -sf ${EIRDIR}/eirmod_ccrm.o ${OBJDIR}

${OBJDIR}/eirmod_cfplk.o:
    ln -sf ${EIRDIR}/eirmod_cfplk.o ${OBJDIR}

${OBJDIR}/eirmod_cgeom.o:
    ln -sf ${EIRDIR}/eirmod_cgeom.o ${OBJDIR}

${OBJDIR}/eirmod_cgrid.o:
    ln -sf ${EIRDIR}/eirmod_cgrid.o ${OBJDIR}

${OBJDIR}/eirmod_cgrptl.o:
    ln -sf ${EIRDIR}/eirmod_cgrptl.o ${OBJDIR}

${OBJDIR}/eirmod_cinit.o:
    ln -sf ${EIRDIR}/eirmod_cinit.o ${OBJDIR}

${OBJDIR}/eirmod_clast.o:
    ln -sf ${EIRDIR}/eirmod_clast.o ${OBJDIR}

${OBJDIR}/eirmod_clgin.o:
    ln -sf ${EIRDIR}/eirmod_clgin.o ${OBJDIR}

${OBJDIR}/eirmod_clogau.o:
    ln -sf ${EIRDIR}/eirmod_clogau.o ${OBJDIR}

${OBJDIR}/eirmod_clmsur.o:
    ln -sf ${EIRDIR}/eirmod_clmsur.o ${OBJDIR}

${OBJDIR}/eirmod_comnml.o:
    ln -sf ${EIRDIR}/eirmod_comnml.o ${OBJDIR}

${OBJDIR}/eirmod_comprt.o:
    ln -sf ${EIRDIR}/eirmod_comprt.o ${OBJDIR}

${OBJDIR}/eirmod_comsig.o:
    ln -sf ${EIRDIR}/eirmod_comsig.o ${OBJDIR}

${OBJDIR}/eirmod_comsou.o:
    ln -sf ${EIRDIR}/eirmod_comsou.o ${OBJDIR}

${OBJDIR}/eirmod_comspl.o:
    ln -sf ${EIRDIR}/eirmod_comspl.o ${OBJDIR}

${OBJDIR}/eirmod_comusr.o:
    ln -sf ${EIRDIR}/eirmod_comusr.o ${OBJDIR}

${OBJDIR}/eirmod_comxs.o:
    ln -sf ${EIRDIR}/eirmod_comxs.o ${OBJDIR}

${OBJDIR}/eirmod_coutau.o:
    ln -sf ${EIRDIR}/eirmod_coutau.o ${OBJDIR}

${OBJDIR}/eirmod_cpes.o:

```

```

ln -sf ${EIRDIR}/eirmod_cpes.o ${OBJDIR}

${OBJDIR}/eirmod_cp13d.o:
ln -sf ${EIRDIR}/eirmod_cp13d.o ${OBJDIR}

${OBJDIR}/eirmod_cp1msk.o:
ln -sf ${EIRDIR}/eirmod_cp1msk.o ${OBJDIR}

${OBJDIR}/eirmod_cplot.o:
ln -sf ${EIRDIR}/eirmod_cplot.o ${OBJDIR}

${OBJDIR}/eirmod_cpolyg.o:
ln -sf ${EIRDIR}/eirmod_cpolyg.o ${OBJDIR}

${OBJDIR}/eirmod_crand.o:
ln -sf ${EIRDIR}/eirmod_crand.o ${OBJDIR}

${OBJDIR}/eirmod_crech.o:
ln -sf ${EIRDIR}/eirmod_crech.o ${OBJDIR}

${OBJDIR}/eirmod_cref.o:
ln -sf ${EIRDIR}/eirmod_cref.o ${OBJDIR}

${OBJDIR}/eirmod_crefmod.o:
ln -sf ${EIRDIR}/eirmod_crefmod.o ${OBJDIR}

${OBJDIR}/eirmod_csdvi.o:
ln -sf ${EIRDIR}/eirmod_csdvi.o ${OBJDIR}

${OBJDIR}/eirmod_csdvi_bgk.o:
ln -sf ${EIRDIR}/eirmod_csdvi_bgk.o ${OBJDIR}

${OBJDIR}/eirmod_csdvi_cop.o:
ln -sf ${EIRDIR}/eirmod_csdvi_cop.o ${OBJDIR}

${OBJDIR}/eirmod_cspei.o:
ln -sf ${EIRDIR}/eirmod_cspei.o ${OBJDIR}

${OBJDIR}/eirmod_cspez.o:
ln -sf ${EIRDIR}/eirmod_cspez.o ${OBJDIR}

${OBJDIR}/eirmod_cstep.o:
ln -sf ${EIRDIR}/eirmod_cstep.o ${OBJDIR}

${OBJDIR}/eirmod_ctetra.o:
ln -sf ${EIRDIR}/eirmod_ctetra.o ${OBJDIR}

${OBJDIR}/eirmod_ctext.o:
ln -sf ${EIRDIR}/eirmod_ctext.o ${OBJDIR}

${OBJDIR}/eirmod_ctrcei.o:
ln -sf ${EIRDIR}/eirmod_ctrcei.o ${OBJDIR}

${OBJDIR}/eirmod_ctrig.o:
ln -sf ${EIRDIR}/eirmod_ctrig.o ${OBJDIR}

${OBJDIR}/eirmod_ctorsf.o:
ln -sf ${EIRDIR}/eirmod_ctorsf.o ${OBJDIR}

${OBJDIR}/eirmod_cupd.o:
ln -sf ${EIRDIR}/eirmod_cupd.o ${OBJDIR}

${OBJDIR}/eirmod_cvarusr.o:
ln -sf ${EIRDIR}/eirmod_cvarusr.o ${OBJDIR}

${OBJDIR}/eirmod_czt1.o:
ln -sf ${EIRDIR}/eirmod_czt1.o ${OBJDIR}

```

```

${OBJDIR}/eirmod_eirbra.o:
    ln -sf ${EIRDIR}/eirmod_eirbra.o ${OBJDIR}

${OBJDIR}/eirmod_infcp.o: ${OBJDIR}/eirmod_cplot.o
    ln -sf ${EIRDIR}/eirmod_infcp.o ${OBJDIR}

${OBJDIR}/eirmod_module_avltree.o:
    ln -sf ${EIRDIR}/eirmod_module_avltree.o ${OBJDIR}

${OBJDIR}/eirmod_mpi.o:
    ln -sf ${EIRDIR}/eirmod_mpi.o ${OBJDIR}

${OBJDIR}/eirmod_octree.o:
    ln -sf ${EIRDIR}/eirmod_octree.o ${OBJDIR}

${OBJDIR}/eirmod_parmmod.o:
    ln -sf ${EIRDIR}/eirmod_parmmod.o ${OBJDIR}

${OBJDIR}/eirmod_precision.o:
    ln -sf ${EIRDIR}/eirmod_precision.o ${OBJDIR}

${OBJDIR}/eirmod_solps.o:
    ln -sf ${EIRDIR}/eirmod_solps.o ${OBJDIR}
else
${OBJDIR}/eirmod_balanced_strategy.${MOD}:
    touch ${OBJDIR}/eirmod_balanced_strategy.${MOD}

${OBJDIR}/eirmod_braeir.${MOD}:
    touch ${OBJDIR}/eirmod_braeir.${MOD}

${OBJDIR}/eirmod_ccona.${MOD}:
    touch ${OBJDIR}/eirmod_ccona.${MOD}

${OBJDIR}/eirmod_ccoupl.${MOD}:
    touch ${OBJDIR}/eirmod_ccoupl.${MOD}

${OBJDIR}/eirmod_cestim.${MOD}:
    touch ${OBJDIR}/eirmod_cestim.${MOD}

${OBJDIR}/eirmod_cinit.${MOD}:
    touch ${OBJDIR}/eirmod_cinit.${MOD}

${OBJDIR}/eirmod_clgin.${MOD}:
    touch ${OBJDIR}/eirmod_clgin.${MOD}

${OBJDIR}/eirmod_clogau.${MOD}:
    touch ${OBJDIR}/eirmod_clogau.${MOD}

${OBJDIR}/eirmod_comprt.${MOD}:
    touch ${OBJDIR}/eirmod_comprt.${MOD}

${OBJDIR}/eirmod_comusr.${MOD}:
    touch ${OBJDIR}/eirmod_comusr.${MOD}

${OBJDIR}/eirmod_coutau.${MOD}:
    touch ${OBJDIR}/eirmod_coutau.${MOD}

${OBJDIR}/eirmod_cpes.${MOD}:
    touch ${OBJDIR}/eirmod_cpes.${MOD}

${OBJDIR}/eirmod_ctrcei.${MOD}:
    touch ${OBJDIR}/eirmod_ctrcei.${MOD}

${OBJDIR}/eirmod_ctrig.${MOD}:
    touch ${OBJDIR}/eirmod_ctrig.${MOD}

```

```

${OBJDIR}/eirmodeirbra.${MOD}:
    touch ${OBJDIR}/eirmodeirbra.${MOD}

${OBJDIR}/eirmodeirdiag.${MOD}:
    touch ${OBJDIR}/eirmodeirdiag.${MOD}

${OBJDIR}/eirmodeextrab25.${MOD}:
    touch ${OBJDIR}/eirmodeextrab25.${MOD}

${OBJDIR}/eirmodeinfcop.${MOD}:
    touch ${OBJDIR}/eirmodeinfcop.${MOD}

${OBJDIR}/eirmodempi.${MOD}:
    touch ${OBJDIR}/eirmodempi.${MOD}

${OBJDIR}/eirmodparmmode.${MOD}:
    touch ${OBJDIR}/eirmodparmmode.${MOD}

${OBJDIR}/eirmodsolps.${MOD}:
    touch ${OBJDIR}/eirmodsolps.${MOD}

${OBJDIR}/eirmodprecision.${MOD}:
    ln -s ${OBJDIR}/precision.${MOD} ${OBJDIR}/eirmodprecision.${MOD}
endif

${MNEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${IMASLIBS} ${LDLIBES} ${LD.CATALYST} ${LDOPTSend}

${AMEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${IMASLIBS} ${LDLIBES} ${LDOPTSend}

${OEEEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${LDLIBES} ${LDOPTSend}

${OPEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${LDLIBES} ${LDOPTSend}

${OQEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${LDLIBES} ${LD.CATALYST} ${LDOPTSend}

${OTEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${LDLIBES} ${LDOPTSend}

${O9EXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${LDLIBES}

${GEEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${GRLIBES} ${LDLIBES} ${LDOPTSend}

${GREXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${GRLIBES} ${LDLIBES} ${LDOPTSend}

${XDEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${OBJDIR}/libsolps4.a ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${OBJDIR}/libsolps4.a ${LCP} ${GRLIBES} ${LDLIBES} ${LDEXTRA} ${LDOPTSend}

${MDEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
    ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o $@ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${

```

```

        LDLIBS} ${LD_MDSPLUS} ${LDOPTSend}

${IDEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${MAKES}
        ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o @$ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MNEXTRA} ${
        IMASLIBS} ${LDLIBES} ${LD_CATALYST} ${LDOPTSend}

${TTEXE}: ${OBJDIR}/%.exe: ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${MAKES}
        ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o @$ ${OBJDIR}/%.o ${OBJDIR}/libb2.a ${LDLIBES} ${
        LDOPTSend}

${NCEXE}: ${NCODIR}/%.exe: ${NCODIR}/nc2text_simple.o ${MAKES}
ifdef LD_NETCDF
        ${LD} ${LDOPTS} ${FFLAGSEXTRA} -o @$ ${NCODIR}/%.o ${LD_NETCDF}
        @-ln -sf ${NCEXE} ${NCODIR}/nc2text_simple
ifeq (, $(findstring nc2text, ${NC2TXT}))
        ln -sf ${NCODIR}/nc2text_simple ${NCODIR}/nc2text
endif
else
        $(warning NETCDF library not present!)
endif

${OBJDIR}/libb2.a: ${LIBOBSJ} ${SRCDIR}/include/git_version_B25.h ${DOCDIR}/b2cdci.F ${DOCDIR}/
        b2cdcn.F
        @$ {BLD} @$ ${LIBOBSJ}

test:    ${TTEXE}

nc2text: ${NCODIR}/nc2text

${NCODIR}/nc2text: ${NCODIR}/nc2text_simple
        ln -sf ${NCODIR}/nc2text_simple ${NCODIR}/nc2text

nc2text_simple: ${NCEXE}

${NCODIR}/nc2text_simple.o: ${NCSDIR}/nc2text_simple.F90
ifdef LD_NETCDF
        @- /bin/rm -f ${NCODIR}/%.o
        @-mkdir -p ${NCODIR}
        -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} $< %.F90
        ${FC} ${FCOPTS} ${FFLAGSEXTRA} -c -o %.o %.F90
else
        $(warning NETCDF library not present!)
endif

${SOLPS4OBSJ}: ${OBJDIR}/%.o: ${SOLPS4}/%.F
        @- /bin/rm -f ${OBJDIR}/%.f ${OBJDIR}/%.o
ifeq ($(strip $(CPP)),)
        ${FC} ${FCOPTS} ${FFLAGSEXTRA} ${DEFINES} ${EQUIVS} ${INCLUDE} ${SOLPS4INCLUDE} -c $<
else
ifeq ($(strip $(SED)),)
        -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} ${SOLPS4INCLUDE} $< ${OBJDIR}/%.f
else
        -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} ${SOLPS4INCLUDE} $< | ${SED} > ${OBJDIR}/%.f
endif
endif
        ${FC} ${FCOPTS} ${FFLAGSEXTRA} -c ${MODINCLUDE} ${INCMODS} ${OBJDEST} ${OBJDIR}/%.f
endif
        @if [ -f %.o ] ; then /bin/mv %.o ${OBJDIR}/ ; fi

${EIR4OBSJ}: ${OBJDIR}/%.o: ${EIR4}/%.F
        ${FC} ${FCOPTS} ${FFLAGSEXTRA} ${DEFINES} ${EQUIVS} ${INCLUDE} ${OBJDEST} -c $?

${EIR4MODS}: ${OBJDIR}/%.${MOD}: ${EIR4}/%.F
        @- /bin/rm -f ${OBJDIR}/%.f ${OBJDIR}/%.o
ifeq ($(strip $(CPP)),)
        ${FC} ${FCOPTS} ${FFLAGSEXTRA} ${DEFINES} ${EQUIVS} ${INCLUDE} ${SOLPS4INCLUDE} -c $<
else
ifeq ($(strip $(SED)),)

```



```

else
    -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} ${SOLPS4INCLUDE} $< ${OBJDIR}/${*.f}
endif
${FC} ${FCOPTS} ${FLAGSEXTRA} -c ${MODINCLUDE} ${INCMODS} ${SOLPS4INCLUDE} ${OBJDEST} ${
    OBJDIR}/${*.f}
endif

${OBJDIR}/libsolps4.a: ${SOLPS4OBSJ} ${EIR4MODS}
    ${BLD} @$ ${SOLPS4OBSJ}

${OBJDIR}/b2rw.o: ${OBJDIR}/eirdiag.${MOD}
${OBJDIR}/default.o: ${OBJDIR}/ceirsrt.${MOD}

ifneq (${MOD},o)
${OBJDIR}/adsp.${MOD}: ${OBJDIR}/cadgeo.${MOD} ${OBJDIR}/clogau.${MOD} ${OBJDIR}/comusr.${MOD} ${
    OBJDIR}/cpes.${MOD} ${OBJDIR}/ctrcei.${MOD} ${OBJDIR}/comprt.${MOD}
${OBJDIR}/avltree.${MOD}: ${OBJDIR}/ccona.${MOD}
${OBJDIR}/caprmc.${MOD}: ${OBJDIR}/cgrid.${MOD} ${OBJDIR}/comxs.${MOD} ${OBJDIR}/comsou.${MOD}
${OBJDIR}/ccflux.${MOD}: ${OBJDIR}/ctrig.${MOD} ${OBJDIR}/cgeom.${MOD}
${OBJDIR}/ccrm.${MOD}: ${OBJDIR}/cestim.${MOD} ${OBJDIR}/csdvi.${MOD} ${OBJDIR}/czt1.${MOD} ${
    OBJDIR}/photon.${MOD}
${OBJDIR}/comxs.${MOD}: ${OBJDIR}/cupd.${MOD}
${OBJDIR}/cpes.${MOD}: ${OBJDIR}/comprt.${MOD} ${OBJDIR}/ctrcei.${MOD} ${OBJDIR}/eirmod-precision
    .${MOD}
${OBJDIR}/cupd.${MOD}: ${OBJDIR}/comsig.${MOD}
${OBJDIR}/eirdiag.${MOD}: ${OBJDIR}/precision.${MOD} ${OBJDIR}/parmmod.${MOD} ${OBJDIR}/braeir.${
    MOD} ${OBJDIR}/ccoupl.${MOD} ${OBJDIR}/clgin.${MOD}
${OBJDIR}/eirgrid-lib.${MOD}: ${OBJDIR}/eirmap.${MOD}
endif
${OBJDIR}/adsp.o: ${OBJDIR}/cadgeo.o ${OBJDIR}/clogau.o ${OBJDIR}/comusr.o ${OBJDIR}/cpes.o ${
    OBJDIR}/ctrcei.o ${OBJDIR}/comprt.o
${OBJDIR}/avltree.o: ${OBJDIR}/ccona.o
${OBJDIR}/caprmc.o: ${OBJDIR}/cgrid.o ${OBJDIR}/comxs.o ${OBJDIR}/comsou.o
${OBJDIR}/ccflux.o: ${OBJDIR}/ctrig.o ${OBJDIR}/cgeom.o
${OBJDIR}/ccrm.o: ${OBJDIR}/cestim.o ${OBJDIR}/csdvi.o ${OBJDIR}/czt1.o ${OBJDIR}/photon.o
${OBJDIR}/comxs.o: ${OBJDIR}/cupd.o
${OBJDIR}/cpes.o: ${OBJDIR}/comprt.o ${OBJDIR}/ctrcei.o ${OBJDIR}/eirmod-precision.o
${OBJDIR}/cupd.o: ${OBJDIR}/comsig.o
${OBJDIR}/eirdiag.o: ${OBJDIR}/precision.o ${OBJDIR}/parmmod.o ${OBJDIR}/braeir.o ${OBJDIR}/
    ccoupl.o ${OBJDIR}/clgin.o
${OBJDIR}/eirgrid-lib.o: ${OBJDIR}/eirmap.o

# target 'clean' cleans up the directory.
clean :
    -mkdir ${OBJDIR}/.delete
    -mv -i ${OBJDIR}/${*.o} ${OBJDIR}/${*.f} ${OBJDIR}/${*.f90} ${OBJDIR}/${*.a} ${OBJDIR}/${*.exe} ${SRCDIR}
        /include/git_version_B25.h ${OBJDIR}/LISTOBJ ${OBJDIR}/dependencies ${DOCDIR}/b2cdci
        .F ${DOCDIR}/b2cdcn.F ${OBJDIR}/.delete >& /dev/null
ifneq (${MOD},o)
    -mv -i ${OBJDIR}/${*}.${MOD} ${OBJDIR}/.delete >& /dev/null
endif
ifdef SOLPSTOP
ifdef LD_NETCDF
    -rm ${NCDIR}/${*.f90} ${NCDIR}/${*.o} ${NCDIR}/${*.exe}
endif
endif
    -rm -rf ${OBJDIR}/.delete &

depend: ${OBJDIR}/LISTOBJ ${B2OBSJ:.o=.F} ${B2F90OBSJ:.o=.F90} ${EXELIST:.o=.F} ${EX90LIST:.o=.
    F90}
    @'which makedepend' -p'${OBJDIR}/' ${DEFINES} -f- ${INCLUDE} $^ | \
    sed 's,^[${OBJDIR}/[^ ]*],\${OBJDIR}/,' | \
    sed 's,: ${SOLPSTOP},: ${SOLPSTOP},' > ${OBJDIR}/dependencies
    @echo '# 1' >> ${OBJDIR}/dependencies
ifneq (${MOD},o)
    @'which makedepend' -p'${OBJDIR}/' ${DEFINES} -f- ${INCLUDE} ${MODLIST} -o.${MOD} | \

```

```

sed 's,^${OBJDIR}/[^ ]*$/,\' | \
sed 's,: ${SOLPSTOP},: ${SOLPSTOP},\' >> ${OBJDIR}/dependencies
@echo '# 2' >> ${OBJDIR}/dependencies
endif
ifeq ($(shell [ -d ${SRCLOCAL} ] && echo yes || echo no ),yes)
@egrep -aiH '^ {6,}use ' ${SRCLOCAL}/*.F | grep -v 'IGNORE' | tr , ' ' | awk '{sub("\\.F
:",".o:",$1);sub("^.*/", "${OBJDIR}/", $1); print $1,"${OBJDIR}/"tolower($3)".${
MOD}"' >> ${OBJDIR}/dependencies
@echo '# 3' >> ${OBJDIR}/dependencies
endif
@egrep -aiH '^ {6,}use ' ${SRCDIR}/*.F | grep -v 'IGNORE' | tr , ' ' | awk '{sub("\\.F
:",".o:",$1);sub("^.*/", "${OBJDIR}/", $1); print $1,"${OBJDIR}/"tolower($3)".${
MOD}"' >> ${OBJDIR}/dependencies
@echo '# 4a' >> ${OBJDIR}/dependencies
@egrep -aiH '^ {0,}use ' ${SRCDIR}/*.F90 | grep -v 'IGNORE' | tr , ' ' | awk '{sub("\\.
F90:",".o:",$1);sub("^.*/", "${OBJDIR}/", $1); print $1,"${OBJDIR}/"tolower($3)".
${MOD}"' >> ${OBJDIR}/dependencies
@echo '# 4b' >> ${OBJDIR}/dependencies
ifneq (${MOD}.o)
@egrep -aiH '^ {6,}use ' ${MODLISTF} | grep -v 'IGNORE' | tr , ' ' | awk '{sub("\\.F:",".
${MOD}:",".o:",$1);sub("\\.f:",". ${MOD}:",".o:",$1);sub("^.*/", "${OBJDIR}/", $1); print $1,"
${OBJDIR}/"tolower($3)".${MOD}"' >> ${OBJDIR}/dependencies
@echo '# 5a' >> ${OBJDIR}/dependencies
ifdef MODLISTF90
@egrep -aiH '^ {0,}use ' ${MODLISTF90} | grep -v 'IGNORE' | tr , ' ' | awk '{sub("\\.F90
:",". ${MOD}:",".o:",$1);sub("\\.f90:",". ${MOD}:",".o:",$1);sub("^.*/", "${OBJDIR}/", $1); print
$1,"${OBJDIR}/"tolower($3)".${MOD}"' >> ${OBJDIR}/dependencies
@echo '# 5b' >> ${OBJDIR}/dependencies
endif
endif

tags:
rm -f ${SRCB2}/TAGS ; etags -o ${SRCB2}/TAGS ${TAGSLIST} || touch ${SRCB2}/TAGS

listobj: ${OBJDIR}/dependencies ${DOCDIR}/b2cdci.F ${DOCDIR}/b2cdcn.F
ifdef USEEIRENE
@rm -f ${OBJDIR}/LISTOBJ; touch ${OBJDIR}/LISTOBJ; l="OBJS ="; \
for d in `echo "${FPATH}" | tr : \ `; do \
l="${l} `(cd $d > /dev/null; echo *.F) `"; \
done; \
for d in `echo "${FFPATH}" | tr : \ `; do \
l="${l} `(cd $d > /dev/null; echo *.F90) `"; \
done; \
l="${l} `(cd ${SRCEIR}/modules > /dev/null; echo *.f) `"; \
l="${l} `(cd ${SRCEIR}/interfaces/couple-SOLPS-ITER > /dev/null; echo eirmod_*.F90 eirmod_
*.f) `"; \
E="-e 's/\.F90/\.o/g' -e 's/\.F/\.o/g' -e 's/\.f/\.o/g'"; for f in ${EXCLUDELIST}; do \
E="${E} -e 's/ $f//'"; \
done; \
echo "$l" | eval sed "$E" > ${OBJDIR}/LISTOBJ
else
@rm -f ${OBJDIR}/LISTOBJ; touch ${OBJDIR}/LISTOBJ; l="OBJS ="; \
for d in `echo "${FPATH}" | tr : \ `; do \
l="${l} `(cd $d > /dev/null; echo *.F) `"; \
done; \
for d in `echo "${FFPATH}" | tr : \ `; do \
l="${l} `(cd $d > /dev/null; echo *.F90) `"; \
done; \
E="-e 's/\.F90/\.o/g' -e 's/\.F/\.o/g'"; for f in ${EXCLUDELIST}; do \
E="${E} -e 's/ $f//'"; \
done; \
echo "$l" | eval sed "$E" > ${OBJDIR}/LISTOBJ
endif
@l="B2OBJS ="; \
for d in `echo "${FPATH}" | tr : \ `; do \
l="${l} `(cd $d > /dev/null; echo *.F) `"; \
done; \

```

```

E="-e 's/\.F90/\.o/g' -e 's/\.F/\.o/g'" ; for f in ${EXCLUDELIST}; do \
    E="$E -e 's/ $f//'" ; \
done ; \
echo "$$1" | eval sed "$$E" >> ${OBJDIR}/LISTOBJ
@l="B2F90OBS =" ; \
for d in `echo "${FFPATH}" | tr : \ ` ; do \
    l="$l '(cd $d > /dev/null; echo *.F90) '" ; \
done ; \
E="-e 's/\.F90/\.o/g'" ; for f in ${EXCLUDELIST}; do \
    E="$E -e 's/ $f//'" ; \
done ; \
echo "$$1" | eval sed "$$E" >> ${OBJDIR}/LISTOBJ

${OBJDIR}/LISTOBJ: listobj

VERSION: ${SRCDIR}/include/git_version_B25.h

${SRCDIR}/include/git_version_B25.h: force
@echo "      character*32 :: git_version_B25 = 'git describe —dirty —always'" > ${
    SRCDIR}/include/git_version_new.h
ifdef SOLPS_CPP
    @echo "      character*32 :: git_version_ADAS = '( cd ${SOLPSTOP}/modules/adas ; git
        describe —dirty —always)'" >> ${SRCDIR}/include/git_version_new.h
else
    @echo "      character*32 :: git_version_ADAS = '0.0.0-0-g0000000'" >> ${SRCDIR}/include/
        git_version_new.h
endif
    @if cmp -s ${SRCDIR}/include/git_version_new.h ${SRCDIR}/include/git_version_B25.h; then
        rm ${SRCDIR}/include/git_version_new.h; else mv ${SRCDIR}/include/git_version_new.h $
        {SRCDIR}/include/git_version_B25.h; fi

${OBJDIR}/dependencies: ${SRCDIR}/modules/.new_modules
ifeq ($(shell [ -d ${OBJDIR} ] && echo yes || echo no ),no)
    -mkdir -p ${OBJDIR}
endif
    touch ${OBJDIR}/dependencies
    ${MAKE} tags
    ${MAKE} VERSION
    ${MAKE} local
    ${MAKE} listobj
    ${MAKE} depend

include ${OBJDIR}/dependencies

ifeq ($(COMPILER),g77)
${OBJDIR}/b2stbc.o : b2stbc.F
    @- /bin/rm -f ${OBJDIR}/b2stbc.f ${OBJDIR}/b2stbc.o
ifeq ($(strip $(CPP)),)
    ${FC} ${FCOPTS} ${FFLAGSEXTRA} ${DEFINES} ${EQUIVS} ${INCLUDE} -c $<
else
ifeq ($(strip $(SED)),)
    -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} $< ${OBJDIR}/b2stbc.f
else
    -${CPP} ${DEFINES} ${EQUIVS} -P ${INCLUDE} $< | ${SED} > ${OBJDIR}/b2stbc.f
endif
endif
    ${FC} ${FCOPTS} ${FFLAGSEXTRA} -O1 -c ${INCMOD}${OBJDIR} ${OBJDIR}/b2stbc.f
endif
    @if [ -f b2stbc.o ] ; then /bin/mv b2stbc.o ${OBJDIR}/ ; fi
endif

echo:
    @echo LD_CATALYST=${LD_CATALYST}
    @echo INCLUDE=${INCLUDE}
    @echo LDLIBS=${LDLIBS}
    @echo DEFINES=${DEFINES}
    @echo EQUIVS=${EQUIVS}
    @echo VPATH=${VPATH}

```

```

@echo FPATH=${FPATH}
@echo OBJDIR=${OBJDIR}
@echo OBJS=${OBJS}
@echo SOLPS4OBJS=${SOLPS4OBJS}
@echo EIR4MODS=${EIR4MODS}
@echo EIR4OBJS=${EIR4OBJS}
@echo MODLIST=${MODLIST}
@echo MODLISTF=${MODLISTF}
@echo MODLISTF90=${MODLISTF90}
@echo MODULES=${MODULES}
@echo MODOBJS=${MODOBJS}
@echo MODMODS=${MODMODS}
@echo MODLOCAL=${MODLOCAL}
@echo $(filter -out ${MODOBJS},${ALLOBJS})
@echo EXCLUDELIST=${EXCLUDELIST}
@echo EXELIST=${EXELIST}
@echo EX90LIST=${EX90LIST}
@echo GREXE=${GREXE}
@echo MNEXE=${MNEXE}
@echo AMEXE=${AMEXE}
@echo XDEXE=${XDEXE}
@echo OTEXE=${OTEXE}
@echo O9EXE=${O9EXE}
@echo IDEXE=${IDEXE}
@echo NCEXE=${NCEXE}

local: ${SRCLOCAL}/b2local.F ${MODLOCAL}/b2mod_local.F ${INCLOCAL}/b2local.h

${SRCLOCAL}/b2local.F:
    mkdir -p ${SRCLOCAL}
    echo "      subroutine b2local" > ${SRCLOCAL}/b2local.F
    echo "c" >> ${SRCLOCAL}/b2local.F
    echo "c store local or locally modified subroutines in this directory" >> ${SRCLOCAL}/
    b2local.F
    echo "c" >> ${SRCLOCAL}/b2local.F
    echo "      use b2mod_local" >> ${SRCLOCAL}/b2local.F
    echo '#include "b2local.h"' >> ${SRCLOCAL}/b2local.F
    echo "c" >> ${SRCLOCAL}/b2local.F
    echo "      end" >> ${SRCLOCAL}/b2local.F
ifndef SOLPS_CPP
ifeq ($(shell [ -d ../solps4-5/src/local_5 ] && echo yes || echo no ),no)
    ln -sf ${SRCLOCAL} ../solps4-5/src/local_5
endif
endif

${MODLOCAL}/b2mod_local.F:
    mkdir -p ${MODLOCAL}
    echo "      module b2mod_local" > ${MODLOCAL}/b2mod_local.F
    echo "c" >> ${MODLOCAL}/b2mod_local.F
    echo "c store local or locally modified modules in this directory" >> ${MODLOCAL}/
    b2mod_local.F
    echo "c" >> ${MODLOCAL}/b2mod_local.F
    echo "      end" >> ${MODLOCAL}/b2mod_local.F
ifndef SOLPS_CPP
ifeq ($(shell [ -d ../solps4-5/src/B2.5_modules.local ] && echo yes || echo no ),no)
    ln -sf ${MODLOCAL} ../solps4-5/src/B2.5_modules.local
endif
endif

${INCLOCAL}/b2local.h:
    mkdir -p ${INCLOCAL}
    echo "c" > ${INCLOCAL}/b2local.h
    echo "c store local or locally modified include files in this directory" >> ${INCLOCAL}/
    b2local.h
    echo "c" >> ${INCLOCAL}/b2local.h
ifndef SOLPS_CPP
ifeq ($(shell [ -d ../solps4-5/src/B2.5_include.local ] && echo yes || echo no ),no)

```

```

ln -sf ${INCLOCAL} ../solps4-5/src/B2.5_include.local
endif
endif

```

and, as a prototypical example, the config.ITER.ifort64 in the config subdirectory:

```

# Test whether necessary variables are set
NODENAME    ?= $(shell echo `hostname`)
LD.NETCDF    ?= `nf-config --flibs`
LD.MSCL      ?= `pkg-config mscl --libs`
ifdef NCARG_ROOT
#LD.NCARG    ?= -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks -lncarg_c -lm -lX11
LD.NCARG     ?= -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks -lncarg_c -lm -lX11 -lXext `pkg-config cairo`
              --libs `--fontconfig --freetype --lpthread --lXrender --lbz2 --lgfortran`
endif
NCDIR        ?= ${EBROOTNETCDFMINFORTRAN}
LD.NAG        ?= -lnag.nag
LD.SONNET     ?= -L${SOLPSLIB} -lsonnet -lstc++

EQUIVS        = -Dsdot=ddot -Dsasum=dasum -Dsaxpy=daxpy -Dsscal=dscal -Dscopy=dcopy -Dsgbtrf=dgbtrf
              -Dsgbtrs=dgbtrs -Dsgetrf=dgetrf -Dsgetri=dgetri -Ds2min=d2min
DEFINES       += -DGRREAL=4 -DDOFLUSH -DNO.OPT
ifndef SOLPS_CPP
DEFINES       += -DNAG22 -DNO_F01AAF
endif

CPP           = /lib/cpp -traditional
CPOPTS        = -P -I${srcdir} ${EQUIVS}
SED           = sed s:\#PREC\#:#:d:g
SED           =
LINK.MOD      =

ifdef USE_MPI
FC            = mpiifort
ifndef MPDIR
MPDIR         = ${EBROOTIMPI}
LD_MPI        += -Wl,-rpath,${MPDIR}/lib64
endif
else
ifdef USE_IMPGYRO
FC            = mpiifort
ifndef MPDIR
MPDIR         = ${EBROOTIMPI}
LD_MPI        += -Wl,-rpath,${MPDIR}/lib64
endif
else
FC            = ifort
endif
endif

COMPILER_MAJOR_VERSION = $(shell ${FC} -V 2>&1 | grep -o "Version [0-9]*" | cut -d ' ' -f 2 )

FCOPTS        = -assume no2underscore -align commons -fPIC
ifdef SOLPS_DEBUG
FCOPTS        += -g -C -debug all -traceback -check all
else
FCOPTS        += -O2 -fp-model precise
endif
ifdef USE_MPI
FCOPTS        += ${MPI.CPP}
else
ifdef USE_IMPGYRO
FCOPTS        += ${MPI.CPP}
endif
endif

ifdef USE_OPENMP

```

```

ifeq ($(shell test ${COMPILER_MAJOR_VERSION} -ge 15; echo $$?),0)
FCOPTS      += -qopenmp
LDOPTS      += -qopenmp
else
FCOPTS      += -openmp
LDOPTS      += -openmp
endif
endif

ifeq ($(shell test ${COMPILER_MAJOR_VERSION} -le 12; echo $$?),0)
FCOPTS      += -vec_report0
LDOPTS      += -i_dynamic
endif

DBLFC       = ${FC}
DBLSED      = sed -e s:\#PREC\#:d:g
DBLSED      =

INCMOD      = -I
MOD         = mod
OBJDEST     = -module ${OBJDIR} -o ${OBJDIR}/${*.o}

BLD         = ar rucv

GRLIBES     = ${LD_NCARG} ${LD_MSCL}
LDLIBES     = ${LD_NETCDF} ${LD_NAG} -lz

ifdef LD_CATALYST
CXX         = c++
CXXFLAGS    = -std=c++11 $(shell paraview-config --include)
endif

#wd: removed; seems to be unused
#ifdef B25_EIRENE
#LDLIBSEXTRA = ${OBJECTCODE}/libeirene.a ${OBJECTCODE}/grib.o -lnag
#endif
ifdef PERFMON
LDLIBES     += -L${SOLPSLIB} -lperf
endif

ifdef IMAS_VERSION
GGD_VERSION ?= ${EBVERSIONGGD}
UAL_MAJOR_VERSION ?= $(shell echo ${UAL_VERSION} | cut -d '.' -f1 )
IMAS_MAJOR_VERSION ?= $(shell echo ${IMAS_VERSION} | cut -d '.' -f1 )
IMAS_MINOR_VERSION ?= $(shell echo ${IMAS_VERSION} | cut -d '.' -f2 )
ifeq ($(shell test ${IMAS_MINOR_VERSION} -ge 17; echo $$?),0)
GGD_MINOR_VERSION ?= $(shell echo ${GGD_VERSION} | cut -d '.' -f2 )
LD_UAL        ?= 'pkg-config imas-ifort ggd amns --libs '
INCMODS      += 'pkg-config imas-ifort ggd amns --cflags '
else
ifeq ($(shell test ${IMAS_MINOR_VERSION} -ge 12; echo $$?),0)
GGD_MINOR_VERSION ?= $(shell echo ${GGD_VERSION} | cut -d '.' -f2 )
LD_UAL        ?= 'pkg-config imas-ifort --libs ' -lggd-ifort
INCMODS      += 'pkg-config imas-ifort --cflags '
else
GGD_MINOR_VERSION ?= 0
LD_UAL        ?= 'pkg-config imas-lowlevel imas-ifort --libs '
INCMODS      += 'pkg-config imas-lowlevel imas-ifort --cflags '
endif
endif
ifndef SOLPS_CPP
DEFINES      += -DIMAS -DAMNS -DUAL_MAJOR_VERSION=${UAL_MAJOR_VERSION} -DIMAS_MINOR_VERSION=${
IMAS_MINOR_VERSION} -DGGD_MINOR_VERSION=${GGD_MINOR_VERSION}
endif
IMASLIBS     = ${LD_UAL}
endif

```

```

LD          = ${FC}
ifdef USE_MPI
LDLIBES     += ${LD_MPI}
else
ifdef USE_IMPGYRO
LDLIBES     += ${LD_MPI}
endif
endif

LCCPP       = ${LD.SONNET}

#### Local Variables :
#### mode: Makefile
#### End:

```

If the user wishes to modify the compiling options used, the creation of a `config.$HOST_NAME.$COMPILER.local` file is recommended, in which to store the modified settings. Since the `Makefile` reads those after the `config/compiler.$HOST_NAME.$COMPILER` file, they need only contain the modified definitions.

J.4 Defines

J.4.1 SOLPS-ITER

We list here the possible optional pre-processor definitions that can be used when compiling the various components of the **SOLPS-ITER** code suite. The optional variables below must therefore, if desired, be declared as part of the `$SOLPS_CPP` environment variable, which is defined in the `$SOLPSTOP/SETUP/config.$HOST_NAME.$COMPILER(.local)` files. Additionally, system-wide compiler flags can be provided by means of the `$FFLAGSEXTRA` environment variable in the same files.

CATALYST Turns on the Paraview Catalyst utility (see the SOLPS-GUI manual for full details)

COMPILER_MAJOR_VERSION Major version number of the compiler being used (set internally by the configuration files)

COMPILER_MINOR_VERSION Minor version number of the compiler being used (set internally by the configuration files)

CRAY To be used with the Cray Fortran compiler

DBG Turns on additional debugging output (set internally by the `Makefile` when requesting a `_debug` compilation)

F2003 To be used if the compiler is compatible with Fortran 2003

F77 Identifies specific code to Fortran-77 compilers

GFORTRAN To be used with `gfortran` compilation

GGD_MINOR_VERSION Minor version number of the IMAS GGD library to which the code is linked (set internally by the configuration files)

G95 To be used with `g95` compilation

HPUX To be used when compiling on HP Unix systems

IBMaix To be used when compiling on IBM AIX computers

IMAS To be used when compiling within the IMAS framework

IMAS_MINOR_VERSION Minor version number of the IMAS library to which the code is linked (set internally by the configuration files)

ITM To be used when compiling within the ITM framework

ITM_ENVIRONMENT_LOADED To be used when wishing to compile with ITM CPOs

MDSPLUS Enables use of MDSP_lus trees

NAGF95 To be used when compiling with the NAG F95 compiler

NAG22 Specific to using the Nag Mark 22 or higher mathematical library

NCAR4 Used to specify NCAR4 rather than NCAR3

NO_CDF No CDF library available

NO_GETENV No GETENV system call available

NO_LONG_LINES Identify those compilers which have a limitation to relatively short output lines

NO_NAG No NAG libraries available

PGF90 To be used when compiling with the Portland Group Fortran-90 compiler

PSR Specific to the Pacific Sierra VAST compiler

UAL_MAJOR_VERSION Major version number of the UAL (Universal Access Layer) library to which the code is linked (set internally by the configuration files)

USE_MPI Set internally by the Makefile and used to identify code necessary for MPI applications

USE_PXFGETENV Uses PXFGETENV instead of GETENV

WINDOWS To be used when compiling on a Windows system

If you want to activate any of these options, you should add `-DOPTION` (where `OPTION` might, for example, be `NO_NAG`) to the `SOLPS_CPP` environment variable with

```
setenv SOLPS_CPP '-DOPTION'
```

and then recompile all of the routines affected (if unsure which routines are affected, recompile them all with

```
gmake clean
gmake depend
gmake
```

). If `SOLPS_CPP` already had a value, you would want to do

```
setenv SOLPS_CPP "$SOLPS_CPP -DOPTION"
```

or

```
SOLPS_CPP += -DOPTION
```

so that the new option is added to the list. (Unsupported `ksh` or `bash` users would use something like

```
export SOLPS_CPP="$SOLPS_CPP -DOPTION"
```

).

J.4.2 Carre

We list here additional possible compiler options that can be used when compiling `Carre`.

BUILDING_CARRE Flags code for extended grids handling routine specific to `Carre`

EXTRA_GRAPHS Turns on additional graphical output of intermediate steps of the `Carre` grid generation

MODIF Allows for manual modifications to the grid generation process during the `Carre` run

READOPT Allows for format-free keyboard input

We list below where the `Carre` compilation switches are used.

name	used in
DBG	carre/doubl.d.F
DBG	carre/saute.F
DBG	fcrr/fcrprp.F
EXTRA_GRAPHS	carre/carre.F
GFORTRAN	carre/rdfrin.F
GFORTRAN	carre/rdfre.F
MODIF	carre/maill.F
MODIF	carre/mailrg.F
READOPT	carre/rdfrch.F
READOPT	carre/rdfrin.F
READOPT	carre/rdfre.F

Table J.1: Defines for `Carre` and where they are used.

J.4.3 DivGeo

We list below where the `DivGeo` compilation switches are used.

name	used in
IMAS_MINOR_VERSION	equtrn/src/rdids.f

Table J.2: Defines for `DivGeo` and where they are used.

J.4.4 B2.5-Eirene

We list here the possible compiler options that can be used when compiling the coupled version of `B2.5-Eirene`. These options are shared across both the `B2.5` and `Eirene` source codes. The options below must therefore, if desired, be declared as part of the `$SOLPS_CPP` environment variable, which is defined in the `$SOLPSTOP/SETUP/config.$HOST_NAME.$COMPILER(.local)` files.

ALLOCATE_AND_NAMELIST To be used if the compiler allows dynamically allocated namelist arrays

B25_EIRENE Set internally by the `Makefile` and used to identify differences between source code for standalone `B2.5` and coupled `B2.5-Eirene` applications

NO_B2_CHEM_SPUT Used to indicate that chemical sputtering is specified through `Eirene` arrays instead of `B2.5`

NO_SAVE Turning off static save of some common blocks

SOLPS4_3 Set internally by the `Makefile` to activates code specific to the `SOLPS4-5 b2sxdx` converter

USE_IMPGYRO Used when coupling to `IMPGYRO`

J.4.5 B2.5

We list here the additional compiler options available that only affect compilation of B2.5. The options below must therefore, if desired, be declared as part of either the `$SOLPS_CPP` or the `$B25_DEFINES` environment variable, which are defined in the `$SOLPSTOP/SETUP/config.$HOST_NAME.$COMPILER(.local)` files.

AMNS Links to AMNS database in IMAS environment

ASTRA To be used if one wishes to run in coupled mode with the core transport code ASTRA

CDF_MOVIE To be used if CDF movie output is desired (turned on by default if the NetCDF library is present)

CLASSICAL_CURRENT Turns on classical current drift term

COMPLICATED Uses a more "complicated" model for feedback schemes

EASY Uses the "easy" model for chord integration

GOT_AMNSPROTO Links to AMNS database in ITM environment

JET Enables some additional output for coupling to the JET post-processor

NO_21 Do not use 21-moment model for thermal and viscous forces

NO_IEEE Provides replacements to IEEE non-standard arithmetic routines

NO_F01AAF No NAG routine F01AAF — uses its replacement instead (needed if linking with Version 17 or later of NAG)

NO_OPENMP Turning off OpenMP parallelization

NO_OPT No compiler optimization

PAR_OPT Runs the code in a special interpretive parameter optimization mode

PERFMON Enable performance monitoring

PLPLOT Use PLPLOT instead of NCAR for plotting

UAL_MAJOR_VERSION Major version number of the UAL (Universal Access Layer) library to which the code is linked (set internally by the configuration files)

USE_CDF_B2STIME Specific for **SOLPS 4.x** CDF tallies

WANT_PROCESS Enables the code for the **SOLPS 5.x** to **SOLPS 4.x** converter

WOS Uses the WOS libraries for accessing experimental data at IPP

If you want to activate any of these options, you should add `-DOPTION` (where `OPTION` might, for example, be `PERFMON`) to either the `SOLPS_CPP` or the `B25_DEFINES` environment variable with

```
setenv B25_DEFINES '-DOPTION'
```

and then recompile all of the routines affected (if unsure which routines are affected, recompile them all with

```
gmake clean
gmake depend
gmake
```

). If `B25_DEFINES` already had a value, you would want to do

```
setenv B25_DEFINES "$B25_DEFINES -DOPTION"
```

or

```
B25_DEFINES += -DOPTION
```

so that the new option is added to the list. (Unsupported `ksh` or `bash` users would use something like

```
export B25_DEFINES="$B25_DEFINES -DOPTION"
```

). If choosing to use SOLPS_CPP instead, the procedure is the same as in the previous section.

We list below where the B2 . 5 compilation switches are used.

name	used in
ALLOCATE_AND_NAMELIST	b2plot/init_wall.F
ALLOCATE_AND_NAMELIST	modules/b2mod_boundary_namelist.F
ALLOCATE_AND_NAMELIST	modules/b2mod_input_profile.F
ALLOCATE_AND_NAMELIST	modules/b2mod_layer.F
ALLOCATE_AND_NAMELIST	modules/b2mod_neutrals_namelist.F
ALLOCATE_AND_NAMELIST	modules/b2mod_numerics_namelist.F
ALLOCATE_AND_NAMELIST	modules/b2mod_sputter.F
ALLOCATE_AND_NAMELIST	modules/b2mod_transport_models.F
ALLOCATE_AND_NAMELIST	modules/b2mod_transport_namelist.F
ALLOCATE_AND_NAMELIST	modules/b2mod_user_namelist.F
ALLOCATE_AND_NAMELIST	modules/b2mod_wall.F
ALLOCATE_AND_NAMELIST	postprocessing/b2yt_wall.F
ALLOCATE_AND_NAMELIST	sources/b2stbr.F
AMNS	preprocessing/b2ardr.F
AMNS	utility/ratadas.F
ASTRA	modules/b2mod_astra_to_b2.F
ASTRA	sources/b2stbc_phys.F
ASTRA	sources/b2stbr.F
ASTRA	sources/b2stcx.F
ASTRA	sources/b2stel.F
ASTRA	transport/b2trcl.F
B25_EIRENE	b2plot/b2pinp.F
B25_EIRENE	b2plot/b2plot.F
B25_EIRENE	b2plot/b2pnwl.F
B25_EIRENE	b2plot/b2ptrdl.F
B25_EIRENE	b2plot/b2ptrgl.F
B25_EIRENE	b2plot/b2pwinm.F
B25_EIRENE	b2plot/b2pwldw.F
B25_EIRENE	b2plot/b2pwlprp.F
B25_EIRENE	b2plot/b2pwsort.F
B25_EIRENE	b2plot/clvp.F
B25_EIRENE	b2plot/copy.F
B25_EIRENE	b2plot/get_from_movie_cdf.F
B25_EIRENE	b2plot/map_eirene_chords.F
B25_EIRENE	b2plot/ngread.F
B25_EIRENE	b2plot/ntread.F
B25_EIRENE	b2plot/pltoutline.F
B25_EIRENE	b2plot/plttriang.F
B25_EIRENE	b2plot/process_eirene_chords_boxcar.F
B25_EIRENE	b2plot/process_eirene_chords.F
B25_EIRENE	b2plot/process_eirene_chord_slice.F
B25_EIRENE	b2plot/psta.F
B25_EIRENE	b2plot/pstv.F
B25_EIRENE	b2plot/triangle_surf.F
B25_EIRENE	b2plot/trimesh.F
B25_EIRENE	convert/b2co.F
B25_EIRENE	convert/b2mds.F
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
B25_EIRENE	convert/b2yt_ngread.F
B25_EIRENE	convert/b2yt_ntread.F
B25_EIRENE	driver/b2mod_driver.F
B25_EIRENE	driver/b2mod_main.F
B25_EIRENE	ids/b2mod_ual_io.F90
B25_EIRENE	modules/b2mod_b2_to_astra.F
B25_EIRENE	modules/b2mod_mwti.F90
B25_EIRENE	modules/b2mod_neutrals_namelist.F
B25_EIRENE	output/b2wdat.F
B25_EIRENE	postprocessing/b2ytdr.F
B25_EIRENE	sources/b2stbc_fb.F
B25_EIRENE	sources/b2stbc_phys.F
B25_EIRENE	sources/b2stbm.F
B25_EIRENE	sources/b2stbr.F
B25_EIRENE	sources/eirene_f30f31.F
B25_EIRENE	sources/eirene_mc.F
B25_EIRENE	sources/eirene_mc_init.F
B25_EIRENE	transport/b2tinnt.F
B25_EIRENE	user/b2blne.F
B25_EIRENE	user/b2blns.F
B25_EIRENE	user/b2trace.F
B25_EIRENE	user/b2usrtrc.F
B25_EIRENE	user/b2wrnt.F
B25_EIRENE	user/b2wrsep.F
B25_EIRENE	user/b2wrsrt.F
B25_EIRENE	user/cdfmovie.F
B25_EIRENE	user/eirflux_map.F
B25_EIRENE	utility/gfsub2.F
B25_EIRENE	utility/rdneutrs.F
B25_EIRENE	utility/xertst.F
BUILDING_CARRE	ids/b2mod_connectivity.F90
CATALYST	driver/b2mod_driver.F
CDF_MOVIE	b2plot/caltraj.F
CDF_MOVIE	b2plot/process_chords.F
CLASSICAL_CURRENT	transport/b2tdia.F
COMPLICATED	sources/b2stbc_fb.F
CRAY	b2plot/b2plot.F
CRAY	equations/b2news.F
CRAY	equations/b2news_.F
CRAY	equations/b2npmo.F
CRAY	modules/b2mod_openmp.F
CRAY	postprocessing/b2ytdr.F
DBG	b2plot/b2ptrdl.F
DBG	b2plot/b2ptrgl.F
DBG	b2plot/b2pwrldl.F
DBG	b2plot/fill.F
DBG	b2plot/init_wall.F
DBG	b2plot/plotcontour.F
DBG	b2plot/rlcomp.F
DBG	equations/b2news_.F
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
DBG	modules/b2mod_input_profile.F
DBG	modules/b2mod_neutrals_namelist.F
DBG	user/b2usrtrc.F
DBG	utility/ma28.F
DBG	utility/ma28copy.F
DBG	utility/my_out.F
DEF_MAXCHORDS	b2plot/process_chords.F
DEF_MAXCHORDS	b2plot/process_eirene_chords.F
EASY	b2plot/chord.F
EASY	b2plot/chord_triangle.F
F2003	convert/b2cdftallies2mds.F
F77	user/b2file.F
GFORTRAN	equations/b2news.F
GFORTRAN	equations/b2news_.F
GFORTRAN	equations/b2npmo.F
GFORTRAN	modules/b2mod_openmp.F
GFORTRAN	utility/ilutern_solps.F
GFORTRAN	utility/ilutern_7_solps.F
GFORTRAN	utility/ilutern_9_solps.F
GFORTRAN	utility/ilutern_11_solps.F
GFORTRAN	utility/ma28.F
GFORTRAN	utility/ma28copy.F
GFORTRAN	utility/ma28copy3.F
GFORTRAN	utility/ma28copy4.F
GGD_MINOR_VERSION	ids/b2mod_ual_io.F90
GGD_MINOR_VERSION	ids/b2mod_ual_io_grid.F90
GOT_AMNSPROTO	preprocessing/b2ardr.F
GOT_AMNSPROTO	utility/ratadas.F
HITS_PER_CHORD	b2plot/process_chords.F
HITS_PER_CHORD	b2plot/process_eirene_chords.F
HPUX	modules/b2mod_eirbra.F
IBMaix	b2plot/b2plot.F
IBMaix	postprocessing/b2ytdr.F
IBMaix	user/b2file.F
IMAS	driver/b2mnds.F
IMAS	ids/b2mod_ual.F90
IMAS	ids/b2mod_ual_io_data.F90
IMAS	ids/b2mod_ual_io_grid.F90
IMAS	ids/b2mod_ual_io.F90
IMAS	ids/b2_ual_write.F90
IMAS	modules/b2mod_b2cmrc.F
IMAS	modules/b2mod_ppout.F
IMAS	preprocessing/b2ardr.F
IMAS	utility/ratadas.F
IMAS_MINOR_VERSION	ids/b2mod_ual.F90
IMAS_MINOR_VERSION	ids/b2mod_ual_io.F90
IMAS_MINOR_VERSION	ids/b2mod_ual_io_data.F90
IMAS_MINOR_VERSION	ids/b2mod_ual_io_grid.F90
IMAS_MINOR_VERSION	ids/b2_ual_write.F90
IMAS_MINOR_VERSION	ids/b2_ual_write_b2mod.F90
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
ITM	driver/b2mod_driver.F
ITM	preprocessing/b2ardr.F
ITM	utility/ratadas.F
ITM_ENVIRONMENT_LOADED	ids/b2mod_ual.F90
ITM_ENVIRONMENT_LOADED	ids/b2mod_ual_io_data.F90
ITM_ENVIRONMENT_LOADED	ids/b2mod_ual_io_grid.F90
ITM_ENVIRONMENT_LOADED	ids/b2mod_ual_io.F90
JET	driver/b2mnastra.F
JET	driver/b2mnds.F
JET	driver/b2mn_opt.F
JET	driver/b2mod_driver.F
JET	driver/b2mod_main.F
JET	ids/b2_ual_write.F90
JET	modules/b2mod_ppout.F
JET	preprocessing/b2agfs.F
JET	preprocessing/b2ardr.F
MDSPLUS	convert/b2cdf2mds.F
MDSPLUS	convert/b2cdfmovies2mds.F
MDSPLUS	convert/b2cdftallies2mds.F
MDSPLUS	convert/b2mddr.F
MDSPLUS	convert/b2mds.F
MDSPLUS	driver/b2rd.F
MDSPLUS	modules/b2mod_mdsplus.F
MDSPLUS	utility/mds_routines.F
NAG22	b2plot/caltraj.F
NAGF95	b2plot/b2plot.F
NAGF95	b2plot/dhdata.F
NAGF95	b2plot/initialize_plotting.F
NAGF95	b2plot/set_extralabel.F
NAGF95	b2plot/spect.F
NAGF95	convert/b2mds.F
NAGF95	driver/b2co.F
NAGF95	modules/b2mod_ppout.F
NAGF95	modules/b2mod_solpstop.F
NAGF95	output/b2wgpj_gnuplot.F
NAGF95	preprocessing/b2agfs.F
NAGF95	utility/open_file.F
NAGF95	utility/usnam.F
NAGFOR	b2plot/b2plot.F
NAGFOR	b2plot/dhdata.F
NAGFOR	b2plot/initialize_plotting.F
NAGFOR	b2plot/set_extralabel.F
NAGFOR	b2plot/spect.F
NAGFOR	convert/b2co.F
NAGFOR	convert/b2co_movies_cdf.F
NAGFOR	convert/b2co_tally_cdf.F
NAGFOR	convert/b2co_time_cdf.F
NAGFOR	convert/b2co_wall_cdf.F
NAGFOR	convert/b2mds.F
NAGFOR	ids/b2_ual_write.F90
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
NAGFOR	modules/b2mod_openmp.F
NAGFOR	modules/b2mod_ppout.F
NAGFOR	modules/b2mod_solpstop.F
NAGFOR	modules/b2mod_types.F
NAGFOR	output/b2wgpj_gnuplot.F
NAGFOR	preprocessing/b2agfs.F
NAGFOR	utility/calc_atomic_data.F
NAGFOR	utility/open_file.F
NAGFOR	utility/usnam.F
NCAR4	b2plot/end_page.F
NCAR4	b2plot/initialize_plotting.F
NCAR4	b2plot/terminate_plotting.F
NO_21	transport/b2trcl.F
NO_B2_CHEM_SPUT	modules/b2mod_neutrals_namelist.F
NO_B2_CHEM_SPUT	postprocessing/b2ytdr.F
NO_B2_CHEM_SPUT	sources/eirene_mc.F
NO_CDF	b2plot/b2plot.F
NO_CDF	b2plot/caltraj.F
NO_CDF	b2plot/get_from_movie_cdf.F
NO_CDF	b2plot/process_chords.F
NO_CDF	convert/b2cdf2mds.F
NO_CDF	convert/b2cdfmovies2mds.F
NO_CDF	convert/b2cdftallies2mds.F
NO_CDF	convert/b2co_movies_cdf.F
NO_CDF	convert/b2co_tally_cdf.F
NO_CDF	convert/b2co_time_cdf.F
NO_CDF	convert/b2co_wall_cdf.F
NO_CDF	convert/process.F
NO_CDF	driver/b2mod_driver.F
NO_CDF	modules/b2mod_balance.F
NO_CDF	modules/b2mod_movies.F
NO_CDF	modules/b2mod_mwti.F
NO_CDF	modules/b2mod_tallies.F
NO_CDF	modules/b2mod_wall.F
NO_CDF	output/tallies.F
NO_CDF	sources/b2stbr.F
NO_CDF	user/cdfmovie.F
NO_CDF	utility/cdf_routines.F
NO_F01AAF	transport/b2tr21.F
NO_GETENV	b2plot/b2plot.F
NO_GETENV	b2plot/spect.F
NO_GETENV	convert/b2co_movies_cdf.F
NO_GETENV	convert/b2co_tally_cdf.F
NO_GETENV	convert/b2co_time_cdf.F
NO_GETENV	convert/b2co_wall_cdf.F
NO_GETENV	convert/b2mds.F
NO_GETENV	driver/b2co.F
NO_GETENV	ids/b2_ual_write.F90
NO_GETENV	modules/b2mod_ppout.F
NO_GETENV	modules/b2mod_solpstop.F
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
NO_GETENV	preprocessing/b2agfs.F
NO_GETENV	utility/open_file.F
NO_IEEE	utility/nancheck.F
NO_LONG_LINES	output/tallies.F
NO_NAG	b2plot/caltraj.F
NO_NAG	sources/heatdiff1D.F
NO_NAG	transport/b2tr21.F
NO_NAG	utility/nagsubst.F
NO_OPENMP_B2NEWS_LOOP1	equations/b2news.F
NO_OPENMP_B2NEWS_LOOP2	equations/b2news.F
NO_OPENMP_B2NEWS_LOOP3	equations/b2news.F
NO_OPENMP_B2NEWS_LOOP4	equations/b2news.F
NO_OPENMP_B2NEWS_UNDERSCORE_LOOP1	equations/b2news_.F
NO_OPENMP_B2NEWS_UNDERSCORE_LOOP2	equations/b2news_.F
NO_OPENMP_B2NEWS_UNDERSCORE_LOOP3	equations/b2news_.F
NO_OPENMP_B2NEWS_UNDERSCORE_LOOP4	equations/b2news_.F
NO_OPENMP_B2NPMO	equations/b2npmo.F
NO_OPENMP_B2NPMO	sources/b2sifrtf.F
NO_OPENMP_B2SIFRTF	sources/b2sifrtf.F
NO_OPENMP_B2SIHS	sources/b2sihs.F
NO_OPENMP_B2SPCX	sources/b2spcx.F
NO_OPENMP_B2SPEL	sources/b2spel.F
NO_OPENMP_B2SQCX	sources/b2sqcx.F
NO_OPENMP_B2SQEL	sources/b2sqel.F
NO_OPENMP_B2SRDT	sources/b2srdt.F
NO_OPENMP_B2SRST	sources/b2srst.F
NO_OPENMP_B2STCX	sources/b2stcx.F
NO_OPENMP_B2STEL	sources/b2stel.F
NO_OPENMP_B2XPFE	b2aux/b2xpfe.F
NO_OPENMP_MYBLAS	utility/myblas.F
NO_OPENMP_SFILL	utility/sfill.F
NO_OPT	b2plot/b2plot.F
NO_OPT	ids/b2mod_ual_io.F90
NO_OPT	postprocessing/b2ytdr.F
NO_SAVE	include/BRAEIR.H
NO_SAVE	include/EIRBRA.H
NO_SAVE	include/EIRDIAG.H
PAR_OPT	sources/b2stbc_phys.F
PAR_OPT	transport/b2tqna.F
PERFMON	utility/prgend.F
PERFMON	utility/prgini.F
PERFMON	utility/subsys.F
PGF90	equations/b2news.F
PGF90	equations/b2news_.F
PGF90	equations/b2npmo.F
PGF90	modules/b2mod_openmp.F
PGF90	modules/b2mod_types.F
PLPLOT	b2plot/b2plot.F
PLPLOT	b2plot/color.F
PLPLOT	b2plot/end_page.F
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
PLPLOT	b2plot/font.F
PLPLOT	b2plot/fx.F
PLPLOT	b2plot/fy.F
PLPLOT	b2plot/gslwsc_d.F
PLPLOT	b2plot/initialize_plotting.F
PLPLOT	b2plot/init_page.F
PLPLOT	b2plot/lined_d.F
PLPLOT	b2plot/mesh.F
PLPLOT	b2plot/meshit.F
PLPLOT	b2plot/plotpatch.F
PLPLOT	b2plot/plottarget.F
PLPLOT	b2plot/plottripatch.F
PLPLOT	b2plot/plot_wall_elements.F
PLPLOT	b2plot/pltoutline.F
PLPLOT	b2plot/pltsptrix.F
PLPLOT	b2plot/plttarget.F
PLPLOT	b2plot/plttriang.F
PLPLOT	b2plot/pltvessel.F
PLPLOT	b2plot/pltwall.F
PLPLOT	b2plot/set_a4l.F
PLPLOT	b2plot/set_a4p.F
PLPLOT	b2plot/set_extralabel.F
PLPLOT	b2plot/terminate_plotting.F
PLPLOT	b2plot/trgw.F
PLPLOT	b2plot/trgz.F
PLPLOT	b2plot/trigmesh.F
PLPLOT	utility/lbfill.F
PLPLOT	utility/ncardummy.F
PLPLOT	utility/pagdef.F
PLPLOT	utility/pagend.F
PLPLOT	utility/pagini.F
PSR	modules/b2mod_b2plot.F
PSR	modules/b2mod_elements.F
PSR	modules/b2mod_sputter.F
SOLPS4_3	modules/b2mod_b2_to_astra.F
SOLPS4_3	modules/b2mod_diag.F
SOLPS4_3	modules/b2mod_mwti.F90
SOLPS4_3	modules/b2mod_neutrals_namelist.F
SOLPS4_3	sources/eirene_f30f31.F
UAL_MAJOR_VERSION	ids/b2mod_ual.F90
USE_CDF_B2STIME	modules/b2mod_comtimpl.F
USE_CDF_B2STIME	common/COUPLE/COMTIMPL.F
USE_IMPGYRO	driver/b2mod_driver.F
USE_IMPGYRO	driver/b2mod_main.F
USE_IMPGYRO	modules/b2mod_impgyro.F
USE_MPI	driver/b2mnastra.F
USE_MPI	driver/b2mn_opt.F
USE_MPI	driver/b2mod_driver.F
USE_MPI	driver/b2mod_main.F
USE_MPI	modules/b2mod_boundary_namelist.F
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
USE_MPI	modules/b2mod_feedback.F
USE_MPI	modules/b2mod_impgyro.F
USE_MPI	modules/b2mod_input_profile.F
USE_MPI	modules/b2mod_neutrals_namelist.F
USE_MPI	modules/b2mod_numerics_namelist.F
USE_MPI	modules/b2mod_sputter.F
USE_MPI	modules/b2mod_transpbgr_namelist.F
USE_MPI	modules/b2mod_transport_models.F
USE_MPI	modules/b2mod_transport_namelist.F
USE_MPI	modules/b2mod_user_namelist.F
USE_MPI	modules/b2mod_wall.F
USE_MPI	utility/xertst.F
USE_PXFGETENV	b2plot/b2plot.F
USE_PXFGETENV	b2plot/initialize_plotting.F
USE_PXFGETENV	b2plot/set_extralabel.F
USE_PXFGETENV	b2plot/spect.F
USE_PXFGETENV	convert/b2mds.F
USE_PXFGETENV	convert/b2co_movies_cdf.F
USE_PXFGETENV	convert/b2co_tally_cdf.F
USE_PXFGETENV	convert/b2co_time_cdf.F
USE_PXFGETENV	convert/b2co_wall_cdf.F
USE_PXFGETENV	driver/b2co.F
USE_PXFGETENV	ids/b2_ual_write.F90
USE_PXFGETENV	modules/b2mod_ppout.F
USE_PXFGETENV	modules/b2mod_solpstop.F
USE_PXFGETENV	preprocessing/b2agfs.F
USE_PXFGETENV	utility/open_file.F
USE_PXFGETENV	utility/usnam.F
WANT_PROCESS	convert/process.F
WINDOWS	b2plot/initialize_plotting.F
WINDOWS	driver/b2obeq.F
WINDOWS	driver/b2obfn.F
WINDOWS	driver/b2oblml.F
WINDOWS	driver/b2op.F
WINDOWS	modules/b2mod_b2plot_values.F
WINDOWS	modules/b2mod_sputter.F
WINDOWS	postprocessing/b2yv.F
WOS	b2plot/additional_elements.F
WOS	b2plot/flux.F

Table J.3: Defines for B2.5 and where they are used.

J.4.6 Eirene

We list here the possible compiler options that can be used when compiling Eirene. The options below must therefore, if desired, be declared as part of either the `$SOLPS_CPP` or the `$EIRENE_DEFINES` environment variable, which are defined in the `$SOLPSTOP/SETUP/config.$HOST_NAME.$COMPILER(.local)` files.

MPI_VERSION Index of MPI version (set internally by the Makefile)

TRACE Turns on tracing output in geometry routines (set internally by the Makefile when requesting a `_debug` compilation)

If you want to activate any of these options, you should add `-DOPTION` (where `OPTION` might, for example, be `TRACE`) to either the `SOLPS_CPP` or the `EIRENE_DEFINES` environment variable with

```
setenv EIRENE_DEFINES '-DOPTION'
```

and then recompile all of the routines affected (if unsure which routines are affected, recompile them all with

```
gmake clean
gmake depend
gmake
```

). If `EIRENE_DEFINES` already has a value, you would want to do

```
setenv EIRENE_DEFINES "$EIRENE_DEFINES -DOPTION"
```

or

```
EIRENE_DEFINES += -DOPTION
```

so that the new option is added to the list. (Unsupported `ksh` or `bash` users would use something like

```
export EIRENE_DEFINES="$EIRENE_DEFINES -DOPTION"
```

). If choosing to use `SOLPS_CPP` instead, the procedure is the same as in the previous section.

We list below where the `Eirene` compilation switches are used.

name	used in
ALLOCATE_AND_NAMELIST	startup-routines/find_param.F
B25_EIRENE	main-routines/eirene.f
B25_EIRENE	interfaces/couple_SOLPS-ITER/eirmod_extrab25.F90
B25_EIRENE	startup-routines/find_param.f
COMPILER_MAJOR_VERSION	assistant/exit_own.f
COMPILER_MINOR_VERSION	assistant/exit_own.f
DBG	interfaces/couple_SOLPS-ITER/eirmod_infcp.F
GFORTTRAN	modules/eirmod_mpi.F90
GFORTTRAN	assistant/exit_own.f
G95	assistant/exit_own.f
F2003	assistant/subtit.f
HPUX	modules/eirmod_cinit.F
MPI_VERSION	modules/eirmod_mpi.F90
NAGFOR	file-handling/get_solpstop.f
NO_B2_CHEM_SPUT	interfaces/couple_SOLPS-ITER/eirmod_infcp.F
NO_GETENV	file-handling/get_solpstop.f
PGF90	assistant/exit_own.f
TRACE	geometry/time-routines/checks.F
TRACE	geometry/time-routines/timea.F
TRACE	geometry/time-routines/timep.F
TRACE	geometry/time-routines/timer.F
TRACE	geometry/time-routines/timet.F
USE_MPI	assistant/exit_own.f
USE_MPI	modules/eirmod_mpi.F90
USE_MPI	photons/eirmod_phot_reflec.f
USE_MPI	startup-routines/input.f
USE_PXFGETENV	file-handling/get_solpstop.f
<i>continued on next page</i>	

<i>continued from previous page</i>	
name	used in
WINDOWS	main-routines/eirene.f

Table J.4: Defines for Eirene and where they are used.

If you want to activate any of these options, you should add `-DOPTION` (where `OPTION` might, for example, be `NO_NAG`) to either the `SOLPS_CPP` or the `EIRENE_DEFINES` environment variable with

```
setenv EIRENE_DEFINES '-DOPTION'
```

and then recompile all of the routines affected (if unsure which routines are affected, recompile them all with

```
gmake clean
gmake depend
gmake
```

). If `EIRENE_DEFINES` already had a value, you would want to do

```
setenv EIRENE_DEFINES "$EIRENE_DEFINES -DOPTION"
```

or

```
EIRENE_DEFINES += -DOPTION
```

so that the new option is added to the list. (Unsupported `ksh` or `bash` users would use something like

```
export EIRENE_DEFINES="$EIRENE_DEFINES -DOPTION"
```

). If choosing to use `SOLPS_CPP` instead, the procedure is the same as in the previous section.

J.5 NAG calls in SOLPS

J.5.1 B2.5

The NAG calls from B2.5 are given in table J.5.

caller routine	calling	function
b2mndr.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
b2mod_input_profile.f	E01BEF	computes a monotonicity-preserving piecewise cubic Hermite interpolant to a set of data points
b2mod_input_profile.f	E01BFF	evaluates a piecewise cubic Hermite interpolant, as computed by E01BEF
b2stbr.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
b2tr21.f	F01AAF	performs a matrix inversion (OBSOLETE)
b2tr21.f	F06QFF	performs the matrix-copy operation
b2tr21.f	F07ADF	(DGETRF) computes the LU factorization of a real m by n matrix.
b2tr21.f	F07AJF	(DGETRI) computes the inverse of a real matrix A, where A has been factorized by F07ADF (DGETRF).
b2ux5p.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
<i>continued on next page</i>		

<i>continued from previous page</i>		
caller routine	calling	function
b2ux7p.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
caltraj.f	G05CAF	returns a pseudo-random number taken from a uniform distribution between 0 and 1
caltraj.f	G05CBF	sets the seeds used by the generator mechanism
caltraj.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
daytim.f	X05AAF	returns the current date and time
daytim.f	X05ABF	converts from a seven-integer format time and date, as returned by X05AAF, into a character string, returned via the routine name
heatdiff1D.f	D03PCF	integrates a system of linear or nonlinear parabolic partial differential equations (PDEs) in one space variable.
ilutern_7_solps.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name
macheps.f	X02AJF	returns the machine precision
machsfr.f	X02AMF	returns the safe range of floating-point arithmetic

Table J.5: NAG calls from B2.5

J.5.2 EIRENE

The NAG calls from `Eirene` are given in table J.6.

erf.f	S15AEF	returns the value of the error function erf, via the routine name.
second_own.f	X05BAF	returns the amount of processor time used since an unspecified previous time, via the routine name

Table J.6: NAG calls from Eirene

J.6 Performance monitoring

This is an experimental feature!

This only works at the moment using the `linux.Fujitsu` or `linux.ifort64` compilers, and then only at Garching!

You need to set

```
setenv PERFMON -DPERFMON
```

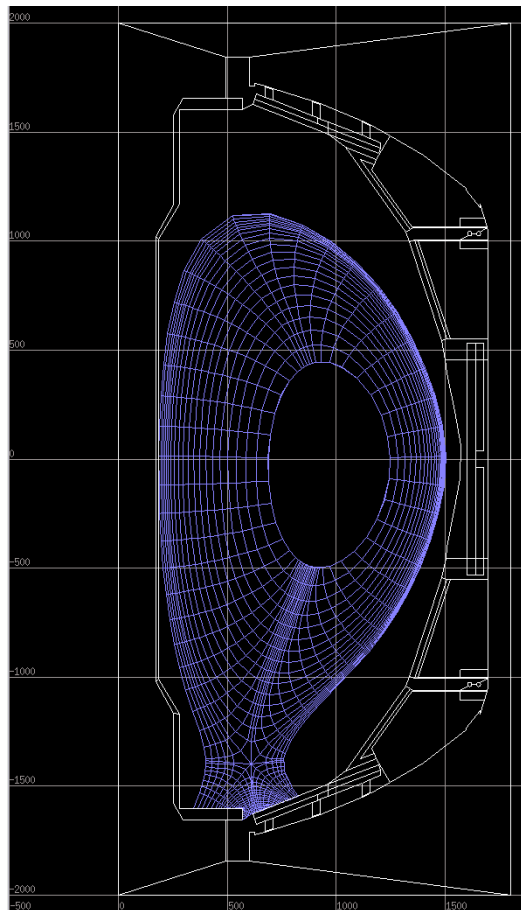
and then do (in `sbb`)

```
rm $OBJECTCODE/prgend.? $OBJECTCODE/prgini.? $OBJECTCODE/subsys.?
gmake
```

Appendix K

DG manual: Version 2.10

A. Kukushkin, K. Kukushkin
ITER Organization
A. Chin, D. Stotler
Princeton Plasma Physics Laboratory



This is the user's manual for the DG geometry definition code. Its primary objective is to simplify the specification of the geometry, and other input data, required for the simulation of the boundary of magnetic fusion devices. The most straightforward application of the code is to divertor tokamak geometries (having one or two X-points), although

in principle other magnetic configurations are possible. The flexible design of DG allows its output to be used with multiple magnetic fusion simulation codes, including B2-Eirene and DEGAS 2. In fact, some of the objects defined in DG and described in this manual may be designed specifically for use by these codes.

K.1 Introduction

This is the user's manual for the DG geometry definition code. Its primary objective is to simplify the specification of the geometry, and other input data, required for the simulation of the boundary of magnetic fusion devices. The most straightforward application of the code is to divertor tokamak geometries (having one or two X-points), although in principle other magnetic configurations are possible. The flexible design of DG allows its output to be used with multiple magnetic fusion simulation codes, including B2-Eirene and DEGAS 2. In fact, some of the objects defined in DG and described in this manual may be designed specifically for use by these codes.

K.1.1 Purpose of DG

DG is the backbone of the **Interactive Graphical Interface** package. It allows user to visualize the model geometry, to modify it, and to specify the geometry-related data necessary for a grid generator and modelling codes. It was originally designed to alleviate transfer of the divertor geometry from the CAD system to the B2-Eirene code package. However, it has proved to become a powerful tool of more general purpose, allowing one, e.g., to specify arbitrary shape of the divertor structures, to check visually the quality of the computational grids produced by a mesh generator, to analyse the error messages from the Monte Carlo routine, to assess easily the compatibility of the plasma equilibrium used in the calculations and the wall geometry of the machine, etc. It can also be very efficient as a sort of very simple drawing program to design the problem geometry from scratch, and not only for B2-Eirene. With use of this code, the most boring and time-consuming part of the preparatory work necessary for the edge plasma modelling in a realistic geometry, becomes really a pleasure. Try it and see for yourself!

K.1.2 DG objects

External objects

The external objects are the files which DG imports or exports. They normally have default extensions by which are used by the code when displaying the lists of files in the I/O control windows. The following table gives a brief description of the files used in DG.

	Description	Extension
Input files:		
DG model	All the data relevant to the model	.dg
Template	Geometry data from CAD drawing or other source	.ogr
Equilibrium	Values of the poloidal magnetic flux on a regular grid	.equ
Mesh	The computational grid to be used by B2-Eirene	
Output files:		
DG model	All the data relevant to the model	.dg
Output file	Specification of the <i>elements</i> and some other information - to be used by the interface routines producing the input files for B2-Eirene	.dgo
Targets file	Specification of the <i>targets</i> - to be used by the interface routines producing the input files for B2-Eirene and Sonnet or Carre	.trg
Structure file	Specification of the structures - to be used by the interface routines producing the input files for Sonnet or Carre	.str
<i>continued on next page</i>		

continued from previous page		
	Description	Extension
Configuration file:		
	The file containing information on the DG <i>layers</i> and <i>variables</i> .	.dgc

Table K.1: DG external objects

Internal objects

The internal objects are those with which you deal when working with DG.

Objects	Description
Drawing objects	
Points	Just points (displayed as red squares) which can be used to create <i>elements</i> or for diagnostic purpose.
Irregular points	Every point which is not regular. A regular point connects exactly two <i>elements</i> having the same orientation (see <i>External normals</i>)
Elements	Straight line segments (displayed in bright white) connecting <i>points</i> . Normally, they represent material surfaces in the neutral transport code ("additional surfaces" in Eirene parlance). You can edit elements (create, delete, split, join, change orientation, etc.) and assign <i>variables</i> to them.
Surfaces	Contours of constant poloidal flux (displayed in yellow) computed from <i>equilibrium</i> data. You can create, move, and delete them. The surfaces allow you to specify the requirements on the radial mesh for the mesh generator (Sonnet or Carre), and also to check the compatibility of your wall structures with the magnetic configuration.
Grid points	Points on the separatrix which represent partition of the magnetic surfaces in different regions for the grid generator (Sonnet or Carre). They are displayed as dashes in magenta, the separatrix being drawn in red.
Separators	Straight line segments (displayed in cyan) connecting the outer nodes of the computational <i>mesh</i> with surrounding <i>structure</i> . The separators also represent additional surfaces, but you can only re-position their outer edges. Their purpose is separation of <i>additional cells</i> - in terms of Eirene. As of V. 1.4 of DG, "separators" have been temporarily disabled.
Sources	Special points which can be grouped together (displayed as white asterisks). They can be used to specify the point sources of neutrals, - and also in calculation of radiation load onto the first wall.
Equilibrium	A simple visualisation of the equilibrium data - red and blue dashes corresponding to negative and positive, respectively, values of the poloidal magnetic flux.
Mesh	Computational mesh, displayed in blue, produced with a grid generator (Sonnet or Carre) and imported into DG. Allows you to check the quality of the mesh and to make the necessary corrections to the input data for the grid generator.
Chords	Straight line segments representing the Eirene and b2plot diagnostic chords.
Cells	Regions outside the computational mesh surrounded by the elements, separators, and the mesh edges (<i>additional cells</i> - in terms of Eirene).
External normals	Are displayed as magenta dashes and depict the orientation of the elements. They point to the right as you move from the beginning to the end of an element. This is the direction of the external normal in terms of Eirene.
Template	Imported graphics from external source (e.g. from CAD drawing or from old Eirene input file). The template is displayed in cyan, and it can be converted into <i>elements</i> .
continued on next page	

continued from previous page	
Objects	Description
Axes	Coordinate axes which can be displayed on the drawing.
Grid	Coordinate grid which can be displayed on the drawing.
X-point	The X-point of the magnetic equilibrium; it effectively separates the divertor from the core plasma. In double-null configurations, there are two X-points which may or may not be on the same flux surface. The code must be told the topology of the equilibrium (see <i>topology</i>) in order to locate the X-points. This must be done before specifying <i>surfaces</i> or <i>grid points</i> .
Topology	The topology of the magnetic separatrix. A set of commonly used topologies is provided with DG. Once the <i>equilibrium</i> and topology have been imported, the code will attempt to locate the X-point(s); if successful, the separatrix (or separatrices) will be drawn in red.
Additional data	
Variables	Typically used to associate various pieces of data with the <i>elements</i> . This information can be then passed to other codes via the output files.
Layers	Groups of variables.
Targets	Bound the <i>surfaces</i> in the poloidal direction, and thus must be specified for the grid generator. They are thus analogous to the actual divertor targets in a tokamak.
Structure	Bound the <i>surfaces</i> in the radial direction. They are thus analogous to the walls of the vacuum vessel in a tokamak.
Target edges	The elements which should be reconnected to the corresponding corners of the calculational mesh in order to avoid mismatches of the mesh and the targets in the Monte Carlo calculations.

Table K.2: DG internal objects

K.2 Using DG

K.2.1 Starting DG

To start DG with a blank window, type:

```
dg &
```

at the UNIX prompt.

To start DG with an existing model (or several models) loaded, type:

```
dg <filename.dg> [...] &
```

After you start DG, the main window will appear.

K.2.2 Main window

After you start DG, the main window appears. It has four areas: menu bar, toolbar, work area and the message line.

Menu bar

The menu bar gives you access to the most of the DG functionality. It contains several buttons, which can be used to open the corresponding menus by clicking the left mouse button on it. Every menu can be torn off and left on the screen after opening it. In order to do this, just click at the dashed line at the top of the menu window. In this document, we will show menu items in *typewriter* font.

The following menus are available at present:

File You use this to start a new model, to work with external objects (to save or open the model; to import templates, equilibrium files, topologies, or computational meshes; to produce the output files which are used for generation of the input files for computational codes), or to quit the application.

Edit Allows you to perform some global operations, such as recovering from errors, selecting and de-selecting all elements, creation or removal of groups of objects, and shift or rotation of the picture.

View Here you can control the overall view of the model and specify which objects should be visible on the screen.

Commands Allows you to perform global commands on the model such as converting a *template* to *elements* and renumbering the elements.

Variables You will use this to specify additional geometry-related information for the output files (eg., location of targets, boundary conditions for the Monte Carlo routine, etc.).

Options From this menu you can tailor the main window and operating mode to fit you preferences. It also allows you to customize the DG variables (*only for experienced users!*).

Window Allows you to open more windows with different views of the same model, to close the current window, to open a window showing the statistics for the model, and to open the `Toolbox`.

Help This is the way to get some online help on DG and to identify its current version.

Toolbar

The toolbar allows you to assign different *tools* to each of the mouse buttons. It has 3 option menus-for the three buttons- that contain the list of available tools. If your mouse has less than three buttons, then only the leftmost option menu(s) are activated.

Work area

The work area is the area where the drawing is displayed. Here you can move around the drawing, and you can use *tools* to edit it. The work area can optionally display coordinate axes and coordinate grid to simplify positioning of objects.

Message line

The message line is located below the work area. It is used to display error messages and information on the objects. Error messages are sometimes displayed in an error box instead of the message line, and in this case you must click OK to proceed.

K.2.3 Using tools

Tools are used to perform actions in the work area. They allow you to create, move, modify, or remove objects, to query information on the objects and to move around the drawing. In this document, we will indicate tools in *italic typewriter* font.

A tool *must* be assigned to a mouse button in order to be used. You can select the tool to be assigned to each of the mouse buttons either from the toolbar, or from the `ToolBox` (accessible from the `Window` menu). There is one small difference between these two options. Operating with the toolbar, only the left mouse button should be used to select the tools associated with the three mouse buttons. On the other hand, when working with the `Toolbox`, a tool is associated with a given mouse button by clicking that button on the tool.

To use a tool, move the mouse pointer into the work area and depress the mouse button to which the tool has been assigned. Then, you can optionally drag the mouse. To complete the mouse drag, release the mouse button with the mouse pointer still inside the work area. To cancel the action associated with the drag, move the mouse pointer out of the work area and release the mouse button.

Some tools require you to 'select' an object. To select an object, move the mouse pointer to the object and depress the mouse button. The object selected will be highlighted in another colour. The *green* colour means OK, while the *red* one indicates that the operation cannot be done. Normally, some explanatory text appears in such cases on the message line.

K.2.4 Using Undo

You can reverse the effect of almost any operation by issuing the `Edit|Undo` menu command. You can also reverse the effect of `Undo` command by `Edit|Redo`. The DG keeps track of practically all the operations you did starting from opening the window, and thus you can recover from almost all the errors you could make in the course of the model editing. To reverse the effect of all the `Undo` commands you issued, use `Edit|Redo` all.

K.3 Working with models

K.3.1 Opening an existing model

To open an existing DG model from DG, use the `File|Open...` menu command. A dialog box will appear prompting you for a filename. You can either select a file from the list or type the filename in the edit field. Then press `Ok`. A new window displaying the selected model will be opened. If you issue the `File|Open...` command immediately after starting DG, the file will be loaded in the current window. If an error happens, the error box appears. Press `Ok` to close the error box.

To start DG with an existing model loaded, specify the filename on the DG command line. If you specify multiple filenames, DG will open a separate window for each model.

K.3.2 Creating a new model

To create a new, empty model with DG, use the `File|New` menu command. A new blank window will be opened.

When you start DG without parameters, the DG window is blank.

K.3.3 Saving a model

To save a model, use the `File|Save` menu command. A dialog box will appear prompting you for a filename. If the current model has a filename assigned to it, this filename will appear in the edit field. In order to save the model under a different name, or to save the newly-created file, you should either enter the new name in the edit field, or select a name from the list (to overwrite an existing file). After that, press `OK`. An auto-save feature automatically saves a version of the file with a `'~'` appended to the name every 5 minutes (see `Options|Save interval`); of course, a new file must be saved before this can work. If you need to use this file (e.g., in the event that DG crashes unexpectedly), you should rename this file to remove the `'~'`.

K.3.4 Quitting DG

To quit DG, you must close all windows.

To close a window, issue the `Window|Close` menu command. You can also use the window manager to close a window. You can also use the window manager to close a window click the "X" at the top right of the menu. With Motif Window Manager, simply double-click the system menu. If you've made changes to the model in this window, a warning box will appear. Select `Save` to save the model. Select `Discard` to close the window without saving changes. Select `Cancel` to leave the window open.

To close all windows, use the `File|Exit` program menu command.

There is a caveat in the current version of DG that iconified windows containing unsaved work are not de-iconified after these commands. You will have to de-iconify them manually in order to see their warning boxes. All other iconified windows will be closed without problems.

K.4 Working with elements

K.4.1 Converting a template

It is often more convenient to convert a template into a set of elements and then edit them to suit your needs, rather than to define all elements by hand.

First, you should import the template. Choose the `File|Template...` menu command. A dialog box will appear prompting for a filename, similar to that for opening DG files. After importing the template, choose `View|Picture` view to adjust the zoom factor in order to make the template visible. Enable `View|Axes` and `View|Grid` to display the coordinate system.

DG lets you adjust the template position by moving it by a fixed distance and/or rotating it 90 degrees around (0,0). To adjust the position, choose the `Edit|Move/Rotate...` menu command. A dialog box will appear. Enable the `Template` check box. To move the template, enter *X*- and *Y*-increments and press `Move`. To rotate, press `Rotate`. If the template moves out of sight, again choose `View|Picture` view. Press `Close` when done.

To convert the template into a set of elements, choose `Commands|Convert template`.

Note that after the conversion, the template remains visible. This is convenient for editing elements, because the difference from the original template is visible. You can hide or display the template by disabling or enabling `View|Template`. To unload the template permanently, choose `Edit|Delete|Template`.

Elements produced in this manner are sometimes improperly connected or have wrong external normals. Manual editing of the elements may be required to remedy these problems.

K.4.2 Creating elements manually

There are several tools for creating new elements.

Use `Add element` to create the first element or to attach an element to a pre-defined point. Move the mouse pointer to the starting point and press the mouse button. Then move the pointer to the ending point and release the mouse button.

Use `Split element` to split an element into two. Select an element. It will split into two. Drag the newly created point to the desired location and release the mouse button.

Use `Connect points` to create an element between two pre-defined points. Select the first point, move the mouse cursor to the second one and release the mouse button. You cannot create more than one element between the same two points.

You can also use the `Edit|Create|Point . . .` menu command to create points that will be later connected using the `Connect points` tool. A dialog box will appear. Enter the desired coordinates and press `Create`. The dialog box will remain on the screen, allowing you to create more than one point without re-opening it. To close the dialog box, press `Cancel`.

K.4.3 Editing elements

There are several tools for modifications of existing elements.

Use `Delete` to remove an element. Select the element and release the mouse button.

Use `Join elements` to join two advanced elements into one (reverse effect of `Split element`). Select the point between the elements and release the mouse button. You can only join elements around a regular point. There must be exactly two elements attached and their external normals must point to the same direction.

Use `Move/Stretch` to move a point with all attached elements. Select the point and drag it to the new location. Then release the mouse button.

Use `Reposition` to attach one end of an element to another point. Select the element near one of its ends, move the mouse pointer to another pre-defined point and release the button.

K.4.4 Changing external normals

Having the external normals of adjacent elements aligned permits them to be easily marked; some operations also require the normals of adjacent elements to be aligned (i.e., for the points involved in the operation to all be regular).

To display or hide external normals, enable or disable `View|External normals`.

Use the `Reverse normals` tool to change directions of external normals. Select an element to change its normal. Moving the mouse pointer to adjacent elements will force their normals to point to the same direction as that of the first element. Release the mouse button to end the action.

K.4.5 Renumbering elements

Each element has a unique integer label associated with it. These determine the order in which the elements are written into the output file. This feature is very useful for debugging. DG uses an internal counter to assign these numbers. Each time the user creates an element, the counter is incremented by one and its value is used as the element number. However, when the user deletes an element, the remaining elements are not renumbered and the counter is not decremented. Since many unused numbers consume disk space and may slow down programs that use DG output, these unused element numbers should be periodically eliminated. Renumbering elements is absolutely safe except when you are doing some debugging and need these numbers to identify elements in the output file.

To display or hide element numbers, enable or disable `View|Display|Numbers`.

To count unused numbers, choose `Window|Statistics` in order to display information about the current file. The 'unused numbers' entry contains this count.

To eliminate unused element numbers, choose `Commands|Renumber elements`.

K.5 Controlling how objects are displayed

K.5.1 Changing the zoom factor and location

You can use the `Zoom` tool to move around the drawing and to change the zoom factor. You can also use commands from the `View` menu.

To move around with the `Zoom` tool, press and release the mouse button anywhere in the work area without moving the mouse. The work area will be adjusted so that the location where the button was pressed is moved to the centre. Alternatively, you can press `Shift+mouse button`, move the mouse pointer and release the button. The work area will be adjusted so that the location where the button was pressed is now shown where the button has been released.

To change the zoom factor with the `Zoom` tool, press the mouse button, move the mouse in order to draw a rectangle around the area to be shown in the work area. Then, release the mouse button. Note that the area actually shown may be enlarged in order to preserve the aspect ratio.

The `View|Zoom in` command increases the zoom factor.

The `View|Zoom out` command decreases the zoom factor.

The `View|Picture view` command adjusts the zoom factor and location so that all objects fit into the work area. This command ignores objects that are hidden because of settings in the `View` menu.

The `View|Selection view` command adjusts the zoom factor and location so that all marked objects fit into the work area.

K.5.2 Hiding and displaying objects

Toggle buttons in the `View` menu let you hide or display the coordinate system and various kinds of objects.

Note that you cannot select an object that is not displayed. In particular, you can only select points that are displayed (a small square is drawn around the point). The `View|Points` setting controls the display of points. When it is disabled, the `View|Irregular points` setting controls the display of irregular points. It is convenient to display only irregular points while correcting a converted template, because irregularities can be seen very easily.

K.5.3 Rotating the view

The display may now be viewed at any angle of rotation. The user may specify the desired angle by using the `Rotate` tool, the commands under the `View|Rotate` menu, or the equivalent keyboard shortcuts to increment ("`Ctrl+(`"), decrement ("`Ctrl+)`"), or zero the angle (`Rectify`, shortcut "`Ctrl+C`"); see Fig. K.1. The rotation does not affect the underlying structures being displayed, but only the drawing procedure. The `Previous view` command will undo rotations. Rotations are saved starting with file format version 115. Files of older formats are still compatible; the default angle will be 0. Note that the `Picture View` command (`View|Picture View`) zeroes the rotation angle.

K.5.4 Stretching and shrinking the view

The aspect ratio may be "unlocked" by enabling the `View|Mode|Stretch` button. In `Stretch` view mode, the `View|Stretch/Shrink` menu item becomes active (see Fig. K.2), and the user may use these tools to stretch or shrink the x and y axes. To stretch or shrink the current view by a factor of 2, use the first four buttons of this menu. The keyboard shortcuts are "`Ctrl+Shift+H`" to stretch horizontally (in the x direction), "`Ctrl+H`" to shrink horizontally, "`Ctrl+Shift+V`" to stretch vertically (in the y direction), and "`Ctrl+V`" to shrink vertically. Precise stretching or shrinking factors can be specified with the `Stretch . . .` dialog. The aspect ratio can be reset with the `Justify` menu item. Additionally, zooming with the highlighted rectangle now zooms to display exactly the contents inside the

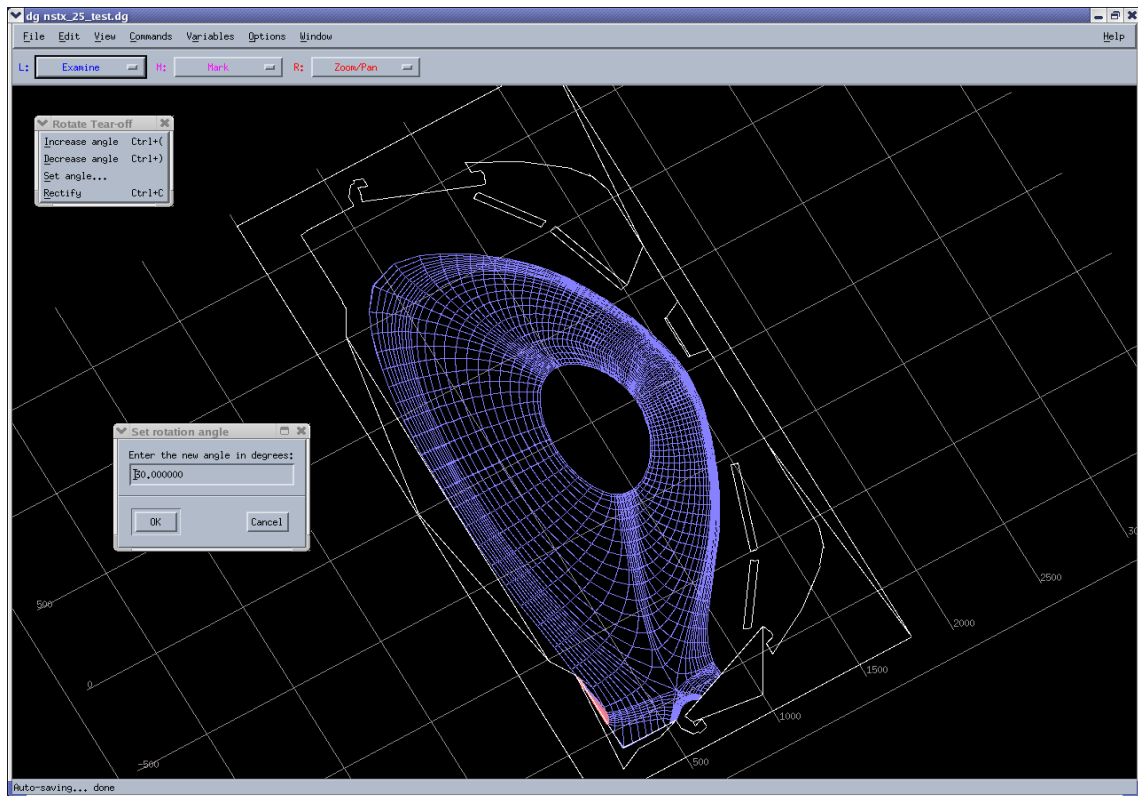


Figure K.1: Demonstration of a rotated view with associated dialog box and tear-off menu.

rectangle, rather than expanding to ensure the proper aspect ratio. Disabling `Stretch` mode redraws the display with proper aspect ratio and preserves the ratio in future changes of view.

The `Stretch` tool works a little differently since it can stretch either horizontally or vertically. If the mouse is within 45° of an imaginary horizontal line through the center of the window, a mouse drag will stretch or shrink the view in the horizontal direction. Likewise, if the mouse is within 45° of the vertical line through the center of the window, a mouse drag stretches or shrinks in the vertical direction. If `Stretch` mode is disabled, the `Stretch` tool does nothing.

Note that the `Picture View` command (`View|Picture View`) resets the aspect ratio, but does not disable `Stretch` mode.

K.5.5 Top-down view

To facilitate visualization and specification of chords with nonzero toroidal extent (see Sec. K.7), a view from above the torus is now available via the `View|Mode|Top-down view` button. When this button is enabled, the main window depicts the view of the present geometry looking down along the y axis. The elements in the model are assumed to be toroidally symmetric. In this view, the elements are displayed as 48 (arbitrarily chosen) straight line segments. The only other objects currently displayed in this view are chords, although working drawing procedures exist for all markable objects. Since displaying every element in the model would likely result in confusion, the user must select the specific elements to be depicted. The variable `Top View Elements` has been defined for this purpose (`Variables|Top View Elements`). The appearance of chords (with some toroidal extent) can be toggled with either `View|Display|Chords` or `View|Display|3dchords`. As in the normal (poloidal) view, `View|Display|Elements` toggles the appearance of elements in the top view. Since top view was intended to be used only for chords, many tools are disabled in this mode. Likewise, since top view uses the same screen coordinates

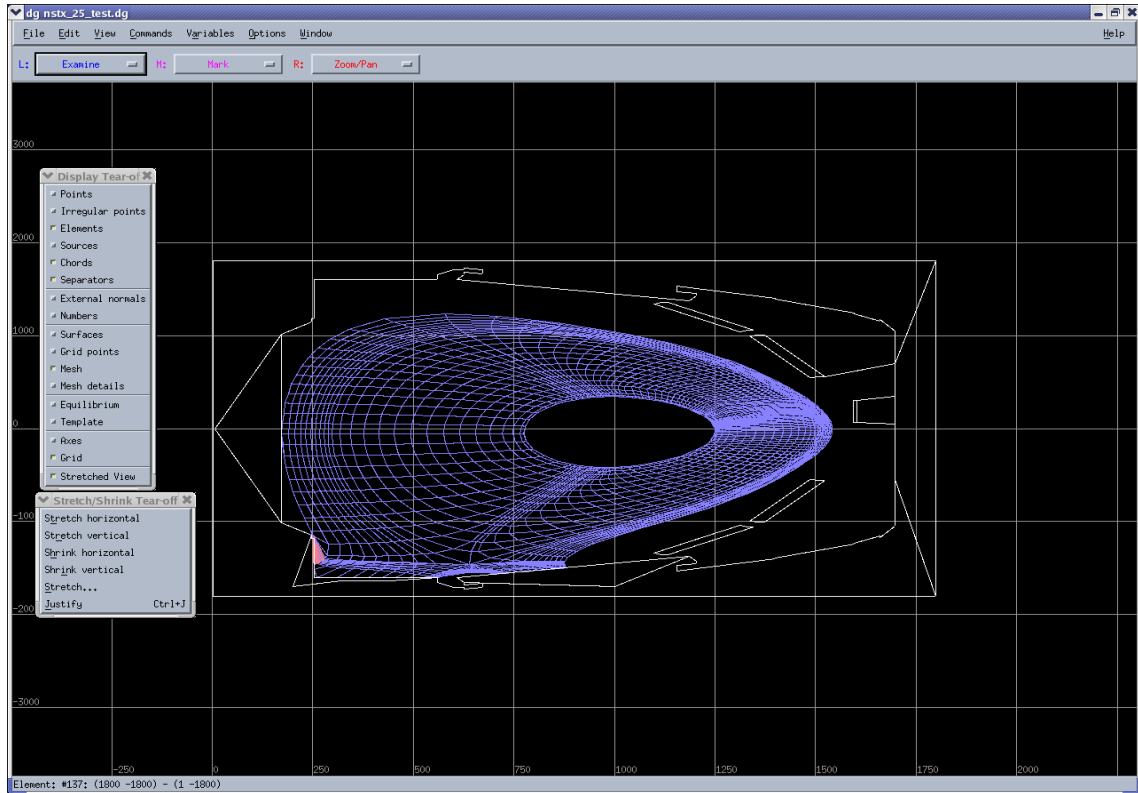


Figure K.2: Demonstration of the variable aspect ratio capability along with the associated tear-off menu.

as the normal view, it may be necessary to re-zoom after switching between the two views. `View|Picture View` can be used for this purpose in either mode.

K.6 Working with magnetic surfaces and grid points

K.6.1 Equilibrium conversion and manipulation

The `$SOLPSTOP/modules/DivGeo/equtrn` directory has some utilities for converting between various widely used formats for magnetic equilibrium files and for making minor modifications to them.

The listings and documentation below may be out of date. An updated documentation of the supported scripts can be found in `$SOLPSTOP/modules/DivGeo/equtrn/doxygen/refman.pdf`.

d2d

This is a script that invokes the `dg2dg` executable. Its sole function is to double the resolution of a DG equilibrium file. To invoke it:

```
d2d <filein>.equ <fileout>.equ
```


d2e

This is a script that invokes the `dg2ef` executable to convert a DG equilibrium file to the EFIT format (specifically, the EFIT "g" file). This is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
d2e <filein>.equ <fileout>.efit
```

Some older DG files lack the magnetic field data needed for this conversion. The default name for the file containing its value is `field.snn` (i.e., using the above invocation). Alternatively, the name of the file containing the field can be specified as a third argument:

```
d2e <filein>.equ <fileout>.efit <fieldfile>
```

d2v

This is a script that invokes the `dg2vr` executable to convert a DG equilibrium file to the Carre (or TdeV) equilibrium format. This is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
d2v <filein>.equ <fileout>.v
```

Some older DG files lack the magnetic field data needed for this conversion. The default name for the file containing its value is `field.snn` (i.e., using the above invocation). Alternatively, the name of the file containing the field can be specified as a third argument:

```
d2v <filein>.equ <fileout>.v <fieldfile>
```

e2d

This is a script that invokes the `ef2dg` executable to convert an EFIT format (specifically, the EFIT "g" file) equilibrium to the format read by DG. This is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
e2d <filein>.efit <fileout>.equ
```

Another program that produce output in an EFIT-like format that is readable by DG using this script is `FreeGS`¹.

n2d

This is a script that invokes the `nk2dg` executable to convert an equilibrium in the "Naka" format to the format read by DG. This is invoked with (the file suffixes are suggested for clarity and are not enforced):

```
n2d <filein>.nk <fileout>.equ
```

rps

This is a script that invokes the `risepsi` executable. Its sole function is to increase the poloidal flux values in the DG equilibrium file by the specified amount. To invoke it:

```
rps <filein>.equ <fileout>.equ <increment>
```

¹Documentation available at <https://freegs.readthedocs.io/en/latest/index.html>.

v2d

This is a script that invokes the `vr2dg` executable to convert an equilibrium in the `Carre` (or `TdeV`) format to the format read by `DG`. This is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
v2d <filein>.v <fileout>.equ
```

By default, this script will look for the value of the magnetic field in a file `field.snn`. Alternatively, the name of the file containing the field can be specified as a third argument:

```
v2d <filein>.v <fileout>.equ <fieldfile>
```

cropequ

This is an executable (no corresponding script) that crops a `DG` equilibrium file. The code is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
cropequ <filein>.equ <fileout>.equ
```

The code will print out the dimension and spatial extent of the input equilibrium grid then request the values for the output grid in the order: number of radial grid points, minimum radius, maximum radius, number of vertical grid points, minimum vertical coordinate, maximum vertical coordinate. The code will require the requested spatial extent to lie within that of the input grid. If it does not, the offending parameter is reset to match the corresponding value from the input grid.

The output file will contain a new equilibrium, re-computed with the new specified grid dimensions, with interpolation of the data to the new grid point locations.

dgcut

This is an executable (no corresponding script) that chops a `DG` equilibrium file. The code is invoked with (the file suffixes are recommended for clarity and are not enforced):

```
dgcut <filein>.equ <fileout>.equ
```

The code will print out the spatial extent of the input equilibrium grid then request the values for the output grid in the order: minimum radius, maximum radius, minimum vertical coordinate, maximum vertical coordinate. The code will require the requested spatial extent to lie within that of the input grid. If it does not, the offending parameter is reset to match the corresponding value from the input grid.

The output file will contain only the data from the original input file that is within the new coordinate bounds being prescribed.

prinequ

This is an executable (no corresponding script) that prints out the contents of a `DG` equilibrium file in tabular format. The file is read from standard input and the result is sent to standard out. So, the preferred invocation would be (the file suffixes are recommended for clarity and are not enforced):

```
prinequ < <filein>.equ > <fileout>.txt
```

K.6.2 Importing equilibrium

An equilibrium file must be imported before defining any surfaces or grid points.

To import an equilibrium file, choose `File|Import|Equilibrium...`. A dialog box will appear prompting for a filename. Select a file and press `Ok`. Select `View|Picture` view to make sure the equilibrium grid is visible in the work area. Red colour represents positive ψ values, while blue represents the negative ones.

K.6.3 Output mode

The detailed procedures for creating and editing surfaces and grid points differ slightly for the `Sonnet` and `Carre` mesh generators. So, before undertaking these tasks, the user should select the appropriate output mode in `Options|Output mode`. Since only the `Carre` code is distributed along with this manual, only procedures pertinent to it will be described.

This dialog box also allows the user to determine which objects are examined for consistency when DG's output is requested (via `File|Output...`). Selecting all of these checks is recommended. Users should override them only if they are aware of the consequences of the object in question being improperly defined.

K.6.4 Creating and editing surfaces

As was noted in Sec. K.1.2, surfaces are drawn between targets and are bounded by structures. The target related variables are single elements set using the variables `Variables|Inner edge of inner target`, etc., where the first "inner" and "outer" refers to the inner and outer edge of the computational mesh, respectively, and the second to the usual inner and outer divertor target. There is an additional set of targets available for double-null geometries; see: `Variables|Add|Extra targets for DN`. The structures are set with the variable `Variables|Structure`. There may be one or more of these.

In `Carre` mode, the user begins the process of surface creation by manually creating the surface that will represent the innermost core surface via the `Add surface` tool. Just click it inside core plasma at the desired location (the tool will not work anywhere else). Its location can be altered via the `Move` tool. It can only be deleted by executing the `Edit|Delete|Surfaces` command (this deletes *all* surfaces).

The next step is to open the `Create surface(s)...` dialog, `Edit|Create|Surfaces...` (see Fig K.3). In `Carre` mode, the user can only create multiple surfaces (i.e., the `Single` option will not work). The `Area` menu determines in which part (topologically) of the equilibrium the surfaces will be drawn. The number of cells must be input manually. The line graph in the dialog box is a simple depiction of the relative spacing of the surfaces. In `Carre` mode, only the `Delta Law` is available. In this case, the two parameters `Delta1` and `Delta2` specify a quadratic function used to determine this spacing (The deltas specify the relative normalized difference between the levels of the first two, and the last two surfaces).

While these parameters can be input directly, the preferred means of setting them is to click and drag the black line in the graph to achieve the desired spacing variation. When the `Create` button is clicked, the surfaces will be generated with this spacing and drawn in DG's main window. If a model has an existing set of surfaces that the user wishes to tweak, click on one with the `Examine` tool and then click the `<<` button in the `Create surface(s)...` dialog; this will import the parameters describing these surfaces.

Note that the dialog will *not* enforce constraints on the number of cells (e.g., that the number of core and PFR cells must be equal); the user must check to be sure that all such constraints are satisfied. `Carre` will perform its own set of checks when invoked as well.

DG's algorithm for calculating the outermost flux surface level for unbounded regions (e.g., PFR and SOL) uses the `Structure and Inner / Outer targets` variables. The only elements that the algorithm is aware of are those specified in the `Structure` variable, shown in Fig. K.3. Additionally, the algorithm recognizes at least two subsets of the structures, the "targets", which consist of all closed loops in the structure on which elements in the `Inner target` and `Outer target` variables lie. All target elements must lie on a closed loop in order that the targets have a well-defined "inside" and "outside". The outermost surface level is then defined as the flux value differing most greatly from the value on the separatrix, whose corresponding flux surface passes through every target but encounters

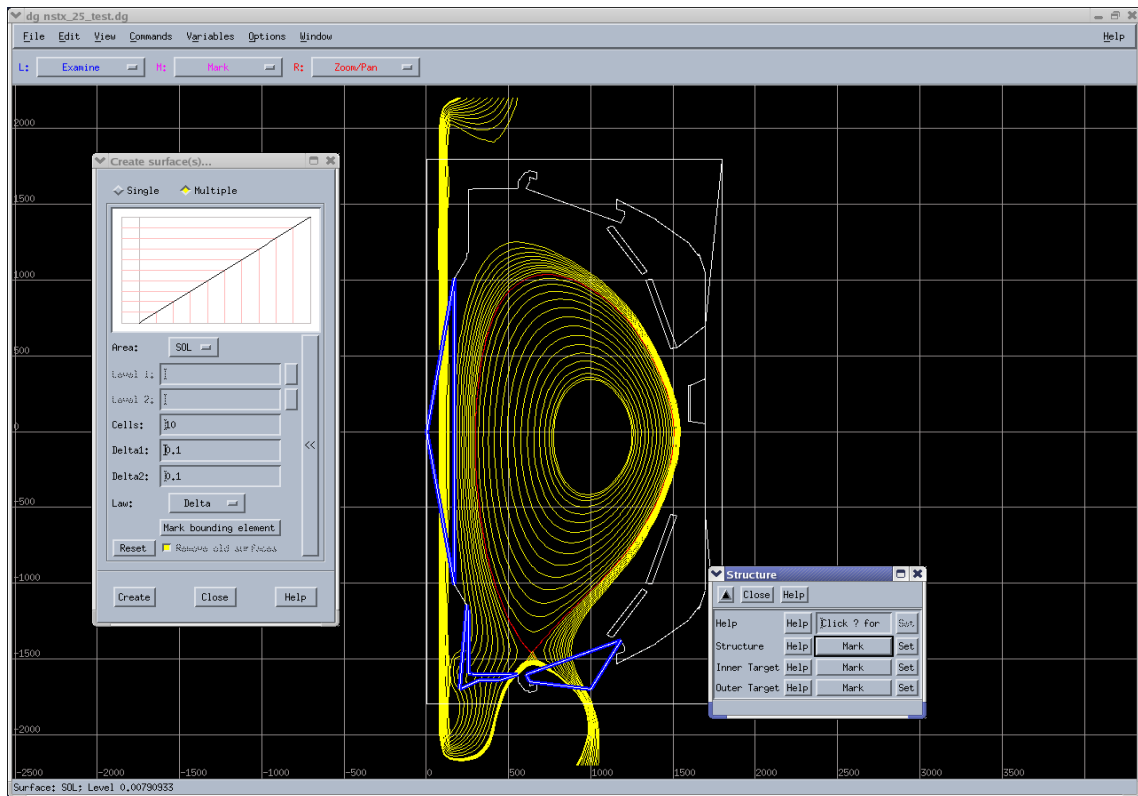


Figure K.3: Create Surface(s) . . . dialog with highlighted Structures.

no other elements in the structure. This value is computed, to as much accuracy as the resolution of the equilibrium allows, as the least of 1) the global maximum on each target and 2) the global minimum on all other elements in the structure, where maximum and minimum are understood to be in terms of absolute deviation from the separatrix flux.

With this procedure, it is clear that the outermost level must be a flux extremum of some element of the structure; we call this element the "bounding element". The capability was added to DG to record this bounding element during surface creation and mark it from within the Create Surface(s) . . . dialog. See, e.g., Fig. K.4. Note that the core region has no bounding element. By identifying what DG considers as the bounding element, we may easily adjust the extent of the flux surfaces by moving the bounding element to change the level of its extremum. Note that once the structure is changed, the bounding element might differ than before.

In case of difficulty . . .

The process of creating surfaces is perhaps the most confusing and difficult aspect of using DG. Moreover, the algorithm it uses for choosing the outermost flux surface level differs in detail from the one used by Carre. If you are having problems getting the surfaces you want, you might benefit from reading the precise description of DG's current algorithm for setting the outermost flux level:

It is to be understood that "maximum" and "minimum" always refer to the absolute flux deviation from the separatrix. First, find the surface zone and the separatrix(ices) of the zone. If there are two separatrices, draw surfaces from one to the other. Else if the zone is marked as "limit by surface", find the farthest surface in the zone and draw surfaces to it. (In practice, only one surface by be added manually in this zone, so that surfaces are always bounded by that one.)

Else, initialize the maximum level to infinity and get the structure variable, which includes the targets. The structure is split into chains of adjacent elements, where each element in a chain may be connected to at most one other element at each endpoint. If there are branching points in any chain, an error is returned. The chains are organized into three

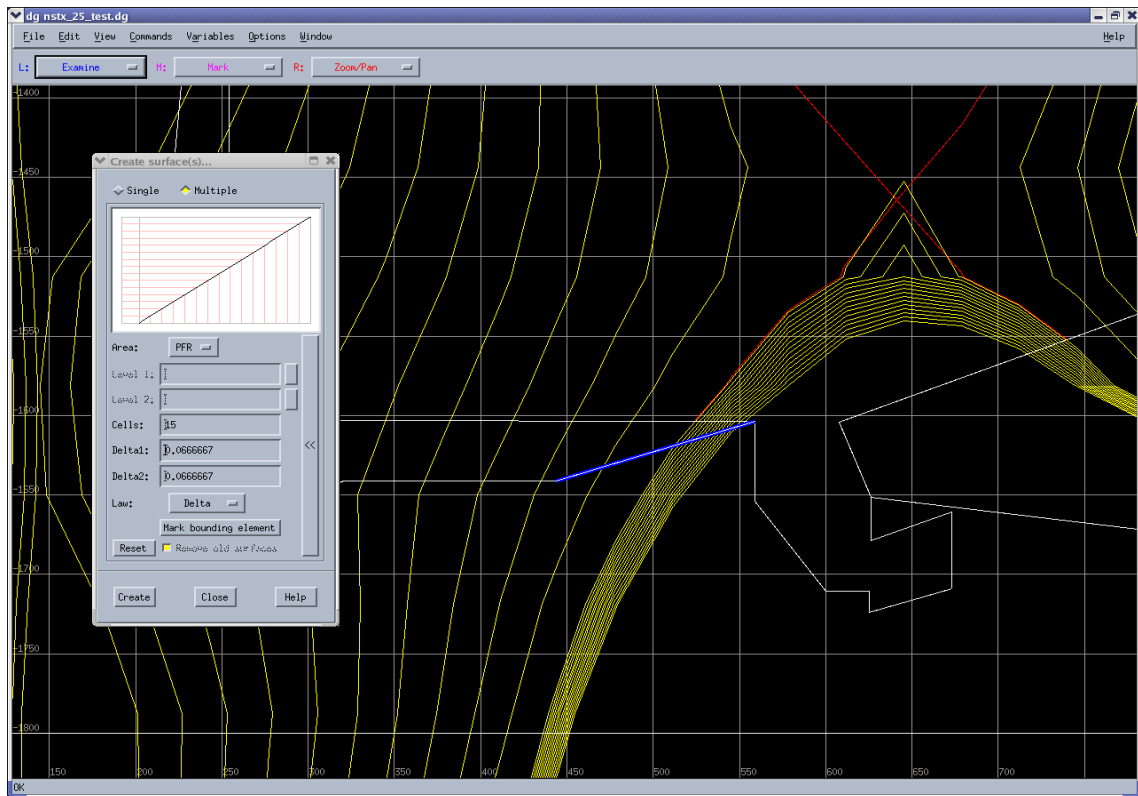


Figure K.4: Zoom of the PFR showing the highlighted "bounding element".

groups: the first group consists of chains which contain elements that have been specified as a target. These chains must contain the entire target variable and must form a closed loop, otherwise an error is returned. The second group consists of chains with no target elements that form a closed loop, and the third group consists of chains with no target elements that do not form a closed loop.

For each chain in the first group, for each element in the chain, if neither endpoint is in the surface zone, skip it, else get the maximum and minimum flux levels on the element from the equilibrium. Find the maximum level on the chain, and set the surface level maximum to the innermost of these levels. Also, find the minimum level on the chain, and set the surface level minimum to the outermost of these levels.

For each of the elements in the second or third group, if the first endpoint is not in the surface zone, skip it, else get the minimum flux level on the element. If it is less than the current surface level minimum, skip it, else if it is still less than the current surface level maximum, set the surface level maximum to this level.

You can use the Add surface tool to create surfaces visually. It is also possible to create surfaces with the Edit|Create|Surface menu command. To move or delete surfaces, use Move/Stretch and Delete tools.

To use the Add surface tool, press the mouse button anywhere in the equilibrium. A surface will appear. Move the mouse pointer to adjust the surface position. When ready, release the mouse button.

The Edit|Create|Surface menu command is similar to Edit|Create|Point. A dialog box will appear prompting for coordinates. After you press Create, a magnetic surface will be drawn through the specified location.

To use Move/Stretch and Delete tools, select a surface. Then drag it to the new location and release the mouse button resp. simply release the mouse button to delete it.

To delete all surfaces, use the Edit|Delete|Surfaces menu command.

K.6.5 Defining the X-point

In some older DG version, you must define the X-point before working with grid points. This is necessary if version of DG used cannot detect the X-point location automatically. By defining it, you give DG a hint how to display grid points and process user input. DG does not write the X-point location into the output file.

For performance reasons, the separatrix is drawn as series of line segments. Because of this, it is not displayed very accurately, especially near the X-point.

First, define the targets so that they are intersected by the separatrix. Then, use the `Set X Point` tool to define a rectangle around the X-point. Press the mouse button at one corner of the rectangle, move the mouse pointer to the opposite corner and release the mouse button. In case of success, the separatrix will be drawn in the work area. Otherwise, an error message will be displayed in the message line. If you get the `Invalid X point location` error message, try selecting a bigger rectangle.

Note that the separatrix inside this rectangle is represented by two intersecting line segments that could even intersect with surfaces!

With an X-point defined, you can use the `Set X Point` tool to redefine it.

K.6.6 Limiter configurations

If you are dealing with a limiter configuration, DG needs to be "tricked", because its internal algorithm requires some quantities that can only be computed in a divertor topology. The implemented solution is to provide two sets of variables, one of which is a dummy only used to satisfy the DG internal requirements, and another set that provides the actual equilibrium information.

The procedure to follow is:

1. Import a single-null Equilibrium. The best would be something of a very different size, so that it lies completely outside the geometry you really want to model. Import the single-null Topology. Create a Surface in the core region of the diverted equilibrium as usual - preferably at about the same depth that you will want for the limiter case grid. Create also two closed polygons to act as targets for this SN equilibrium. Declare them as `Inner target` and `Outer target`, and include them as part of the Structure in the usual way, as described in Section K.6.4.
2. Add a `Limiter` configuration variable, and cut and paste the name of the limiter configuration equilibrium file into the `Real equilibrium` there. Specify the depth of the grid you wish to obtain by setting the `Core width` (in units of m).
3. Include the limiter walls in the Structure, plus some grid-limiting parts if needed.
4. The `Target specification` variables refer to the two halves of the limiter on either side of the tangency point. Provide there all relevant material and sputtering properties. `Targets #1` and `#2` should be positioned so that you travel clockwise around the plasma when going from `Target #1` to `Target #2`. Use the real elements you want to be reconnected to the grid corners as the SOL edges. For the PFR edges, create two fake elements and specify 0 as the Surface Type in the General Surface Data for these two elements. Although these may be fictitious elements, they must NOT be included in the `Elements not for Eirene` list.
5. Create the Surfaces and Grid points as usual. Only the Core zone for the Grid points and the SOL and Core areas for the Surfaces are meaningful.
6. In Global Eirene data, make sure that the Major radius is set to its real positive value (in units of cm).

7. As usual, define all other variables (Surface data, Gas puff, Plasma species, etc...) as needed. When defining the TRIA-EIRENE parameters, define the contour indices as they pertain to the limiter configuration equilibrium.

Once you are finished setting up all the case variables, follow the usual procedure: Output the case files, use the `lms` script to establish the necessary symbolic links (the scripts checks if you are doing a limiter case, and, if so, will have `dg.equ` link to the file you indicated as the Real equilibrium), then use the `carre` - script to create the grid, Import the Grid and inspect it. You may have to play with the usual settings for the grid generation, such as the Surfaces and Grid points together with the Carre guard length in the Target specifications, and also with the Core width in the Limiter configuration in order to get the grid which suits you and is acceptable for the plasma modelling code (no pink segments in the grid). If you find it impossible to create the necessary distribution of the surfaces or the grid points because of local extrema appearing on the visualisation curve, then try to adjust the Radial scale or Poloidal scale in the Limiter configuration layer.

K.6.7 Creating and editing grid points

The Add grid point tool lets you visually define grid points. In addition, the Edit|Create|Grid point(s) menu command lets you define a grid point by specifying its value or fill a zone on the separatrix with a given number of grid points. In Carre mode, the only available method for creating grid points is with the multiple option of the Create grid point(s) . . . dialog box. In particular, the Add grid point tool will not work, nor will the Move or Delete tools have any effect on the grid points.

In Carre mode, the operation of this dialog box is directly analogous to that used to create surfaces (Sec. K.6.4). The "zone" refers to the section of the separatrix along which grid points are to be created. The number of cells must be input manually. The line graph in the dialog box is a simple depiction of the relative spacing of the grid points. In Carre mode, only the Delta Law is available. In this case, the two parameters Delta1 and Delta2 specify a quadratic function used to determine this spacing. While these parameters can be input directly, the preferred means of setting them is to click and drag the black line in the graph to achieve the desired spacing variation. When the Create button is clicked, the grid points will be generated with this spacing and drawn in DG's main window. If a model has an existing set of grid points that the user wishes to tweak, click on one of these grid points with the Examine tool and then click the << button in the Create grid point(s) . . . dialog; this will import the parameters describing these grid points. Note that the dialog will *not* enforce constraints on the number of cells (e.g., that the number of grid points on either side of the PFR must be equal); the user must check to be sure that all such constraints are satisfied. Again, Carre will perform its own series of such checks when invoked.

With the Add grid point tool, move the mouse pointer to the desired location on the separatrix and press the mouse button. A grid point will be created at this location. You can move the mouse pointer to adjust its position. When finished, release the mouse button.

The Edit|Create|Grid point(s) menu command brings up a dialog box prompting you for a separatrix zone and alternatively, number of grid points to create or a grid point value. Two radio buttons let you choose between filling the zone with grid points and creating a grid point with an exact value. After entering values, press Create. Filling a zone with grid points will remove all previous grid points from that zone.

Use the Move/Stretch tool to move a grid point or to 'stretch' grid points of one zone. To move a single grid point, select it, then drag it to the new location and release the mouse button. To 'stretch' grid points, move the mouse pointer to one of them, press Shift+mouse button and drag it to the new location. All other grid points in that zone will be 'stretched'.

Use the Delete tool to remove grid points. Select the grid point to delete and release the mouse button.

Use the Edit|Delete|Grid points menu command to delete all grid points.

K.7 Working with chords

Chords are presently the only objects in DG that are not confined to the poloidal plane (i.e., that may be non-axisymmetric). For purposes of defining chords in three dimensions, two coordinate modes have been established. In "Cartesian" mode, Z is the vertical axis in the "top-down view" (Sec. K.5.5); the horizontal axis in the poloidal plane is X and the vertical axis is Y (note: this is a left-handed coordinate system). In "Cylindrical" mode, Z is the vertical coordinate in the poloidal plane, identical to Y in "Cartesian" mode. In the "top-down view", the horizontal axis is designated as R and the toroidal angle as φ (counterclockwise relative to the horizontal axis) is entered in degrees.

Chords can be displayed with either `View|Display|Chords` or `View|Display|3dchords`. For chords confined to the poloidal plane, these are identical, and chords will not be visible in the "top-down view". For chords with a toroidal extent, selecting `View|Display|Chords` in the poloidal plane results in straight line segments connecting the X and Y "Cartesian" coordinates of the chord end points. Equivalently in the "Cylindrical" mode, these are the R and Z coordinates at the end points. In contrast, `View|Display|3dchords` shows the projection of the toroidal path of the chords on the $Z = 0$ ("Cartesian") or $\varphi = 0$ ("Cylindrical") plane. In the "top-down view", both `View|Display|Chords` and `View|Display|3dchords` show the straight line paths of the chords in the X - Z ("Cartesian") or $R \cos \varphi$ - $R \sin \varphi$ ("Cylindrical") plane.

Starting with file format version 115, chords with X , Y , and Z components may be saved and loaded.

K.7.1 Creating chords

The `Add Chord` tool allows you to create chords graphically. To create a chord, move the mouse pointer to the desired location of its beginning and press the mouse button the tool is assigned to. Without releasing the mouse button, move the mouse pointer to the desired location of the other end of the chord and release the mouse button. Chords can also be created from marked Elements via the `Commands|Convert|Elements to chords` command. Duplicate chords are not produced and the original elements are deleted.

The third method for creating chords is with the `Edit|Create|Chord...` dialog.

K.7.2 Modifying chords

The `Move` tool allows chords to be manipulated in either the poloidal plane or in "top-down view". The user can manipulate either the `Chords` object (displayed by `View|Display|Chords`) or the `3dchords` object (displayed by `View|Display|3dchords`). The `Adjust chord` tool moves an end point of the chord while keeping its radial coordinate constant. Holding down the "shift" key while using the `Move` tool replicates this behavior, but with slightly different graphics that conform to the rest of the `Move` tool.

Chords have an orientation, just like elements; this orientation is displayed with `View|Display|External normals`. The `Reverse normals` tool can be used to change this orientation. Note that when this tool is dragged, it will only operate on objects of the same type as the first object encountered, either chords or elements.

The `Extend chord` tool linearly extends (i.e., replaces) the second endpoint of a chord to lie on the nearest element, or the nearest marked element if elements have been marked. The `Edit|Extend chords` commands applies this for all marked chords. The `Edit|Mark all chords` command can be used to mark all chords in preparation for extension.

Note: as of August 2007, the "mathematically correct" procedure for extending chords has not been debugged. However, two macros defined in `chords.c` may be used to select a different preferred implementation of the procedure. `EXTCHORD_2D` selects the 2-D procedure, which does not support chords with nonzero toroidal extent, and `EXTCHORD_OLD` selects the partially correct, stable 3-D procedure, which does not work for horizontal chords and ignores horizontal elements. (This is enabled in the current release of DG). If neither macro is defined, the code defaults to the incomplete procedure, which may show unpredictable behavior during use.

The `Edit|Delete|Chords` command deletes all chords.

K.8 Working with sources

K.8.1 Creating and editing sources

DG allows the user to define sources and to attach variables to them.

To create sources, use the `Add source` tool. Move the mouse pointer to the desired location in the work area and press the mouse button. A source will be created at this location. You can move the mouse pointer to adjust its position. When finished, release the mouse button.

To create a source by entering its coordinates, select `Edit|Create|Source...`. A dialog box will appear prompting for coordinates. Enter values and press `Create`. A source will be created. The dialog box will remain on the screen, allowing you to define multiple sources without re-opening it. Press `Cancel` to close the dialog box.

To move a source to a new location, use the `Move/Stretch` tool. Select the source to move, drag it to the new location and release the mouse button.

To delete a source, use the `Delete` tool. Select the source to delete and release the mouse button.

To delete all sources, issue the `Edit|Delete|Sources` menu command.

K.9 Working with meshes

K.9.1 Importing a SONNET format mesh

DG allows you to import a mesh in the `Sonnet` format. `Carre`'s output mesh can be converted into this format, as can meshes produced by other codes. At present, it can only read the ASCII version of the grid file. The grid can then be displayed in the working area.

To import the mesh, use the `File|Import|Mesh...` menu command. A dialog box will appear prompting you for a filename. Select a file from the list or type the filename in the edit field. Then press `Load`. The mesh will be imported and displayed in the work area. If nothing is displayed, select `View|Picture` view to enlarge the area displayed. Also, make sure that `View|Display|Mesh` is enabled.

To hide or display the mesh, toggle `View|Display|Mesh`. To unload it, select `Edit|Delete|Mesh`.

K.9.2 Modifying a mesh

DG's graphical interface permits limited modification of a mesh that has been imported. The altered mesh can be written out to a text file (in the `Sonnet` format) for use by other codes. The primary tool for this purpose is the `Move mesh point` tool. The actions of this tool are governed by the settings specified in the `Options|Mesh editing...` dialog. They only affect one-dimensional movements of mesh points.

`Slide along` specifies how movements of mesh points are confined. When `Magnetic surfaces` is in effect, the movement is confined to a magnetic surface passing through the original location of the mesh point. This method should work fine except for the vicinity of the X-point, where magnetic surfaces drawn by DG become distorted. When `Splines` is in effect, the movement is confined to a spline passing through a row of existing mesh points. This method may be useful in the vicinity of the X-point.

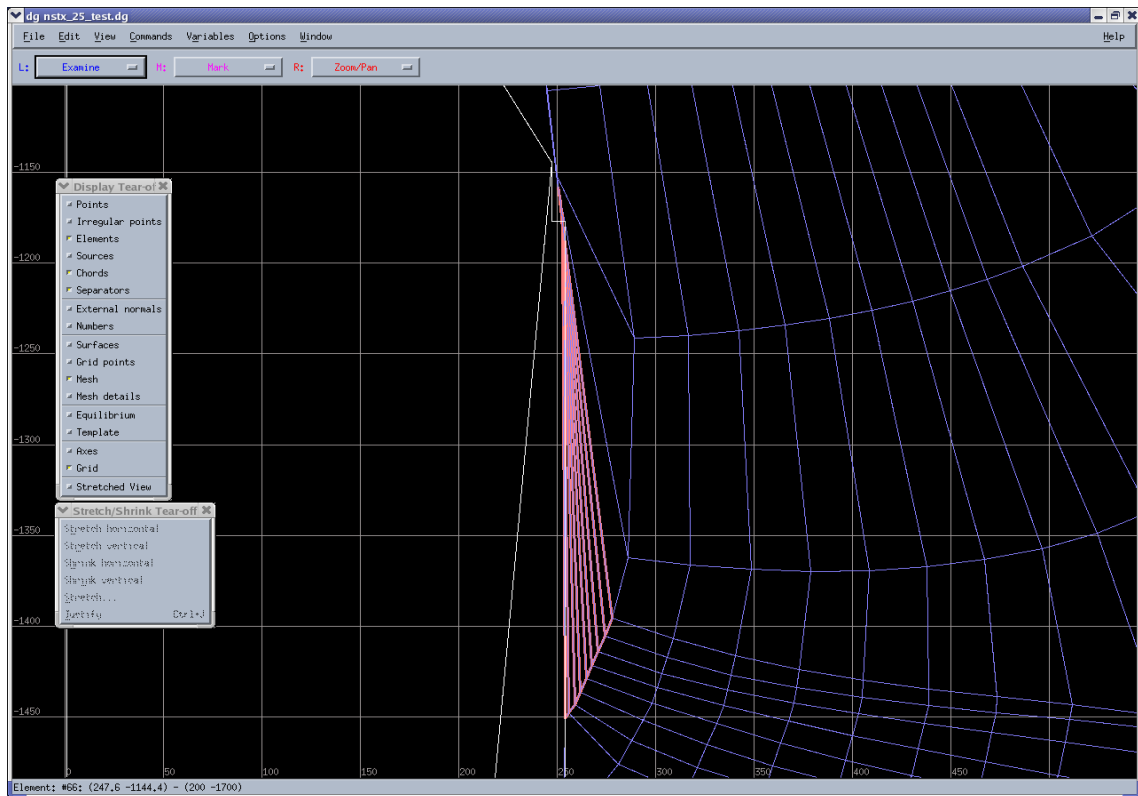


Figure K.5: Zoom of the twisted mesh cells near the inner divertor target of an NSTX geometry.

`Margin` specifies the minimum distance between adjacent mesh points. If this limit is about to get exceeded, adjacent points start moving in order to "make room". The `Double border` setting, when enabled, instructs DG to treat outermost mesh cells as a double border. They would be resized together with adjacent "regular" cells.

A modified mesh can be written out with the `File|Export|Mesh...` dialog.

K.9.3 Repairing mesh cells using rotated view and variable aspect ratio

Mesh cells highlighted in pink in DG's main window are either concave (have an interior angle greater than 180 degrees) or consist of overlap with a neighboring cell. These problems typically arise where the mesh generator (Carre or otherwise) is forcing the mesh to conform to the specified divertor target shape and usually involve cells that are stretched significantly in one direction.

The ability to rotate and stretch the view allows the structure of the problem cells to be discerned and repaired. As an example, we consider a certain mesh generated for the NSTX experiment which contains twisted cells near the inner divertor, Fig. K.5. By zooming, stretching, and rotating the view using the tools described in Secs. K.5.3 and K.5.4, we arrive at a much clearer depiction of the problem, Fig. K.6. We see that the edges of the cells crossed the enforced boundary, twisting the mesh cells in the process. At this point, the user can straightforwardly use the `Move mesh point` tool to manually repair the cells.

K.9.4 Defining cells

With DG, you can define cells between the `Sonnet` grid and the structure (both objects must be present). You define cells by creating separators between them. You create separators with the `Edit|Create|Separators` menu

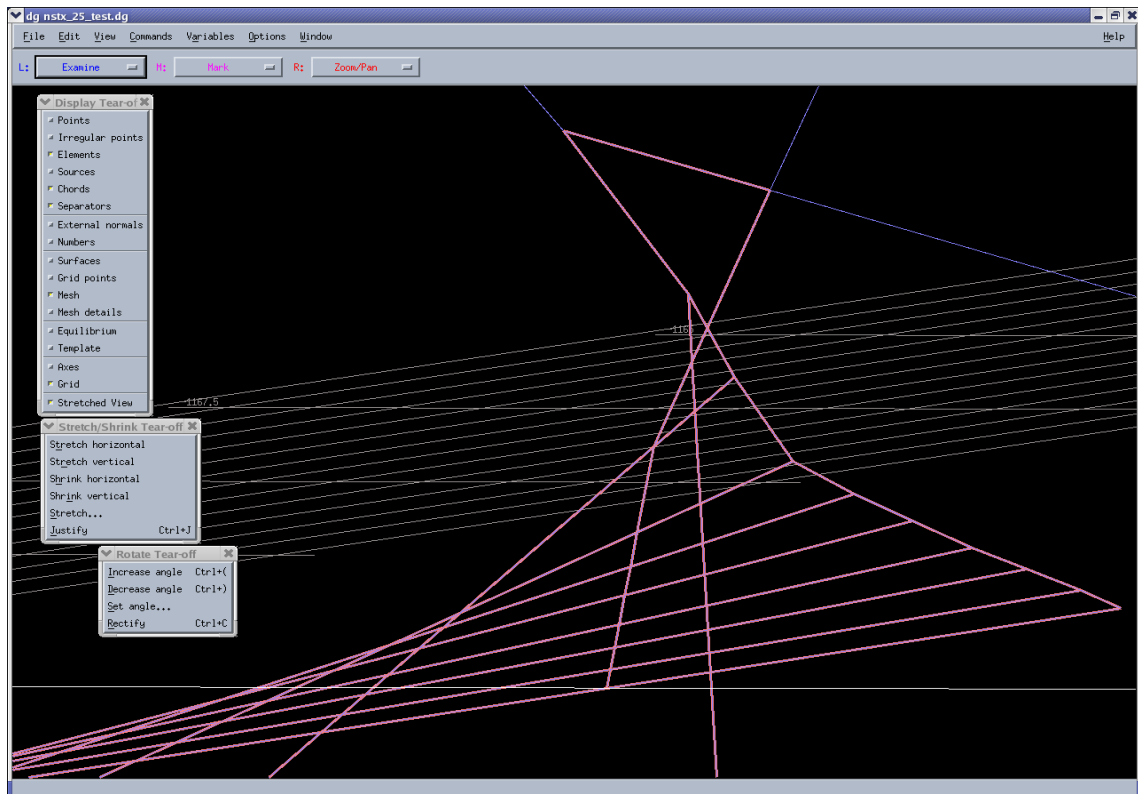


Figure K.6: After stretching and rotation, the overlapping of the cells in the upper part of the region cells depicted in Fig. K.5 becomes clear.

command. To adjust individual separators, use the `Reposition` tool. The `Edit|Delete|Separators` menu command removes all separators. You cannot add or remove individual separators with DG.

To work with separators, you must define a suitable structure and import a Sonnet grid.

Use the `Edit|Create|Separators` menu command to create separators between the structure and the Sonnet grid.

Use the `Reposition` tool to attach a separator to a different structure point. Select the separator and move the mouse cursor a structure point. Then release the mouse button.

Use the `Edit|Delete|Separators` menu command to delete all separators.

K.10 Working with variables

K.10.1 Basics

With DG, you can use `Variables` to specify information that cannot be edited visually in the work area. Variables are written into the output file and can be processed by other programs. Some variables (targets and structure) also have important functions in DG.

Variables must be set up by a power user before using them. This procedure is described in **Customizing DG** (section K.12). Each variable has following attributes:

Name is written into the output file and is otherwise invisible.

Description is the name that is shown to the user.

Type defines type of values the variable can hold.

Scope specifies what objects the variable applies for.

Export/No export determines whether values are written into the output file.

Default value is the default value for newly-created objects.

Help text is displayed to the user when s/he asks for it.

These attributes are normally defined by a power user while setting up variables. A normal user should have some idea of the type and the scope in order to be able to set correct values for correct objects.

Each variable has a type. A variable can hold either a text value (integer, real or text) or a set of elements (element, set of elements, target, structure, part of structure). DG prevents the user from entering values of a wrong type.

The *scope* determines objects the variable is attached to. There are 4 possible scopes:

Layers means that the variable has one value for the whole layer.

Elements means the variable has a separate value for each element.

Sources means the variable has a separate value for each source.

Chords means the variable has a separate value for each chord.

As noted above, variables are organized into *layers* with DG displaying each layer in a separate dialog box.

The power user defines *layer types* and the minimum and the maximum number of layers of each type.

K.10.2 Working with layers

Layers may have one or more instantiations in a particular model. For some layers, a single instance is all that is needed to specify the desired information; DG will not permit the creation of additional instances of this same layer. In the latter case, DG will assign an integer count as the instances are created. Some layer instances are predefined (e.g., *Structure*) and cannot be created or destroyed by the user, power or otherwise.

To create an instance of a specific layer, select the layer from the *Variables|Add* sub-menu. A dialog box for editing the variables in this layer instance will appear. If the maximum number of instances of a specific layer already exist, this layer will not be displayed in the *Add* menu. If this menu becomes empty, the *Variables|Add* cascade button will be grayed.

To destroy an instance of a layer, select its name in the *Variables|Remove* sub-menu. If there are multiple instances of a layer, there will be a sub-menu associated with that layer, listing the integer indices of those instances. If only the minimum number of instances of a specific layer already exist, layers of this type will not be displayed in this menu. If this menu becomes empty, the *Variables|Remove* cascade button will be grayed.

To display the dialog box for an existing layer instance, select its name and instance index (if applicable) under the *Variables* menu.

To hide the dialog box for a layer, press the *Close* button in that dialog box.

K.10.3 Setting variables

To set variables in a layer, you need to display the dialog box for that layer first.

The dialog box for each instance of a layer displays its variables, allowing them to be set, viewed, and modified. The title of this dialog box is the name of the layer; if there are multiple instances, the integer index of the particular index

is also displayed here. At the top of the dialog box, there is a row of buttons that affect the entire box. Below this are fields for the individual variables. Finally, there is a status line at the bottom of the dialog box.

The "△" button on top of the dialog box allows you to "roll it up" so that the box occupies less screen space. In order to restore the dialog box to its full size, press this button again.

The dialog box can be resized using the window manager. If it is reduced in size, scrollbars appear, allowing you to scroll the content. The functionality of the dialog box is NOT affected by resizing it or rolling it up. In particular, the `Set all` button also affects variables that are invisible.

For each variable, the dialog box contains its name, a `Help` button, a text field for entering the new value (for text-like variables) or a `Mark` button (for `set of objects` variables), and a `Set` button. The `Help` button, when pressed, displays help about the variable. The text field is used to enter new values for text variables. When the text field has keyboard focus, the status line displays the current value of the variable. For text variables that apply to elements, chords or sources, only variables that apply to marked objects are accessed. `Filename` variables differ from `Text` in that the user can select the file interactively (press the `Enter` key on your keyboard in the text field to bring up the file selection dialog); the variable is not valid unless the file exists and is readable at the time of the variable check.

The `Mark` button, available only for `Set of objects` variables, marks the objects contained in the variable. The `Set` button sets the variable to the new value. For text variables, the new value must be entered in the text field. For variables storing sets of objects, the objects must be marked.

The `Set all` button in the top row, when pressed, has exactly the same effect as pressing the `Set` button of each variable except that empty new values for variables that apply to elements, sources or chords are ignored. This button is only available if the layer consists of text variables only.

The `Reset all` button, when pressed, resets all text fields to present values of corresponding variables. It is only available in layers consisting of text variables only. The text field is cleared for variables that apply to elements, sources or chords when marked objects have different values.

The `Close` button closes the dialog box.

Pressing the right mouse button over a text field or a button associated with a variable brings up a pop-up menu. If the mouse has less than 3 buttons, the menu can be displayed by pressing "Ctrl+left" button. The `Reset` command sets the text field to the current value of the variable, similar to the `Reset all` button but affecting only one variable. For variables storing sets of objects, this command acts exactly like the `Mark` button. The `Display values` command displays values of variables that apply to elements, chords or sources next to corresponding objects. In order to remove these "labels", highlight any object or use the `View|Remove labels` command from the main menu. Commands in the `Compare` sub-menu compare values of the variable to the value entered in the edit field and mark matching objects. Both the `Display values` command and the `Compare` sub-menu are only available for variables that apply to elements, sources or chords.

To set a value for a variable that contains a set of elements, mark the desired elements and press its `>` button.

To set a value for a text variable that applies to the layer, enter the value in the `New Value` text field and press the `>` button.

To set a value for a text variable that applies to elements or sources, first mark elements or sources the value should be assigned to. To set a new value, enter it in the `New Value` text field and press the `>` button.

Current values for variables are always displayed in corresponding `Old Value` text boxes. For variables that apply to elements or sources, only values for marked objects are considered. If all values are the same, a value will be displayed, otherwise the word `Different`.

To copy the current value of a variable into its `New Value` text field, press its `<` button. For a variable containing a set of elements, the elements belonging to the variable will be marked.

The `Accept` button has exactly the same effect as pressing all `>` buttons except that empty new values are ignored.

The `Reset` button has exactly the same effect as pressing all `<` buttons.

The `Hold` toggle button controls how `New Value` fields of variables that apply to elements or sources are updated. With `Hold` disabled, these fields are updated every time objects are marked and unmarked. When it is enabled, no such updates happen.

You can easily copy values between different elements or sources. Disable `Hold` and mark objects containing the desired value. Then enable `Hold` and mark objects the values should be assigned to. Press `Accept` to set new values.

K.11 Creating output files

K.11.1 Types of output files

DG produces 3 output files:

Structure file contains the structure, with `.str` extension

Targets file contains targets, surfaces and grid points, with `.trg` extension

Output file contains elements, sources, separators, and the values of the variables associated with all of the layer instances for the current model, with `.dgo` extension

The current version of DG requires structure, targets, surfaces, and grid points to be defined. Also, all `integer` and `real` variables must be set to non-empty values.

K.11.2 Producing output files

DG generates names of output files by attaching different extensions to the name of the DG file. If the current file has no name yet, save it with the `File|Save` command before producing the output.

To produce the output files, choose the `File|Output` menu command. A dialog box will appear asking for confirmation. Press `Ok` to create the files.

K.12 Customizing DG

K.12.1 Installing DG

To install DG, you need to:

- Compile and link its sources
- Make sure that the help file exists and has the right name
- Make sure that the configuration file exists and has the right name

DG is an X Windows application. The correct `DISPLAY` setting is necessary.

Some useful infos on installing are contained in the **Customizing DG** section ([K.12](#)).

K.12.2 Files needed by DG

dg Executable file

dg.dgh Help file

dg.dgc Configuration file

K.12.3 Specifying directory masks

You can customize directory masks DG displays in file selection dialog boxes by setting the following environment variables:

DG.OPEN_MASK Directory mask for `File|Open` (defaults to the current directory)

DG.SAVE_MASK Directory mask for `File|Save` (defaults to the current directory)

DG.EQUIL_MASK Directory mask for `File|Equilibrium` (defaults to the current directory)

DG.TEMPLATE_MASK Directory mask for `File|Template` (defaults to the `templates/` subdirectory (if present) or to the current directory)

DG.SONNETMASK Directory mask for `File|Sonnet grid` (defaults to `${SOLPSTOP}/modules/DivGeo/device/${DEVICE}`)

K.12.4 The configuration file

There is a special DG file, the **configuration file**. Its filename is normally produced by appending `.dgc` to the name of the executable. The configuration file has two functions:

1. When the `File|New` command is executed, the configuration file is loaded into the new window.
2. When a DG file is being opened, the configuration file is used to update its definitions of variables and layers.

DG normally writes these definitions in DG files. When a DG file is opened, they are normally ignored and definitions from the configuration file are used instead. This ensures that all files loaded in DG have identical definitions of layers and variables.

Default values are not updated in the current version of DG.

Specifying the configuration file

Sometimes it is necessary to use a different configuration file or not to use any at all.

To use a different configuration file, start DG with the following command:

```
dg -cfg <filename-with-extension>
```

To use DG without loading a configuration file, type

```
dg -nocfg
```

NOTE: There are some bugs in DG that appear only when it is started without any configuration and sometimes cause an exception.

K.12.5 Creating or modifying the configuration file

Use the `Options|Setup` menu command to create/modify the configuration file.

The `Setup` dialog box and all dialog boxes you open during setup are modal. For this reason, you can work only with the topmost dialog box.

Choose the `Variables...` sub-menu in order to edit definitions of variables and layers in the current file. The `Layer types` dialog box will appear.

Press `Save as configuration` to save the current file as a configuration file. A dialog box will appear prompting for a filename. Normally, it already contains the right name. Select `Ok` to save the file in the current window as configuration.

Press `Close` to close the dialog box.

Adding, modifying and removing layer types

To define a layer type, choose `Add` in the `Layer types` dialog box. The new `Layer type options` dialog box will appear.

Enter the name of the layer type to create in the `Name` text field. The name should contain only alphanumeric characters. This name will be written into the output file.

Enter the description in the `Description` text field. In the `MinInstances/MaxInstances` text field, enter the minimum and the maximum number of layers of this type. You will have to use 0 for `MinInstances` because there are no layers of this type yet. You can change `MinInstances` later after creating the minimum number of layers.

Press the `Ok` button to create the layer type.

To remove a layer type, select it in the list and press `Delete`.

To modify a layer type, select it in the list and press `Modify`. The `Layer type options` dialog box will appear. You can change name, description, `Min/MaxInstances` here. Note that if you change the name, layers contained in `DG` files under the old name will be ignored.

To close the `Layer types` dialog box, press `Close`.

Setting up variables

To define the variables that will constitute a layer, first display its `Layer type options` dialog box (`Modify`). This dialog box contains a rectangular area with buttons. These buttons represent row/column positions of variables in dialog boxes. Buttons containing the string `Empty` mean that no variable is defined at this position.

To create a variable at a specific position, press the `Empty` button at that position. The `New variable` dialog box will appear.

Enter the name of the variable to be created in the `Name` field. The name must contain only alphanumeric character.

Select the type and the scope in corresponding option menus. Check the `Do not export` toggle button to prevent the variable from being written into the output file. Structure and targets are normally not exported because they are written into separate files.

Press `Ok` to create the variable.

To modify a variable definition, press its button in the `Layer type options` dialog box. The `Variable options` dialog box will appear. After making changes, press `Ok`. Note that if you change the name, variables contained in `DG` files under the old name will be ignored.

To remove a variable, press its button in the `Layer type options` dialog box. The `Variable options` dialog box will appear. Press `Remove` to remove the variable.

Adding help text to variables

`DG` lets you add help text to variables. The user can display the help text by pressing the `Help` button located near the name of the variable in the `Variables` dialog box.

To edit a help text for a variable, display its `Variable options` dialog box and press `Edit help`. The `Help for variable editor` dialog box with a text editing field will appear. Edit the help text in the editing field and press `Ok`.

Setting default values

DG lets you define default values for variables. When a new layer, element or source is created, the default values are used for the newly created object.

It is suggested that you define default values for all variables because DG will not generate output files when some variables have empty values. A search for empty values can be very difficult and cost a lot of time.

To define default values for variables in a layer type, display its `Layer type options` dialog box and choose the variable for which you wish to set a default. In its `Variables :` dialog box there is a `Default value` field for this purpose.

Default values from the configuration file are used only to create new DG files. When opening an existing file, default values from the configuration are ignored.

K.13 File formats

K.13.1 Simple template file format

Besides the `Neutral File` exported from PATRAN, DG can also import templates in a very simple format where the geometry is specified as a set of polygons (in general, open). In this case, each line of the input file either contains a pair of the co-ordinates of the next corner, or is empty.

K.13.2 Structure file format

The first line in the structure file contains the number of closed chains in the structure.

Then, for each chain, starting with the outermost, the following block appears:

```
<number of points in the chain>
<x1> , <y1>
...
```

For the outermost chain, the bounding rectangle is added.

Points are output in counterclockwise order when the external normals of the elements in the chain point outwards. Otherwise, they are output in clockwise order.

K.13.3 Targets file format

The targets file (for a single-null model) has the following format:

```
target 1
line
  <x> , <y>
  ...
target 2
line
  <x> , <y>
```

```

...
region 1
levels
  <level>
...
region 2
levels
  <level>
...
region 3
levels
  <level>
...
zone 0
points
  <value>
...
zone 1
points
  <value>
...
zone 2
points
  <value>
...
<eof>

```

The `target1` and `target2` blocks contain point coordinates for targets.

The `region 1 . . . 3` blocks contain levels of surfaces. Regions have following meaning:

region 1 Main Plasma

region 2 SOL

region 3 Private Flux

The `zone 1 . . . 3` blocks contain values of grid points (between 0 and 1). Zones have following meaning:

zone 1 Main chamber

zone 2 Inner divertor

zone 3 Outer divertor

K.13.4 Output file format

The output file has the following format:

```

p1
  <x> , <y> , <z>
...
p2
  <x> , <y> , <z>
misselem
  <number>

```

```

...
sprtrs
  <number>
...
sources
  <x> , <y>
cells <cellnum>
  <number>
...
...
<layer name> <layer number>
  <variable>
...
...
finish
userdata

```

p1 and p2 sections contain coordinates of origins and ends of elements and/or separators, sorted after element number. For unused numbers, dummy coordinates are output. The *Z* coordinate is always 0.

The `misselem` section contains unused element numbers. Elements with these numbers **must** be ignored.

The `sprtrs` section lists numbers of separators in p1 and p2 arrays.

The `sources` section lists sources.

If separators are defined, there is a `cells` section for each cell. Each section lists numbers of elements and separators included in its cell.

For each layer, a `<layer name> <layer number>` section is present. Layer name is the name of the layer type. Layer number is used to distinguish between layers of the same type. Each section contains a `<variable>` entry for each variable in the layer.

The `<variable>` block has following formats:

- Text variables that apply to the layer:

```

<variable name> <value>

```

- Text variables that apply to elements or sources:

```

<variable name>
  <value>
...

```

- For variables that apply to elements, a value for each element number is included. Dummy values are used for unused numbers. For variables that apply to sources, a value for each source is included. Values are written in the same order as sources in the `sources` section.

- "Set of elements" variables that apply to the layer:

```

<variable name>
  <element number>
...

```

Appendix L

Some information from the SOLPS 2002 Course Notes

The following information is extracted from the SOLPS Course Notes given at JET in 2002.

L.1 Breakdown of B2.5 programs

B2.5 consists of a number of programs

b2ag prepares the geometry and magnetic field (see 3.1)

b2ah prepares the default physics parameters (see 3.2)

b2ai prepares the initial plasma state (see 3.4)

b2ar prepares a table of atomic rate coefficients (see 3.3)

b2co one option when running the main program is to write out a short restart file periodically; this program converts such a restart file into the normal version (see 3.15)

b2fu the opposite of **b2uf**; it converts a formatted plasma file `b2fplasmf` to the normal unformatted `b2fplasma`

b2md saves the SOLPS 5.x output to the MDSPplus server (see 6)

b2mn the main program (see 3.5)

b2pl the plot program (see 3.6 and Appendix I)

b2rd retrieves the SOLPS 5.x output from the MDSPplus server (see 6)

b2ts an example post-processing program; it reads the basic output files produced by **b2mn** and prints some simple geometric quantities

b2uf the normal `b2fplasma` file (used for plotting) is written using FORTRAN unformatted writes; this program reads such a file and writes it out as a formatted file, `b2fplasmf`. Typically used for moving the `b2fplasma` to a different computer architecture.

b2xd was used in the transition stage from SOLPS 4.x to SOLPS 5.x — it converts from the SOLPS 5.x plasma state file to that used by SOLPS 4.x

b2ydm post processes the heat load profiles on the dome and evaluates outlet coolant temperature (see 3.20)

b2ye computes an error assessment of a coupled B2.5-Eirene run

b2yg plots the geometry and magnetic field

b2yh prints the table of physics parameters

b2yi plots the plasma state

b2yi_gnuplot a version of **b2yi** using `gnuplot` instead of NCAR

b2ym claims to produce a movie

b2ymb post processes the heat load profiles on the targets and creates input files for the monoblock heat transfer model (see 3.18)

b2yn plots the progress of the inner B2 .5 iterations

b2yp displays the plasma state

b2yq claims to provide a quick display of the evolution data

b2yr plots the atomic rate coefficients

b2yrp post processes the heat load profiles on the reflector plates and creates input files for the RPs heat transfer model (see 3.19)

b2yt transfers the plasma from one mesh to another mesh of a different size (see 3.14)

b2_ual.write writes out the plasma state in IDS format (see 3.21)

L.2 B2.5 ".dat" and ".prt" files

filename	formatted?	read by
b2ag.dat	formatted	b2ag b2ai
b2ah.dat	formatted	b2ah
b2ai.dat	formatted	b2ai
b2ar.dat	formatted	b2ar
b2md.dat	formatted	b2md
b2mn.dat	formatted	b2md b2mn b2pl b2yi b2yi_gnuplot b2yn b2yp b2yt b2_ual_write
b2ydm.dat	formatted	b2ydm
b2yg.dat	formatted	b2yg
b2yh.dat	formatted	b2yh
b2yi.dat	formatted	b2yi
b2ym.dat	formatted	b2ym
b2ymb.dat	formatted	b2ymb
b2yn.dat	formatted	b2yn
b2yp.dat	formatted	b2yp
b2yq.dat	formatted	b2yq
b2yr.dat	formatted	b2yr
b2yrp.dat	formatted	b2yrp
b2yt.dat	formatted	b2yt
stra.dat	formatted	b2ar
weis.dat	formatted	b2ar

filename	formatted?	written by
b2ag.prt	formatted	b2ag
b2ah.prt	formatted	b2ah
b2ai.prt	formatted	b2ai
b2ar.prt	formatted	b2ar
b2co.prt	formatted	b2co
b2fu.prt	formatted	b2fu
b2md.prt	formatted	b2md
b2mn.prt	formatted	b2mn
b2rd.prt	formatted	b2rd
b2uf.prt	formatted	b2uf
b2ydm.prt	formatted	b2ydm
b2ye.prt	formatted	b2ye
b2yg.prt	formatted	b2yg
b2yh.prt	formatted	b2yh
b2yi.prt	formatted	b2yi
b2ym.prt	formatted	b2ym
b2ymb.prt	formatted	b2ymb
b2yn.prt	formatted	b2yn
b2yp.prt	formatted	b2yp
b2yq.prt	formatted	b2yq
b2yr.prt	formatted	b2yr
b2yrp.prt	formatted	b2yrp
b2yt.prt	formatted	b2yt

L.3 B2.5 communication files

filename	formatted?	written by	read by	contents
b2fgmtry	un*formatted	b2ag	b2ah b2ai b2co b2fu b2md b2mn b2pl b2ts b2uf b2xd b2yg b2yh b2yp b2yt b2_ual_write	geometry
b2fmovie	un*formatted	b2mn	b2ym	movie of plasma evolution
b2fparam	un*formatted	b2mn	b2md b2fu b2pl b2ts b2uf b2xd b2yh b2yp b2_ual_write	run parameters used for plotting
b2fpardf	un*formatted	b2ah	b2co b2mn b2yh b2yt	transport parameters
b2fplasma	unformatted	b2fu b2mn	b2md b2pl b2ts b2uf b2xd b2_ual_write	plasma state used for plotting
b2fplasmf	formatted	b2uf	b2fu	plasma state used for plotting
b2frates	un*formatted	b2ar	b2md b2mn b2pl b2yn b2yp b2yr b2yt b2_ual_write	atomic physics rates
b2fstate	un*formatted	b2co b2mn b2rd	b2fu b2md b2pl b2ts b2uf b2xd b2yi b2yi_gnuplot b2yp b2yr b2yt b2_ual_write	final plasma state
b2fstati	un*formatted	b2ai	b2co b2mn b2yi b2yi_gnuplot	initial plasma state
b2fstatt	un*formatted	b2yt		plasma state on a new grid
b2fgmtrt	un*formatted	b2rd		recovered geometry file
b2ftrace	un*formatted	b2mn	b2yq	residuals
b2ftrack	un*formatted	b2mn	b2yn	residuals
b2specp	formatted	b2yp		sources
b2faveri	formatted		b2mn	initial averaged plasma solution
b2favere	formatted	b2mn	b2ye	final averaged plasma solution
tran	formatted	b2mn		plasma state used for archiving at JET

Appendix M

Subversion repository logs

M.1 SOLPS-ITER

2558728 Caught a few missing switches from generic list of species to list of actually used nuclear sequences in Uinp when preparing bundled input files
5a3eb35 Further fixes when Uinp adds elastic collisions but no AMDS file is used
ca122fe work around a problem with broken shell initialization scripts
faf44e5 Pointing to Eirene submodule with safety for missing fort.13 file
1f43536 – Corrected mtv_3d scripts – Pointing to B2.5 submodule with new summary IDS quantities – Pointing to Eirene submodule with clean-up distribution table print-out
3bf2b37 Fixes to Triang compilation if GR library is missing
271aedd Pointing to Eirene submodule with conditioned non-standard Fortran usage
cf09bc4 – Adding automatic H/H2/He elastic collisions in Uinp – Pointing to B2.5 submodule with correction from IMAS-3195 – Pointing to DivGeo submodule with correction from IMAS-3202 – Pointing to Eirene submodule with MPI broadcast of variables moved to their respective modules – Consequences of Eirene changes on Triang dependencies required a rethinking of Triang compilation
5fd6a93 Updating 98-species example and documentation for 3.0.7 master version release
90a8942 Final complete installation of graphical libraries at JFRS (by Shohei Yamoto)
1a58bb9 Fixes for EireneX script examples
e18d355 Corrected link instruction for MSCL on Cray JFRS compiler (provided by Shohei Yamoto)
2614206 Adding a default AMDS Cray configuration file
2a969c5 Fixes from JFRS Cray compilation and permissions
6d3f14a Removing use of setup.csh in SLURM submission scripts and adding ITER baseline equilibrium IDS to converted cases
54ef336 Completing compilation at JFRS for gfortran and Intel compilers (thanks to Shohei Yamoto)
32973cb Improved compilation with graphics at JFRS
c0dd85a Improvements for central installations
3252088 Upgrading capability of rescaling density for isonuclear sequences
37a60d9 Minor Makefile clean-up
977990c Removing nc2text from multi-threaded compilation and compiling a single copy with additional links
173d18d Merge branch 'release/3.0.7' of /project/MISONIC/solps-iter into release/3.0.7
c1087e7 Second pass at attempting to link to NCAR libraries and changing JFRS default compiler to gfortran
2ab2b5f More updated examples and added angular momentum tracing data
663d121 MDSPLUS support and partial NCAR support at JFRS
1c83166 Pointing to B2.5 with safeties for obsolete b2time.nc files and Eirene coupled cases with no atoms, molecules, or test ions
41c5d76 Added MPI clean.triang instructions
e9a4e20 Fixing an inconsistency in the Makefile (spotted by Jeremy Lore)
e284d24 Minor script mods and additions
443955e NERSC configuration files from Jeremy Lore and improved Eirene tally algebra from Petra Boerner
6c55083 Pointing to B2.5 submodule with drift_style=2 treatment for visq current
1a06794 Improved treatment for NetCDF4 files

a9a8853 Pointing to B2.5 submodule with new drift_style.eq.2 treatment of currents and diamagnetic terms, updated documentation
f72d6f0 Correction to print_versions script following change of Uinp GIT hash include file
ceafd47 One more updated example and corrected backward compatibility for OnedBsqc array in B2.5
b774270 Merge branch 'release/3.0.7' of ssh://git.iter.org/bnd/solps-iter into release/3.0.7
5a0b508 Cleaned up treatment and documentation of drift switches (following discussions with Ilya Senichenkov and Serguey Voskoboinikov)
91c1d11 Removed underflow data in AUG D+C+He benchmark example and manual corrections provided by Jeremy Lore
f1d3299 Bug fixes for ITM build of 3.0.7
64f3cd9 update modules; add amns module
494a5fe Pointing to B2.5 submodule with bug fixes and additional safeties for linear geometries and some updated examples
71c70e7 Pointing to B2.5 submodule with Makefile typo correction
79c0f25 Dealing with long Uinp GIT hashes and updating some more examples
37b93f4 Merge branch 'develop' into release/3.0.7
fb578dd Fixing a bug when building LISTOBJ files (spotted by Stefano Carli)
ac8094d Fixing a bug when building LISTOBJ files (spotted by Stefano Carli)
17d9ed1 Better tests of GLLHOME
00e1352 Adding compilation warnings when graphical library definitions are missing
40a6509 Pointing to corrected Eirene submodule
8elf8b7 Adjustments for missing GKS library
ac76d6d Pointing to B2.5 submodule with corrected Makefile
b9d597a Added safeties when the Motif or GR libraries are not present
9dflc1b Improvements to compilation when NCAR library is missing
171eeb9 Cleaned up examples and removing Eirene Makefile targets for stand-alone runs
89fdd65 Two more 3.0.7 examples and pointing to Eirene submodule with bug fixes from FZJ
e1e4ca3 Bug fixes from gfortran debugger
3e3f9f3 Bug fixes from gfortran debugger
1737ac7 Pointing to B2.5 submodule with corrected location of separatrix in fy
56cd487 Pointing to B2.5 submodule with corrected separatrix position in fy.F
deel38a Fix units for FFBZ function
642f58e Bug fixes and safeties added from IDS write-up of SPB 3.0.7 narrow SOL runs
ccfcfa5 Correcting environment variables for GA (provided by Alberto Gallo)
fd7a62d Removing obsolete IPPITM hostname
1a0cd2b bring reality into congruence with the documentation
c2d3bcb documentation for the new marconi submit script
04958b4 batch scripts for Marconi
bd6ee6e Correcting environment variables for GA (provided by Alverto Gallo)
c85ea44 Removing obsolete IPPITM hostname
d798afb bring reality into congruence with the documentation
14cab64 documentation for the new marconi submit script
4bc12fe batch scripts for Marconi
1561c2d Removing obsolescent -vec.report0 compilation flag for MARCONI Intel compiler
58df192 updates for Marconi
02d6380 Removing obsolescent -vec.report0 compilation flag for MARCONI Intel compiler
3f99907 Correct the information about which branch is being checked out
fb5c380 updates for Marconi
677b523 Updated 3.0.7 Release Notes and handling of 'whereami' script following User Forum discussions, pointing to DivGeo submodule with bug fixes for centreline vessel units
499aeb4 Third pass at improving nc2text rule
cf2609e Third time is a charm: Better rule for nc2text linking not being repeated every time
9fcec4f Fix to nc2text.simple to handle very large NetCDF files
1e584ac Fix to nc2text.simple to handle very large NetCDF files
046d823 Better rule for nc2text utility build
cf18fed Some updated examples, improved documentation, better rule for nc2text usage, corrected a conversion rule for when both core and gas puff feedbacks are simulatenously active
b7aeb52 add a note to the manual about the mapping from cell-faces to cell-centres
a0f2552 add a note to the manual about the mapping from cell-faces to cell-centres
33f9dc2 Clarification on usage of 'gmake clean'
c0558a8 Pointing to B2.5 submodule with added OpenMP run diagnostics
b3bbd04 Updated submodules with improved nc2text build rule
3133011 Added caution about old n2ctext utility in Release Notes
15e328e Cray compilation fixes for OpenMP runs
3702393 Merge branch 'release/3.0.7' of ssh://git.iter.org/bnd/solps-iter into release/3.0.7
9e3176d Adding Release Notes document for version 3.0.7
7f8348f Merge branch 'release/3.0.7' into feature/OpenMP

6f8b9b1 Merge branch 'release/3.0.7' into feature/consistent_energy_flux_3.0.7
5985421 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
926a4af Merge branch 'feature/fix_recomb+redef_gmtry' into feature/consistent_energy_flux_3.0.7
56d101c Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
c711975 Clean-up of config file to help with compilation on USTC domain
c487dab Updated references, moved to IMAS/3.27.0-4.7.1 at ITER, pointing to Eirene and B2.5 submodules with latest small corrections
d375337 First attempt at 8-th pass of FZJ updates: 'embarrassingly parallel' scheme and use of NPRL switch
ff3ad8a Further tweaks for compilation speed-up capability
9c75f94 JFRS IFERC-CSC installation with compilation speed-up, updated documentation, and configuration files clean-up
796d12f Including an energy flux calculation into Appendix G
1a27d55 Merge branch 'feature/fix_recomb+redef_gmtry' into feature/consistent_energy_flux_3.0.7
ccc04a2 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
aa2f442 Pointing to Eirene submodule with sixth old gfortran compilation fix
3d7ef0c Pointing to Eirene submodule with next old gfortran compilation fix
0417360 Pointing to AMNS/1.3.3 at ITER and Eirene submodule with more compilation fixes
bfb3c4 Pointing to Eirene submodule with third old gfortran compilation fix
c819369 Pointing to Eirene submodule with next old gfortran compilation fix
edfb5b0 Updated installation instructions and compilation fix for old gfortran in Eirene
0c87518 Merge branch 'feature/fix_recomb+redef_gmtry' into feature/consistent_energy_flux_3.0.7
429db80 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
d511a89 Pointing to updated B2.5, Eirene, Carre, and DivGeo submodules
ab296a4 Pointing to updated B2.5, Carre and DivGeo submodules
9d98fee Pointing to B2.5 submodule with bug fixes and code clean-up
648e816 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
e8ef494 Bringing in AMNS support
c003bb0 Clean-up of config file to help with compilation on USTC domain
b79c704 Provision for older GIT versions
f439c02 Updated documentation (correction provided by Sergei Makarov and Serguey Voskoboinikov)
e848d26 Updated documentation (correction provided by Sergei Makarov and Serguey Voskoboinikov)
a611a3e Fixes for DN and fluid neutral diagnostics (reported by Sarah Newton)
e8c951d Fixes for DN and fluid neutral diagnostics (reported by Sarah Newton)
7498024 Pointing to updated Eirene module with KUL config files for release 3.0.7 version.
64bed7a Pointing to B2.5 submodule with safeguard for stand-alone compilation
14eb277 Adding an ADAS git hash for traceability
21c6ee7 Pointing to ADAS submodule with corrected ADAS96 Ne ionization data
55081a2 Pointing to ADAS module with corrected ADAS96 Ne ionization data
2822085 New b2yt capability of flipping poloidal grid direction when mapping between USN and LSN topologies and other updated conversion rules, clean-up of Makefile options
ca5a17f Merge branch 'feature/fix_recomb+redef_gmtry' into feature/consistent_energy_flux_3.0.7
94df55b Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
500f0ca Updating whereami-script for new KUL cluster
cb87cea Further refinements to the plot script documentation
246c171 Pointing to updated B2.5 module
4bc0327 add a '-h' option to plot; I might have missed some features or got some wrong
0da5bee providing the documentation requested by Xavier as well as a bonus
cdbc7fd routines for calculating what the sputtering would have been based on a run saved in MDSplus on solps-mdsplus.aug.ipp.mpg.de
a2c5cb1 providing the documentation requested by Xavier as well as a bonus
3edcfd8 Updating whereami-script for new KUL cluster
89f5c7f routines for calculating what the sputtering would have been based on a run saved in MDSplus on solps-mdsplus.aug.ipp.mpg.de
ade4cb9 Dealing with IMAS/3.28.0-4.7.2 at ITER and new wall IDS features
367b7c2 Relying on the native MPI support for PGF90 compiler and removing superfluous MPI code from Uinp programs
9bd2914 Merge branch 'release/3.0.7' into feature/OpenMP
7b55666 Relying on the native MPI support for PGF90 compiler and removing superfluous MPI code from Uinp programs
e462696 Merge branch 'release/3.0.7' into feature/OpenMP
d5fe343 Correcting GGD library address for gfortran MPI compiler
717b5e7 A little more credit and second version of DIV_chords figure
f1e20e3 Updating the Div.chords figure in the manual, and correction spotted by Jeremy Lore
3a9a25c A little more credit and second version of DIV_chords figure
9d03246 Updating the Div.chords figure in the manual, and correction spotted by Jeremy Lore
a45042c Merge branch 'release/3.0.7' into feature/OpenMP

433367b Pointing to B2.5 and DivGeo submodules with stand-alone compilation fixes
 5a33eea Merge branch 'release/3.0.7' into g2dpc/feature/fix-patch.py
 6bc44ac fix patch.py to take into account changes in matplotlib
 877f08b Pointing to updated DivGeo submodule with streamlined configuration files for IMAS backward compatibility
 2baa313 - Streamlined configuration files and code fixes for IMAS backward compatibility - Use of Version_GGD instead EBVERSIONGDD throughout - Updated JET setup files as per release/3.0.7 branch - Added a safety warning to resave_mds
 0362a7c Changing Intel NetCDF library on Gateway to version 4
 94aec39 Added printout of actual running average analysis in res_av script
 066a5c7 Added funding acknowledgment in EIRENE averaging manual
 a06fdef Adapting text to indicate scalar support for nc2text_simple
 fbc758b Fixes to nc2text_simple to handle 0D data and match old nc2text executable
 33f9846 Changes in script files after introducing b2batch.nc file to store batch averages; small change in manual for Eirene averaging schemes
 a261296 Pointing to the B2.5 module with NetCDF4 usage and updated documentation
 cfb128e change to using netcdf4
 d69f1f7 - Streamlined configuration files and code fixes for IMAS backward compatibility - Use of Version_GGD instead EBVERSIONGDD throughout - Updated JET setup files as per release/3.0.7 branch - Added a safety warning to resave_mds
 d3b81f4 Merge branch 'release/3.0.7' into feature/dpc-fixes-imas-updates-for-JET
 f8eff77 update IMAS configuration for JET
 1202620 Changing Intel NetCDF library on Gateway to version 4
 e31df12 Pointing to B2.5 submodule with corrected opening of b2batch.nc file
 f965ced Adding '-r' option to (re)save_mds script to pass variable redefinitions to Makefile.plot through the script
 7ecadf7 Added printout of actual running average analysis in res_av script
 6f455a4 Merge branch 'release/3.0.7' into feature/OpenMP
 4358cfb Merge branch 'release/3.0.7' into feature/LSNvsUSN.labelling
 88b778a Merge branch 'develop' into feature/NetCDF4+batch_averaging
 152d13c Merge branch 'release/3.0.7' into feature/OpenMP
 a80f893 Pointing to B2.5 submodule with IMAS bug fix for older versions
 a0e0ea2 Merge branch 'release/3.0.7' into feature/OpenMP
 88e997a Second pass at configuration files clean-up and adjusting compilation flags for ASIPP case
 2b8b4e4 Clean-up of configuration files for the case when the MPI compiler is different from the serial one
 48a0460 Pointing to updated Carre and DivGeo submodules
 84fbb95 Accompanying changes to IPP SOLPS-ITER setup files update
 28e5693 Merge branch 'develop' into feature/develop-USN.labels
 53407ec update doxygen version; use imas and ggd
 a24d10f update doxygen version
 10ee598 Added funding acknowledgment in EIRENE averaging manual
 c3c9bfc Adapting text to indicate scalar support for nc2text_simple
 966ccc4 Fixes to nc2text_simple to handle 0D data and match old nc2text executable
 0594dd6 Changes in script files after introducing b2batch.nc file to store batch averages; small change in manual for Eirene averaging schemes
 8e70dd0 Pointing to the B2.5 module with NetCDF4 usage and updated documentation
 76804a0 change to using netcdf4
 a07c8c5 fix for imas missing access to stdc++
 f098f73 Second pass at configuration files clean-up and adjusting compilation flags for ASIPP case
 fd3146a Second pass at configuration files clean-up and adjusting compilation flags for ASIPP case
 d9c349b Clean-up of configuration files for the case when the MPI compiler is different from the serial one
 2c4270e Clean-up of configuration files for the case when the MPI compiler is different from the serial one
 7471b62 Pointing to B2.5 and Eirene submodules with compilation fixes for older gfortran versions
 1bd00da Pointing to B2.5 submodule with OpenMP bug fix for writing fort.31 (provided by Stefano Carli)
 a0d307f Merge branch 'release/3.0.7' into feature/LSNvsUSN.labelling
 dc51b7c Pointing to B2.5 submodule with SAVEing b2time.nc filename to fix OpenMP runs
 43cf54e Merge branch 'develop' into feature/develop-USN.labels
 17265df Pointing to B2.5 submodule with SAVEing of b2time.nc filename for OpenMP runs
 95b0977 Pointing to updated submodules B2.5 with b2time.nc filename bug fix, Eirene, and Carre with comments clean-up

02532ed Second pass at modifications and documentation for USN topologies
 925701a Second pass at modifications and documentation for USN topologies
 2b06fdd First pass at clarifying labels and comments for LSN vs USN target locations, added Slack forum in documentation
 5290d1c First pass at clarifying labels and comments for LSN vs USN target locations, added Slack forum in documentation
 22210eb Merge branch 'release/3.0.7' into feature/OpenMP
 6a6c7ed Adding a missing allocation for description IDS workflow variable (spotted by Jae-Sun Park)
 33d124c Updated ASIPP config file (provided by Hang Si)
 16f54ae Merge branch 'release/3.0.7' into feature/OpenMP
 7ea202a Updated ASIPP config file (provided by Hang Si)
 a66d452 Pointing to B2.5 submodule with added OpenMP validated loops following tests by Stefano Carli
 c4efb18 Merge branch 'release/3.0.7' into feature/OpenMP
 ee53758 Further correction to documentation
 6e05be2 Makefile clean-up from OpenMP branch
 8252918 Merge branch 'release/3.0.7' into feature/OpenMP
 d75b9b9 Uinp bug fix for BGK species numbering in presence of bundled charge states
 12237af – Fixes for bundled charge states in Uinp – Continued change to "database" IMAS argument name – Corrected "bolo.cmd" and "bolo.dbg" scripts
 904571a Pointing to B2.5 with particle balance tracing bug fix from Ilya Senichenkov
 4a5f1bf Pointing to B2.5 submodule with updated b2mod_elements functionality
 97de71d Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 1b2bec5 Adding links to new needed routines for b2sxdx converter
 9dc4e20 Adding bundling scheme information in Eirene input files from Uinp case build-up
 558afc0 Adjustments for IMAS/3.27.0–4.6.0
 c845f7e Pointing to Eirene submodule with diagnostic format correction
 477b698 Fixed Uinp ordering of chords according to their label in Eirene input file
 f679047 Changing IMAS argument name from device to database
 7e699bf Small corrections to equations in manual
 6406e9f Pointing to B2.5 submodule with expanded copy of Eirene data to B2.5 data when using eirc command in b2plot
 74a2ac4 Pointing to B2.5 submodule with trajectory plots bug fixes and updated equations document
 7dd59a0 Pointing to B2.5 submodule with expanded copy of Eirene data to B2.5 data when using eirc command in b2plot
 fc9b1d0 Pointing to B2.5 submodule with trajectory plots bug fixes and updated equations document
 4a38f28 Using ub2p program for better write-up of chords files for b2plot
 b696eba Using ub2p program for better write-up of chords files for b2plot
 8db8f23 Using ub2p program for better write-up of chords files for b2plot
 95a4a49 Pointing to Eirene submodule with removed non-standard MOLD allocation feature
 4b9d6bc Pointing to Eirene submodule with removed non-standard MOLD allocation feature
 76379a8 Pointing to Eirene submodule replacing fatal error in timer with simple particle trajectory kill
 81f3b59 Pointing to Eirene submodule with fix to avoid fatal error in timer, replaced with killed trajectories
 984da57 Pointing to Eirene submodule with fix to avoid fatal error in timer, replaced with killed trajectories
 7d783d8 Pointing to B2.5 submodule with renamed qz b2wdat output files and clean-up of chord data variables in Uinp
 ae282a5 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 877ad94 New capability of Uinp to output chord data for both Eirene and b2plot use, and pointing to B2.5 submodule with corrected qz b2wdat output files
 373bc1d New command-line superseding options available for b2_ual_write
 d2b4cba Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 183fe88 Pointing to Eirene and B2.5 with support of ADAS bundled data files and minor bug fixes
 5e6c734 Pointing to B2.5 and Eirene submodules with bug fixes for IDS write-up and stand-alone runs
 fc055af Pointing to B2.5 and DivGeo submodule with UAL status error handling (IMAS–2859)
 1d8a85e Pointing to DivGeo submodule with ids2dg modified to be able to translate equilibrium and wall IDSs separately
 0e4d4f4 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 52e6398 Moving to IMAS/3.26.0–4.4.0 and pointing to DivGeo submodule with new dg2ids and ids2dg utility programs and numerics IDS write-out in B2.5 submodule prgram b2_ual_write
 07d3c58 Corrected linked libraries for ITM Intel compilation with IMAS
 07060b6 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 062ea32 Brexit

629ebc0 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
1d9c82d Pointing to B2.5 submodule with added eirene_mc_init allocation safeties
3710fcc Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
0bb78b0 Removing obsolete imasdb calls
d5266e0 Brexit
2c271a4 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
7a48b5c Updating B2.5 and Eirene submodules after rebase
27350cd Adding Makefile objects for pure MPI or pure OpenMP compilation
37a476d Removing obsolete STATIS_BGK variables in Eirene and pointing to updated ADAS submodule
22fc8a9 Removing statis_cop variables from Eirene, adding Planckian col-rad model to Uinp, adding AMNS module at ITER
190334a Removing statis_cop variables from Eirene, adding Planckian col-rad model to Uinp, adding AMNS module at ITER
6a2ecd5 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
dbc8668 Pointing to B2.5 submodule with new potential solver method and corrected H 89 PRC ADAS module
09f9432 Corrected also ITM submit files for OpenMP
da5c69a Updated SLURM submit file for correct OpenMP job submission
d29fc11 Pointing to Eirene submodule with corrected fort.15 write-up
71f79ea Pointing to updated ADAS submodule with corrected H 89 PRC data
b8f295c Pointing to B2.5 submodule with ion-neutral conductivity bug fix provided by Sergey Voskoboynikov
cc400b6 Pointing to B2.5 submodule with ion-neutral conductivity bug fix from Serguey Voskoboynikov
b10a86b Pointing to B2.5 submodule with ion-neutral conductivity bug fix and updated ADAS submodule
8c9176a Correcting b2.neutrals.parameters.stencil setting for species_end for time stratum (error spotted by Bob Wilcox) and pointing to updated ADAS module
de27358 Correcting b2.neutrals.parameters.stencil setting for species_end for time stratum (error spotted by Bob Wilcox) and pointing to updated ADAS module
adb4c34 Pointing to B2.5 submodule with corrected output for neutral average energy in b2pwwd.F
e96f03a Pointing to B2.5 submodule with corrected output of average energy for neutral particles in b2pwwd.F
6aa6b68 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
f87c2a7 Fixes for MPI job submission on ITM Gateway
314e4f5 Fixes for MPI job submission on ITM Gateway
7c7114b Pointing to B2.5 and Eirene submodules with updated calc_atomic_data code and corrected usage for correlated sampling with block 14 recycling strata
f0f437e Pointing to B2.5 submodule with further corrections to particle balance from transparent surfaces
1d90c17 Pointing to B2.5 submodule with removal of transparent surfaces in particle balance calculation
8dfc7a2 Pointing to B2.5 submodule with fix for fluxes through transparent surfaces
21047d0 Pointing to updated ADAS module with fuller adf15 file list
cd23b48 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
cbf48fc Updating ITER IMAS modules to UAL/4.4.0 and pointing to Eirene and B2.5 with bug fixes, Arrhenius rates, and new treatment for plasma background Eirene input cards
eaaac92 Pointing to B2.5 with corrected peak target flux calculation for B2.5 standalone runs
d579262 Pointing to B2.5 submodule with added particle source stabilization in guard cells
b87elee Pointing to Eirene submodule with moved call to dealloc_timea
1ecbf33 Pointing to Eirene submodule with various bug fixes from Petra Boerner at FZJ
f5de298 Pointing to Eirene submodule with profn bug fix to avoid division by zero
c560d94 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
42165c9 Pointing to Eirene submodule with added missing alternate return replacements
f1f7378 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
17bb73d Pointing to Eirene submodule with new Arrhenius reaction data format and additional changes from FZJ develop branch
3c5671f Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
4044142 Pointing to B2.5 submodule with merge fix
5e1686e Updating ITM gfortran module to IMAS/3.25.0/gcc/7.3.0
a659dc0 New ITM modules compiled with GCC/7.3.0
6aa2362 Merge branch 'release/3.0.7' into FZJ_attempt.8
d2b7c9a Eighth pass at importing FZJ Eirene changes, moving NLTRIMESH to block 14 and removing obsolete debugging switches
c8e6237 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
30dc46b update the documentation about the nodes in the SOLPS MDSplus tree
21944ea update the documentation about the nodes in the SOLPS MDSplus tree

ad780a3 First part of seventh pass at importing FZJ update changes: Move of NLTRIMESH, updated submission and plotting scripts, use of PGI-19.4 at ITER

03fd963 First step for re-introduction of Eirene FZJ updates

4b5ea24 Pointing to B2.5 submodule with improved LIONSCL default settings

1562550 Pointing to Eirene submodule with MPI debugging warning removal

efe488e Pointing to B2.5 submodules with modified potential equation convergence criterion

1a0021d Pointing to B2.5 submodule with modified potential equation convergence criterion

c9b93c7 Added usage of FFLAGSEXTRA compiler flag switches variable and COCOS number in manual

91f5f92 Pointing to Eirene submodule with bugfix from Jeremy Lore

303dacf Pointing to Eirene submodule with bugfix provided by Jeremy Lore

5aa2d18 Pointing to B2.5 submodule with added initialization of electron drift velocities

548db8e Pointing to B2.5 submodule with added initialization of electron drift velocities

12b09a4 Pointing to B2.5 submodule with additional correction of faulty Be input in SOLPS4.3 files

65e61ce Pointing to B2.5 submodule with an additional SOLPS4.3 input data correction

2cfa4f8 Pointing to B2.5 submodule with open_file compilation fix

1aa0651 Pointing to B2.5 submodule with open_file compilation fix

a713ac5 Pointing to B2.5 submodule with open_file bug fix

6dceee3 Pointing to B2.5 submodule with open_file bug fix

7272b3d Pointing to B2.5 submodule with cherry-pick correction

3444ecd Pointing to B2.5 and Eirene submodules with fixes for cases and conversions when hydrogenic species are missing

d2b13c3 Pointing to Eirene submodule with bug fix for coupling quantities

4fcddd8 Pointing to B2.5 and Eirene submodules with critical bug fixes

a04ec44 Further improvements to conversion scripts

74ce4c4 Pointing to B2.5 submodule with NAG compiler adjustments

c672e33 Improvements to import_files_from_convert script and pointing to B2.5 and Eirene submodules with corrections to handle cases without hydrogenic species, molecules, or molecular ions

6965af0 Pointing to Eirene submodule with bug fix for multiple ion temperatures plasma background input

7b47d4e Pointing to Eirene submodule with bug fix from ammonia test case development

3650eb7 Correction to ITER MSCL module

fc51ed8 Back to ifort64-compiled MSCL module

06d5209 Must use MSCL module with ifort64 compiler, other minor imports from other branches

8691259 Pointing to B2.5 and Carre submodules with fixes for getting Carre2 to work on ITM Gateway within SOLPS-ITER

8456e01 Updating ITM IMAS intel module and adding EBVERSIONGGD variable since it is not provided

c3137d3 Updating ITM IMAS intel module and adding EBVERSIONGGD variable since it is not provided

6eeba40 Updating for IMAS module at ITER and pointing to B2.5 submodule with release/3.0.7 fixes

56abacd Updated IMAS module at ITER and pointing to B2.5 submodule with changes to match Carre2

0f56c87 - Updating configuration files to deal with GGD/1.9.0 - Pointing to B2.5, Carre and Eirene submodules with bug fixes and compilation fixes from Carre2

03c0093 Updating IDS write_up from 4.3 archive and pointing to latest B2.5, Eirene and Carre modules with various bug fixes

c04e3e9 Safekeeping the replacement information for He fuelling cases

35c253b Updating to GGD/1.9.0 at ITER

7780de4 Updating to GGD/1.9.0 at ITER

4bfff26b Pointing to B2.5 submodule with updated GGD linking instruction

0f2fe10 Updating ITER modules to GGD/1.9.0

cf7fff32 Pointing to submodules with updated LEUVEN configuration files

2d9eff9 Changes to compile with new NX nodes of VSC

f370a74 Correcting for the IDS sign convention for Ip and B0

95820d6 Corrected handling of He-fuelled ITER cases IDS write-up

2922775 Bringing up a compilable Carre2 module

1916cc4 Pointing to B2.5 and Eirene submodules with bug fixes from removing labels spotted by FZ-Juelich

4058180 Bug fixes from FZ-Juelich after removal of loop labels

b2188b3 Using math symbols for some b2plot labels

a48fled Using math symbols for some b2plot labels

7124dc3 Correcting the IMAS and GGD modules for ITER implementation

4af7238 Modified SETUP files for IMAS and GGD modules at ITER

fffe08a Pointing to B2.5 submodule with new b2plot features

e4dfe85 Improvements to 'new spectrometer' function and addition of math font in b2plot

1e4a360 Pointing to Eirene submodule with ammonia chemistry test runs bug fixes from Detlev Reiter

26d5583 Pointing to Eirene submodule with ammonia chemistry bug fixes from Detlev Reiter

4da2428 Pointing to Eirene submodule with bug fixes from Detlev Reiter
 6b12113 Catching up with B2.5 submodule
 6cbc749 Further improvements to import_files_from_convert script
 289c1a4 Further improvements to import_files_from_convert script
 17ca31d Further improvements to import_files_from_convert script
 07e2736 Pointing to B2.5 submodule with added conversion rule for LIONSCL and LMOLSCL
 99570bb Pointing to B2.5 submodule with added rule for LMOLSCL and LIONSCL conversion
 5e4ba33 Pointing to B2.5 submodule with added LIONSCL and LMOLSCL conversion rule
 4307135 Pointing to B2.5 submodule with better molecular ion scaling
 84e7f96 Pointing to B2.5 submodule with conversion rules for Eirene molecular ion data
 d784555 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 ede86ad Pointing to B2.5 submodule with molecular ion composition variable MICMP and new
 conversion rules emanating from it
 a3a0bdc Pointing to B2.5 submodule after code clean-up
 94fa0a9 Pointing to B2.5 submodule with new MICMP array for molecular ion composition
 a9b2921 Merge branch 'release/3.0.7' into feature/fix_recomb+redef_gmtry
 bfef87e Merge branch 'release/3.0.7' into IMAS-2525+IMAS-2581-IDS.saves
 4a4cd0a Pointing to B2.5 submodule with IDS geometry fixes and GA stand-alone compilation
 definitions
 7ada2f4 updating GA setup.csh.ifort64/gfortran and setup.ksh.ifort64/gfortran files; primary
 change is to use ifort2017 instead of ifort2016 (the default at the moment on GA's computer)
 0554abb Pointing to B2.5 submodule with fixes to tracing data files
 2194ee2 Pointing to B2.5 submodule with IDS write-up fix when balance.nc is not used
 ed381c5 updating GA setup.csh.ifort64/gfortran and setup.ksh.ifort64/gfortran files; primary
 change is to use ifort2017 instead of ifort2016 (the default at the moment on GA's computer)
 90ffabf Pointing to B2.5 submodule with treatment of unmatched molecular ions for IDS write-up
 81849bb Pointing to B2.5 submodule with added scaling factor definition
 11791ab Pointing to B2.5 and Eirene submodules with IDS bug fixes
 94fe1e4 Pointing to Eirene submodule with output bug fix from Detlev Reiter
 0a8d6b5 Pointing to Eirene submodule with print-out bug fix spotted by Detlev Reiter
 4d0319c Fixes to convert_from_solps4.3 script to adapt to central installations
 4b4093d Fixes to convert_from_solps4.3 script to adapt to central installations
 2c60a6d Documenting new b2yr graphs with and without fix_recomb correction
 347d3dd Pointing to B2.5 submodule with new qc interpolation from Wouter Dekeyser
 221052c Pointing to B2.5 submodule with change of qc interpolation (provided by Wouter Dekeyser)
 859ec03 Pointing to B2.5 submodule adding external sources in b2stbr for balance diagnostics
 f37ed2e Pointing to B2.5 submodule with b2stbr bug fix provided by Ilya Senichenkov
 1cac72d Pointing to B2.5 submodule with b2stbr bug fix from Ilya Senichenkov
 fa14c07 Pointing to B2.5 submodule with corrected error message for BCCON=25
 6654afa Populating edge_profiles, edge_transport, edge_sources and radiation IDS for SOLPS-ITER
 cases to a much fuller extent (in response to IMAS-2525, and including fixes from IMAS-2581)
 and creating the associated watcher and yaml files
 401a239 Adding description of limitations of Pfirsch-Schlueter flows and diamagnetic velocities
 2f61a61 Adding description of limitations of Pfirsch-Schlueter flows and diamagnetic velocities
 dd7c088 Fixes to PGFortran ITER configuration files
 bf3b11c Pointing to B2.5 submodule with inclusion of molecules in core neutral flux for BCs
 7d1e883 Adding creation of .yaml and .watcher files to IDS creation when converting a SOLPS4.3
 case
 530aaf0 Pointing to B2.5 submodule with corrected conditions for conversion of new sputtering
 fort.44 fields
 994e05a Added get_git_name script and second attempt at eirdiag.nds array conversion
 159aa0b Pointing to B2.5 submodule with fixed conversion algorithm for eirdiag.nds arrays
 11da148 ITER modules clean-up
 223f12f Code and ITER modules clean-up
 348ba34 Fixes for b2yt conversion (reported by Stephan Gloeggler), ITM compilation fixes,
 imported Eirene TRIM files from SOLPS5.0 distribution, improved debugging options for Triang
 c4a24aa Fixes for b2yt conversion (reported by Stephan Gloeggler), ITM compilation fixes,
 imported Eirene TRIM files from SOLPS5.0 distribution, improved debugging options for Triang
 e132e0b Adding dataset_description IDS write-up
 c200506 Pointing to B2.5 submodule with fixes for nagsubst and energy sources
 6600895 Pointing to B2.5 submodule with nagsubst bug fix
 1506a82 Fixing a syntax error in the manual
 1b036fc Tweaks to improve SOLPS4.3 conversion and IDS write-up script
 543b730 Adding a hard-wired correction for faulty original ANK data for Be ion descriptions in
 Eirene input files
 d77f3b4 Updating to IMAS-3.24.0 and adding SUMMARY IDS information where possible
 73597cb New 'fcol' and 'bold' functionalities in b2plot (bolding and colouring graph frames, axes

, ticks, and labels)

24a23d5 Pointing to B2.5 submodule with dcol bug fix in b2plot

42386cc Pointing to B2.5 submodule with b2plot radial graphs fix

0254de4 Bug fix for radial plots

55d2aed Bug fixes for conversion and running of ITER Helium cases

1661bcc Further bug fixes for IDS conversion of SOLPS4.3 cases

af3c98c Further bug fixes for IDS conversion of SOLPS4.3 cases

df176f0 Automated SOLPS4.3 conversion script now also writes out corresponding IDS

10fec78 Automated SOLPS4.3 conversion script now also writes out corresponding IDS

ecb85c0 Pointing to B2.5 and DivGeo submodules with configuration files for USTC and KE–Leuven domains

c702032 Configuration for the HPC cluster of the University of Science and Technology, China

9bcbf13 Configuration for the HPC cluster of the University of Science and Technology, China

e630397 Pointing to B2.5 submodule with molecular sputtering indexing bug fix

b90a450 Pointing to B2.5 submodule with molecular sputtering indexing bug fix

42bd1ea Pointing to B2.5 submodule with b2blns bug fix spotted by Ilya Senichenkov

118b839 New run transfer script from Anil Tyagi and rules for partition choice when submitting jobs at ITER

894fb88 New run transfer script from Anil Tyagi and rules for partition choice when submitting jobs at ITER

d5ba40e Clean-up for import from other branches

4b42c65 Updated AUG D+N and D+Ne new drift examples from Ilya Senichenkov

40cd248 Merge branch 'release/3.0.7' into senichi/AUG28903_drift_example

fde60b8 Merge branch 'senichi/AUG28903_drift_example' of ssh://git.iter.org/bnd/solps-iter into senichi/AUG28903_drift_example

2cdff4e Second attempt to upload examples

64f586b change for newest Gateway

c1963c5 Pointing to Eirene submodule with broadcast of saved grid dimensions

17ded9c Bug fixes for plotting PUX,PUY,PVX,PVY arrays from Eirene in b2plot

f0cc707 Pointing to B2.5 and Eirene submodules with bug fixes for treatment of PUX,PUY,PVX,PVY arrays

07a613a queue names changed on the new Gateway

91bbdae queue names changed on the new Gateway

5252b2c Moving the AUG28903.D+N+drifts example to the main directory and adding it to the documentation

b9d7d7f Merge branch 'release/3.0.7' into senichi/AUG28903_drift_example

76b9ec3 AUG28903.D+N_drifts example

b58ab2b Pointing to B2.5 submodule with bug fixes from Garching workshop

3bc531e Pointing to B2.5 submodule with various bug fixes from Garching workshop

d6cf4ae Merge branch 'release/3.0.7' into senichi/loop_scripts

d1be6e7 Further updates to loop scripts and updated documentation and drift input files from Ilya Senichenkov

1623219 Correcting some mistakes in loop submission scripts

dcc8e46 Fourth pass at itersubmit loop submission fix and third pass at external sources treatment for speed-up

0ce2c66 Protection against null string characters in last10 script (spotted by Stephan Gl gler)

e51620b Adding documentation of balance analysis Matlab scripts by David Moulton and pointing to B2.5 submodule with particle source speed-up scheme modified to account for external sources

4b6e70b Protection against null string characters in last10 script (spotted by Stephan Gl gler)

c52873a Adding documentation of balance analysis Matlab scripts by David Moulton and pointing to B2.5 submodule with particle source speed-up scheme modified to account for external sources

3ff1f71 Third pass at itersubmit script fix

937c5c7 Second pass at itersubmit fix

8e66994 Second pass at itersubmit fix

29a6fc4 Pointing to B2.5 submodule with fix to momentum terms for friction force in b2plot

2863aa4 Fixes to previous commits

a2a0e4f Storing momentum sources due to centrifugal, thermal and friction forces for display in b2plot

5e312f3 Adding note in manual about looping scripts

abdd795 Storing momentum sources due to centrifugal, thermal and friction forces for display in b2plot

8d0622b Adding note in manual about looping scripts

290eea3 Merge branch 'senichi/2019.06.26' of ssh://git.iter.org/bnd/solps-iter into senichi/2019.06.26

36d3019 Merge branch 'release/3.0.7' into senichi/2019.06.26

ee4c2f6 Removed unwanted balance scripts

2791a73 Linked to new balance features in B2.5 and included new matlab balance scripts in

\$SOLPSTOP/scripts/Balance
 672fd16 Corrections to SPBSTU cluster submission scripts
 1b02950 Bug fixes for b2yt scripts
 6276ea5 Corrections to IPP loop submission scripts
 daa7a08 Merge branch 'develop' into feature/balance_dev
 14313c9 Pointing to B2.5 submodule with update balance diagnostics
 82d0df4 Adding new sample drifts b2mn.dat file provided by Ilya Senichenkov, fixes to b2co
 handling of Eirene output files, new scripts to plot currents
 2b75f90 Adding new sample drifts b2mn.dat file provided by Ilya Senichenkov, fixes to b2co
 handling of Eirene output files, new scripts to plot currents
 4888db1 Loop submission scripts for ITER HPC cluster
 9c6406b Removing obsolete call to QSUB in itersubmit (spotted by Ilya Senichenkov)
 22e2ddb Removing obsolete call to QSUB in itersubmit (spotted by Ilya Senichenkov)
 4f6d391 Removed unwanted balance scripts
 564d58c Linked to new balance features in B2.5 and included new matlab balance scripts in
 \$SOLPSTOP/scripts/Balance
 41e2e9c Pointing to B2.5 submodule with b2blns bug fix from Ilya Senichenkov
 3e6b119 Clean-up of unnecessary modules in b2plasmastate_inspect
 f07bca5 Addind inspect.plasmastate script and updating documentation about drift-compatible
 sheath boundary conditions
 3e9aa4d Importing fixes from Ilya Senichenkov during Garching workshop
 c1846cd Correcting a typo in Appendix G
 982fbd5 Pointing to Eirene submodule with corrected distribution table printout
 c059716 Fixes for G95 compilation
 5dfc7d0 Configuration fixes for ITM environment
 0d1042e Pointing to Eirene submodule with merge fix
 e92f4e2 Merge branch 'release/3.0.7' into feature/radiation_IDS
 d4811d6 Configuration fixes for ITM environment
 133e1b5 Correcting AUG_16151.D benchmark example reference
 8258e78 Correction to GGD module for ITER PGF90 IMAS MPI compilation
 ef5759b CHANGES TO DEFAULT BEHAVIOUR: Importing mulitple new features and bug fixes from release
 /3.0.7 branch – See B2.5 and Eirene submdules for fuller listing of new features – Pointing
 to ADAS modules with corrected and updated data – Changes to config files to make compilation
 independent of IMAS and UAL versions number – Adding radiation tallies from Eirene molecules
 and molecular ions – Adding nc2text_simple utility – Adding default data for cross-sections
 used for fluid neutral diffusivities – Replacing ITER submission scripts from QSUB to SLURM –
 Added safety to Carre script to make sure the necessary input files are locally present –
 New scripts for monitoring of external sources, currents and Carre rzpsi data plotting –
 Updated documentation, including new description of equations and b2wdat output files
 3fb3fc6 Adapting to IMAS/3.23.2–4.1.0 and updating documentation about use of b2plot radiation
 tallies
 d87b417 Pointing to updated ADAS module with corrections from M. O’Mullane and D. Coster
 daaeeec Bug and environment fixes for IMAS/3.23.2–4.1.0
 c07147a Merge branch 'release/3.0.7' into 3.0.7_beta
 cbee65e Updating to IMAS-UAL/4.1.0
 48f5b03 Updating scripts and AUG_16151 benchmarks to use CODATA–2018 physical constants
 759bdeb fix the comments
 aa9e9f8 Merge branch 'release/3.0.7' into g2dpc/feature/newest_gateway
 b9a2d14 Documentation for some of the to-do scripts
 1b66fdc Other undocumented scripts added to the manual list
 1139674 Merge branch 'release/3.0.7' into g2dpc/feature/external_source
 0a6ff20 Updating scripts and AUG_16151 benchmarks to use CODATA–2018 physical constants
 b5592f0 fix the comments
 a88494b SLURMs
 03690bc update IPP SLURMs
 aad6461 documentation added for b2sext* scripts
 57881fb update commented lines
 6b46013 Adapting fields to IMAS/3.22.0–4.0.3 and making code compilation backward compatible to
 older IMAS versions
 58f08fd Pointing to B2.5 and Eirene submodules with script updates to Python3
 89fd925 Had forgotten one module
 51ce988 Translation of scripts to Python3 and adaptation of setup.csh files
 467925b Compiling with IMAS/3.21.1–4.0.1 and first part of radiation IDS write-up (incomplete)
 8d94aa5 Pointing to B2.5 and Eirene submodules with radiation tallies proposal
 96908f1 Pointing to B2.5 and Eirene submodules with improved radiation tallies calculation (
 including increased precision of He ionization energy)
 5b1c14e Pointing to updated Eirene and B2.5 submodules with fifth pass at importing FZJ Eirene

updates

d23b8af Fifth pass at import of FZJ Eirene updates: photon processes, no XDR support, no photon reflection

f998251 Pointing to updated Eirene and B2.5 submodules with fourth pass at importing FZJ Eirene updates

53fea91 Fourth pass at introducing Eirene FZJ updates: — New collisional—radiative models and photon transport — Change of treatment for computing H—alpha emissivities — Changed default path of TRIM files to Eirene Database directory

c5a43cd Pointing to updated Eirene and B2.5 submodules with third pass at importing FZJ Eirene updates

d8ac877 Third pass at importing FZJ Eirene updates — Using FORT/FORTLC variables for default filenames — Adding an OpenMP Uinp executable — Updated all submodules — Mdsplus database version 26 with corrected dspar calculation

61792e8 Adding separate OpenMP compiled executable for Uinp

ce7a28d Second pass at FZJ updates: Renaming plotting routines

e7cbc4b First step for re—introduction of Eirene FZJ updates

1d3a9db change for newest Gateway

2c2ba01 Adapting fields to IMAS/3.22.0—4.0.3 and making code compilation backward compatible to older IMAS versions

0054069 Had forgotten one module

22f6a53 Translation of scripts to Python3 and adaptation of setup.csh files

7bd9b5b Compiling with IMAS/3.21.1—4.0.1 and first part of radiation IDS write—up (incomplete)

feeld1e Pointing to B2.5 and Eirene submodules with improved radiation tallies calculation (including increased precision of He ionization energy)

1759595 First step for re—introduction of Eirene FZJ updates

3efc566 modifications/additions of scripts for external sources

0a95d4a Sixth pass at implementing new FZJ updates: Moved XMCT timing arrays to fort.14 and new random number generator in Eirene

1619761 First step for re—introduction of Eirene FZJ updates

c05b246 Pointing to B2.5 and Eirene submodules and reducing distance with other branches

645205c Fifth pass at import of FZJ Eirene updates: photon processes, no XDR support, no photon reflection

e483134 Fourth pass at introducing Eirene FZJ updates: — New collisional—radiative models and photon transport — Change of treatment for computing H—alpha emissivities — Changed default path of TRIM files to Eirene Database directory

c995481 Third pass at importing FZJ Eirene updates — Using FORT/FORTLC variables for default filenames — Adding an OpenMP Uinp executable — Updated all submodules — Mdsplus database version 26 with corrected dspar calculation

15b2971 Adding separate OpenMP compiled executable for Uinp

d9aa491 Second pass at FZJ updates: Renaming plotting routines

5f942ae First step for re—introduction of Eirene FZJ updates

b3a4134 Allowing for MPI compilation at GA

55f362a Pointing to B2.5 submodule with further fix for reading older Sonnet files

715d751 Pointing to B2.5 submodule with further fix for Sonnet file read

edfa6b7 Pointing to B2.5 submodule with gfortran compiler correction

a6a8933 CHANGE TO DEFAULT BEHAVIOUR: Pointing to B2.5 submodule with revised definition of plasma sound speed not including fluid neutrals Updated documentation and AUG_16151 benchmark cases

Importing corrections to submission scripts from Ilya Senichenkov

7654fb6 Pointing to B2.5 submodule with bug fix for b2mod.b2.to-astra

935b9bc Fixing a cherry—pick mistake

5378a24 Pointing to B2.5 submodule with fixes to qc calculation and b2tfcc

8d874ce Pointing to B2.5 submodule with backward compatible read of Sonnet grid files

9a53394 Pointing to B2.5 submodule with backward compatible read of Sonnet file

69c9d62 Pointing to updated submodules with bug fixes for Carre and B2.5 and updated DivGeo documentation

98b2bb5 Pointing to B2.5 submodule with bug fix for DDNU non—commensurate conversion

d5f4cec — Pointing to Carre submodule with adapted step size for very large machines — Pointing to B2.5 submodule with corresponding format change for Sonnet grid file — Updated DivGeo documentation

5825e8f Fixes for submission scripts and plotting

fdc5055 Pointing to updated Carre and DivGeo submodules with corrected GA config files

22a7bd0 Fixes for submission scripts and plotting

6747448 Fixes for GA compilation and case conversion: — Clean—up of GA setup and config files — Fix for manual build—up latex executable choice

7d1f0a5 Updating MSCL module at ITER to version 1.2.0

8081173 Check for which PDF to PS converter is available

97816f7 Check for which PDF to PS converter is available

73cb155 get MPI to work ...
278ef92 get MPI to work ...
b97ffb9 updates to the slurm configuration
335fef6 updates to the slurm configuration
a219c61 Merge branch 'release/3.0.7' into dpc/feature/example-pollution-avoidance
c3fd24b use ddt for debugging
c2564d7 avoid pollution of the examples directory if the tar wasn't done at the right level
2605b38 Rebuilt archives for AUG_16151 benchmark cases one level up
1064c80 Correction to B2.5 and AUG_16151 benchmark cases to not include smban term when both ExB
and diamagnetic drifts are not turned on
7db95cc Updated modules for ITER HPC to avoid MATLAB conflicts (spotted by Steve Lisgo)
0c0d2bb Correction to itersubmit script for setting of CPU job limit
3abbde3 Pointing to B2.5 submodule with compilation fixes for GFORTRAN and NAG, including OpenMP
pragmas
87aaa86 Compilation fixes for ITM and IPP environments
84f172f missed the removal of a library
c85df68 Pointing to updated and cleaned-up B2.5 submodule
29bb691 Compilations fixes for ITM
84b0a0e Adding OpenMP options to submission scripts
4a6f810 Generalized '-h', '-T' and '-H' options for submissions scripts and updated documentation
889f2c0 expand documentation of ippsubmit
bf18992 use -M and -T tp set the required memory and time rather than -g and -t
8a7931f changes needed for new IPP cluster (SLES.15 + SLURM)
3f93942 Merge branch 'dpc/feature/new_IPP_cluster' of ssh://git.iter.org/bnd/solps-iter into dpc/
feature/new_IPP_cluster
2fb0112 Generalized '-h', '-T' and '-H' options for submissions scripts and updated documentation
cc3a4b1 expand documentation of ippsubmit
1f56caf use -M and -T tp set the required memory and time rather than -g and -t
8f43db7 changes needed for new IPP cluster (SLES.15 + SLURM)
916c5ef Bringing back ITM gfortran config files
c59936d Pointing to B2.5 submodule with IDS write of ion label and improved resolution in Carre
when searching for the grid frontier
8c99ca1 Pointing to B2.5 submodule with pressure term correction switch
9d8a010 Adding an OpenMP compilation for Uinp and following changes for TRIM pathname in Uinp
7fd8a90 Pointing to updated submodules
891f4b2 Some corrections to manual regarding the total energy balance (typos) and the output by
b2wdat.iout=4
c30225d remove extra \
d96ebf5 add documentation for the new diagnostic for currents
71efbb7 add a diagnostic for currents
ea47efa Pointing to B2.5 submodule with version 26 of MDSplus database
8d922a0 Compilation fix for IMAS ITM gfortran cases
4ecd137 Replacing assert calls in B2.5 submodule with xertst when possible
3c9afa4 Getting non-IMAS gfortran MPI compilation to work on ITM Gateway
d36f5c6 Corrected ITM gfortran MPI settings and solps4-5 converter compilation
f31451b Fixes for gfortran ITM modules
50cf35d Pointing to B2.5 submodule with artificial neutral source speed-up scheme updates
801fe8a Improvements to OpenMP environment and run submission
52b4d0e Modifications to the ITM gfortran configuration files
38a6061 Pointing to submodules with GA gfortran configuration files
a0dafc1 Adding gfortran config files
531bbd2 Bug fixes for Uinp
5521100 Debugging of Uinp while building raised X-points ITER case
4110810 Pointing to updated modules with ITM gfortran config files and improved treatment for
partial solving of the B2.5 equations
c7f55ba Adding ITM gfortran config files
9fe28a2 Rewrite of the b2plot manual section about radiation plots
c6d1a3e Adding OpenMP listobj and depend B2.5 builds in main Makefile
90f6a79 Pointing to B2.5 submodule with Eirene routine name clash avoided
de18423 Pointing to B2.5 submodule with Eirene routine name clash avoidance
65691e5 Pointing to B2.5 submodule with bug fixes from Sergey Voskoboynikov
a3bfc2e Pointing to B2.5 submodule with bug fixes from Sergey Voskoboynikov
8e2973f Pointing to B2.5 submodule with bug fixes from Sergey Voskoboynikov
8c36aff Pointing to Eirene submodule with NAG compilation fix
876b27f Updating OpenMP testing examples
5a501af Pointing to updated B2.5 and Carre submodules
69b91d6 Pointing to B2.5 submodule with revised normalization for psi function plots

e6aa60f Updates to manual provided by Ilya Senichenkov and pointing to B2.5 submodule with Eirene averaging bug fix from Wouter Dekeyser

1f016de Dealing with missing or empty fort.l5 files

fb3f807 Rewrote setup.ksh to mirror setup.csh. Couldn't get awk commands for env variable double entry removal (lines 141–145) to work in bash. Commented out for now.

258ff65 updated GA config files since workshop

527ad47 Pointing to B2.5 submodule with minor corrections to previous commits

3b6ac93 Updated documentation

a296b5e Pointing to B2.5 submodule with OpenMP directives amended with new variables

1a8237e Pointing to B2.5 submodule with get_residuals capability

f4dcab5 Pointing to B2.5 and Eirene submodules with OpenMP fixes and inclusion of sputtered species in artificial source speed-up scheme, fixing typo in manual for ion-ion collisions Coulomb logarithm (spotted by Peter Stangeby)

3473175 Pointing to B2.5 submodule with b2blns fixes from release/3.0.7 branch

5316627 Adding collision cross-sections for default Be neutral diffusivity

70a1b55 Adding capabilities for plotting against full flux function and rho_poloidal

f0ed370 Pointing to B2.5 submodule with IDS ion specification write-up fixes

76a93c6 Added mention about SOLPS-GUI manual in documentation

ee3a4ae – Pointing to B2.5 and Carre submodules with new capability for plotting the magnetic flux functions and radial profiles against normalised flux. – Added default cross-sections for neutral N and Ar transport coefficients in b2sxdx converter – Modified carre script to use specified path from SOLPSTOP environment – Updated documentation

4110221 Pointing to B2.5 submodule with UAL write bug fix

59e11a1 More robust store directory management in Carre script

8c83615 Pointing to B2.5 and Eirene submodules with bug fixes from GA workshop

e2ea17c Bug fix for Carre script

eff1076 Updating solps-iter_update_3.0.7_release script to match DivGeo branch name change

1b41732 Pointing to B2.5 and Eirene submodules with fixes from day 2 of GA workshop

d75ef60 Pointing to DivGeo and Carre submodules with compilation clean-up fixes

1calla4 Bug fix and clarified documentation from GA SOLPS-ITER workshop

6dca807 Moved line 15 for EXTERNALDIR variable as it make more sense for it to be in the ELSE block.

9253805 Makefile now properly sets the EXTERNALDIR to either local path, is the user is located on ITER cluster, or a scp path for local machines.

2ed0aab Moved line 15 for EXTERNALDIR variable as it make more sense for it to be in the ELSE block.

da701bc Makefile now properly sets the EXTERNALDIR to either local path, is the user is located on ITER cluster, or a scp path for local machines.

f5c5640 Conform to /bin/sh when tests only

822b86c Reconfigure ExternalData dependencies to be time relative to cached.

e4aaf01 Conform to /bin/sh when tests only

60c423e Pointing to B2.5 submodule with bug fix for zero toroidal field cases

839dc80 Reconfigure ExternalData dependencies to be time relative to cached.

157427d Bug fix for nc2text_simple to get 2d_profiles to work

44a6b51 Merge branch 'release/3.0.7' into feature/OpenMP_rebuild

dfb1e1 Pointing to B2.5 and Eirene submodules with OpenMP debugging

78825d9 Pointing to Eirene submodule with further debugging output

f219706 Pointing to most recent DivGeo module, awaiting pull request

86b6dd0 Merge branch 'release/3.0.7' into feature/OpenMP_rebuild

588e506 Pointing to Eirene submodule with compiling fixes

33282fe Merge branch 'release/3.0.7' into feature/OpenMP_rebuild

4816afe Pointing to B2.5 and Eirene submodules with various bug fixes

e290236 Adding 3.0.7 Release update script

3c2b83c Updating IMAS and GGD modules at ITER to ensure compatibility

dc7478a – Partial MPI debugging fixes and stricter debugging flags – Fixes for NAG compilation – Pointing to B2.5 submodule with b2mxar bug fix – Documentation of new ADD_TE.CORR.TO_PO switches

3alc942 Adding scripts directory to mpi script

83364a2 Streamlining NAG compilation

e4e4e79 Adding compilation and documentation of nc2text_simple utility program

da6fb72 Fixed bug for 1D arrays with subscript

44e9cc4 whereami now recognizes GA Iris/Saturn worker nodes, not just login nodes

4ff3e56 Updating to loading MATLAB/2018a module on ITER HPC to avoid a conflict

2787390 Add first pass at simple nc2text

5ef92f5 Updated description of b2blnc.SPb tracing files

c2f5366 Revised set of setup files for GA (Iris) environment

2ebelca Switch test script from csh to tcsh

8f67617 Update ORNL config
 4d07bd6 Removing obsolete QSUB ITER submission scripts and updating documentation
 b0f41a3 Updated scripts and runs for AUG_16151 benchmarks
 978b738 Updating AUG_16151_D benchmark
 703a3ca Code clean-up for triainq
 11637f9 Fixes to AUG_16151_D+C+He benchmark scripts
 838ec13 Pointing to release/3.0.7 submodules and accompanying fixes to Makefiles
 939b610 Pointing to B2.5 and Eirene submodules with various bug fixes
 34c9a7a Merge branch 'develop' into feature/OpenMP
 12f3322 Pointing to updated submodules
 1af16b3 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 ee4d9b6 – Partial MPI debugging fixes and stricter debugging flags – Fixes for NAG compilation –
 Pointing to B2.5 submodule with b2mxar bug fix – Documentation of new ADD_TE_CORR_TO_PO
 switches
 764c4b5 Had forgotten the graphics case for Triang compilation fix
 4bba2cd Fixes for NAG compilation
 3e30f0a Merge branch 'develop' into coveleb/bugfix/whereami_edit
 baf892b Compiler fixes required by G95 compiler
 65b3b96 whereami now recognizes GA Iris/Saturn worker nodes, not just login nodes
 0678ca4 Updating to loading MATLAB/2018a module on ITER HPC to avoid a conflict
 28eb11d Pointing to updated submodules
 62e1809 Updated description of b2blnc.SPb tracing files
 1e3d05c Pointing to Eirene submodule with ITM name clash resolution
 ce6b849 Pointing to Eirene submodule with corrected ITER PGF90 config file
 7ac78a9 Pointing to Eirene submodule with corrected ITER PGF90 config file
 13f7f5a Pointing to updated submodules
 6e5ff99 Merge branch 'develop' into jlore/setup
 ebb1971 Pointing to the submodules with GA setup files added
 1c158d4 Revised set of setup files for GA (Iris) environment
 b5721c4 Switch test script from csh to tcsh
 376a69d Update ORNL config
 b7955b0 Merge branch 'develop' into feature/OpenMP
 252af49 Pointing to updated submodules
 04ebbcf – Added in all submodules compiler flags for when using non-Intel compilers for module
 stand-alone compilation. – Removed QSUB.iter* files and updated documentation about SLURM
 submission at ITER. – Added SLURM.eir Eirene standalone submission file.
 63663f7 Updated scripts and runs for AUG_16151 benchmarks
 31bec01 Merge branch 'develop' into feature/OpenMP
 3433ab7 Pointing to Eirene submodule with bug fixes from FZJ_synchronization merge
 b2fb39b Pointing to Eirene submodule with fixes after FZJ_synchronization merge
 2f04c9c Updating AUG_16151_D benchmark
 255f5f7 Pointing to B2.5 submodule with corrected LBCSTRA computation for strata repetition per
 species
 2f3421b Code clean-up for triainq
 88510dd Merge branch 'develop' into feature/OpenMP
 be48d81 Pointing to merged B2.5 and Eirene submodules
 ef60f4f Fixes to AUG_16151_D+C+He benchmark scripts
 2a9dlaf Missing files in previous commit
 464cc9c Adjustments for MPI compilation with PGF90 compiler at ITER
 5846526 Pointing to B2.5 and Eirene submodules with FZJ synchronization updates
 8ca9465 Pointing to updated submodules and adapting Uinp and Triang Makefiles for changes from
 Eirene
 4bf4138 Removing usage of Hollmann files graphite_ext.dat and mo_ext.dat
 23ddcda Compilation fixes for Triang using new FORT variable setting for HPUX filenames
 0c5756d Applying the FORT variable to allow for HPUX default filenames
 99f33fa Following changes from Eirene FZJ master version: – Moving nltrimesh to block 14 of the
 Eirene input file – Changing LCHKQUD to LCOARSE in block 14 of the Eirene input file – Timing
 information for MPI balanced parallelization strategy moved to fort.14 – Changing default
 value of NFILE to 356 – Changing default density range for b2frates to 1e14–1e22, and
 increased default size of rtnn to 32
 7b1bb0e Changes to follow Eirene synchronization
 1de37c7 Pointing to B2.5 submodule with LBCSTRA bug fixes
 7815faa Adjustment for compilation in ITM environment
 965ce80 – Pointing to B2.5 and Eirene submodules with artificial neutral source speed-up method –
 Updated documentation, including new description document for b2blnc.SPb tracing file –
 Updated ITER setup.csh files for IMAS/3.21.0–3.8.4 (partial) – Updated AUG_16151 benchmark
 cases

2224fea Pointing to submodules with final OpenMP implementation
 a453d5f Adding optional extensions for executable paths in Makefile
 48e584d Further changes for OpenMP compilation and updated documentation
 0a5af3a Adding GFORTRAN define for compilation
 43c3636 Pointing to Carre and Eirene submodules with GFORTRAN compiler flag changes
 c762cc1 Merge branch 'develop' into feature/OpenMP
 6baa0ed Adding GFORTRAN define for compilation
 bdd7d03 Second pass at cleaning up ITER setup.csh files
 e1db188 Merge branch 'develop' into feature/OpenMP
 6713b23 Adapting other changes for OpenMP scripts
 2968643 Removing support for old ITER cluster (now decommissioned)
 fec8078 Changes to have CI server also use new cluster environment at ITER
 fcd2e8f Updated documentation on boundary conditions
 a10ffab Second pass at previous commit
 cbclffa Trying to prevent SOLPSTOP name clashes with mpi, debug, openmp and ig scripts
 6c2efaa Further code clean-up from NAG compiler
 b1295fe Recovering capability to launch several B2 executables with a single b2run command
 77ac33e Fixes for MPI PGF90 compilation at ITER
 c9b6e0e Bug fix for enforcing constant flux tube magnetic area in private flux regions
 6e34b07 Not applying corr_core_dn to neutral species
 5b24cf7 Updating documentation and warnings for cases with drifts
 022543f update of whereami for JET
 2068ced Updated AUG.16151.D benchmark case
 e9065b0 Adding python modules to ITER old cluster configuration settings
 ff5f124 Tweaking the Uinp config files
 fedf07e Fixes for more general manual build
 aa3ab3d remove the obsoleted -vec-report0
 329fe3f Pointing to submodules with -vec-report0 removed config.JET.ifort64 files
 5dab80d remove the obsoleted -vec-report0
 9cb34c4 Merge branch 'develop' into feature/OpenMP
 24f1fe2 Pointing to B2.5 submodule with min_max_df0 fix
 f2bca7f Merge branch 'develop' into feature/dpc-fixes-add_script
 f432532 needed by mtv_3d.nf
 845d17c Merge branch 'develop' into feature/OpenMP
 75ac5a4 Updated AUG.16151.D and AUG.16151.D+C+He benchmark cases
 247e75e Merge branch 'develop' into feature/OpenMP
 1f7f1b4 Pointing to updated B2.5 submodule
 7ea3753 Further tweaks to regression test code
 d72c233 Tweaks to regression testing script provided by Tamas Feher
 f52abeb Pointing to B2.5 submodule with Windows adaptations provided by Serguey Voskoboynikov
 17c42a9 Merge branch 'develop' into feature/OpenMP
 edbb865 Pointing to updated B2.5 and Eirene modules
 a3c266c Third pass at avoiding infinite loops when block 13A in the Eirene input file is present
 6f56205 Improving handling of debugging option for save_mds and resave_mds scripts
 8b7f71e Fix for handling of shotnumber.history file
 6694d9c Fix to Makefile for b2md handling of shotnumber.history file; pointing to updated submodules
 0ble139 Second pass at b2run bug fixing
 91df43d - Bug fix to b2run for when the default debugger is invoked for other executables than b2mn - Pointing to B2.5 submodule with bug fix in b2rd for new quantities (provided by David Coster)
 9cbc6ff Adding Makefile dependencies on setup.csh file
 d3fb6fd Merge branch 'develop' into dcoster0/feature/MARCONI
 828b98e Pointing to B2.5 submodule after b2ag debugging
 0092ab2 use 2018 intel; disable imas until JET imas version updated; build on xelatex fix
 0ea2238 Pointing to Eirene submodule with fix for reading optional block 13A in input file
 b5da894 distinguish between Gateway and Marconi on the presence of /afs
 0129594 Attempt to address IMAS-1831 to better deal with character encoding
 aa47e42 Using dedicated B2.5 routines for the computation of the Pfirsch-Schlueter currents
 d6a9e6a Merge branch 'develop' into feature/OpenMP
 335cad3 Pointing to B2.5 after merge of fixes for 5.0 mimic behaviour
 b676fd0 New switches for 5.0-mimic runs
 2127838 Pointing to B2.5 submodule with bug fixes for averaging scheme and statistical post-processing
 68de55e Fixed bug in res_av script + updated documentation for EIRENE averaging schemes
 ab143be Replacing *.spbau with *.spbstu in whereami
 1e771af Fixing some issues with SNCARG.ROOT and -DNCAR4 in configuration files for SPb cluster (

required for b2plot)
 9168cd5 Adding mpi.openmp Makefile objects
 2ed5115 Avoiding an endless loop when reading Eirene input file
 05b9d67 Adding safeties in b2co for corrupted tracing files
 d41e607 Adding safeties when etags is not present
 01c4393 – Pointing to B2.5 submodule with position fix for extra label in multi-plot pages –
 Providing fix for IMAS-1814
 1ddc29f Pointing to B2.5 submodule with new CI testing scripts
 ffb7704 – Adding safeties to scripts in case binary characters snuck into the run.log file (as
 suggested by Stephan Gloeggler) – Attempting to avoid printing string nulls in sps_id from
 Uinp – Pointing to B2.5 submodule with other bug fixes
 7e31635 New 'glsz' b2plot command for setting global header font size
 d5d8270 Pointing to B2.5 bug fix for testing against impurities in BCENI=23/24
 e3062a7 Adding the date to the front page of the manual
 07335bd Fixes due to an MSCL variable mismatch (see IMAS-1742) and other bug fixes in b2yt
 434fe4d Same safety in Triang for (U)inp option and loading corrected MSCL modules
 e15e78a Added safety check in triang to suggest b2ag when successful triangulation is followed by
 a new carre mesh
 50e2509 – Adding safety checks for various parameter sizes in Uinp. – Have Uinp stop when the
 contours fail to close. – Change triang script to not proceed by default if Uinp has failed.
 e6f18a1 Pointing to bug fixes in Carre and DivGeo
 ecaa66c Adding mention of new SOLPS-ITER Confluence Wiki page in Download and Installation
 Instructions document
 18b6138 – Fixing rps script and underlying risepsi program. – Pointing to Carre submodule with
 bug fix for pntrat_old usage. – Added SOLPS-ITER Confluence web page reference to
 documentation.
 8fd277c Merge branch 'develop' into feature/OpenMP
 7cefdb1 Further adjustments for OpenMP and hybrid MPI runs
 ef2936a – Enforcing that b2.boundary.parameters be used for DN cases – Fixing a round-off error
 in Eirene when computing contours – Adding diagnostics to triangulation Eirene run – Updated
 documentation
 9fd7f74 Merge branch 'develop' into feature/CX_Nitrogen
 1ae2dde access the recompiled and reinstalled ggd library
 29188b4 Pointing to new '96 CX ADAS data for Nitrogen provided by Martin O'Mullane
 f73221c Bug fixes for Carre
 56e5795 Second pass at bug fix for mapping to midplane in b2plot
 437fcaf Bug fix for mapping to midplane b2plot radial graphs from divertor regions
 ee6ccd0 Adding Ti to bolometer output data file from b2plot '-l3 user' function
 90a9182 Work-around because of gfortran compilation issue
 43cfff Improvement of '-l3 user' b2plot function for bolometer support
 3445a77 Pointing to Eirene submodule with bugfix for tfluxm indmpi call
 aa03c3d Changing to compiling with IMAS/3.20.0 with Intel compiler on ITM Gateway
 b85a0ff Merge branch 'develop' into feature/OpenMP
 51ale2a Corrections to MPI job submission given by Tamas Feher
 db599cd needed for post-bdw Gateway
 950bf9f Changing the NAG compiler module at ITER
 94d71f6 Merge branch 'develop' into feature/OpenMP
 cal450b – Adding OpenMP compiling options and submission scripts – Pointing to B2.5 submodule
 with OpenMP improvements – Updated documentation
 9bc3dd2 Pointing to submodules with CUMULUS config files
 612217a Merge branch 'develop' into feature/cumulus-freia-config
 443a06f Configuration files for CCFE Cumulus cluster
 19022a6 Bug fix in B2.5 spotted by Tamas Feher
 2b5c429 Typo spotted in B2.5 by G95 compiler
 8ea07c8 Additional file for PostScript manual build-up on Marconi
 elaa62c Change in B2.5 to safeguard against inconsistent user input for feedback schemes
 febab6e More manual fixes for Marconi build
 6e65ebe Pointing to submodules with MARCONI config files
 fflaf2a Merge branch 'develop' into dcoster0/feature/MARCONI
 c3283ad configuration files for Marconi-Fusion
 7f34135 Package files for manual build-up on Marconi
 7342d51 New location for SOLPS4.3 archive
 fd80373 Pointing to B2.5 submodule with correction to b2mod_numerics_namelist
 9ca1883 no g95 for Marconi
 c71c96b Fixing thermo-electric coefficient flux limit to match documentation in manual (spotted
 by Egbert Westerhof and Menglong Zhao)
 d8307a7 config files for Marconi

0bdccd3 Using new plt42 ADAS data files
 d7c5e78 Fixes for D+He+W+Kr CFETR case (reported by Guozhang Jia)
 09941a6 Cleaning up trailing spaces
 5ecbf3d Merge branch 'develop' into feature/OpenMP
 8a9f617 Format fix in Unip for cases with 100+ plasma background species
 b269213 Removing the LibTIFF module causing problem for the new ITER HPC system
 2882b66 Clean-up of setup.csh files for ITER
 de03839 Fixing print-out of B2.5 GIT hash so b2fstate_OK still works
 d7769bd Adding computation of cflags and output of vertex_data.out file in b2ag
 a2e42b2 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 cb18ccf - Adding gnuplot module for job submission on new ITR cluster (suggested by Simon Pinches)
 - Clarified some datasets descriptions as requested by Zhongping Chen
 45b9419 Fixes for Gateway compilation with IMAS/3.12.1
 6f7662d Additional printing; additional region
 91b8599 Backward compatibility adjustments when module package variables are unavailable
 76bdb5a Changes requested by Simon Pinches for better compatibility with modules
 ea72f46 Change suggested by Simon Pinches for GR library location
 b70e6af Updating the gfortran modules for ITER new HPC cluster
 24d02fd A few more final tweaks from the old cluster side
 bebc9d3 Final compilation tweaks
 89a0180 Fixes for Eirene modules compilation when using NAG compiler
 8922018 Backward compatibility fixes for old ITER cluster
 b564f01 Merge branch 'develop' into feature/new_ITER_cluster
 0ead6e4 Simplifying IMAS compilation flags as suggested by Simon Pinches
 7523198 Further compilation adjustments and clean-up
 9ddea32 Merge branch 'feature/IMAS-1330-new-hpc-cluster' of ssh://git.iter.org/bnd/solps-iter
 into feature/IMAS-1330-new-hpc-cluster
 0709c16 Changes for job submission on new ITER cluster
 35c92cc Pointing to B2.5 submodule with momentum sources fix
 ala5aed Further tuning of configuration for compilation on new ITER HPC cluster
 ec78d52 Tweak setup for new ITER cluster
 c4255e1 - Improvements to new ITER HPC job submission scripts - Pointing to B2.5 submodule with
 IMAS 3.12.0 backward compatibility fixes
 40c9e37 Pointing to B2.5 submodules with fixes for cases with zero toroidal field
 845dc28 Pointing to B2.5 submodule with compilation change for old cluster
 74elc85 Merge branch 'feature/new_ITER_cluster' into feature/IMAS-1330-new-hpc-cluster
 9774a10 Further improvements for compilation on new ITER cluster
 0cf4466 New setup script for environment on new ITER HPC cluster
 51de932 Merge branch 'develop' into eldond/feature/ga_install
 4cff0e8 Cleanup GA entries in switch in whereami
 25d8bd6 Identify GA servers iris & saturn in whereami
 fa98ab2 Typo in the previous commit
 f9f2a9b Fix for cases with zero toroidal field (bug reported by Michele Sala)
 e330fc7 Fixes for ITM compilation
 f7edefa Activating ramping of facdrift switches in b2news_ (bug reported by Mirko Wensing)
 8c5b00b Fixes to LD_LIBRARY_PATH to remove a warning from gwprompt on Marconi Gateway (identified
 by Dejan Penko)
 2996390 Adding recombination energy to target loading post-processing files (provided by Stefano
 Carli)
 f072954 Further compilation changes for new ITER HPC cluster
 a554460 Adding missing variables from block 13 in Eirene input file for converter (reported by
 Stefano Carli)
 66066e3 Removing a duplicate SAVE
 23b88b5 - Added some instructions about Eirene speed-up scheme in manual. - Pointing to B2.5
 submodule with feedback scheme bug fixes.
 f354703 Fixing merge issues for Makefile
 2b3c818 Merge branch 'develop' into feature/new_ITER_cluster
 59505a5 Pointing to merged submodules with modification for new cluster
 1b67f0c Adding SLURM ITER job submissions scripts for new cluster
 7fd9b16 Forgot a SAVE statement in previous commit
 be82b88 Pointing to B2.5 submodule with corrections for multiple feedback schemes
 34448d1 Adding other hpc-login0* nodes to new ITER HPC cluster and adjusting list of modules in
 setup files
 e509b7d Fix (for He and up) of a double counting issue in calculating the particle balance when
 the ionizing_core option is used with B.C. 8; note: an error can still exists for H/D/T
 c7dd126 Adding an all.nox.mpi compile target
 0392e52 Pointing to B2.5 submodule with updated feedback scheme for average impurity

concentration

1e3236e New ADAS plt41 and '50 year data', and B2.5 safeties for ADAS local files

3a821a0 Pointing to B2.5 submodule with fix for finding local and standard adf15 files

cd64e16 Pointing to B2.5 submodule with b2nxfx bug fix

225859e Pointing to Eirene submodule with IIMINT bug fix from Petra Boerner (reported by Sarah Newton)

1077aae – Pointing to B2.5 submodule with fixes for post-processing of linear cases – Pointing to DivGeo fix to be more flexible about EFIT headers – Adding a divgeo_nox compiler target

1f21970 Changes following discussions with users at PSI 2018

5f31609 Various bug fixes for saving to MDSplus database

bc67924 Merge branch 'develop' into feature/new_ITER_cluster

bc03ef4 – Merging with develop branch

b2379e2 New b2plot option for plotting total source terms or individual contributions

8ccd37f Changes for partial forward-compatibility with new ADAS formats

2677369 Adding Summers ADAS diagnostic line data and reviving b2plot 'spec' option

fc4038f Updates to documentation for 28th SOLPS-ITER User Forum

49a3bbf Changing default N recombination reaction to ADAS

3bf4d0b Pointing to B2.5 submodules with corrections about ExB electron velocity calculation

a9815ae Adding rdneureg to display_tallies

445bf33 Using b2stel_phm0 as a multiplier to sources due to atomic physics (suggested by Serguey Voskoboynikov)

6165e94 Update ORNL config

16b46ba Adding a B2.5 standalone test slab orthogonal example for benchmarking (provided by Wouter Dekeyser)

180bc0e Bug fix for lambda_q calculation

fa73c0d Forgot one line in previous commit

3306cdc Adding computation of lambda_q and correcting target heat fluxes for energy carried back into the plasma by recycled neutrals and sputtered material

de5316a Merge branch 'develop' into feature/new_ITER_cluster

bf48d50 IMAS 3.18.0

cb93555 Optional output switches b2npc7_iout and b2npc11_iout added (provided by Serguey Voskoboynikov)

e79d632 Changing default value of b2mndr_hz to 1.0 to match SOLPS5.2 physics model

002d62f Changes proposed to Serguey Voskoboynikov for treatment of perpendicular viscosity drift terms

0c4e696 Second pass at correcting conversion rule for lpfr[bt]_[io]

403ea3e Correcting conversion rule for lpfr[bt]_[io]

bd7c9cc Added neutral radiation tally rdneureg

c95ef95 Added derivation of parallel viscosity term in manual

59635f7 Added same in README file

fde1266 Adding documentation about the SETUP/setup.csh.HOST_NAME.local file possibility

f19c669 Update config and whereami to allow for ORNL cluster

fb8eac2 Correcting logic for ADAS reactions in Uinp

abd4132 Linearization of perpendicular viscosity current contribution to Eirene fluxes

4a2b747 Updating manual and equations document to reflect changes in treatment of perpendicular viscosity current from Serguey Voskoboynikov

77b2b54 Merge branch 'develop' into voskobs/feature/jvisper_in_fna

41ddf17 Fixing errors in fort.44 data description from manual (spotted by Ilya Senichenkov) and improvements to 2dvi script for SOLPS-ITER use

c7026ff Bug fixes for Uinp computations of l_h_mol and latmscl

0862457 Adapting scripts for case when ds* files are not yet in run directory but only in b2mn.exe.dir

039f7cb Pointing to B2.5 repository with taking to account jvisper in fna

1d84c34 – Changed definition of kinetic energy to only include parallel velocity (and updated documentation accordingly). – Updated manual and equations document.

cda3bf0 Superseding b2tvspa_vis_par switch with b2tfhe_vis_par (as suggested by Serguey Voskoboynikov)

9aa8aa0 Further improvements for dealing with NetCDF files when using b2co

b560143 Fix to b2co script to grab *.nc files

46b9aef Had forgotten a switch in b2ag.dat for limiter cases (reported by David Coster)

ef35450 Cleanup of equilibrium conversion scripts

57859cc Adding fnex and fney quantities in b2plot and changing default carbon recombination reaction to ADAS

8022cb0 Bug fix for when the DEVICE environment variable has not been defined

3bb2878 b2plot 'adas' command survives a missing data file

d854600 Further consistency checks for use of eirene_ionising_core

5fbea04 Modified documentation for eirene_ionising_core switch

7fc9a3a Ionising core modulation (part 2)
d1b6181 Modulating ionising_core flux according to poloidal density variation to be consistent with drift cases
2363df0 Updated documentation on user command in b2plot + fixed bug in PFC freefem model
0ff2501 Changing the NCL module at ITER as per IMAS-1577
c0d1249 Changing the NCL module at ITER as per IMAS-1577
f710a25 Further changes for compilation on new ITER HPC cluster
0804338 Merge branch 'develop' into feature/new_ITER_cluster
1eae8eb Pointing to merged submodules with limiter configuration and Chinese training session changes
e57d9d8 Adapting run environments to new IMAS releases
630ec9c Various improvements from Chinese training session: — Pointing to B2.5, Eirene, DivGeo, Carre, and ADAS submodules with Chinese training session changes and bug fixes. — Added a NOQUAD compiler option when quadruple precision is not available. — Uinp now adds Eirene surface tallies to be printed for surfaces where tallies are requested to be computed. — Changes to Uinp to create cases with bundled charged states. — Carre, Uinp and Triang scripts now check for presence of a SOLPSTOP file to decide which executables to use. — Earlier definition of SOLPSLIB in setup.[ck]sh to allow for its use in host setup and config files. — Modified example files from manual to reflect new default choices of transport coefficients. — Documented new 'nice' b2plot toggle switch. — Canada signs an MoU with the ITER Organization. — Other improvements to the manual to resolve inaccuracies/ambiguities and provide additional information. — Cleaned up config files.
6e3b654 Adding electron temperature and density information at the peak power locations
0a66cad Stand-alone compilation settings for IPP requested by Fabio Subba
32e141a Fixing output formats for large census arrays
a2955dd Pointing to Eirene submodule with bug fix for reading non-empty one-line block 13 input file
f5b90ed Pointing to ADAS submodule with bundled data
08285d9 Correction for solps4-5 conversion cases where fort.30 file is not already present
a6c2d1d Fixes for solps4-5 converter compilation
562bfd8 Adding ITER.2298_Honly.20MW case to list of examples
eff0395 Adjustments to be able to deal with limiter cases — Fix to lns script to point to the limiter equilibrium when present — Fixes to Uinp when dealing with a limiter configuration — Pointing to modified DivGeo, Carre and Eirene submodules with limiter configuration changes — Included documentation in SOLPS-ITER manual for DG limiter configuration procedure — Uinp output prettification
19d088b Moving to IMAS/3.17.0/ual/3.8.0
a13c0e8 Merge branch 'develop' into feature/new_ITER_cluster
8dc2b6b Further adaptation to new grid_ggd objects and IMAS/3.17.0
9455d1a Updating documentation
2f3ece5 Pointing to B2.5 submodule with fixes to total heat load calculations
d219857 Other fixes for Marconi Gateway runs
f9a86d6 Fix to 2d_plots for Marconi Gateway
5204e6f Pointing to B2.5 submodule with NFILE and L_NEUTFLUX conversion rule changes
e96165a Pointing to B2.5 submodule with b2yt bug fix when removing species
11f47ef Pointing to B2.5 submodule without extension of Eirene volumetric losses quantities in b2plot
3feba69 Pointing to Eirene submodule with trace diagnostics bug fixes
195d994 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
6cbb4ef — Improvements to IPPITM environment — Adding new ITER #2275 example (low power D-fuelled) case — Recommended MPI setting for ITER HPC MPI runs with gfortran — Pointing to B2.5 and Eirene submodule with bug fixes to b2yt when 'shaving off' species
2d769da Merge branch 'develop_jlore_py' of ssh://git.iter.org/bnd/solps-iter into develop_jlore_py
ca406d9 Manual notes for python scripts
08e97ca Update to make python 3 and 2.7 compatible.
3227f63 Extend 2dt to work with strictly 1d arrays
3e84961 cleaned up comments a bit in python scripts
1193607 Added resall_D python version
5953c3c 2dt python version
4778b63 Update to make python 3 and 2.7 compatible.
0eac528 Extend 2dt to work with strictly 1d arrays
20df1bb cleaned up comments a bit in python scripts
58d16d5 Added resall_D python version
38b18eb — Added NATM, NMOL and NION as quantities available in b2plot — Pointing to Eirene submodule with bug fix for scaling of volume-averaged tallies — Corrected b2mn.dat.5.2 example file following report from Jeremy Lore — Pointing to DivGeo with updated ORNL config

file

f117134 Updating UAL to point to new imas_constants

1f8bb88 2dt python version

53202db reset stty after triang

1e02b84 Update ORNL config

7b864a8 Updating and cleaning up examples after changes to CODATA 2014 constants

531d1db Merge branch 'develop' into feature/new_ITER_cluster

03e4ae9 Improvement for NAG compiler options and further adjustments for new GGD structures in B2.5 submodule

0647b7e Adapting to IMAS/3.17.0

d5098ab Modified b2mod_constants and emissivity_constants to use CODATA 2014 values

a3c3d2b Pointing to B2.5 submodule with static grid objects and exporting NODENAME and SOLPS_CPP variables to Makefiles

8bfcc11 Update to SPB speed-up documentation provided by Lisa Kaveeva

ef9b119 Merge branch 'develop' into feature/SPB_speed-up

2d6bb39 Merge branch 'develop' into senichi/feature/LOOP_submission

5f9ba61 Clean-up of SLURM submission scripts after consultation with CPT team

f08a71a Pointing to B2.5 submodule with SPB pressure correction speed-up scheme and supplemental documentation

b7b6d05 Pointing to B2.5 and Eirene submodules with bug fixes coming from NAG compilation code clean-up and avoiding an Anaconda3 module error

cbal0d3 Further code clean-up from NAG compilation on new ITER HPC cluster

cac279c Tweaks to Ilya's proposal for loop submission scripts

d010aae Removing double entries to environment variables \$PATH, \$MANPATH, \$LD_LIBRARY_PATH etc inside the setup.csh to avoid crashes in loop submission

3d7fb63 Merge branch 'develop' into senichi/feature/LOOP_submission

blae6da Compilation of Uinp and Triang for new HPC cluster and pointing to B2.5 and Eirene submodules with further improvements for compilation on new HPC cluster

b24514e Further progress towards new ITER cluster installation

f104780 Pointing to B2.5 and Carre submodules with compilation improvements

e6232b8 Pointing to B2.5 submodule with b2yt fix for track_chem_sput

0efd781 Pointing to B2.5 submodule with b2yt fixes for SOLPS5.1 conversions

797ac57 – Improvements to b2yt and associated scripts, as discussed in 25th User Forum –

Recommitted ITER #2171 D+He+Be+Ne example case – Pointing to Carre submodule with bug fix for deep core grids

b464953 Pointing to B2.5 with corrected Makefile

7bcc40d – Pointing to Carre submodule with bug fix for limiter cases – Pointing to B2.5 submodule with additional input check for density feedback schemes

ec0dc49 Updated to use same var name as XB

f917166 Update config for multiple ORNL systems, switch default compiler to gfortran

c64a5fc When setting up convert, missing AMJUEL means b2mn will be called

8c7ac43 Adding ITER #2299 example

a33669d Changes for intel MPI compilation

59c62f0 Merge branch 'develop' into feature/new_ITER_cluster

9653179 Pointing to B2.5 merged submodule

cd44e9b MPI compilation with OpenMPI/pgf90

11ef848 Documenting the two new cases

7b3f688 Moving ITER #2264 case to proper directory

6fe74de New ITER examples: #2264 Ne injection 100MW and #2360 Helium majority 40MW scenarios

8231c80 Manual corrections and B2.5 bug fixes provided by Jeremy Lore

3ef5a04 Continued adjustment to new IMAS version

ac7eefc Continued work towards compilation in both ITER cluster environments

12d4888 Continuing fixing for ITER cluster compilation in mixed environment

db250a2 Merge branch 'develop' into feature/new_ITER_cluster

89ddef5 Pointing to merged B2.5 submodule

213a88d Further advances towards installation on new ITER HPC cluster

c8a1ea7 Avoiding to source setup.csh in loop scripts, for testing and debugging

d261c07 Pointing to B2.5 submodule with debugged b2usr_loads calculation

1a446c5 Further improvement to loop submission scripts for Marconi Gateway and SPbSTU cluster

dc751db Dealing with NetCDF malfunction in B2.5 submodule

173ddda Further adaptations for new ITER cluster with new module list

288f3fa More tweaks for installation on new ITER HPC clusters

8a35f1c Adding MPI conditional options for new HPC cluster

800cc00 Further improvements towards compilation on the new ITER HPC cluster

fdcabaf Further adaptations for SLURM submission on new HPC cluster

2f193db Further fixes for ksh setup files at ITER

aa752ac Adding a setup.ksh.ITER.ifort64 file (still untested)

0871274 Preparatory work for new HPC cluster installation
 d4810ff Debugging b2co and associated scripts
 1f9f7ac – Pointing to B2.5 submodule with improvements to b2co to work as advertised – Walking back to IMAS/3.12.0 (because of IMAS–1398) – New script for b2co to fetch tracing files whether they are in b2mn.exe.dir or in the run directory – Updated documentation
 82d62cb Sturdier imasdb.solps–iter script when DEVICE is not defined
 5e47b06 Updating the IMAS module load at ITER
 f99a154 Pointing to B2.5 submodule with bugfix for BCCON=16,22,26
 77005c5 Merge branch 'develop' into feature/SLURM.Gateway
 25b4f90 A few more tweaks to SLURM submission scripts
 83bbdba Corrections for the MPI mode and improvements to loop scripts
 ad6ac21 Pointing to B2.5 submodule with user–specified precision of *.dat files
 15717fe Fixes for when compiling MPI executables only
 5a2a6bc Merge branch 'feature/SLURM.Gateway' of ssh://git.iter.org/bnd/solps–iter into feature/SLURM.Gateway
 5247c8f Improvements to SLURM Gateway submission scripts for MPI runs
 41639ca Adding job error and output files to SLURM submission scripts
 b45d8f9 A loop submission script for standalone runs on TOKS cluster in Garching
 90f7c4d Improvements to SLURM.itm and SLURM.itm.loop
 b2a81c3 Changing queue name again and removing account statement in job submission scripts
 f4cce7b Correcting the queue name
 d343787 First attempt at ITM Gateway SLURM submission scripts
 738cdd1 Code cleanup
 38d5585 A loop submission script for standalone runs on TOKS cluster in Garching
 2770ed4 Improvements to SLURM.itm and SLURM.itm.loop
 38bbf73 Changing queue name again and removing account statement in job submission scripts
 db28d79 Correcting the queue name
 0b41ccd First attempt at ITM Gateway SLURM submission scripts
 38e941d Pointing to B2.5 submodule with full precision print–out of .dat files in debug mode
 0f821b5 Safety to imasdb.solps–iter script when the DEVICE is not declared
 28844d2 Allowing for PGF90 compilation when using quadruple precision
 d7c8bee A better strategy to handle forked develop branches in Carre and DivGeo
 d3fd48f Divgeo and Carre develop branches were not in fast–forward state
 6836183 Trying instead with 'branch = .' as suggested by Simon Pinches
 3cedfbc Making .gitmodules file consistent with new submodule branch naming
 9139881 Updating the download and installation instructions
 b4ce12a Clean–up of master and develop branches for submodules
 74d8ecb Pointing to B2.5 submodule with fixes for CI compilation
 3cac7e1 Pointing to merge B2.5 submodule
 10c6bf2 Pointing to B2.5 submodule enforcing a 4–digit limit for IDS run number
 96f1da3 Making sure the proper IMAS database is used for IDS write
 8ea181d Updating code download and installation instructions to reflect the new branch structure in SOLPS–ITER v3.0.6
 4fcbeab Attempting to fix manual build on CI server and addind specificity to some loaded modules at ITER to avoid conflicts
 bf694a7 Pointing to B2.5 submodule with corrections for CW and CCW wall loading structures
 40b46cd Attempting to have the tar examples extract to their own directories
 9bf99b5 Pointing to B2.5 submodule with bug fix by S. Carli for wall loading routines when the area of interest starts or end in the plasma wetted part of the targets
 7fb4e0a Adding triangle grid files to ITER_Be–WD+T+He+Ne example
 b6c5c59 Improvement of manual description of b2_ual_write
 3c776a5 Pointing to B2.5 submodule with fixes to chord integration routine on Eirene triangle grid
 eab571c Correctly set the default value for b2mndr.device
 b37c8a1 Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps–iter into feature/IDS
 62ba03d Fixed unintentional typo caused in previous commits.
 954306b Added section 3.21 and 5.6 to solps.tex, about b2_ual_write
 83b133a Pointing to B2.5 submodule with ITM compilation fix with AMNS.PROTO is absent
 dbffd2e Fixes for running b2_ual_write with Totalview (still does not complete)
 8224b0f Adding Release Notes document
 3e3af43 Pointing to prepared Eirene submodule for 3.0.6 release
 14b82a7 Corrected treatment of input file b2frates for Makefile call of b2_ual_write
 3e9471d Partial debugging attempt of b2_ual_write
 e6dlb10 Pointing to B2.5 submodule with merged feature/IDS branch
 dc5fe41 Pointing to B2.5 submodule with bug fix from Stefano Carli for radiation heat load calculation onto the walls
 c271b8e One misplaced file being put right

37f3ede Merge branch '3.0.6-beta' into feature/IDS
 9fe52b5 Stand alone variant of b2_ual_write now works correctly by invoking 'b2run -s b2_ual_write'
 1e02c6b Minor tweaks before final merge
 4941fb2 Adding benchmark cases to runs/examples directory and updating manual
 8f57540 Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 a3ab58a b2_ual_write now saves the output data of a b2mn run.
 adff793 Upgrade routine 'B25_process_IDS' to (optionally) work with different time slices.
 55ab517 Merge branch '3.0.6-beta' into feature/ExternalData
 c58d8d7 A disk-space saving move
 bb21e7f Merge branch '3.0.6-beta' into feature/ExternalData
 da36524 b2_ual_write now writes to IDS. It is run normally with b2run b2_ual_write.
 bcaae25 Pointing to B2.5 submodule with fixes for tracing files
 90da2b4 Uncomment code. Compile fix.
 df26c7f - Copy and rename b2_ual_write.F90 to b2_ual_write_archive.F90. - Move b2_ual_write_archive.F90 and b2_ual_write_gsl.F90 to folder 'archive'. - Move 'put_ids_edge' routine from b2_ual_write_b2mod.F90 to b2mod_ual.F90 to be used as a routine inside module. - Syntax edit of b2mod_ual.F90. - Create new b2_ual_write.F90 to be used solely for writing data to IDS, obtained by b2mn (b2run), within SOLPS-GUI.
 d45a184 When untaring update creation date.
 4b43d87 Pointing to B2.5 submodule with fixes for appending new batch data to b2time.nc files
 a5589a6 Pointing to B2.5 submodule with corrected treatment for tracing file headers
 0458013 Pointing to B2.5 submodule with fixes for stand-alone MPI compilation on the ITER CI server
 8ce435a Merge branch '3.0.6-beta' into feature/ExternalData
 f7e1b4a - Pointing to B2.5 with new use of MA28copy3 for potential equation solver and modification of example files accordingly - Clean-up of Uinp config files
 83e7e63 Fix step argument usage. In documentation edit paths '/home/ITER/penkod/solps-iter' to '/home/ITER/penkod/solps-iter'.
 d42fb52 Merge branch '3.0.6-beta' into feature/ExternalData
 862e6dc Remove help for realclean
 6f1d32f Remove ASSETDIR dependency
 e3fe27c Use plain Makefile to mimic ExternalData instead of using CMake
 0645bf3 Merge branch 'feature/ExternalData' of ssh://git.iter.org/bnd/solps-iter into feature/ExternalData
 3ea2e51 Pointing to B2.5 submodule with b2co fix when running cases without Eirene
 7d830de Further fixes for patches to backward compatibility of J.H.AT and L.H.MOL settings
 076cfbe Merge branch '3.0.6-beta' into feature/ExternalData
 09315f6 ITER tutorial files updated.
 db0b731 Removing reference to GGD submodule
 26751ab Avoiding multiple rebuilds of the manual
 4ac85ae - Fixes for H isotopic mixtures (changing dimensions for J.H.AT and L.H.MOL) - Fixes to Uinp for including non-generic species - Adding the GIT hashes of the corresponding code repositories to the SOLPS manual
 1873603 Further improvements for b2time.nc files under gfortran
 c5ba059 Merge branch '3.0.6-beta' into feature/IDS
 606ef43 Additional fixes for very large mesh cases
 89aa2f6 Updated and added example files for ITER baseline case
 8459780 Improve the 'write_quantity' routine to set and write separate plasma state data for grid subsets Core, SOL, Inner Divertor and Outer Divertor. Minor syntax edit and documentation additions.
 b0e6fe0 Corrected b2mn.dat files to hold lower b2mndr_ntim
 ee72e65 Forgot to correct b2mn.dat in case files for C-Mod InitializeRun
 70063a6 Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 bb6db57 - Improve the use of command line arguments. Add also argument for number of steps for b2mn_step() routine. - Update the documentation with the above listed changes.
 b63e7e2 Fixed data for C-mod case, for the meshing case
 9df7c52 Updated C-Mod-InitializeRun
 1ad78cb Prepared initialized run for C-mod case.
 e918fb4 Merge branch 'feature/ExternalData' of ssh://git.iter.org/bnd/solps-iter into feature/ExternalData
 18f8453 Merge branch '3.0.6-beta' into feature/ExternalData
 13ec8b8 Fixes for cases with large meshes
 dbf9e1c Updated C-mod case with additional stencil input files.
 f9b6655 Fixes for cases with large meshes
 ff2c320 Add references to SOLPS-GUI documentation and tutorial video.
 8d6b40b Rename 'write_ids' routine to 'B25_process_ids'. Minor documentation edits and additions.

7940ede Pointing to Carre submodule with more obvious error messages in b2agfz

2527215 Merging feature/input-builder branch in B2.5 submodule

9f9761d "b2_ual_write_b2mod.F90: - Implement the use of command line arguments for variables 'shot', 'run', 'username', 'device' and 'version'. - Update the documentation with the changes listed above. - Add sections 'Compiling and setting the environment' to the in-code documentation. README.md: - Fix the display of doxygen generated list of arguments. - Update section 'Running the code'. - Add main description of the code. - Add section 'Documentation' - Minor changes in section 'Compiling and setting the environment'. b2mod_ual_io_grid: - Minor code cleanup."

53eab09 - Point to B2.5 submodule with improved handling of tracing and optional files. - Point to DivGeo submodule with shotfile2dg program added to IPPITM domain. - Define totalview as default debugger for Intel compiler on Gateway. - Added overseas territories for list of allowed domains.

052cc6c Merge branch '3.0.6-beta' into feature/ExternalData

93f1c98 Debugging mtv_3d script and other related ones

cb70faa Merge branch '3.0.6-beta' of ssh://git.iter.org/bnd/solps-iter into 3.0.6-beta

ee973ac Fixed files for tutorial-DivGeo_C-mod

fe8274a Merge branch '3.0.6-beta' into feature/ExternalData

a7678a4 Fixing B2.5 compilation on the CI server without MDSplus

f00e6b6 Updated tutorial-DivGeo_C-Mod.tar.gz.md5 hash

246dd4c Removing an extraneous mdslib.inc reference

05d0cc0 - New option to append tracing and NetCDF files when continuing a run. - Improved treatment of optional run files. - Crash recovery program b2co synchronizes the tracing and NetCDF files with the recovered state file. - New "-b" option for b2co to obtain a provisional b2fstate file while the simulation is running. - Updated documentation.

fc7820a Safety for 0-timestep runs to do no balance averaging

7aa66aa Pointing to updated B2.5 and Eirene submodules

313f41d Minor code syntax edit.

af4b4cd - b2_ual_write_b2mod: Additions to documentation, comparison of CPO (B2.5 ITM) and IDS (B2.5 IDS) data structure nodes. - Other: Rename (and shorten in order to avoid possible routine name limit when compiling) some routines to better view their names in documentation (e.g. b2IMASFillGridDescription -> b2_IMAS_Fill_Grid_Desc). Doxygen, when dealing with Fortran files, does not distinguish between upper and lowercase letters.

d3fdd7d Changes to fix some SOLPS-GUI demo glitches

13a38f6 Further documentation modifications, upgrades, syntax edit, code cleanup etc.

2e2ea1d - b2mod_ual_io_grid: Use gmap% variables for allocation of 'objects_per_dimension(:). object(:)' structures instead of 'raw calculated' values. - Further documentation update, syntax edit, cone cleanup etc.

c797337 Further documentation modifications, upgrades, syntax edit, code cleanup etc.

3c3a3f9 Correcting script header by adding -f to make them running on ITM

ff90f93 Documentation modifications: - Set new documentation pages for b2_ual_write, b2_ual_write_gsl and b2_ual_write_b2mod - Split 'Program description' and 'Program contents' pages/sections - Add references to related document pages/codes

6b29d84 Documentation update, code cleanup and syntax edit.

6aa2c5f Documentation update, code cleanup and syntax edit.

221d75f - Documentation modifications and updates (Doxygen 1.8.8). - b2_ual_write and b2_ual_write_gsl: Rename 'read_b2fgmtry_b2fstate' routine to 'read_b2fgmtry'. Rename 'read_additional' routine to 'read_b2fstate'. Code cleanup.

27d1aee Modify inside code documentation to follow Doxygen 1.8.8 keywords.

c07e9c1 Adding libxml2 to ITM compilation

ffa5a73 Documentation update.

a55ee3c Merge branch '3.0.6-beta' into feature/IDS

c352594 Moving Eirene fort.75 file to fort.85 to avoid a clash with B2.5 tracing files and other tracing file treatment fixes

f90ee97 Documentation update.

40b43b2 Forgot in the previous commit.

c68e2d5 - Add Doxyfile (forgot to include in the previous commit) - Modify .F90 codes comments to follow syntax compatibility with Doxygen

20a880a Create basic template for generation of B2.5 IDS documentation using Doxygen. Start to modify .F90 codes comments to follow syntax compatibility with Doxygen.

c45a3da Remove Sphinx related files, docs and settings as sphinx-fortran proved to be insufficient.

990fba3 Delete the unnecessary file.

4316218 Upgrade documentation: - Create .rst for every .F90 file and required index.rst files. - Modify 'sources/index.rst' to follow the new layout of the source files.

dee1e9b Documentation update. README.md: - Fix display of terminal commands. - Update "Generating HTML Fortran Documentation" section. .rst and .py documentation sources: - Include the

remaining b2_ual_write and b2mod files in the documentation. .F90 files: – Minor syntax edit.
 Remove old local ggd from the modules list.

456d9b9 – Create basic template for generation of B2.5 IDS documentation with the use of Sphinx–Fortran package. – Minor syntax edit.

fdf0499 – Corrected SOLPS4–5 converter for cases with simultaneous density feedback core and SOL control (required to translate case #2360 as part of task IMAS–1246) – Fixes for optional file handling – Module corrections for pgf90 ITER environment – Pointing to B2.5 and Eirene submodules with memory leak and debugging fixes

0189201 Use b2mn_step(0) routine and write the processed data to IDS. Code cleanup and syntax edit

69c3d82 Update documentation and code cleanup.

2d1bf7f Updating to Gateway module imasenv/3.12.1 from 3.9.1

c0ealf5 Merge branch '3.0.6–beta' of ssh://git.iter.org/bnd/solps–iter into 3.0.6–beta

4a3491b File handling changes in B2.5 submodule

70b5063 Introducing 'b2tinnt_fchin_in_core' switch to improve convergence of cases with drifts

0b9969c Rename routine 'gridWriteDataVectorComponents' to 'B2gridWriteDataVectorComponents' to avoid possible GGD routine name collision in the future. Remove GGD as submodule.

fdab5f2 Fix code writing magnetic field vector 'diamagnetic'.

5b15955 Typo in previous commit

8a885ca Removal of obsolete switches in converter

09c64b5 – Upgrade 'write_cell_vector_component' and 'gridWriteDataVectorComponent' subroutines to enable writing to all FLT_ID leafs of type(ids_generic_grid_vector_components) IDS node. – Fix code writing three unit basis of the magnetic field vector (poloidal, radial and toroidal vector components).

8709c48 Merge with 3.0.6–beta.

05f701b – Remove deallocation of already set data in .ion(:) IDS node. – Include and modify 'computeCoordinateUnitVectors' subroutine to work with IDS. – Improve 'write_cell_vector_component' and 'gridWriteDataVectorComponents' subroutines. Done in a way that writing data field for other vector components will be possible in the future (work in progress).

f53ee5e Fix code that writes electron temperature, ion temperature and electric potential quantity data fields (for corresponding grid subsets) to IDS.

e81d9ec Additionally deal with cases where nrknot is already known or fort.33 is empty

8adc754 Correction for b2ag detecting format to use when writing fort.30

476726c Fix the code intended for writing Ion Parallel Velocity to edge_profiles IDS. Create new subroutines 'write_cell_vector_component' and 'gridWriteDataVectorComponents' in the process.

f962022 Fixes for installations on UNKNOWN platforms

203d337 – Fix code that writes ion density data fields (including edge_transport ion fluxes and ion source). – Start to adapt the code, used to write the remaining data fields, to IMAS IDS.

2218fd0 b2_ual_write_b2mod: – Fix code used to write set data to edge_transport and edge_sources. Note: Electron flux data fields are now temporarily written to edge_transport % model(:) % ggd(:) % electrons % energy % flux(:) until the flux related nodes are to be moved to edge_profiles IDS (IMAS 1165, point 3.). b2mod_ual_io: – Set basic data for edge_sources – Adapt 'write_cell_vector' subroutine.

13bfe57 Small corrections to model equations (provided by Serguey Voskoboinikov)

c3f2512 Staging space for shared libraries on VSC

52c8771 b2_ual_write_b2mod: – Add subroutine 'checkFileAndDelete' used to delete old b2fparam, b2mn.prt, b2fstate, b2fmovie, b2ftrace and b2ftrack files before running the 'b2mn_init' routine. b2mod_ual_io: – Fix code writing electron density data fields for Inner Midplane and Outer Midplane grid subsets to IDS. – Start to adapt the code, used to write ion density data fields, to IDS.

be888a8 Merge branch '3.0.6–beta' into feature/ExternalData

35326c7 And yet more tweaking of modules

5fe66df Another attempt at compilation fixes to deal with IMAS and Anaconda3 module conflicts

0469634 Another attempt at compilation fixes to deal with IMAS and Anaconda3 module conflicts

1db1623 Fix C–mod tutorial hash

106814a DivGeo tutorials for C–Mod and ITER baseline scenario

a43e5e2 Move the protobuf overwrite fix (made in the previous commit) to the end of the setup.csh .ITER.ifort64 script. Fix now works properly.

2191494 Makefile: – Include the required Catalyst related libraries in order to compile b2_ual_write_b2mod writer. b2_ual_write_b2mod: – Implement the use of the 'b2mod_main' module and its 'b2mn_init' routine instead of the previously used custom made routines for processing the b2 data. – Remove the now unnecessary routines, module statements, variable statements etc.

491b13c Move the protobuf overwrite fix (made in the previous commit) to the end of the setup.csh .ITER.ifort64 script. Fix now works properly.

a486163 Fix broken Anaconda3 ABI for protobuf by overwriting it with the working paraview

protobuf.
 da00c3e Fix broken Anaconda3 ABI for protobuf by overwriting it with the working paraview
 protobuf.
 0808fba Merge branch '3.0.6-beta' into senichi/bugfix/2d_plots
 536cad3 IMAS modules must be loaded first to avoid conflicts
 b68463f Merge branch '3.0.6-beta' into senichi/bugfix/2d_plots
 5b3c22d Replace the header `#!/usr/bin/env tcsh` with `#!/bin/tcsh -f` in the 2d_plot script to
 avoid problems on ITM Marconi cluster
 56f97e8 Merge branch '3.0.6-beta' into feature/python3
 3a54578 Merge branch '3.0.6-beta' into feature/IDS
 a627cf2 Compilation fixes for the ITM environment
 4cc8456 Add missing 'B2.SUBGRID_*' and 'B2.GSUBSET_*' variables. The 'B2.GSUBSET' variables are
 intended for possible future use.
 dc3d11f Rename the 'B2.GSUBSET' variables labels to 'B2.SUBGRID'.
 7a24d9f Add the missing B2.GENERIC.SUBGRID_COUNT variable under '#ifdef ITM'.
 5b02b9f Pointing to corrected IDS executables in B2.5 submodule
 a100104 Upgrade all scripts to Python3 and maintain Python2 compatibility. Commands to upgrade
 were: `sed -i -e '1a\from __future__ import print.function' *.py` `sed -i -e '/print /s/print`
`\(.*\)/print(\1)/' *.py` `sed -i -e "s/''.join/b''.join/g" *.py` `sed -i -e "/b'".join/s/strip()/`
`strip().decode('utf-8')/" *.py`
 59814f0 Pointing to B2.5 submodule after merge with feature/IDS
 d08bc4a Pointing to B2.5 submodule after merge with feature/IDS
 0a4d78d Correction for ITM g95 GGD compilation
 fef73cf Import 2013-05-01 version from <https://bitbucket.org/robertodealmeida/pupynere/> althoug
 there is SciPy maintained version at [https://github.com/scipy/scipy/blob/master/scipy/io/](https://github.com/scipy/scipy/blob/master/scipy/io/netcdf.py)
 netcdf.py but that needs scipy types.
 6093a06 b2_ual_write_b2mod: - Rename 'put_ids_edge_profiles' to 'put_ids_edge' and modify it
 to support also edge.sources and edge.transport IDSs b2mod_ual_io: - Attempt to write
 basic data to edge.sources and edge.transport. - Start to implement the writing of
 electron density data for grid subsets 18 and 1. - Fix subroutine 'write_face_vector'
 a285fbd Fixes for wall loading calculations in B2.5 and Eirene submodules
 441a513 - Fix writing of electron density data field for Cells grid subset to IDS as previously
 the writer wrote only empty arrays (zeros) to IDS. - b2_ual_write_b2mod: Include and modify '
 read_b2fplasma' routine, located in b2_ual_write.F90. - b2mod_interp: Minor syntax edit.
 c3c6f77 Merge branch '3.0.6-beta' into feature/IDS
 b4961c7 Pointing to B2.5 and DivGeo submodules with additional compilation fixes
 e976a23 Moving to have the GGD as an IMAS module and compilation fixes
 71240b6 Fix writing of electron density data field for Cells grid subset to IDS (done by fixing
 assert routines in b2mod_interp, fix and improve assert checks in '
 b2IMASTransfortDataB2toIDSGeneral' subroutine). Basic syntax edits
 f89a8d0 Pointing to updated Eirene and DivGeo submodules
 c31ad0d Merge branch '3.0.6-beta' into bugfix/read_ft44_ft46
 c06081e Trim trailing white space.
 d5009ba Adapt Matlab routines for solps-iter version 3.0.6, part 2
 73bd6f3 Adapt Matlab routines for solps-iter version 3.0.6
 3ba64f7 Update GGD branch name in solps-iter-update scripts
 da8a274 Forgot one calling instance of b2tvspa with cvsa-cl instead of cvsa
 c7083d6 - Pointing to B2.5 submodule with various changes from SPB colleagues. - Added linking to
 IMAS ITER scenario database. - Updated documentation and example b2mn.dat files.
 a0d09b6 Align GGD to release/1.6.0
 6aa701f Merge branch '3.0.6-beta' into feature/IDS
 6998a9d Merge branch '3.0.6-beta' into feature/ExternalData
 a3b8f2e Switch to HTTPS supported CMake
 28ce692 Switch to HTTPS supported CMake
 7858470 Describe ExternalData in Markdown
 61e0dec Unloading Anaconda3 module (that comes surreptitiously from IMAS and causes untold
 trouble and incompatibilities), as recommended by Leon Kos
 9919790 Adapting to the change of GGD branch to develop
 48deeea Add RPATH linking to GGD library, required by b2_ual_write_gsl, b2_ual_write_b2mod and
 other modules in order to work.
 78fafc9 Merge branch 'feature/IDS' of [ssh://git.iter.org/bnd/solps-iter](https://git.iter.org/bnd/solps-iter) into feature/IDS
 6557aa2 README.md: Fix example13 name. Example13: Add the use of the GRID.SUBSET_NODES parameter,
 located in ids_grid.common. Upgrade the example13.
 aae3fb3 Fixes for solps4-5 converter compilation within IDS branch
 6d547dd Pointing to updated GGD, Eirene and B2.5 submodules (fixing ifort debug compilation)
 34a725c Adding a -20 b2plot user option and removal of dummy additional wall in some older cases
 f7062ad Tweaking SPB output files in B2.5 submodule

015d4d9 – Matlab routines for comparing SOLPS data and JETPPFs (provided by Debasmita Samaddar) –
 Linking to DivGeo submodule with embedded X-Win solution
 93d8018 Broadcasting NRISTQ to fix an SPB bug (reported by Ilya Senichenkov)
 2a0a43b Update he b2_ual_write_gsl. Add RPATH linking to GGD library and Mark local GGD builds as
 dirty
 a72c1e9 Merge branch '3.0.6-beta' into feature/ammonia
 91fd8a3 Adding a safety in Uinp for DivGeo surfaces with no defined material
 3e396cb AMMONX database update and pointing to Eirene
 4d8af9e Set IMAS GGD namespaces to be under #ifdef IMAS preprocessor. Obsolete GGD cleanup.
 1ba2a1d Obsolete GGD fix and clanup. Update b2_ual_write_gsl.
 8b40a31 Bug fixes caught by gfortran debug run (after report by Serguey Voskoboynikov)
 cfb9a76 Rename the existing 'gridSetup2dSpace' routine to 'gridSetupUnstructured2dSpace'. Create
 new example routine 'examplePutIDS' in the source file ids_grid_example. Upgrade the
 ids_grid_example_14_2dstructured.servicelibrary with the use of 'exampleCreateIDS' and '
 examplePutIDS' routines, update the name of the 'gridSetup2dSpace' routine.
 3eb387b Fix documentation project name.
 e8cd4ba Update the imas version used for documentation generation to imas/3.11.0/ual/3.6.2.
 3598b6f Update ids_grid_example1 and ids_grid_example16. Fix the use of the ids_types (R8 and DP
 -> IDS_real).
 debcf0d Fix the use of ids_types (R8 -> IDS_real).
 b92bc4d Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 2bbe948 Rename ids_grid_example13_subsets to ids_grid_example13_gridsubsets. Update and upgrade
 the example 13. README.md update.
 313a89d Rename ids_grid_example13_subsets to ids_grid_example13_gridsubsets. Update and upgrade
 the example 13.
 578c1f8 Adding Dejan Penko's MSc thesis on IDS visualization and pointing to updated B2.5 and GGD
 submodules
 799a8c1 Change to residuals.trc file header (REQUIRES HUMAN INTERVENTION FOR CONTINUATION OF AN
 OLDER RUN) and use of dedicated routines for computation of currents in b2tfch
 bc1950b Merge IDS against 3.0.6-beta
 3782f9e Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 805c12e Align GGD build procedure with IMAS conventions
 alb4e49 b2mod_ual_io: Start to adapt the code used to write the ne, te, ti... data fields to IMAS
 IDS. Start to adapt the write_quantity and write_face_vector subroutines to IMAS IDS.
 b2mod_interp: Uncomment the functions interpolateToAllVertices and interpolateToVertices. Add
 temporary 'use ... ! IGNORE' arguments.
 842dd00 Fix and upgrade assert subroutine usage. Basic syntax edit.
 817b1f2 Start with the #ifdef IMAS section and create IMAS IDS subroutine variations of pre-
 existing ITM subroutines. Syntax edit (spacing, line length of 80 characters per line etc.).
 58e4a0f b2mod_ual_io: Continue modifying and adapting the write_ids subroutine to IMAS IDS. Start
 to modify and adapt the write_quantity subroutine to IMAS IDS. Syntax edit (spacing, line
 length of 80 characters per line etc.). b2mod_grid_mapping: Syntax edit (spacing, line length
 of 80 characters per line etc.).
 57928b3 Modify and adapt/add the findMidplaneCells, collectRadialVertexIndexList and
 collectRadialVertexIndexListSubroutine subroutines to IMAS IDS. Modify and adapt the code
 writing the last two missing grid subsets (Inner Midplane and Outer Midplane).
 c8522f8 Changing a write statement to make it compilable with gfortran
 6095d56 Syntax edit (spacing, line length of 80 characters per line etc.).
 d3395a7 Optimize handling of tracing files to prepare for making them appendable
 a808a22 Getting FILEDATA working as advertised and other fixes
 e02c31d Modify and adapt the code writing of 13 more grid subsets to IDS. Create and use
 subroutine collectIndexListForRegionSubroutine, a subroutine variation of function
 collectIndexListForRegion.
 a664b37 Condition ion-neutral current to hydrogenic species and not applying c_core_corr_dn on
 the core boundary ring (as requested by Serguey Voskoboynikov)
 ebbd906 Re-ordering variables in namelists produced by Uinp
 97c45ff Pointing to GGD modules with stand-alone compilation improvements
 91ec6b1 Compilation with new GGD module
 e6c1730 Merge branch 'feature/IDS' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 0a8cfb8 Merging with Ljublana contributions (still does not compile)
 d0382b2 Align submodules to 3.0.6-beta
 313c9f1 config.ITER.gfortran: Update the GSL library linking. b2_ual_io_grid: Use array variable
 arguments instead when calling createExplicitObjectListSingleSpace GSL routine (more GNU
 Fortran compiler friendly).
 c505d0a Set and use new variable edge_profiles=imas-obj.edge_profiles intended for better code
 interpretation. Rename few variables to more appropriate labels. Update in-code documentation

061e71b Statis ITER server now includes project name for download
 93601fc Further improvements for stochastic current treatment
 0e31bde Further updates to model equations (provided by Serguey Voskoboynikov)
 5d0a6c2 Adding FILEDATA flags to b2.user.namelist to select which tracing files are produced
 18f8a8e Update of manual equations with explicit splitting of classical and anomalous viscosity terms
 2996022 Include IMAS static server at ITER as ExternalData repository
 277fbcc Merge work done on b2mod modules, intended for writing data from B2.5 result files to IDS
 .
 e7aed99 Catalyst now requires `-std=c++11` for all C++ compilers.
 98654f7 Force sh shell for CMake. Module CMake added on ITER
 bc58e34 Merge branch '3.0.6-beta' into feature/ExternalData
 1a48f9e CMake ExternalData of for fetching large examples
 4c6016b Pointing to B2.5 submodule with b2plot fort.46 bug fixes
 71a058c New Matlab scripts from SPB and more updated AMDS Ammonia files
 c405af2 - Adding a gfortran debug trap for denormalized variables to catch passing and using of
 initialized variables as subroutine arguments - Pointing to submodules with same debug trap
 implemented - Dealing with empty tracing files to avoid errors when attempting to append to
 them - Updating Ammonia-metastables and N2 AMDS files according to new database version -
 Correcting Uinp to not double-count ground states and metastable description states -
 Allowing for an N2-only chemistry model
 3dd9b43 Build change to support GGD
 3134f2f Merge B2.5 branch '3.0.6-beta' into feature/IDS
 986eb31 Sync IMAS version with ParaView 5.4.1 plugin installed
 8b029c2 Merge branch '3.0.6-beta' into feature/IDS
 b2a4c84 AMMONX: new database version
 b32d1df Dealing with ITM environment changes
 d9f9203 AMMONX: new database version
 d152e18 Updated Ammonia database reports (provided by Sylvain Touchard)
 fb0724c Adding option for a simple N2 AMDS molecular model as well as an additive ammonia model
 b46b864 Adding new tracing data fields requested by Ilya Senichenkov and Lisa Kaveeva
 a9ffdl1d Re-ordering loading of modules in an attempt to fix ITS-74874 and removing a module
 inconsistency
 968ee05 Dealing with IMAS-1120 changing the paraview module from 5.2.0 to 5.4.1
 497a673 Uniformize the setting of b2tfnb-vis-q (spotted by Serguey Voskoboynikov)
 c12cad2 New output description file from Lisa Kaveeva
 caf250b Removing style.shei switch and equipartition term deleted in guard cells (proposed by
 Serguey Voskoboynikov)
 3a4471a B2.5 transport model tweaks from Serguey Voskoboynikov
 3387de2 Adding ion current contributions to balance
 bb1113f Adding the ion-neutral friction current to the list of currents carried by ions
 6d1a6bc Modified equations in manual to reflect new ion current contributions
 a21dd88 Patch provided by Leon Kos for use with SOLPS-GUI
 25583fa Bug fix for inclined gas puff launch surfaces within the computational grid (reported by
 Guozhang Jia)
 6396040 Pointing to B2.5 submodule with b2pwlw format fixes
 44abc41 Fix to eisum B2.5 diagnostic and some aesthetic changes
 0b97438 Conditioning output about velocity restriction (requested by Ilya Senichenkov)
 0807813 Fixes for DDN-up cases (after a bug report from Ivan Paradela-Perez)
 07546c3 Moving the optional B2.5 output to output/ folder (proposed by Serguey Voskoboynikov)
 f3a763d Removing duplicate my_outi.F routine in B2.5 (spotted by Serguey Voskoboynikov)
 68361f6 Dealing with HP default filenames
 77813ea Adding triangle volumes and cell center unit vectors to fort.46 (as requested by Ilya
 Senichenkov and Wouter Dekeyser)
 90457f5 Merge branch '3.0.6-beta' into senichi/config_SPbSTU
 4fb0904 Adding treatment of averaged solution files to loop scripts
 5e8cdec Bug fix spotted by Serguey Voskoboynikov
 be6699d Fixes to scripts from Egbert Westerhof and debugging setup.ksh
 1617147 Tiny updates to SPbSTU configuration scripts plus loop submission script for both SPbSTU
 and ITM
 7c73f5b Better treatment for slab and annular cases
 d64b86d Bug fix for isolated leg case of David Moulton's
 8dc155b Further tweaks to momentum source terms in manual
 97c5c65 Adding momentum source term for radial anomalous current (provided by Serguey
 Voskoboynikov)
 f15b0e1 Extending use of cvs_a_mltpl switch (provided by Serguey Voskoboynikov)
 95cfa7b Forgot one instance in the previous commit

d32309b Fetching and pruning GIT submodules branches when updating SOLPS-ITER
df27942 Uinp output prettifying
210f2ee – New version of SOLPS5.2 model equations document (provided by Sergey Voskoboynikov) – Documentation modified to indicate association of BCCON=14 with BCPOT=11 and BCENE/I=15
dc3a202 Output modifications from Lisa Kaveeva for stochastic current
f29998e Further usage of BoRiS switch for heating terms in 5.2 physics model and bug fix of mis-assignment of friction heating to Joule heating
ed9a85e Re-ordering b2aidr to read geometry before old b2fstate file (bug spotted by Irina Veselova)
e05d331 Adding stochastic current terms (provided by Sergey Voskoboynikov and Lisa Kaveeva)
a97373e Fix to balscript.m provided by David Moulton
32d6ace Clean-up of Balance script files and addition of MATLAB module on hpc-app1 ITER HPC node
33007bf Additional input test for KSNS array
339e382 Improved documentation and treatment of LSNS and KSNS arrays
0535c97 Adding an input consistency check between ksns and lsns arrays
994dd46 Fix from gfortran compilation
075c8e3 Removing molecules with unmapped constituents in b2yt
b9b4be4 Fixes for B2.5 standalone compilation
8fc8c17 Conditioning reading of fort.44 and fort.46 files to use_eirene switch value (suggested by Felix Reimold)
e924b3c – Further fixes to feedback schemes from SOLPS4.3 – Adding a CPU-time limit for ITER job submission scripts (default was too small)
f744c5a Pointing to B2.5 submodule with symmetrized ue computation before b2npmo call
50d859e Removing BLOCK statements (from Sergey Voskoboynikov)
8ecb848 – Adding a fort.46 save-file – Adding scripts to move from master to develop branches and back – Pointing to B2.5 submodule with feedback-namelist diagnostics – Pointing to Eirene submodule with fixes to spectral sums
2908d62 Putting back a bibtex rule for manual build
7e4eb54 Improvements to manual build when bibliography files are modified and removal of some build warnings
bdc5529 Fix to b2co auxiliary script
35c8a25 Second pass at redef_pbs bug fix
d34accf Bug fix for b2yt ion thermal reflection species and ue output in b2news..F (latter by Sergey Voskoboynikov)
b5beb10 Addressing a PGF90 compiler warning in Eirene
84f8e46 Bug fixes for redef_pbs algorithm after report from Ilya Senichenkov
00b7fcb Additional documentation for b2wdat.iout use (provided by Ilya Senichenkov and Lisa Kaveeva)
594e57b Another fix pointed out by Sergey Voskoboynikov
ba986b5 Correct config file for mpi-compilation on VSC
1ac17cb – Re-activation of energy spectra diagnostics in Eirene (with new logarithmic binning, conversion rules, and input build-up) – Corrections to 3.0.6-beta as suggested by Sergey Voskoboynikov and Stefano Carli
d533309 Correcting b2yt conversion rule for block 10F of Eirene input files
d6adeb2 Adding a missing b2yt conversion rule for ESCD1\$ spotted by Stephan Gloeggler and Felix Reimold
1a46f0c SOLPS-GUI module not available for g95
436b568 ITM-paraview module not available for g95
dd4aaf4 Replacing hard-coded values of NBDC, NNISOD, and NNCUTMAX by DEF.NBC, DEF.NISO and DEF.NCUT (as suggested by David Moulton)
a43b9dd Response to comments by D. Moulton and S. Carli for 3.0.6-beta
5834094 – Fixing order of setup instructions for Marconi Gateway – Adding dependency on style files for manual build
dd2b9d8 Adding a sometimes missing LaTeX style file
d9e0b26 Fixes for g95 compilation (Eirene side)
d9e341a Improvements for g95 compilation
7988f27 Config files for Leuven VSC cluster
c00f0c1 Pointing to B2.5 submodule with fixed redefinition of pbs
83f4552 Improving debugging for triangulation
6943028 Clean-up before re-running benchmarks
595423d – Pointing to submodules for version 3.0.6-beta after Makefile clean-up and debugging – Updated example files – Module load and pgf90 compilation fixes for runs at ITER
c5a97d6 Merging with ammonia database and metastable treatment branch
8444e22 Pointing to B2.5 and Eirene submodules after merge of ammonia database and atomic metastables treatments
2698161 Adding flux limiting factors to b2plot
6ba9727 Pointing to B2.5 submodule with conditioned d[ax][12] file writing

f448db3 Dealing with problematic style files in the manual build
 ccee660 Heating sources corresponding to new Zhdanov treatment for thermal force and friction terms
 fd8e983 Adding cflim(4) default to Uinp
 747c8c3 Adding values of new flux limiters for friction force and thermo-electric coefficient
 92404cb Adding new treatment for friction and thermal force terms from SPB
 f7c517f Version 3.0.6, new default value for cflim(3)
 9b7ec54 Introducing flux limit for thermo-electric coefficient and updating documentation
 eb441f4 Zhdanov parallel transport for electrons and revised documentation
 575875c Correcting reflection energy formula for recycled fluid neutrals after discussion with D. Tskhakaya
 bda6e9a Pointing to B2.5 submodule with new routines for electric field computation
 d0dlf0a – Corrections to Uinp config files – Matlab script library from SPB – Pointing to updated Divgeo module with new defaults – Pointing to Eirene submodule with SPB config file – Other SPB configurations
 4a24fde Merging with Eirene source averaging branch
 lbca39f Pointing to B2.5 and Eirene submodules after merge of Eirene source averaging scheme
 49749bf Pointing to Eirene submodule with LEVGEO=3 bug fix from Felix Reimold
 c37f558 Changes for adaptation to cases with metastable species (natm>nspecies and 3-product reactions)
 65b8509 Correction to momentum source b2plot labels provided by Mirko Wensing
 442435d Bug fixes found while coding for metastable atoms (second pass)
 ad908df Bug fixes found while coding for metastable atoms
 b638033 Second pass at compilation fix
 adea2c7 Fix for gfortran compilation
 a604453 Pointing to B2.5 submodule with sped-up compilation when linking to Eirene
 f9428ff Adding fort.13 and fort.15 savefiles
 0edee41 PGF90 compilation at ITER
 6e42ede PGF90 compilation at ITER
 4dl6cdc PGF90 compilation at ITER
 4fb5ded Pointing to Eirene submodule with single-pass compilation fixes
 26afc09 Fixing a faulty compilation
 0b591c3 Pointing to B2.5 submodule with new default settings for LHETRGTs
 d5flaa9 Replacing /usr/bin/tcsh with /bin/tcsh in submission scripts
 121c841 Pointing to B2.5 submodule with further compilation speed-up with linking to Eirene library only when needed
 39b64df Pointing to B2.5 submodule with B2.5 compilation speed-up without Eirene library when superfluous
 bcabd16 Pointing to B2.5 submodule with corrected typo in Makefile for single-pass compilation
 072ffd7 Pointing to B2.5 submodule with sped-up compilation with Catalyst
 cf005b2 Pointing to B2.5 with sped-up compilation for Catalyst
 1c011e6 Pointing to B2.5 submodule with sped-up compilation with Catalyst
 bb880a0 Clean-up of Triang compiler configuration files
 9a22bfd Second pass at Triang build fix
 6d8be2b Fixing compilation for Triang build
 effe6bb Adding J.NE.AT variable to b2.user.parameters file from Uinp
 c7b5675 Adding a recover_initial_files script
 6cacd93 Further fixes to triageom
 d322251 Further fixes to triageom
 d95e07e Further fixes to triageom
 1a5b4bb Track changes in B2.5 OpenMP submodule
 cfb3e67 Pointing to B2.5 submodule with NO.CDF and GGD modules compilation fixes
 2979f3b Fixes to triageom
 49769a0 Fixes to triageom
 2b64d93 Fixes to triageom
 08b502e Pointing to B2.5 submodule with cleaned-up GGD modules and NO.CDF compilation, and 5/2 formalism for total energy flux in b2mod.mwti
 6118b0d Pointing to B2.5 submodule with cleaned-up GGD modules and NO.CDF compilation
 4d295a6 Pointing to B2.5 submodule with fet 5/2 calculation and NO.NAG compilation fix
 c5440e0 Adding a statement in manual that the averaging scheme does not contribute to code stabilization
 1f83d29 – Added postprocessing tool b2ye and bug fix for pressure correction term
 9f23373 Pointing to B2.5 submodule with pressure correction bug fix
 9aff3de Pointing to B2.5 submodule with bug fix to b2upco pressure correction smoothing procedure
 b47960d Pointing to B2.5 submodule with 5/2 formulation of total energy fluxes and Eirene submodule with debugging options for BGK runs
 00b61db Pointing to updated DivGeo submodule with JET Freia config files

fe6ff95 Pointing to B2.5 submodule with fix for standalone average balance arrays
 9c4f3ee Correctly find ny independently on ncdump version
 2739b6c Pointing to B2.5 submodule with 5/2 formulation for heat fluxes to targets for temperature calculation
 2cf2581 Pointing to B2.5 submodule with changed definition of 'fht' to 5/2 formulation
 2db6306 Ensure NFILE is at least 30000 for second- and third-stage Eirene input files produced by Uinp
 f7b474f AMMONX: 3 new reactions in amd files and pointing to new Eirene submodule
 5900575 Pointing to Eirene submodule with BGK debugging
 d5c8f39 Pointing to B2.5 submodule with averaged balance arrays provided by D. Moulton
 8f4990c Pointing to B2.5 submodule with averaged balance arrays from D. Moulton
 d8ac86a Adding a recover_initial_files script
 399f125 Adding a recover_initial_files script
 523c1f5 Fixing a cut-and-paste error (spotted by Sven Wiesen)
 1d2d87e Fixing a cut-and-paste error (spotted by Sven Wiesen)
 d15a12c Handling optional files in runs Makefile
 57610e6 Pointing to B2.5 submodule with default GAMMAI value set to 1.0
 fe5f253 Changing default value of GAMMAI to 1.0, following recommendation from D. Tskhakaya
 9a0c473 Adding a 2d_plots_av script
 7d82d64 Removing error messages from runs Makefile
 7281598 Clean-up of Triang compiler configuration files
 b63e695 Second pass at Triang build fix
 80a1482 Fixing compilation for Triang build
 3c9c80f Tweaks to Makefile to fix non-repeat compilation
 145a859 Tweaking runs Makefiles and documentation, adding scripts, and updating B2.5 submodule
 2698024 Corrections to run_av script
 73eb580 Adding automation scripts and prepared input files for Eirene averaging scheme
 acee4af Using non-default NINITL for recycling strata
 ddef590 Write res.log from res_av script. Add Matlab-routine to read b2faver[e/i] files.
 80d2d60 Dealing with batch average files for b2co
 9e7bb2c Documenting res_av script
 65403b5 res_av calculates the residuals for the averaged plasma and source profiles produced during the running average phase
 4f39312 Adding handling of averaging procedure files and directories
 35002af applying a 2dt bug fix to 2dt_av
 99660e4 Adding documentation about averaging scheme
 9269f78 Adding batch-averaged quantities in b2time.nc, and a script 2dt_av to visualize them in real-time during a simulation.
 11d1020 Correcting header in b2ag.dat from Uinp
 46b8fa3 Adapting runs/Makefile to new averaging files
 43ea005 Pointing to B2.5, Eirene and DivGeo submodules for Eirene averaging schemes and changes to NINITL defaults in Uinp
 58816c6 Pointing to Eirene submodule with more obvious messages when coupling sizes mismatches occur
 bblce43 Pointing to B2.5 submodule with corrected Makefile
 b8af123 Pointing to Eirene submodule with corrected mpi_version.mk file creation instructions
 adf00c5 Adding back-up copies of fort.l[0-5] files
 668c236 Pointing to Eirene submodule with revised restart fix for cases with neutral-neutral interactions (provided by Sven Wiesen)
 357318c Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 43e4fee Adding /usr/lib64 to LD_LIBRARY_PATH at ITER
 b859b87 Pointing to B2.5 with corrected target load file formats
 08f3b68 Correcting units for wavelength_to_energy script (spotted by Mirko Wensing)
 65510ed More precise description of some NetCDF output quantities
 6ald58b Bringing back description of geometry switches in Appendix A.4 of manual
 f0593b9 Corrections to b2ag.dat from solps4-5
 b9b3043 Pointing to Eirene submodule with second pass at correction for restart effects with internal iterations
 03fd78f Was pointing to the wrong branch of B2.5
 2cd8406 Pointing to B2.5 submodule with smoothing of pressure correction on closed field lines
 76c062d Pointing to B2.5 submodule with smoothing of pressure correction on closed field lines
 9ab2b2b Adding pointers to modified submodules for same purpose of get SOLPS-ITER to run on Freia at JET
 bf48fe0 XPB wanted a compile time option moved
 548551f getting JET working again ... switch to Freia (dropping JACs); switch to ifort64 (pgi compiler dies on some of the code)
 1a7d514 Fixing a reaction product in Ammonia_metastables.amd

4a475f6 Pointing to Eirene submodule with bug fix to avoid restart effect when using neutral-neutral interactions (provided by Irina Veselova)

5231671 Pointing to Eirene submodule with bug fix to avoid restart effect when using neutral-neutral interactions (contributed by Irina Veselova)

719ddc3 Pointing to B2.5 submodule with corrected LHMOL description

2cb0150 Added description and example of b2yr.dat file in manual (as suggested by Stefano Carli)

b00412e Improving error messages about species number and dealing with gfortran block data idiosyncracies

710fe97 Pointing to fixed Carre submodule and updated NFILE settings in DivGeo

fe12d07 Getting Uinp to handle nitrogen metastables and various other corrections

bc2c973 Improving error messages about species number and dealing with gfortran block data idiosyncracies

45a7e2c Previous commit file contained a corrupted version of the pdf FILE

808c950 Including final report on Ammonia database contract

4bfd82d Allowing H.1/H.2 pairings for II type reactions and adapting default NFILE value when BGK species are present

bbe86fd Merge branch 'feature/ammonia' of ssh://git.iter.org/bnd/solps-iter into feature/ammonia

2007975 Pointing to new Eirene submodule with new AMMONX latex file (23s)

15c14b5 Second pass at bug fix for in-grid surface gas puffs (reported by Guozhang Jia)

dc211b0 Second pass at bug fix for in-grid surface gas puffs (reported by Guozhang Jia)

b5babda Fixing a bug for in-grid surface gas puffs (reported by Guozhang Jia)

2d0b8a6 Pointing to new Eirene submodule with new AMMONX latex file

a2c32de New AMMONX database files including cross section for non metastables scheme

bacc5ae Adding missing data for nitrogen-bearing compounds

3506dd4 Pointing to updated Carre and DivGeo submodules

ba0c1ef Adding BGK species in Uinp and revised ammonia database

8793fba AMMONX update: new reaction (23s) $N_2^+ + e = N + N$ to include N_2^+ dissociation for scheme without metastables

718fc8c Correction for isotope masses

572e549 Ability to import Ammonia chemistry AMDS files

a306eec Data corrections in adms files for 29-02-ZN-36-40-ZZ-21-22-33-46-50 reactions; A/B reagents inversion in adms files for 33-46-50-23-24-25-27-28-31-20-21-22 reactions

e4acdf7 New files for AMMONX database

8c6dd6f Update: all polynoms depend on eV temperature ; for constant values only one coefficient (better fits) ; double temperature axis for graphs

36eb0b8 Adding AMMONX handles for Uinp and b2yt

e507095 Adding AMMONX file to setup_baserun_eirene_links and in runs Makefiles

f5a34bc Change to MDSplus module load to follow IMAS module default

55d872d Getting AMDS to use data from the AMMONX file

ad1cbf5 Pointing to Eirene submodule with the AMMONX database

0b547d1 Adding new AMMONIAEX database report from LSPM-CNRS

6cf3f68 Added reference in documentation for the heat transfer model manual

ea3d870 Added mmanual for heat transfer model and corrections to freefer scripts

0d3688e Adding a src.local directory to Carre and bug fix to avoid infinite loops

70f5952 Australia and Kazakhstan signed MoUs with the IO

975d753 Adding IN-DA config files (provided by Anil Tyagi)

7d5fc02 Changed order in regression script to do check_b2_output before testing against residuals

fff10bb Adding a regression test script

89c1494 Further tweaks to run environments settings scripts

eeb9c75 Changes for OpenMP compilation and run environment

70d43ae Pointing to B2.5 OpenMP branch

d21eb2d Pointing to updated Eirene submodule

cla83b6 Changing default setting of NLOLDTRAN to T in Uinp (following Detlev Reiter's advice)

20e3345 Suggestion to use intel/17.0 and impi/2017 in Garching, which seem to be free of bugs with pmi-proxy, mpiexec and related things

c4998a0 update statistics on mdsplus server

93877b2 Adding additional safeties against too large number of walls in Uinp

61f4fc0 remove extra 'module load impi' which should be done by the setup.csh

2ad2821 Added NO_DIV switch for b2.transport.inputfile and changes to SPB comfig files

f13a661 Removing explicit MPI load for ITM submission scripts (after reporting by Ilya Senichenkov)

eb8d8c1 Reorganized Section 2 (Structure) of the manual and emphasized re-compilation when creating a local DIMENSIONS.F file

33a0395 Improved handling for restart files

9d37d5f Adapting to new ITER HPC modules

eb0f032 Pointing to B2.5 submodule with g95 compilation adjustments

2805fcb Pointing to B2.5 submodule with g95 handling of new modules

474f8e2 Adding a fort.44.i file for reruns and fixing typos in manual
 fd01a87 Pointing to B2.5 submodule with common blocks replaced by modules
 d87f430 Adjustments for running with gfortran
 3f0738a – Fixing a Uinp bug reported by Lisa Sytova when the number of species and strata happen to be the same – Bug fix for translation of L.SPCSRF array – Pointing to B2.5 submodule with updated documentation and conversion of longer Eirene reaction cards with 2D extrapolations – Corrections to manual descriptions for BCENE/I=15
 0867cd5 Generalizing the treatment for finding the value of SOLPSTOP to all submission scripts
 ab1c9b7 b2run modification for case where running in separate workspace but no SOLPSTOP file present
 61867b0 – Added automatic creation of BGK species by Uinp – Changed default value of TRCSRC(0) by Uinp to T (after discussion with Lisa Sytova and David Coster) – Corrected comments and documentation in B2.5 and Eirene submodules
 c16419b Forgot the SBATCH.spbstu files in the previous commit
 11d42c7 Removing stdin input for MPI jobs to avoid hang-ups (many thanks to Tamas Feher for figuring out this one!)
 37a8448 Uinp debugs and improvement from Ammonia database
 01c254d Replace use of 'pwd -L' which is not a standard Unix command
 7f65f6a Pointing to B2.5 submodule with ntim.save bug fix proposed by Wouter Dekeyser
 dd597f8 Pointing to Eirene bug fix from Irina Veselova about allocation of octree surfaces
 2f0e9e0 g95 compiler supports Fortran 2003 features
 c3ba099 Modifications for gfortran compilation with IMAS/3.9.0
 cc694bd Updating manual about compiler options and make NetCDF and MDSPLUS compiler options automatically detected
 fe536f2 Pointing to Carre submodule with tgarde array re-ordering
 4ed1854 b2plot commands files searched in scripts.local instead of data.local and added M. Baelmans PhD thesis to the documentation
 bde3ade Pointing to B2.5 with BCCON=19 and 27 fixes
 fla4e0d Pointing to B2.5 with BCCON=27 fix and new scripts for plotting momentum terms
 8b515c0 Pointing to B2.5 submodule with hbout character argument length fix
 7283e40 g95 compiler needed another change to be happy about the last commit
 ca49743 Adding links to submodules
 789ba8b added fzj config files for ifort including -ffreestanding option
 4ee0a91 added config files for fzj ifort64 compiler
 dd2262a changed default fzj compiler to ifort64
 b218e61 Pointing to B2.5 module with re-arranged feedback schemes
 ea5c26c Pointing to Eirene submodule with g95 inout argument corrections
 43b6a97 Pointing to B2.5 submodule with corrected dsp file read
 bf618dd Adding new sputtering and pumping data in fort.44 from Stefano Carli
 fc82436 Accompanying geometry/time-routines files re-organization in Eirene
 4cf017e Adding 2dvi and get.parm.bpl scripts for production of DIVIMP start-up files
 b726a60 Debugging and cleaning up b2plot user functions
 753c61c Doing Eirene submodule separately because of a local tree memory issue
 cb1e955 Reviving the b2plot user functions from SOLPS4.3
 213f468 Compiler options for ITER HPC Kepler use
 4a681c2 Dealing with g95 compilation on Marconi Gateway in preparation for Kepler actors
 51e6a9c Adding impi/5.1.3 module to setup.csh..IPPITM.ifort64 to emphasize a bug in Eirene
 6077796 Pointing to Eirene submodule with balanced MPI optimization strategy bug fix from Ilya Senichenkov and Tamas Feher
 4a8ff61 Change to MDSplus module load to follow IMAS module default
 8491cbe New ifort compiler option for Kepler actors and corrected b2run script for Marconi Gateway
 59244d2 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 705e9dc Compilation options for g95 on Marconi Gateway
 39da411 adding summary plots for htm
 c7ed63f Bugfix in heat transfer model; added scripts for plotting htm results and core internal energy balance
 ace35f8 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 5f24a33 Fix for Uinp to write the correct Hydrogen data in b2.user.parameters
 5ec0d6e Merging with changes from JET installation
 fbc5f89 Pointing to B2.5 and Eirene submodules with latest commits
 2e0aee5 Pointing to B2.5 submodule with debug warning about call to INTERFACE addressed
 078fdc5 Merging with update from changes by Jeremy Lore for central installation
 5cd837b Changes for config files at JET and code compilation with pgf90
 0f758b1 Pointing to B2.5 submodule with b2mod.mwti imax bug fix
 457fc87 Fixed recursion when calling cnvtria from triang script, which added additional local directory to DEVICE

94cdcd9 Changing the IMAS module at ITER
 720c295 Fix for Kepler actor required when compiling with Catalyst
 2c4471e Corrected file handling for b2md Makefile
 44fee2d Providing a better location for SOLPSWORK
 5edc4e7 Making solps4-5 Makefile more silent
 96e19f9 Adding a RUN_OPTIONS flag to the runs Makefiles and updating the list of directories for the B2.5 and Eirene submodules in the manual
 b717621 More changes for G95 debug runs
 85d3133 Making the dependencies build silent and fixes for simple-pass compilation for G95 on Gateway and file-handling fix in B2.5
 423d754 Further modifications for g95 compilation when neither IMAS nor ITM are defined
 1877de8 Correcting runs Makefile for new file handling in Kepler actor context
 f7c7db2 File handling compatible with Kepler actors
 e43b77d Changing the b2.boundary.parameters.stencil file to use zero-gradient instead of zero-velocity boundary conditions at the PFR, far SOL and core boundaries
 f0a5000 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 7237345 Adding sz array in b2plot
 4bfe15f Further tweaks for g95 compilation on Marconi
 6b22c40 Changes for g95 compilation on Marconi
 26e23d1 Pointing to B2.5 submodule with g95 compilation fixes
 1773ce1 Correcting the ADAS repository branch name in the SOLPS-ITER manual
 1604e7e Moving the ADAS branch from master to develop
 75bf290 Change to use AMNS-ADAS rather than SOLPS-ADAS submodule (IMAS-804)
 281aca6 Pointing to DivGeo module with ITM g95 compilation fixes
 d14fbd1 Documentation and update script accompanying the change of ADAS repository
 f01ee90 Pointing to Eirene submodule with newunit workaround
 4140002 Pointing to B2.5 submodule with fixed Makefile for single-pass compilation
 56e39a0 Pointing to Eirene submodule with ITM g95 config file
 3be6508 Adding PLVTK option from Eirene and removing some diacritic characters in comments
 85e8425 Following changes for Eirene rebasing effort
 1148ce2 Bringing in Eirene update (Sept.-Nov. 2016) from FZJ: 2D reaction interpolations and code clean-up
 bae778b Pointing to B2.5 with correction to Makefile for CI server
 0d91f19 Pointing to B2.5 submodule with Makefile correction and Carre submodule with config files for Marconi Gateway
 a8ba01e Pointing to B2.5 and Eirene submodules with changes for new modules
 2851887 Pointing to B2.5 submodule with Makefile changes for new modules
 62239b0 Pointing to B2.5 submodule with BCENI=27 bug fix from Ilya Senichenkov
 0ac5b2f restore calls to postprocessing scripts
 1692eec Updating the configuration scripts for SPBSTU cluster and ippitmsubmit submission script for TOK-P cluster
 eb18cc4 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 2693ea2 Pointing to B2.5 submodule with eirene_fixmap obsolete switch removed from converter
 6540839 Pointing to B2.5 submodule with cone0_mdf bug fix
 0405c99 Somehow the rule for b2cdcr had disappeared
 410bcea Further tweaks to the b2mwti_2dwrite documentation
 33c8a2f Added documentation for b2mwti_2dwrite outputs
 0ccab25 Pointing to B2.5 submodule with balance allocation bug fix
 5d10be9 Only building the doxygen documentation once
 ef2ffb6 Pointing to DivGeo module with new dgcut and cr2dg executables and automatic doxygen documentation build, improved documentation about centralized installation
 4a776ae DEVICE/USER change already captured in triang.conf
 d6f3f40 Updates to allow for centralized installation
 14ad52f Adding fc2k, Kepler, and xmllib modules when possible
 feb9f07 IMAS installation on Marconi, as per IMAS Code Camp
 8591eb4 Pointing to B2.5 submodule with added dependencies to b2mod_mwti
 b4b0070 Correction to description of BCMOM=15 b.c. in manual, provided by Serguey Voskoboynikov
 b1bb533 Merge branch 'feature/input-builder' of ssh://git.iter.org/bnd/solps-iter into feature/input-builder
 5c67758 Clean obsolete XSLT and XML that was moved to SOLPS-GUI and B2.5/documentation respectively
 4d70eac Adding rule for b2cdcn.F build in manual
 9130e3b Pointing to B2.5 submodule with further b2usrtrc corrections
 e415e5c Improvements for b2cdcn build from XML file
 689eb40 Pointing to Eirene submodule with corrected typos from Ilya's coding
 4235d48 Pointing to B2.5 and Eirene submodules with various bug fixes: lpfr[bt]_[io] variables, b2.user.parameters conversion, pfluxm, and NO_CDF compilation for balance arrays

4b3a522 Adding Maddison report on recycling source terms and convergence (provided by Detlev Reiter)

7c3d545 Changes to have radial and vertical coordinates in heat transfer model output for WALLDYN

22c22cc Pointing to B2.5 submodule with tweaks to surface heating model postprocessing programs from Stefano Carli

514c3b4 Pointing to B2.5 submodule with correction to 'gmake local' to avoid circular references

3ff2875 Pointing to B2.5 submodule with corrections from Jeremy Lore

48cff9d Pointing to B2.5 submodule with heat wall loading postprocessing improvements from Stefano Carli

c10f7ce Adding local B2 include directory links in solps4-5 Makefile

f66da98 Pointing to B2.5 submodule with new 2D array NetCDF write capability from Jeremy Lore

7e16d65 Added new all_nox object to main Makefile

366d62a Modification to b2sxdx due to b2mod_user_namelist bug fix

157f72f Pointing to B2.5 submodule with bug fix when b2.user.parameters is not used

b8c9fab Pointing to B2.5 submodule with bug fix for calls to b2npx

7e6d66c Clean obsolete XSLT and XML that was moved to SOLPS-GUI and B2.5/documentation respectively

0e7a835 Merge branch 'develop' into feature/input-builder

5aaa32b Sync to latest B2.5

e94c80a Re-arranging scripts list in manual

66e442a Updates for heat transfer model

bd34645 Finalizing heat transfer model

aa20483 Adding rule for b2cdcn.F build in manual

e07aebb Pointing to B2.5 submodule with optimization bug fix in b2ytdr

e54968b Update config files for compilation with catalyst

fe77566 Allowing for DEVICE to be preset by SOLPS-GUI

19937c7 Adding a Makefile.local option for manual build

d3075aa Clarifications of some boundary conditions descriptions following suggestions by Felix Reimold

8bb6ad3 Adapting solps4-5 compiler to new b2mod_mwti.F90 module

918c03d Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

2c39c57 Pointing to B2.5 submodule with NetCDF changes from Jeremy Lore

7fa2b0f Updated config files to remove DNETCDF4.3 precompiler flag, no longer required after update to b2mwti

1ed42aa Changing the Paraview error message in setup.csh.* files

86499ef Linked to balance update. All components of b2stbr accounted for in scripts. Labels updated for tex interpreter for use in earlier matlab versions.

24cb35d Adding a safety for when LD_LIBRARY_PATH starts out empty

f23c63e Pointing to B2.5 submodule with wldsp_res array read bug fix

c0d15e4 Pointing to B2.5 submodule with species name added to 'eavg'

15956f2 Pointing to B2.5 submodule with new volume average option for Eirene data in b2plot

930b438 Pointing to B2.5 and Eirene submodules with proper transmission of EIRENE_STEP.CPU

51be469 Adding a safety against faulty builds of b2cdci.F that compiles!

15bbfcc Adding a safety against faulty builds of b2cdci.F

73416ee Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

3a658c9 adjusting some submission scripts like QSUB.* for myself

97f7ce4 Move LD.CATALYST to setup.csh

84498fc Patches provided by Leon Kos and generalized

c446a4b another update for Ilya

6f560c5 Pointing to Carre submodule with final fixed version of Makefile

70afb5f Pointing to Carre submodule with yet another fix to Makefile

91280f2 - Pointing to B2.5 submodule with 1-D cases fixes - Pointing to Carre submodule with improved Makefile treatment of graphics/nox compilation

d398c48 Pointing to Carre submodule with improved compilation handling of switching between graphics and nox executables

9164bee Pointing to Carre submodule with corrected 'nox' compilation flag

c1e2734 Adding missing documentation and correction to python script for reading CDF files from David Coster

6277514 job submission for IPP gateway

92dbc40 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into dpc/feature/IPP-ITM

7afb647 Pointing to B2.5 and Eirene submodules with neutral fluxes for ion-neutral friction current computation (provided by Ilya Senichenkov)

24afc59 Documenting the IPPITM and Marconi ITM Gateway domain names

b7ff29d Bringing back ability to specify the number of processors for Marconi device

b407466 Was pointing to a local branch for Eirene

0fa8c48 Pointing to all submodules with new IPPITM config files

b6fafb7 Recreating the ITM environment on the IPP Gateway as IPPITM

4492fe7 Pointing to B2.5 submodule with re-applied removal of x05baf
 9ac6050 Adding wldsptr.res array in fort.44
 a038848 Recreating the ITM environment on the IPP Gateway as IPPITM
 9ed5d28 Pointing to B2.5 submodule with b2sihs source balance bug fix
 7593420 We were pointing to a local copy of the B2.5 submodule
 5166bc0 Merge branch 'feature/balance' of ssh://git.iter.org/bnd/solps-iter into feature/balance
 c305b3b Pointing to B2.5 submodule for feature/balance after rebase
 c6896c4 Pointing to B2.5 submodule with better b2cdci.F build-up instructions
 1584138 Clean-up before rebase
 67b6b17 Pointing to B2.5 for feature/balance pull request
 e151737 small modification to balscript.m
 edf2efe Link to new b2.5 balance routines which store balance arrays in balance.nc. Also modified scripts to read balance.nc file
 a566555 Adding README file suggested by David Moulton (author of Balance scripts)
 125c67c Balance scripts included in scripts/Balance
 f045902 Working particle balance version. Ready to divide EIRENE sources.
 2e064ff Pointing to B2.5 for feature/balance pull request
 4480381 Fixes for species removal in b2yt (reported by Stephan Gloeggler)
 de2d354 Linked to b25 fix for balance allocation
 f4a8e70 small modification to balscript.m
 97edce0 Merge branch 'feature/balance' of ssh://git.iter.org/bnd/solps-iter into feature/balance
 891a111 Pointing to merged version of B25
 8a08f3a Link to new b2.5 balance routines which store balance arrays in balance.nc. Also modified scripts to read balance.nc file
 a374ec5 Pointing to Eirene submodule with fixed work distribution table formats
 e74df69 Allowing for gas puff sources that are not contour walls in Uinp
 35fba23 Adding some missing units to the data description of fort.44 in manual
 839005e Pointing to Carre submodule with moved git include format
 d13df83 Fix to print_versions because of change in Carre GIT version include file
 3d138b3 Searching the log file deeper for nnreg values
 623ce10 Pointing to Eirene submodule with corrected volumes for srcml and edissml
 363adb3 Pointing to B2.5 submodule with merged input builder file
 4420347 - Listing of variables for fort.44 and fort.46 files in manual - Fixes to energy_balance_coupled script after enquiries from Fabio Subba
 33c6b4a Pointing to B2.5 submodule with added factor of 5/3 multiplying cvla for fna_he computation
 0fa0bc4 Merge branch 'feature/input-builder' of ssh://git.iter.org/bnd/solps-iter into feature/input-builder
 86be600 Move b2cdci.F generation into B2.5 submodule
 c17de5f Restructure solps-input.xml to follow b2cdci.F ordering
 7291d26 Generate b2cdci.F from XML
 957ceae Generate tooltips.py for solpsinput.py
 eb0029f Create basic HTML help for input parameters
 6270e9a Using XML file for building b2cdci in manual
 42af92b Move b2cdci.F generation into B2.5 submodule
 26ef0fe Restructure solps-input.xml to follow b2cdci.F ordering
 e8ad5c6 Generate b2cdci.F from XML
 531c05e Refactored documentation .xslt templates.
 8b057fc Fix path for XSLT
 6eb830a Create tuples of equal lengths
 7416428 Add names to parametergroups for hierarchical menus
 db1f20b Add b2mndr.b2time
 b37bb36 Handle parameter groups translation into tuples
 a26ecee b2mn.dat switch all listed and finished - 28.9.2016
 be35582 Update 1, 28.9.2016
 113eaf2 Update 27.9.2016
 2c2e1b3 Update 26.9.2016
 8625209 Generate tooltips.py for solpsinput.py
 fb0116d Fix unicode chars
 caacfe5 Update 23.7.2016
 3571871 Output parameters in LaTeX
 08d80ff Create basic HTML help for input parameters
 19aff0a Updated with more parameters
 efaafc5 Added all of the parameters/switches from the basic b2mn.dat file
 2d1d05c Initial builder parameter description
 894113c Merge branch 'g2dpc/feature/new-gateway' of ssh://git.iter.org/bnd/solps-iter into g2dpc/feature/new-gateway

968a027 new EUROfusion gateway
 99aad47 new EUROfusion gateway
 f719afc new EUROfusion gateway
 6166b07 new EUROfusion gateway
 1dfa2f0 additional scripts
 47c4392 Revert "new EUROfusion gateway"
 5910456 Pickup the new EUROfusion gateway changes
 78e33ec new EUROfusion gateway
 0a40ale new EUROfusion gateway
 5ab5f0e additional scripts
 fc9b840 Merge branch 'feature/input-builder' of ssh://git.iter.org/bnd/solps-iter into feature/
 input-builder
 92cbae3 Using XML file for building b2cdci in manual
 87f503a Move b2cdci.F generation into B2.5 submodule
 fbc7faf Restructure solps-input.xml to follow b2cdci.F ordering
 ca0c60a Generate b2cdci.F from XML
 57941df Refactored documentation .xslt templates.
 5534d5c Fix path for XSLT
 cb842c0 Create tuples of equal lengths
 9f65332 Add names to parametergroups for hierarchical menus
 a00d438 Add b2mndr.b2time
 541da26 Handle parameter groups translation into tuples
 8400511 b2mn.dat switch all listed and finished – 28.9.2016
 587677d Update 1, 28.9.2016
 83cd261 Update 27.9.2016
 ad7dfc9 Update 26.9.2016
 13f2c4d Generate tooltips.py for solpsinput.py
 def0b66 Fix unicode chars
 46027c9 Update 23.7.2016
 ffe09c8 Output parameters in LaTeX
 4a4d84b Create basic HTML help for input parameters
 7de56e1 Updated with more parameters
 c16f96c Added all of the parameters/switches from the basic b2mn.dat file
 2c90910 Initial builder parameter description
 27e49d2 Using XML file for building b2cdci in manual
 c1d2f07 Pointing to B2.5 submodule with corrected jxi documentation
 719a04c Lifting the ambiguity in the manual about volume integration in derivation of energy
 sources for Eirene coupling
 2e4f484 Move b2cdci.F generation into B2.5 submodule
 e7b3eda Restructure solps-input.xml to follow b2cdci.F ordering
 0d26dbf Pointing to B2.5 submodule with bug fixes for ntim.save.eq.0 and 1-D slab cases
 11c6cf9 Manual update on Delta_t study
 a175c5f Pointing to B2.5 submodule with bug fixes for ntim.save.eq.0 and 1-D slab cases
 97fd33e Manual update on Delta_t study
 6cc06f2 A better solution for linking to AMJUEL automatically
 7ceb023 A better solution for linking to AMJUEL automatically
 499d84e Generate b2cdci.F from XML
 02d39f5 Added links to Eirene database files for pre-loading H-alpha reactions for stand-alone
 runs
 6124dbd Added links to Eirene database files for pre-loading H-alpha reactions for stand-alone
 runs
 df141af Pointing to B2.5 submodule with added links to Eirene modules for B2.5 compilation
 6b9ec0a Pointing to B2.5 submodule with added links to Eirene modules for B2.5 compilation
 b096624 – Pre-loading H-alpha reactions from AMJUEL for coupled executables not running with
 Eirene – Adding chords example files from manual
 0f6f2a7 – Pre-loading H-alpha reactions from AMJUEL for coupled executables not running with
 Eirene – Adding chords example files from manual
 81ec42f Pointing to B2.5 submodule with iargc bug fix for pgfortran
 b391b44 Merge branch 'develop' into feature/input-builder
 0c6bb6d Refactored documentation .xslt templates.
 fa97541 Adding new program for plotting rates in AMDS using existing b2plot routines
 cf9c7c5 Ignoring an AMDS auxiliary compilation file
 10db97b Adding new program for plotting rates in AMDS using existing b2plot routines
 5c3f323 Ignoring an AMDS auxiliary compilation file
 f314b0e Pointing to B2.5 submodule with converter fixes for new geometry quantities and surface
 group data from SOLPS4
 6040994 Pointing to B2.5 submodule with converter fixes for new geometry quantities and surface

group data from SOLPS4
 c08487b Pointing to B2.5 submodule with spect bug fix from Jeremy Lore
 1986ef0 update ORNL setup file
 793b9e5 Fixing clean for amds dependencies
 8373cbc update ORNL setup file
 d0ad846 Fixing clean for amds dependencies
 9e86afa Adding README file suggested by David Moulton (author of Balance scripts)
 38b654e Balance scripts included in scripts/Balance
 f88ebd1 Working particle balance version. Ready to divide EIRENE sources.
 flclbf6 Better treatment of default additional reactions for H-alpha calculation
 5d2787d Pointing to B2.5 submodule with zeroed out unused ADAS arrays
 0728d70 Dealing with consequences of default H-alpha calculation in Eirene and fixes to double-
 null array size checks
 62b2aec Adding fort.46 file to run archive files
 4255d47 Module acrobatics for gfortran compilation at ITER
 ac357de Lots of fixes for Korea Training session
 5a9035e Various bug fixes from Korea Training Workshop
 426cc13 Adaptation of modules for ITER IMAS environment
 8e82ae4 Requesting more memory for ITER queue jobs and pointing to B2.5 submodule with my_intcor
 fix
 0860026 Avoiding name clash of smin for solps4-5 converter
 b546feb Debugs to converter and manual discovered during Korea Training Session
 6d78bfe Pointing to B2.5 submodule with cflags bounds bug fix
 e60a337 Pointing to B2.5 submodule with bug fix from a report by David Moulton
 333ca08 Correcting a bug discovered during Korea training session
 c214d6e Compilation without NAG for ITER gfortran compiler
 5c3eeae Change divgeo module pointer to ITER-develop
 8cf4d97 Adding README file suggested by David Moulton (author of Balance scripts)
 fc269e6 Balance scripts included in scripts/Balance
 8990031 Working particle balance version. Ready to divide EIRENE sources.
 2555f2f An addition to the previous commit
 33240cd Adding automated debug options when SOLPS_DEBUG is on
 1f8eb2c Pointing to submodules with automated debug settings
 285bc87 Fix path for XSLT
 a7a8167 Pointing to B2.5 submodule with fixes to b2ag for new quantities from feature/IDS update
 521b88c updated paraview libraries
 8b0c50d Same as previous commit but for setup.ksh
 242c7b5 Definition of PYTHONPATH must occur prior to SETUP/setup.xxx call when not predefined
 77b1bc5 Pointing to B2.5 submodule with corrected 5/2 energy fluxes to targets in post-processing
 086e6a7 Pointing to B2.5 submodule with CXXFLAGS defined for all config files
 22e01c0 Fixes to Uinp config.ITER.gfortran and adding b2sign function in B2.5 submodule
 dce3647 Pointing to B2.5 and DivGeo modules with bug fixes reported from users
 b1613b7 MPI compilation for gfortran at ITER
 09f4055 Pointing to B2.5 submodule with non-MPI namelist read error handling
 b71034b Pointing to B2.5 submodule with vertex array bounds fix
 a82895d Exporting PARAVIEW_LIB variable to Makefiles
 09224d9 Adjusting list of loaded modules at ITER
 be0012a Adaptations for gfortran compilation, PATH settings and setup files cleanup
 b03850a Adjust for the ParaView 5.2 and new NCAR/6.3.0 modules
 29be8da Adjust for the ParaView 5.2 and new NCAR/6.3.0 modules
 d581338 Update all setup and config scripts with Catalyst and IMAS autodetect
 c9bf392 Setup for Catalyst on ITER
 1c6df8e Getting b2sxdx to compile
 875206b Fix ITM config typo
 2671a24 Update ITER config for Catalyst installed locally
 c14741e Update ITM config for Catalyst and IMAS as optional components that are built if
 available
 376df36 Update ITM config for IMAS and Catalyst
 c980706 Fixes to IDS branch compilation
 9c56aaf Set variables needed for .cxx compilation in modules/B2.5.
 eb488ea Add LIBRARY_PATH to QSUB.iter
 467792c Correct Catalyst library paths in config.ITER.ifort64.
 eccd5e7 Remove -c from b2run arguments and update submodule B2.5.
 0a6f03a b2run now takes argument -c in order to run ParaView Catalyst.
 e5124b3 Update config.ITER.ifort64 and setup.csh.ITER.ifort64 with libraries necessary for
 linking with ParaView Catalyst.
 e386da8 Add variables for catalyst and paraview to config.ITER.ifort64 and change python version

to 2.7/11 in setup.csh.ITER.ifort64.

e4ca283 Progress towards GGD data read/write from IDS

3550f50 – Pointing to B2.5 submodule with some routine imports for IDS grid write-up – Corrected ids_get into read_ids as requested by Simon Pinches – Added a –DITM compiler flag

8d64355 Python script for reading edge-profiles ids

834a090 Compiling with links to IMAS libraries

c6f819b Pointing to B2.5 submodule with specIng bug fix

0a77ee6 Manual section labels suggested from 17th User Forum

18c14a1 Pointing to Carre submodule with improvements for gfortran compilation

866a728 Pointing to Eirene submodule with MPI gfortran compilation improvements

c296e06 – Corrected V[XYZ]DEN[AMI] to be momentum densities – Corrected PATH modification for MPI and DEBUG switches – Updated manual for new neutral quantities

90136f8 Change names of some plotting routines to avoid problems with case-insensitive file systems

13clf97 – Reducing the tolerance for colinearity check to find corners in triageom – Correcting toroidal approximation setting in DivGeo/Uinp – Handling of local links to DivGeo files for Uinp

8898f0f Do not conflate 'decay length' and 'leakage' when describing boundary conditions

69628a9 Renaming two prohibited switches in b2mn.dat.drifts (as spotted by YiPing Chen)

2bfb5e0 Modified sonnet makefile to ignore error if paths change

07be07d Further tweaks to the documentation about boundary conditions after conferring with Ilya Senichenkov, and adding linking to ioflush for Uinp compilation as requested by Jeremy Lore

cd9b5ff Manual changes to reflect update of sign convention for BCCON=14

3f5d1c0 Pointing to B2.5 submodule with bracketed Eirene routine call

42033fe Pointing to B2.5 submodule with b2yt format correction for Eirene.96 files

2058a63 Modified converter script for clean re-runs and changed b2tqca setting to match model-consistency update

dbef803 Allowing for 100+ species in b2sxdx

be85c93 Getting the converter to create correct b2ag.dat and b2ai.dat files, avoiding double (de) allocation of NMASSA and NCHARA

d2f11a2 Pointing to B2.5 submodule after rebase for model consistency changes from Wouter Dekeyser

576a6fc Pointing to B2.5 module with correction to Eirene ion energy sources

7d10457 Pointing to B2.5 and Eirene submodules with GIT hash information added to fort.44 and fort.46 files and documentation about surface group variables and Eirene stack plotting commands

7cdc153 Pointing to B2.5 submodule with triangle geometrical data fields added to b2plot

67e2cac Pointing to B2.5 submodule with further fixes to Eirene stack data plots

b3e8d54 Pointing to debugged submodules of plotting Eirene data in b2plot

3961197 Documenting b2plot ability to plot Eirene data on the triangular grid

743661f Pointing to MPI bugfix from Tamas Feher and corrected b2mn.dat.drifts example file

fb16e24 Compiling solps4–5 converter with new fields from model-consistency branch

1c63eeb Grabbing local changes before rebase

11fb61f Set of tools for postprocessing SOLPS–ITER results with Matlab

ce8e746 Pointing to modules with updates for internal consistency; updates to documentation and example files consistent with code updates

0a1b56b Pointing to B2.5 submodule with tracing routines bug fixes from Sven Wiesen

dc40ac1 Pointing to Eirene submodule with improved IMAS–619 bug fix

3a06bbd Pointing to B2.5 submodule with BCENI=11 bug fix from Ilya Senichenkov

70b0d9b Ukraine joins EUROfusion

0b210c7 Pointing to B2.5 submodule with BCPOT=11 bug fix from Ilya Senichenkov

69b9e8b Pointing to Eirene submodule with attempted fix for issue IMAS–619

a65e6b6 Pointing to B2.5 submodule with de-bundling b2yt

33cab6f Pointing to B2.5 submodule with b2yt bundling species

11b7500 Modifications to QSUB.sge* files as requested by IPP

9466791 Adding default fluid neutral boundary conditions in Uinp

9ef31e8 New multi-species particle conservation script for coupled runs provided by Stefano Carli

c3739b7 Pointing to Eirene submodule with bugfix from Tamas Feher

4b9c74c Correcting typos in sgesubmit spotted by Stephan Gloeggler

577cd92 Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop

1ce311d Correcting typos in sgesubmit spotted by Stephan Gloeggler

3179384 modified wall loading routines using fhe_/fhi_eir

eeeaf23 Additional changes to get correct molecular ion dissociation reaction cards

bd25135 Adding a b2mn.dat.drifts example stencil file

50b1de3 Indicating that the GR library is now hosted on the IO Git Repository

0e97841 Making Uinp compatible with bundled charge state inputs

18d6f97 Fixes to Uinp for new molecular dissociation reaction

68be8e4 Pointing to Eirene submodule with LD_NAG fix in Makefile

845621b – Added dependencies on the config files to the Makefiles (also in submodules) – Debugged some older b2plot command files – Fixes to "import_files_from_convert" script for conversion changing the species order – Fixes to runs Makefile for runs with debugger – Updated b2yt converter section in manual

0a97911 Removing a prohibited switch from b2mn.dat examples from manual

3ecae37 Pointing to B2.5 submodule with b2yt bug fix

4d5ef29 Adding safeties if DivGeo model is too big for Uinp

debb0ad Adding mention of the SOLPS-ITER Materials IDM folder in manual and welcome message

a1657db Fixing typo in quotes for itmsubmit

19a65bf Various fixes from 3rd day of SOLPS-ITER EU Training session: – Adding a variable job name to the job submission scripts – Fixes to Uinp for produced bundled format b2ah.dat.stencil and corrected dissociation reaction of D2+ pointed out by Felix Reimold – Corrections to manual – Pointing to B2.5 with reaction conversion fixes

e679973 Adding NAG module load for shotfile2dg compilation on IPP and ITM domains

920a0a9 Fixes for debugging slab cases from David Moulton

6c57573 missing ctrans

349828f Fixing a bug reported by Felix Reimold

b197e67 Adding SOLPS-GUI module load to ITM Gateway setup.csh files

0534be3 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

4c131d3 Debugs coming from EU SOLPS-ITER Training session and adding Eirene submission script

e5a1b7e Correction to the b2mn.dat.5.2 example file

64c011d The bad node at ITER has changed number

ea9c1f3 Pointing to B2.5 submodule with MPI-compatible error handling of faulty namelist reads

0e59788 Additional gfortran Makefile tweaks for Triang

5549363 Adding a safety in the Makefiles to grab SOLPS-ITER environment settings if compiling from local directory instead of SOLPSTOP (provided by Sven Wiesen)

80bc3c9 Another bug fix in b2sxdx

aabadbd Create tuples of equal lengths

dcb2281 Additional correction for plasma species strata conversion

e9a5f8d Add names to parametergroups for hierarchical menus

1eeal5c Getting species_start and species_end to convert properly

a4e5452 Debugging of solps4-5 converter and gfortran runs

70elec5 Adding a safety in runs/Makefile to get fort.3[3-5] files from baserun or TRIANG directories if they do not exist in the local run directory already

61d4cf3 Pointing to B2.5 submodule with fixes to b2usrtrc

4e82d2a Fixes for conversion from SOLPS4

5aldee7 Pointing to B2.5 submodule with removed duplicate feedback namelist write-up

9682ad7 Adding the B2.5 and Eirene corresponding submodules and rebasing

1dd161f gfortran b2sxdx compilation changes

5e7ca71 Compiler configuration, environment setup and launching script updates for MEPHI cluster

e6bf704 Pointing to DivGeo submodule with cleaned-up topology files

c332176 Various changes for gfortran compilation, nox compilation and solps4-5 compilation

e7149d2 Add b2mndr.b2time

0089269 Adding comment in documentation that b2.transport.parameters does not apply to fluid neutral species

3559695 Avoiding a bad MPI node on the ITER HPC cluster

07903c9 Handle parameter groups translation into tuples

fbf567e Adding safeties against empty files in scripts

43303b2 Another correction to localsubmit submitted by Leon Kos

505c0f4 Pointing to Eirene submodule with MPI output file redirection and B2.5 submodules gfortran corrections

a2aee34 Fixes for gfortran compilation of b2sxdx at MEPHI

59c849f Additional corrections to the submission scripts spotted by Leon

b3cdb98 Fix = comparison for /bin/sh interpreter

0e4727c Patch provided by Leon Kos for use with SOLPS-GUI

ee87329 Pointing all submodules with new MEPHI config files

c62cdec MEPHI configuration files, revised

8alc371 MEPHI configuration

7fbf440 Getting the (P)lot option in triang to work

950eafc Fixes for ADAS element names

591c59f Adding a default gfortran config file for AMDS

a7addf0 Pointing to B2.5 modules with changes for dealing with tungsten and neutral-neutral interactions

7dbf570 Adding the print-out of a b2ah.dat.stencil file by Uinp

c3331f2 Dealing with Eirene MPI output.* files

21e5185 Safety against absent LD_LIBRARY_PATH

47cc215 Fixes to the Eirene dependencies file
 8db2962 Correcting to MPI scripts submission
 e5762c6 Pointing to B2.5 submodule with fixes to plot zones in plasma-wetted areas
 242a0f5 More fixes for new Eirene MPI strategies
 248f6b8 Adding a manual description for choosing the MPI parallelization strategy for Eirene
 0563b7c Adjusting the Makefile tags for included files
 ae250d6 Adding a Makefile style tag to config files
 7c5e7e2 Removing cause of multiple rounds of 'gmake depend' in solps-iter.update: putting the burden on the committer
 4f68340 Pointing to Eirene submodule with added F90 tag
 a411c4d Adapting ITM setup.csh files to new environment
 8ee70e Pointing to Eirene submodule with new MPI balancing strategies
 1eaacf4 Removing b2plot.commands file from repository as it is now automatically built
 631e683 Correcting a typo from the previous commit
 6cd8fdc Improvements to triang script and pointing to Eirene submodule with cleaned-up code wrt string mismatches
 f7cd280 Pointing to B2.5 submodule with string manipulation fixes
 5ed9cf0 Pointing to B2.5 submodule with bug fix for chord evaluation
 76b1366 Pointing to B2.5 submodules with new features provided by SPB: – Rescaling of fluxes to match integral across boundaries for modified and original fluxes – Corrections to computation of fluxes for Eirene coupling – New boundary conditions (BCCON=25,26,27 and BCENI/E=26,27) and applying BCCON=19 to highest ionisation stages – Updated documentation
 6229ded Correcting an error in the momentum equation in Appendix B of the manual
 1b113ae Updated documentation
 612eed8 New submission scripts from SPBSTU
 ce6beb5 Manual modifications for new SPB boundary conditions
 cf4da41 Pointing to B2.5 submodule with a bug fix
 34e99c4 Pointing to fixes in the B2.5 and Eirene submodules from the previous commits
 27c4bf5 – Added gfortran patch for reading missing formatted logicals in Eirene input file – Patch to triageom for very elongated cells – New "Inquire" function in triang script
 9136310 Resolving a module conflict in ITER setup.csh files
 ae38a7e – Adding creation of post-processing input files in SOLPS4–5 converter – Pointing to B2.5 submodule with cleaned-up post-processing programs – Updated documentation in manual about post-processing programs
 c81cec8 updated postprocessing routines for HT model
 b406e6d Adding help messages to b2run, b2plot and b2sxdx
 bae8247 Fixes to internal.energy.balance.coupled script
 c40480e Adding lnblnk routine to Uinp compilation
 7380d4f Fixes to energy balance scripts
 c0c0b3d Added documentation about specific SOLPS–ITER coupling blocks in Eirene input files
 9c2e8c3 Not using recombination reactions to determine molecular composition in Uinp
 d7dc753 Using electron–impact reactions to build molecule composition array
 72e3eee Adding mlmp field to b2.neutrals.parameters.stencil produced by Uinp
 c8d973b Fixes to Eirene sources and added ADAS file handle in Uinp
 0939f5b Replacing energy.analysis by energy.analysis.total in summarize-run
 589d61f Corrected scripts to allow for missing NREG file and updated documentation
 b14193e Getting the species labels from the log file if SOLPS.FLUIDS is not present
 4df23f3 Pointing to Eirene submodule preparing ITER N example
 4481ab2 Clean-up of b2sxdx and creation of a valid b2ag.dat and a dummy b2ag.prt file upon conversion from SOLPS4.x
 6c067df Pointing to Carre submodule with added safety proposed by David Elder
 7a06045 – Added new Uinp capability to treat all elements with default ADAS ionization/ recombination reaction data – Pointing to B2.5 submodule with new defaults dimensions for a H +D+T+He+Be+C case and b2stbr tfiles reactivated – Pointing to Eirene submodule with corrected ADAS file content explanation – Added printout of lkindi, lkindm, nhst?, lmolscl and lionscl variables from Uinp – Corrected species.start, species.end, and lstrasc1 Uinp output – Added DBG debug compilation option for Uinp – Fixes to Uinp compilation – Added call chain for Uinp when fatal errors occur – Prettified Uinp output – Corrected b2plot manual description of smag (spotted by Felix Reimold) – Added AMDS IPP.ifort64 config file
 a0fced8 b2mn.dat switch all listed and finished – 28.9.2016
 37df504 Update 1, 28.9.2016
 dfbfcfe Update 27.9.2016
 2319df7 Pointing to B2.5 submodule with bug fix in b2tinnt from Ilya Senichenkov
 2401cf1 Update 26.9.2016
 d9c7216 Generate tooltips.py for solpsinput.py
 3d64c9a Fix unicode chars
 014dfe4 Merge branch 'feature/input-builder' of ssh://git.iter.org/bnd/solps-iter into feature/

input-builder
 5a6fb85 Update 23.7.2016
 e5a9610 Forgot to add the link to new B2.5 submodule in previous commit
 0032624 Tweaks to the Sonnet-light Makefiles
 aa7c0fd Changes to allow compilation on Macs (provided by Eric Meier)
 fe0a864 Updated documentation for colour tables
 2f463f3 Output parameters in LaTeX
 5941141 Create basic HTML help for input parameters
 b4be897 Corrected typo in variable
 ede9a33 Tweaks to the heat transfer model documentation
 13e013a updated documentation on RPs post processing routine
 0a82c24 added routines and scripts for heat transfer model, modified documentation and Makefile.
 plot
 8d5a1bc added two scripts and routines for postprocessing needed by heat transfer model
 00970e5 Pointing to B2.5 submodule with typo correction from Jeremy Lore
 06ada15 added ORNL config file for amds gfortran
 4ace782 – Pointing to Eirene submodule with correction of fort.44 array sizes for B2 integrated
 tallies – Pointing to B2.5 submodule with corrections to neoclassical treatments, 9-pt
 stencil bug fix, cell-centered pressure correction term, ITM-AMNS support – Clean-up of case
 conversion scripts
 0795bd0 Setting .eps files as binary
 1847826 Adding documentation about new palettes files provided by Jeremy Lore
 7079890 Added color tables
 f277f29 Clean-up of AMDS Makefile
 f4a6c06 Totalview now available as default debugger at ITER
 14b4876 Pointing to B2.5 submodule with b2fplasmf.xxxx files from Jeremy Lore
 b913770 Pointing to B2.5 submodule with corrected b2_chem-sput description and conversion
 318ceb5 Clean-up of save_mds procedure
 29d9a2f Adding fort.30 to setup_baserun_eirene_links script
 c5e78ac Correction for conversion of core feedback schemes
 d1c3362 Pointing to B2.5 submodule with fixes for shadowing structure
 1263752 Updated documentation and fixed missing line returns
 d284cc8 Adding xcb library for ITM Gateway compilation
 91a0412 Various fixes for Eric Meier's runs of DIII-D with drifts
 5da3f5c Pointing to B2.5 submodule with corrected boundary conditions for parallel velocity sign
 convention change
 da89393 Changes towards version 3.0.5 with new quantities in restart files
 62300bb Addition of vecrb and vedia arrays to b2fstate files, conversion, and b2plot
 5a5dedd Changes towards version 3.0.5 with new quantities in restart files
 13e88af Adding fluxes b2plot script
 10c9d6c Adding fluxes b2plot script
 9ac6eb2 Updated documentation for input builder, better treatment of optional Eirene files,
 remote debugging
 2d292c6 Updated with more parameters
 5866ac3 Fixes to IDS branch compilation
 2332456 Updated documentation with new SOLPS-ITER reference papers
 a82e19c Pointing to B2.5 submodule with b2ytdr recombination stratum conversion fix
 147b8e2 Added all of the parameters/switches from the basic b2mn.dat file
 3df28d5 Set variables needed for .cxx compilation in modules/B2.5.
 2b135f3 Update submodule B2.5.
 1579981 Merging with develop branch.
 9f8c228 Adding examples section to manual and point to B2.5 7-point stencil fix
 0361412 Dealing with additional grid files when converting
 61392a4 Dealing with converters with no triangular grid in Eirene
 1a6e160 Handling conversion from SOLPS4.x with Eirene.96 input files
 993ec6b Initial builder parameter description
 c92b384 Further improvement to the NSTRA post-couple check
 6ab9b01 Add LIBRARY_PATH to QSUB.iter
 3c3e78c Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 b706e78 Correct Catalyst library paths in config.ITER.ifort64.
 a4300d0 Remove -c from b2run arguments and update submodule B2.5.
 b156392 Pointing to B2.5 and Eirene submodules with fix for the post-couple NSTRAI check
 7e0dfb9 b2run now takes argument -c in order to run ParaView Catalyst.
 412f3c3 Update config.ITER.ifort64 and setup.csh.ITER.ifort64 with libraries necessary for
 linking with ParaView Catalyst.
 115b0b0 Pointing to fixed B2.5 cherry-pick for automatic matrix refresh and corrected Makefile
 for tracing directory

24001ea Dealing with missing tracing directories
 69e0e9a Keeping MA28 as 5.0 potential equation solver
 99990d6 Pointing to B2.5 submodule with BCCON=9 sources sign checked
 6d2ed11 Pointing to B2.5 submodule with corrected typo
 dd216f9 – Cleaned up and corrected documentation about b2plot commands – Better treatment of
 b2_to_astra in runs/Makefile – Added "fixbb" script for better PostScript bounding box
 handling – Pointing to B2.5 submodule with updated b2plot commands and BCMOM=10 Pfirsch–
 Schlueter sign fix
 9dfdd00 Update B2.5 submodule.
 bb31924 Add variables for catalyst and paraview to config.ITER.ifort64 and change python version
 to 2.7/11 in setup.csh.ITER.ifort64.
 609488e Pointing to B2.5 and Eirene module with bug fix for when eirdiag_nds arrays are empty
 71c4c1c Dealing with compilation after removed Fortran files in the update script
 4097aad Compilation fix for the previous commit
 6435f2f – Additional correction to JET setup files – Pointing to Eirene submodule with variables
 renaming from Juelich – Pointing to B2.5 submodule with better scheme for refreshing ma28
 when solver switches change
 ac5abb1 Adding needed CPP option for FZJ config file and definition of FLUSH environment variable
 for JET
 23fa397 Pointing to Eirene submodule with removed duplicate MODBGK call
 5991f72 Adding checks for existence of Carre directory and files in triang
 c8a0021 Correcting mtv_3d for f.3d.P plots
 ba3fd66 Add merged submodule B2.5 (feature/IDS with develop branch) to feature/IDS branch.
 e08badd Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into feature/IDS
 abdff18 fix for cpu with new text
 f178be2 Pointing to Eirene submodule with corrected sums
 9f599ab Pointing to B2.5 submodule with changed mmid b2plot command
 457958a Progress towards GGD data read/write from IDS
 44637fc – Pointing to B2.5 submodule with some routine imports for IDS grid write-up – Corrected
 ids_get into read_ids as requested by Simon Pinches – Added a -DITM compiler flag
 e0c51e2 Python script for reading edge_profiles ids
 af7fe82 Compiling with links to IMAS libraries
 97af480 – Fix to Uinp for the right ordering of secondaries for CX reactions – Removing isotopic
 mass corrections in Uinp since the mass scaling happens already in Eirene – Pointing to fix
 for IMAS-400 bug report in B2.5 – Pointing to Eirene submodule with call to MODBGK with no
 BGK species removed
 014d27e Pointing to B2.5 submodule with additional work module dependencies
 be7c141 Pointing to B2.5 submodule with correction quantities included and Eirene submodule with
 bug fix for multi-segment recycling strata
 7079777 Pointing to B2.5 submodule with fluid neutral compatible wall loading routines
 5a4c3f6 Adding fort.44 save files
 4af7fbe Fixes to b2co after bug report from Wouter
 1805805 MDS clean-up
 f4a35c3 Corrected typo in B2.5 submodule
 28e7ac5 Pointing to B2.5 submodule with modified treatment of rescaling fluid neutral rates
 dc81e18 Pointing to B2.5 submodule with updated dependencies in Makefile
 90b4e89 Adding a description of the b2yr.plt file in the manual
 1b53255 Pointing to B2.5 submodule with more wall loading fixes (from C-Mod example)
 cb2f81f Pointing to B2.5 submodule with compilation correction from last commit
 1acc712 Pointing to B2.5 submodule with relaxed convergence error message
 957fa2e Pointing to B2.5 submodule with corrected Eirene temperature conversion
 375f173 Pointing to B2.5 submodule with improved solver message formats
 260be29 Pointing to Eirene submodule with bug fix for fort.78 contour write-up when the
 bottommost point is also the last point of the contour
 0a5402f Pointing to B2.5 submodule with rescaled fluid neutral transport
 270c1d6 Removing b2news_vis_per obsolete switch from example files
 992cced Pointing to B2.5–Eirene modules with wall loading routines corrections
 6ffe282 Additional commit from Eirene
 25d3b52 Pointing to Eirene submodule with Juelich update
 f333077 Pointing to B2.5 submodule with new safety for better code convergence provided by Sergey
 Voskoboinikov
 c2d6d3c Pointing to B2.5 submodule with BCMOM=16 bug fix and some fixes to wall loading routines
 6880145 Pointing to B2.5 submodule with corrected sheath B.C.'s arising from parallel velocity
 sign convention change after debugging from Stefano Carli
 266f5d0 Pointing to B2.5 submodule with ion current contributions in BCENI=8
 783479a – Safer handling of param.dg file for b2plot – Pointing to updated B2.5 submodule with
 multiple ifmt input files

94692d0 Adding amds config directory to first_setup script
 819e43f Adding AMDS config ITM file and correcting sheath heat transmission coefficients in Uinp
 a82d8ef Pointing to B2.5 submodule with shadowing structure read in b2prdn
 7bb18d3 Pointing to B2.5 submodule with changed size for wall loading arrays
 5e9c14f Fixing some bugs found during the 12th SOLPS-ITER User Forum morning session
 8elc03e Pointing to B2.5 submodule with corrected b2tfed routine
 48d9a80 Pointing to B2.5 submodule with new b2tfed routine for computing electron drift velocities
 f00e93b Debugging the -s option for *submit scripts
 3ad45cd Adding -s option to *submit scripts
 1664d01 Pointing to B2.5 submodule with na_feedback scheme bug fixes
 95a5935 Fixing b2co treatment of b2.*_save.parameters files and adding conversion of ffb_strata to nesepm_istrai when defaults are used
 1194021 Pointing to B2.5 and Eirene submodules with latest contributions from SPB and Juelich
 be74c68 - New Uinp features: H/D/T isotope mixes support, species-separated strata and block 10 of the Eirene input file now produced, added C and N reactions - Corrected AMDS files now including C+H CX reactions, and better C and N ionizations and recombination - Pointing to B2.5 submodule with updated DIMENSIONS.F for species-separated strata
 5e3c2fc Pointing to B2.5 submodule with cleaned-up b2md code
 bcl13bf Pointing to B2.5 submodule with b2tfnb_anomalous_core_only switch removed and updated documentation
 c2ba22f Added new str2ogr script provided by David Coster
 697d145 Pointing to B2.5 submodule with corrections to b2trcl
 91d6b93 Pointing to B2.5 submodule with new wlld b2plot options
 60ae6e8 Pointing to B2.5 submodule with compilation fix for b2usrtrc
 efd6365 Pointing to B2.5 submodule with warning message about missing mesh.extra file
 27e63fe Correcting the triageom format for fort.33
 5clf6d1 Pointing to B2.5 submodule with some debugging changes from SPB
 955aca6 Updating MDSplus module for ITER setup files
 f32025f Pointing to B2.5 submodule with nxt[lr] replacing nxiso in b2usrtrc
 c6a9c48 Improved scripts to deal with little bugs on IPP TOK-S system
 c7a0572 Addinf MDSPLUS settings for IPP config files
 675a5f6 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 9373bba Removing iner_par switch and changing default value of ycur to 1
 be9ed4d Making csig=1e-7 by default in b2sxdr and allowing random order of arguments for submission scripts
 a9fld03 Some run Makefile cleanup
 3e9d200 Better path for executables in Makefiles
 14da37a Added environment variables for IPP and pointing to B2.5 submodule with fix attempt for standalone B2.5 fort.31 write-up
 badae33 Fixes to Uinp using AMDS files and pointing to B2.5 submodule with guard_flows treatment of additional currents
 da2ec51 Pointing to B2.5 submodule with optimization error in b2ytdr fixed
 7a3c70e New tflux script and cleaned up b2sxdr Makefile
 09e80c9 Pointing to B2.5 submodule with b2ytdr debugging fixes and consistent file closures
 f35efd2 Adding dealing with b2fgmtry.2 file in b2yt Makefile instructions
 106f766 Pointing to B2.5 submodule with BCMOM=3 fix using sign(bb(:,0))
 bbddb53 Pointing to Eirene submodule with corrected typo
 0e9159f - Adding treatment of b2fgmtry.2 file in conversion script - Pointing to B2.5 module with various bug fixes - Pointing to Eirene submodule with nee Juelich updated comments
 451c06f Attempting to fix some scripts following a report by Anil Tyagi
 1067bd0 Pointing to B2.5 submodule with savestratum definition fix
 002eafc Corrected ITM Gateway submission scripts
 3f15f63 Fixing LD_LIBRARY_PATH for UNKNOWN.ifort64 setup
 342d45b Pointing to caught-up Eirene submodule from FZJ with new comments
 45bd63a Pointing to Eirene submodule with corrected stop statement in post_couple
 7f0cc7a Pointing to Eirene submodule with ISTRAI/ISTRAI debugging in mcarlo spotted by David Coster
 b61809d Adding SPBSTU settings for default_compiler and whereami, and completed list of EU member countries
 fblf8e7 Adding pointers to submodules with SPB config files
 1471589 Configuration and setup files for Saint Petersburg Polytechnic University cluster
 b3bdb40 Pointing to B2.5 submodule with SPB config files and write_nml feature
 4c9775c New treatment of MPI submission for ITM
 6f89cc7 Pointing to Eirene submodule with RCONA earlier allocation
 8409f9c Adding comments to know the list of needed modules
 c9bfca6 Changes from HPC-MNI test build

4804743 Pointing to B2.5 submodule with last adjustments for the ua sign convention change
 7047d3a Pointing to the B2.5 submodule with conversion of fort.44 after b2yt geometry changes
 dd05e4b – Various fixes for DN cases in Uinp – Removing reference to fort.55 file in runs/
 Makefile – Pointing to Eirene submodule with symmetric DN fixes – Pointing to B2.5 submodule
 with some b2stbc-phys fixes towards merge with SPB version – Pointing to cleaned up DivGeo
 module
 6f887ee Config and setup files for new Keio Cluster system
 2c4c8da Moving the deallocation calls after the writing of the fort.31 file in solps4–5 converter
 77f147e Pointing to the B2.5 and Eirene submodules with completed bug fixes from Jeremy Lore
 1dbf2b8 Pointing to Eirene and B2.5 submodules with safeguards against dead cell neighbours
 cc6f381 Pointing to Eirene submodule with typo correction from previous commit
 369057e Pointing to Eirene submodule with MPI file unit handling fixes
 0a012c9 Fixing a CI compilation issue
 79c149d Pointing to Eirene and B2.5 submodule with plasma–species resolved tallies EAPL, EIPL,
 EMPL, EPPL and EPHPL
 ee79a33 Corrected manual description for BCCON=22,23
 d28a2f3 Pointing to B2.5 module with linearisation of BCCON=8 boundary sources
 e09489e Pointing to Eirene submodule with corrected He+ recombination rate
 c3a3e5e Going back to ifmt=1 in B2.5
 ea7caf2 Pointing to B2.5 module with fixed fchanml computation
 3ab2cab Adding b2fstat[ie].add file to runs/Makefile
 5dabd07 added make option b25eirene_ig
 6e6b097 runs/Makefile for IMPGYRO run
 0ece496 runs/Makefile for IMPGYRO run
 bdc21f2 Adding implicit none and cleanup implicit vars
 alb9513 Added implicit none, declared vars that were implicit, combined integer and parameter
 statements
 6a2c55f Some variables were not saved and could go out of scope. E.g., nptcm_b
 2747b75 Fixed typo in format generation
 f41b371 Pointing to the Eirene submodule with curpoi deassociation correction
 958d811 Pointing to the Eirene submodule with curpoi conditional deallocation
 4148e2b Pointing to B2.5 submodule with multiple scaling input profiles
 89eae60 Fixing a git merge malfunction
 ffa5ee1 Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 aa60ddd Fixes required to make CI server happier
 32121fd Cleaning up config files and Makefile
 1076a21 Adding an initial clean for CI build targets
 f4ae418 Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 39ea0da Pointing to corrected B2.5 submodule with removed duplicate ua inversion
 c46eadd Pointing to Eirene submodule with ITS.COUNT issue solved
 0b4cc22 Pointing to eirene submodule with update AMJUEL file
 4b20f19 corrected makefile in \$SOLPSTOP/runs
 b6c15ed Pointing to B2.5 and Eirene submodules with Makefile corrections
 de48207 Pointing to Eirene submodule with completed rate_coeff interface
 8542cc7 Adding .new.modules files in solps–iter.update
 dal4417 Compiling SOLPS–ITER_IMPGYRO
 d8aflb5 adding ksh aliases
 b58e167 gmake b25eirene_ig creates dedicated build directory for impgyro coupled run by adding
 build/*.ig directory
 878ce7a added make option b25eirene_ig
 fea5096 runs/Makefile for IMPGYRO run
 a6b560b B2.5–IMPGYRO coupling calculation with the option USE_IMPGYRO
 953e5fb added make option b25eirene_ig
 4bd773b adding aliases set_ig and unset_ig
 f7cc683 Turn on and off switches for IMPGYRO coupled mode
 667c764 runs/Makefile for B2.5–IMPGYRO coupled calculation
 657b12d runs/Makefile for IMPGYRO run
 884359a Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 56c97ce Pointing to latest patches in B2.5 and Eirene following the large Eirene update
 cb1856f Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 d665323 Modifying print_versions script to deal with Eirene update changes
 c885124 Update ORNL setup files
 9ff051b Pointing to Eirene updated submodule with additional compile fixes
 840a45c Getting the Eirene update to compile
 726143c Large Eirene update
 9a5e4d7 Fix to carre script when storing the first file with a given stem name
 a608aa0 – Adding H.0 elastic collisions to AMDS models – Correcting Uinp to handle plasma–wetted

targets consisting of only two walls – Getting b2sxdx to create a new fort.31 matching the SOLPS-ITER format – Adding the AMDS directory to the SOLPS-ITER path and an associated script
 – Corrected manual following comments from Fabio Subba – Pointing to B2.5 submodule with corrected boundary conditions, b2mod_external and b2yt treatment of inverted velocity sign convention (i.e. fixing sloppy SPB coding) – Pointing to cleaned up Eirene and DivGeo submodules to prepare for IMPGYRO merge
 fcccac0 Correction to the manual concerning sput_dst spotted by Fabio Subba
 5856f69 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 f6ecee8 Changing the MPI module at IPP and on the ITM Gateway to fix an MPI job submission issue with HYDRA
 3c6dff3 Pointing to Eirene submodule with corrected time surface data
 7f62ce3 Pointing to DivGeo submodule with updated ORNL compiler options
 e807d51 Fixes to converter for core feedback schemes and boundary conditions
 b634505 Clarifying radial vs perpendicular in the manual
 8e40f20 Various fixes to Uinp to run through in debug mode
 3316681 Defining SOLPSWORK for ITM Gateway
 9084e84 – Corrections to Uinp: writing a b2.user.parameters file, skipping "not-for-Eirene" wall materials – Adding conversion of new variables from SOLPS4.3 in b2.user.parameters for neutral group processing – Pointing to B2.5 submodule with new neutral group processing code
 5c51fcf Pointing to B2.5 submodule with corrected Luciani flux limits
 d8829c7 Removing a limit on the length of some lines in converted namelist files
 f508983 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 331533b Pointing to ADAS submodule with updated files
 72f2932 Fixes to carre and Triang scripts to grab the proper last sequence number
 703b041 Pointing to B2.5 submodule with b2plot new default settings
 22abde1 Pointing to B2.5 module with underflow error avoidance
 42d75ff Pointing to B2.5 module with ncut=0 corrections
 ddi988b Pointing to B2.5 module with argument mismatch corrections
 5af8f2f – Adding default value to SOLPSWORK as SOLPSTOP/runs – Pointing to B2.5 submodule with partial IMAS-253 fix
 1a83a68 Pointing to updated B2.5 and Eirene submodules
 dbd8c94 Importing changes necessary for b2sxdx compilation
 af30954 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 a60af07 Adding Neon collision cross-section data to b2sxdx converter
 372121b Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 7754ed7 – Pointing to DivGeo module for jetppf2dg fixes – Adding a dummy submission for ITER MPI jobs to keep bad nodes busy
 0050008 deal with JET's new 64 bit cluster
 783d88f don't apply JET's special action if you are in the baserun directory
 e3db4ad allow gks to find it's fonts
 7df801f no use just printing that you will load SETUP/setup.csh.HOST.NAME.local
 eb95b2b Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 ec08c05 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 719f76d Small corrections to setup files
 36f6led Pointing to B2.5 submodule with fixes to fort.44 for restarting 5.0 case coupled to Eirene facelift
 4b445c9 – Pointing to Eirene submodule with debugged MPI error-handling – Fixed grf script and adding grep.lis files to .gitignore
 d050d25 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 8fb8c29 Pointing to the various submodules with JET 64bit machines setup files
 d946c39 Add ifort64 as an option for 64bit JACs
 67d6f07 get job submission to work at JET
 1311d74 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 8c5e838 Add ifort64 as an option for 64bit JACs
 918d46e get job submission to work at JET
 a8e798c Cahnging ITER submission files to ask for only 1 processor by default, but parsing MPI options to change this upon user request
 a28713f changes for JET pgf90
 9cdd137 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 3cbb7ba changes for JET pgf90
 c907afa Pointing up to cleaned up submodules and new use of na_min in b2mndr and b2news_
 d82b379 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 44b28c4 New syntax for command line b2run MPI submission
 3f48272 Adding a SOLPSWORK environment variable for where to send the converted SOLPS4.3 runs and providing a SOLPSTOP file for the converter
 a707eed Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 ef618de – Changed the convert_from_solps4.3 script to write new cases in /work/projects/solps-

iter/{USER}/{DEVICE} – Made b2sxdx script insensitive to argument list order and search for SOLPSTOP file

fc2c978 Pointing to B2.5 submodule with corrected conversion for volume recombination and gas puffing strata

a052829 Adding a link to eirmo-precision for triang-nox compilation

0010a64 Linking to Eirene and ADAS submodules after changes from ITM Gateway import

e7fe3fc Adding ADAS submodule to solps-iter-update script

04f9343 Pointing to cleaned up ADAS submodule with added Li emissivity file reference

dalfc96 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

c216fb9 Pointing to Eirene submodule with improved NEMODS handling

21f4337 Pointing to B2.5 submodule with added dependencies found when compiling on ITM Gateway

322a776 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

97b67a2 better expression for SOLPS4 directory definition

00e555b Pointing to B2.5 submodule with diagnostic modifications to b2nph9

5a594fd Bringing in B2.5 pull request from Eric Meier

85d89a5 typo correction

969f998 Pointing to B2.5 submodule with added flush statements

40de579 Pointing to B2.5 and Eirene submodules with latest interface corrections provided by Sergey

f93121a Pointing to latest Eirene module

d5112a7 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

5cf5578 Fixes for converter and compilation

90695b4 Corrected MPI module for ITER gfortran compilation and updated setup files

6a03e89 Added aliases rmmmin and rmmmax to b2plot manual

875e955 – Pointing to B2.5 submodule with fort.44 file conversion/resizing – Added PDF version of AMDS manual

904a171 Don't create empty input.dat if not found

f6a1c1a Pointing to B2.5 submodule fixing b2rd bug

db93a3a Update ORNL setup file

c954dc6 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

4c523fd Adding python module

904da4f Merge remote-tracking branch 'origin/develop' into feature/ITER-InfiniBand-MPI

4991317 Add path to find libibverbs during linking

0e5e6a9 Use version of EIRENE that compiles on ITER CI build agents

1bb121b Adding status broadcast to SOLPS-GUI from submit scripts (contributed by Leon Kos)

2159e16 Adding compilation of b2xd SOLPS5 to SOLPS4 converter

fe31e09 Sturdier SOLPS-ITER update script and test correction in b2tfhe_ from Aaron Sontag

2036c09 Pointing to B2.5 submodule with tracked species format fix

be90d94 Pointing to B2.5 submodule with corrections from Ilya Senichenkov for BCCON=19 and 23

9eb7d4a Pointing to B2.5 submodule with latest requested changes from St.Petersburg and updated documentation

3e0a53d A more definitive fix for solps4-5 converter compilation

a74ce38 Trying a better end condition for the update script (2nd attempt)

4be7809 Trying a better end condition for the update script

fe583f6 Putting the update script stop conditions at a better place

dd8a5b0 Adding a NETCDF4.3 compiler flag for fixing illegal stride issue

8e33df4 Testing solps-iter-update with stopping conditions if git status is not clean

16f6c75 – Reworded documentation about sign conventions in equations chapter of manual – Removed ignorable error messages from runs Makefile – Added saving of both b2fstati_with+/-ua files – Improvements to solps4-5 compilation from top directory – Improvements to Uinp compilation, including with MPI – Added MPI-compliant error-handling in Uinp – Modified ITER submission file to accept 16 processes per node – Added SOLPS-GUI module for ITER setups – Pointing to B2.5 and Eirene submodules with MPI-compliant error handling

19dc63a Pointing to Eirene submodule with removed duplicates

2e13b81 Adding one more missing dependency

d326ff0 Adding a lot of missing dependencies

32693c2 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

727fdb1 Fixing typos in stand-alone ITER submission files

152ae26 New default Y format for gnuplot scripts

fcffb55 A better fix for removing the legacy links

fdf4209 Attempting to deal with legacy link directories

215da91 Fixing an infinite link loop

093e156 New print.versions script

5cc53ac Fixes for stand-alone b2plot runs

5d5ac44 Some Makefile fixes and pointing to newly tagged submodules

af450bf Added some files to Eirene source list that are exclusively needed for the ITER SOLPS4.3 version for the solps4-5 converter. Also added the localsubmit script for use with the SOLPS-

GUI.

9fbf61e A much better way to automate the solps4-5 links

ceb178a - Trying to clean up the links to the SOLPS4.3 source code

a450a78 Moved the default location of SOLPS4.3 for solps4-5 converter to match that of the ITER repository and provided an automated way to displace those links for local implementations

cb3afd3 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

485d53c Adding new files for compiling solps4-5 converter with ITER version of SOLPS4.3

d492cf9 - Making b2run silent about missing files when cleaning up at the end of a run - Pointing to B2.5 submodule with latest fixes of feedback schemes and warnings about inverted parallel velocity

fcc53e5 One more dependency for solps4-5

0ac24e9 Typo in the last commit

efe98a0 Adding a missing dependency

20a8d89 Fixes to solps4-5 conversion scripts

d7f3bf9 Fixes to the solps4-5 config files

486657a - Needed to add capability for converter to read the full Eirene input file to extract some data from it. - Corrected computation of nspze and nszpi. - Modified scripts to handle the consequences of reading the Eirene input file. - Cleaned up config files and Makefile for solps4-5 converter compilation. - Pointing to B2.5 submodule with corrected conversion of b2.neutrals.parameters variables. - Continued updating the documentation about the sign convention for parallel variables.

a893d03 Pointing to DivGeo and Eirene submodules with old unwanted files cleaned up

d56957a Updated documentation about sign definitions

ee0df3a - Rewrote equations in manual about neutral flux limits to diffusion coefficient - Added treatment of b2fstati-with-ua in run Makefile - Pointing to updated B2.5 submodule with corrected neutral flux limits and inverse-ua switch

7de78ef - Fixes to the job submission scripts - Handling the TRIANG directory files - Pointing to the B2.5 submodule with the ITER ELM 98-species fixes

2315109 Some fixes for MPI infiniband compilation and compatibility with SOLPS-GUI

b56fd95 Pointing to B2.5 submodule with corrected computation for second-order corrections for internal energy sources due to Eirene

0df9fef Second attempt to grab B2.5 corrections

2ad49b7 Pointing to the B2.5 module with the feedback consistency check fixed

9459bc9 Pointing to B2.5 submodule with wlld bug fixes and better use of strings in Uinp. Also handling some additional files in the run scripts

563fcf4 - Changes coming from debugging Uinp - Added module use command in ITER setup file following suggestion from Leon Kos - Pointing to updated B2.5 module with feedback consistency check and restricted parallel viscosity current to hydrogenic ions - Pointing to Eirene module with corrected format for timer printout

df1edd6 Fix for reading only first two characters of char*8 strings

6355474 Pointing to B2.5 submodule with new switch for turning off perpendicular Joule heating

e6103ee Correcting a factor of 4/3 in the conversion of the viscous flux limiter

c0c3e72 Fixing an error introduced in the last commit

e5a6a26 Pointing to Eirene submodule with IMAS-243 fixes provided by Petra Boerner

743b954 Pointing to committed B2.5 module with new default values for alfteEh and fch-pTe

d9a01e1 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

45bd113 Changes required for g95 compilation on ITM Gateway

6f11e15 Fixes for solps4-5 compilation

8def2ee - Pointing to B2.5 submodule with consistent use of csig and calf - Added local option to top level Makefile - Updated documentation

c4703ff Added conversion of SOLPS4.3 feedback schemes and some tweaks to KSTAR config files

1247a13 Pointing to B2.5 submodule with deallocation routines for memory management

a8288ce - Added stdc++ library to Sonnet linker instructions - Pointing to B2.5 submodule with corrections for LCBS consistency check and b2ytdr file handling

48e41b6 Added treatment for fort.l[0-5] files in run and convert scripts and changed default value of NFILE in Uinp to 400

4a1878e - Additional config files for the KSTAR case - Pointing to Eirene submodule with MPI debugs from HLST - Pointing to B2.5 submodule with corrections to converter and improved protection against singular matrices - Clean-up of some equations in the manual

f36707e Pointing to B2.5 submodule with correction on recalculated sources

819d6c1 Added KSTAR config files

a5efb26 Pointing to B2.5 submodule with negative fluxes in BCCON=16

b0904b8 Pointing to B2.5 submodule with safeguards against singular matrix rows

e57bd38 Pointing to B2.5 submodules with fixes to b2yt converter and use of totalview

c5723a2 Adding the doxygen module when running at IPP

6043ae0 - Streamlining b2sxdr compilation - Rebuilding the git_version_*.h files after a code update - Pointing to Carre submodule with changes for DIVIMP output format

650bab4 Pointing to modified B2.5 and Eirene modules for latest changes
4e04d95 A typo in the previous commit
e976c4b Pointing to Eirene submodule with ioflush interface for Windows
4a7ce8f Pointing to Eirene submodule with latest argument list correction from Sergey
cb25b7e Some users insist on separate git-version files for each submodule
71d6c9d Undoing a bad change
adf6cfc – Documentation for BCPOT=16 boundary condition (feedback for constant potential with a given current) – Using git_version variable name everywhere – Other various small bug fixes (a busy week-end for Sergey)
816015d Having the b2sxdx Makefile deal with the fort.30 file
ff09f9b – Changes for b2sxdx to write out a fort.30 file – Added a b2ar.dat.4.3 example file
d6b9394 – Allowed for passing of MPI options through the *submit scripts – Pointing to B2.5 modules with corrected sizes for Eirene source averaging arrays
91dafb5 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
3ed3aca Makefile typo correction in clean_b25eirene_mpi option
2bfa117 Fixes for -z option in job submission scripts
5aff6db Changing the config files for solps4-5 converter compilation
d9ae33b Correcting a typo in the previous commit
a2e774e Hopefully this is all the needed modules!
2d2f7d3 Adding precision to solps4-5 Makefile
f2554aa Adding clgin to solps4-5 Makefile
05a479e Changes to solps-4-5 compilation files
3e2efd1 Adding a line of DEFINES to switch to double-precision BLAS routines in solps4-5 converter for IPP
4e7cf5e Adding a single-precision myblax.F routine for solps4-5 converter compilation
48a6dfa – Corrected examples b2mn.dat files for new use of b2mndr_na_min and b2mndr_na_new instead of b2mndr_na0eps – Changed start-up value for neutral densities in solps4-5 converter from 1 to 1e8 – Pointing to B2.5 submodule with modified interpolation of densities to corners to ensure positivity
0feb8d2 – Changes towards improved compilation of b2sxdx – Updated documentation
5d8f133 Removing some compiler warnings
d24c6bf More whitespace errors preventing a rebase
d642d9e Answer to IMAS-222 with import_files_from_convert script
489fc53 Making use of the new script import_files_from_convert in convert_from_solps4.3
d40c2bb Answer to IMAS-222 with import_files_from_convert script
d056ac4 A proposed fix by david Coster for b2sxdx compilation at IPP
e79f845 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
fccac71 Pointing to corrected B2.5 modules with long comment lines
dab8bcc Loading up GCC module for gfortran compilation at ITER
7b14726 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
2ed95f8 Towards resolution of IMAS-87: further changes to improves compilation with gfortran at ITER. Still blocked by eirmod_octree module in Eirene and errors when linking to MSCL library
73182bc A fix to the b2mn.dat.5.2 example spotted by Aaron Sontag
4456127 Various fixes (for Sergey and gfortran compilation at ITER [except for Eirene])
d40cf41 Added an IMAS compilation flag for future use
b2edd6c Increased set of derived quantities for zero timesteps runs
9850c87 B2plot fix for comment lines
773319f Increased automation of conversion of older 4.3 files
b28bb6b Modifying solps-iter-update script so that debug compilation also succeeds when new files are added
fd3774b Pointing to Eirene submodule tagged for version 3.0.4
78e8c0b Documentation to go with new 3.0.4 version format of bulk materials properties files
9419334 – Pointing to version 3.0.4-a of B2.5 submodule – Added bcpot and potpar to b2.boundary.parameters.stencil – New default setting sets BCPOT=12 for core boundary condition and BCPOT=13 for PFR and North boundaries – Removed -DWANT.THIS compilation flag – Updated documentation
5a1bfe Matlab routines and output description of the data files
f5b5b55 Pointing to B2.5 module with fix for skipping extrapolated reactions in converter
dbf9f86 Updated Web address of SOLPS-ITER Sharepoint site in documentation
73667aa Updating link to B2.5 submodule after importing fixes to heat transfer module
edac029 Pointing to B2.5 submodule with working computational domain contour plots
aa429b9 Some steps towards a more correct translation of the feedback schemes from SOLPS4.3
79d99d7 Improvements for MPI and solps4-5 compilation
21791bd Updated documentation
907989d Pointing to B2.5 submodule with PFC heat transfer model debugged
f4ef255 Pointing to B2.5 submodule with b2mndr_elapsed end message
e834d63 Pointing to B2.5 module with standalone compilation fix

497f3fb Pointing to B2.5 and Eirene submodules with fixes discovered to get the JT-60U reference case for IMPGYRO coupling to run

4052c3c Pointing to B2.5 submodule with corrected NO_CDF bounds

525cc5c Updated documentation: equations and variables

2624799 Including new b2news_no_b2sral-call and b2news_re_eval_prtls_fluxes switches in the example b2mn.dat.?.? files

43b9a13 Adapting to new name for netcdf-fortran module

71eb86b Pointing to B2.5 module with performance monitoring routines

cbd68fe Additional documentation about submission scripts

d1ac70e Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

9e472e7 - Added '-z' option to submission scripts to compress output log file - Added JET submission script - Made b2run search and use a SOLPSTOP file for remote run submission

b8774d1 - Pointing to the Eirene and B2.5 with correct IPP MPI compiler definitions

1c41cb0 - Various fixes to submission scripts to answer complaints by Ilya Senichenkov - Updates to the documentation to the same effect - Pointing to Eirene submodule with corrected mapping of densities and fluxes being passed to B2.5

825e5f0 Fixing the passing of arguments to mpi (and debugger) through b2run.

cec4100 Pointing to modified Eirene and B2.5 modules with various fixes

b5ae8b7 Pointing to B2.5 module with IMAS-223 converter fix to remove 0.1 reactions and support of multiple fort.30 formats

f241eb0 Pointing to IMAS-223 fix in Eirene and reading old fort.30 formats in triageom

1b89afe Modifications to the submissions scripts as proposed by Ilya Senichenkov and Wouter Dekeyser and associated documentation

0ade34a Modified README file to use solps-iter.update script

63571c1 More piecewise improvements to handling old fort.30 format

3f2643c Pointing to B2.5 and Eirene submodules with cleaned-up MPI compilation flags

22a6432 Additional documentation about the itersubmit script provided by Ilya Senichenkov

ca42fa7 Adding copying of old fort.30 file in convert_from_solps4.3

db9a2f3 Changing the interpolation scheme for velocities in b2sxd

f9feb77 Merge branch 'carlis/bugfix/uinp' of ssh://git.iter.org/bnd/solps-iter into carlis/bugfix/uinp

5183959 Fixes for ITER pkgconfig utility

f126ac6 Ability to use older fort.30 formats

66da5c5 Improved treatment of wall materials TRIM file list

60809df Some bug fixes for Uinp (incomplete)

4ebbede Fixes for ITER pkgconfig utility

e019b3b compilation with mpi-infiniband at ITER

c146be4 Option for running with mpi using b2run

119e0eb Ability to use older fort.30 formats

ebbb654 Improved treatment of wall materials TRIM file list

947a5ef Some bug fixes for Uinp (incomplete)

b85322d - Adding a new solps-iter.update script - Fixing some issues with pkgconfig at ITER

34298ff Pointing to new DivGeo submodule with eqdim.inc file

7d0fc9f Pointing to DivGeo submodule with new utility program to shift equilibria

efdb43a More b2ytdr modifications to point to

e8c8609 Pointing to B2.5 submodule with improved conversion of block 11

e5754be Pointing to B2.5 submodule with improved conversion of block 11

7916997 Pointing to B2.5 submodule with new iyref default value for b2plot

5201a71 Pointing to B2.5 submodule with additional SONNET file format reads

a6fc1cc Additional changes to Keio config file

e8d0332 Additional changes to Keio config file

3f703ae Pointing to B2.5 submodule with new Carre translation formats

388426f Adapating correct_b2yt_timestamps to look also in baserun directory for files

2b5b39c Adding the isolating cut data to b2.boundary.parameters.stencil

c4c2da9 Adding creation of a b2.boundary.parameters.stencil file by Uinp

358dd84 Pointing to B2.5 submodule with latest fix

5ca5839 Changing the default B2.5 run label to SOLPS-ITER

9e6d705 Getting Uinp to indicate the minimum number of histories given in DG and new number of test histories for strata in block 14

6dd3e56 Adding GLLHOME definition to Keio setup file

4c5af8a Pointing to B2.5 submodule with improved capability for b2ag to find grid files

c06ad6b Modifying compiler options for Keio cluster

41d184b More finding of k-shell for scripts

219ec3d Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

fe8ee04 Looking for ksh environment in a more robust way

703f687 Pointing to B2.5 module with sign error in b2tfnb corrected

99bc56e Finishing up Keio files merge

9f154b5 Pointing to B2.5 module with Keio configuration files
 e4153fc Pointing to Eirene module with Keio compilation files
 6f26113 Pointing to Carre module with Keio compilation files
 dff95a1 Pointing to DivGeo module with Keio compilation files
 8ab2cd3 Files needed for compilation at Keio.
 929d191 add help option to Makefile
 b33280d Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 2a30a61 Pointing to B2.5 module with fixes requested by Ilya Senichenkov and test against zero
 ion heat conductivity
 7ca65f9 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 a66f6c0 Needed -r8 flag for compilation of SOLPS4 library in solps4-5 converter
 1a5cf1c Second pass at same debugging
 4dbeb6b Pointing to debugged obsolete reaction removal B2.5 submodule
 1cb5409 Pointing to debugged B2.5 submodule
 0a2ca6a Pointing to B2.5 submodule with iffboundx/y allocation moved to b2mndt
 d56216b Pointing to B2.5 submodule with boundary namelist arrays allocation moved earlier
 2ec47c6 Previous commit reason redux
 688b8cd Pointing to B2.5 submodule with corrected compiler error noticed by Sergey
 ebccd39 Bug fix to the previous commit
 fe50aea Small changes to the manual equations required by Sergey
 c1bf6bc Pointing to B2.5 submodule with workspace arguments list fixed after notice by Sergey
 9371105 Pointing to the B2.5 module with debugged Eirene source averaging schemes
 2c7ed44 Pointing to B2.5 submodule with Eirene source averaging moved to eirene_mc
 1e84a09 Pointing to B2.5 submodule with latest bug fix from Eric Meier about Eirene source
 averaging schemes
 b76e0cd Pointing to B2.5 submodule with modified Eirene sources averaging schemes
 de08a21 Pointer to B2.5 with generalized wall elements areas
 c096d94 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 8b5d785 Corrected compilation error for coupled SOLPS-ITER
 92b3b2b Pointing to B2.5 submodule with Eirene source underrelaxation scheme
 a8007f9 Added documentation about averaging schemes for Eirene sources
 c39ed19 Changed default device at SWIP from asdex to hl2m
 04d2e6b Pointing to B2.5 submodule with neutral sources averaging documentation
 9bbaf1d Pointing to DivGeo submodule with corrected dg-symm_dg library dependencies
 38b3cb5 Pointing to B2.5 submodule with moved nspecies from b2mod.diag to b2mod.b2cmpa
 21c4ef8 Pointing to B2.5 submodule with corrected eb2sper test
 dalfb41 Changing the AMDS file extension in Uinp from .amds to .amo
 b34e766 Changing the AMDS file extension in Uinp to .amo
 5cffba1 Pointing to B2.5 submodule with version 21 of MDSPlus database
 1772330 Pointing to B2.5 submodule with wlld working even without a fort.44 file
 484f2ec Getting AMDS to compile at ITER and adding GIT version to its output
 48a36fa changes to amds to allow compilation and run from script
 f9edd3c Pointing to B2.5 submodule with corrected mtab format and not overwriting shotnumber 0
 a819619 Pointing to B2.5 module with modified default values for eb2sper and use_recyceir
 0d6ea16 Bringing in new density feedback actuator scripts from 5.0
 cd9951b Code clean-up and pointing to submodules with same
 61b0da4 Update ORNL config files for MPI build
 90801db Added writing of b2.neutrals.parameters.stencil and fixed save bug in uimdvr
 47de72b Pointing to submodule B2.5 with label field populated for wall loading diagnostics
 bdb3704 Pointing to modified B2.5 module with corrected target radiation load calculation
 9552176 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 d290b5b Pointing to modified Carre module dealing with perfectly symmetrical CDN equilibria
 205dedf add PLOT_FROM_FILE to a bunch of scripts
 e57b751 Added tmpdir and check to preserve original pause 3600 behavior until all routines are
 updated
 307680e Allow for responsive gnuplot windows with mouse control that do not close immediately on
 some systems
 902fe94 Pointing to B2.5 submodule with corrected fch_p and ue computations
 b3e3382 Pointing to modified B2.5 and DivGeo modules with changes from 5.0 trunk
 8de2763 Changes to manual from 5.0 relevant to SOLPS-ITER
 fcce38a Fixing a table float option in the manual
 eac5478 Adding the Doxygen file creation to manual make
 23e1e91 Pointing to DivGeo submodule with cleaned up Doxygen files
 e607020 Pointing to B2.5 module with corrected b2tcpa routine that strongly reduces the restart
 effect in 5.2-like runs
 ba0a8af Second attempt at activating the Eirene tags
 d46640b Trying to activate the tags

30c5acf Pointing to Eirene module with modified check for fluxes
 5b0bfc5 Second pass at bringing in the modifications to the manual equations
 6b97bee Resolving the conflicts in a better way so the manual builds
 31c3096 – New version of manual with latest changes indicated by Sergey Voskoboynikov
 e48f87b updates to the runs/Makefile so that all files are processed; fix for some scripts; add the missing label in solps.tex
 bb37e36 – Adding a forgotten file in setup_baserun_eirene_links – Pointing to modified B2.5 module with latest MDSplus additions – Modifying the runs Makefile to avoid premature terminations
 d9f1221 updates to the runs/Makefile so that all files are processed; fix for some scripts; add the missing label in solps.tex
 73bbffe Pointing to B2.5 module with correction to potential and velocity corrections in guard cells
 499449d Modified MSCL module reference as per suggestion by Simon Pinches for ITER ifort64 environment
 4f04e14 Added some files to SOLPS manual clean list
 944f069 update to ORNL config files to use modules
 4fe18e7 Correcting instructions for radiation plot in manual as noticed by Anil Tyagi
 519a694 Adding links to three more sources files from SOLPS4 Eirene for the SOLPS4–5 converter
 9bd1ae0 Changing the default B2PLOT_DEV value to 'x11 ps' to get b2plot graphs both on the screen and in the PostScript b2plot.ps file
 cf9b3ed Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 4de6f95 Adding capability in Uinp Makefile for standalone MPI compilation
 3f0fd42 Removing real promotion to avoid a problem when linking to NCARG.GKS (solves IMAS–173 & IMAS–180)
 3a7ef40 Pointing to B2.5 submodule with correction to b2siav and b2cdcn
 4caaf04 Pointing to updated B2.5 submodule with b2trcl fix notified by Sergey
 f7d5319 Pointing to B2.5 submodule with b2tinnt bugfix
 d509ec1 Pointing to B2.5 module with corrected viscosity calculation
 242b0a9 add the ability to run totalview for save_mds and resave_mds; additional early output from Eirene broke mds_id
 e9722e3 Catching up with Python script developments in 5.0
 ba55dc4 so that b2plot will work
 3fe295f Grabbing the B2.5 submodule with correct neutral load wlld calculation
 1e7569a Pointing to latest modifications in a desperate attempt to get git to work as intended with the submodules
 0322005 Pointing to b2mds fix provided by David Coster
 fc33333 Attempting a fix to fort.44 with respect to the number of plasma species
 d53fe09 Catching up with Python script developments in 5.0
 ec193d2 Pointing to B2.5 module with corrected snedt allocation
 aldfdfc so that b2plot will work
 510fdb1 Fixing usage of fort.44.2 file and trying to limit error messages from runs Makefile
 5f7aa6f Pointing to B2.5 modules with snedt bug fix
 cc00686 Pointing to corrected Eirene module and modified Makefiles
 a5f80f8 Remove duplicate line from script to toggle mpi
 fc51415 – Pointing to Eirene module with proper nltrimesh=.true. termination – Pointing to B2.5 module with various bug fixes suggested by Sergey – Fixes to the b2sxdx Makefile – Addition of the b2_to_astra.dat file to the runs Makefile
 0d74e18 Modifying PPPL setup file to load all necessary modules. (Had been loading modules manually.)
 f3734f8 Pointing to the corrected B2.5 module for tracing files
 89653d1 Pointing to corrected allocation call to b2wrsepi in B2.5 module
 e08ef1a add python script for comparing 2 cases; fixes for saving to MDSplus server; add python library; add python lib to PYTHONPATH; amend README.EXAMPLES to refer to the solps–user examples
 629a129 add python script for comparing 2 cases; fixes for saving to MDSplus server; add python library; add python lib to PYTHONPATH; amend README.EXAMPLES to refer to the solps–user examples
 643989b – Proper handling of local files for b2sxdx converter – Added conversion of old fort.44 files – Re–sizing some arrays in Uinp to use the number of walls from DG and not from Eirene – Pointing to B2.5 and Eirene modules with latest fixes from Wouter and David (see next commit)
 bf46c18 – Pointing to DivGeo module with new set of topology files – Pointing to B2.5 module with improved 0–timestep handling and 'wlld' partial debugging – Added handling of param.dg file in run directory setup scripts
 9d91e04 Merge branch 'develop' of ssh://git.iter.org/bnd/solps–iter into develop
 1803cc2 Additional modifications to the b2mn.dat.5.2 example file after consultation with Sergey

Voskoboynikov

48a2692 Updated documentation: revised equations and lists of switches

054325f First draft of new b2mn.dat.4.3 example file

f24074b Pointing to the module with ITM MPI compilation fix in B2.5

b5c3307 Pointing to corrected ADAS module and new ITM MPI config files

5adbd4b Pointing to corrected B2.5 module

b013c31 Pointing to re-corrected B2.5 module

f602943 – Pointing to B2.5 module with corrected heatdiff1d for coatings – Pointing to ADAS module with corrected N and Ne data and new Si data – Small additional changes to example files b2mn.dat.5.0 and b2mn.dat.5.2

669cfbd Addind a new download and installation instructions document

107ee8f Getting the 5.2 model to run, with example switch set

0e0f661 Pointing to bug fix from Sergey Voskoboynikov

8585a99 Reproducing SOLPS5.0 behaviour with SOLPS-ITER

f087449 Reverting a commit on DivGeo that causes it to break

3f63581 Version 3.0.3

fcf9e95 Fixes for 5.0 model in B2.5

0c2512a Linking to latest pulls from Jeremy Lore fixing some gfortran issues

48e4415 Grabbing the 5.0 fix for B2.5

0634913 Link to updated B2.5 module

a7cd087 Updated DivGeo module with test for flush routine and architecture-independent integer pointers

3flcd67 Previous merge did not grab the most recent Eirene module pointer

e198d8d New consistent set of modules for NetCDF compilation at ITER (part II)

069a6ab New consistent set of modules for NetCDF compilation at ITER

f32ec53 Changes to better handle netCDF integration

b7fe64d Adapting the sonfig files for NetCDF at ITER

d1a4138 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

3400763 Updated B2.5 code with revived output

1ea4439 Update B2.5 module to (try to) compile with netcdf on CI server

722e56d Improved compilation for standalone B2.5

a48fcfd Link to better set of default switches for running B2.5

068ee25 Compilation with local NetCDF copy

d7324e0 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

e32518e IMAS-138 fix

0088907 Silencing the 'which nc2text' test

0d55b8c Improved fort.44 treatment

662820b Uinp Makefile cleanup and eirene_repeat_first_call 1 by default

0a47159 Link to working default fluid neutral B2.5

ce54dc2 Better handling of the IMAS installer script

80fd46f Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop

8371205 Version 03.001.002

39556c4 Added modules/amds to the path for AMDS files in Uinp

856106e Adding notes about job submission scripts in manual

6101e85 Adding the possibility for a scripts.local directory

2544543 Allowing for the SOLPSTOP file to be above the run directory

8b59ad4 – Allowing for zipped b2ftrace file – Loading the IMASDB module

9ee53bf New streamlined set-up for submission scripts, inspired from 5.0 treatment by David Coster

363c84b Now with working submission scripts at ITER

d61f794 Adding make module to list

b8ffcbe Temporary ITER submission scripts

2742509 Bringing in various bug fixes from Jeremy Lore

c9ad777 itersubmit script

4e0053a Changes for the AUG-16151.D benchmark case

3c545ee Compilation fix for Eirene graphics

9553477 Fix to gr_dummy compilation conditions

168775a Added mention of .quit file in manual as per IMAS-116

b0abeb4 Better fix for b2plot

1de2d55 Bringing in changes from the ORNL Training session

eb77848 Fixing Uinp compilation.

977b2e9 Pointing to new Eirene config files

d252494 Pointing to B2.5 with comments on b2???.dat files and slight cosmetic changes to carre and triang scripts

72e9e10 Second attempt at fixing DERF issue in Eirene

7e9bf40 update default value so correct version of geomd is used

c4a9ab5 Pointing to the Eirene module with corrected nox compilation

1ef5fb7 Connected to B2.5 module with earlier b2stbr_init call
 f73a2b7 Importing corrected Eirene graphics config file for PPPL
 c671cb9 Pointing to corrected Eirene module
 fdca286 Pointing to the Eirene module with corrected if0prm for ibrad,ibpol,ibtor
 03549fa Added three default values to uinp
 9325ed3 Added description of equations Chapter to manual
 92f525c Updated Eirene module link
 9088de4 Linking to the B2.5 with corrected electron heat fluxes
 4c8c738 Version 3.0.1 with new 5.2 physics model and added conversion of Eirene input file from SOLPS4.3 to SOLPS-ITER format
 4bca47b Pointing to fixed up modules
 d0729bf Corrected symbol choices for test.eir input files
 7e776a2 Eirene modules with graphics
 8ece760 Fixes for the Eirene merge
 e1b5e62 Updated documentation
 a9415a7 – Changed sputtering switches from SOLPS4.3 style to new Juelich style (ESPUTS/ESPUTC) – Modified Uinp accordingly – Modified some scripts according to SOLPS5.0 IPP repository, including error bar option for plots – Modified SOLPSTOP/runs/Makefile to deal with debugger – Hard-wired HDF5 module version number awaiting for resolution of issue IMAS-95
 6526898 Cherry-picked Eirene code changes from J lich
 10311ae Makedepend does not like hashes as continuation marks
 d4aa758 Adding ntrfrm,ibrad,... to Uinp
 5b68eaf Modifications to Eirene module for compilation
 c8d36e3 With a compiling Eirene submodule
 1d09071 Adding LCHKQUD to Uinp
 b488547 Bringing in version 3.0.1 with POB field in fort.44
 f9eea6b Fixing a potential issue reported by Larry Owen for the b2yt converter
 edf5355 update for ornl config files with graphics
 3f4171b Modified ncdump parsing to extract subarrays in 2d_profiles and 2d_profiles_extended. Much slower than nc2text, but should work.
 2447b36 Updating the manual to include a mention of the new conversion script
 6a206b3 New script for conversion from SOLPS4.3 to SOLPS-ITER
 da06436 Merging with Jeremy's replacement for nc2text
 cd8ae1b Merge branch 'jlore/feature/2dt-no-nc2text' of ssh://git.iter.org/bnd/solps-iter into jlore/feature/2dt-no-nc2text
 10140e2 Adding an alternate to nc2text for the scripts
 3f4ef83 added check for nc2text
 4146283 Pointing to a shared hash for the Eirene submodule
 edf608b added 2dt without nc2text calls
 1f741d4 Importing some code updates from Juelich and IPP
 1067df8 updated ORNL config files
 9d118ba Submodule modules/adas points to ITER-develop.
 f561dc0 – Modified build of dependencies to include relative SOLPSTOP addresses – Corrected Uinp to always run with assuming triangulated regions – Added new ITER-develop reference branches in modules/Carre and modules/DivGeo – Pointing to updated submodules (including Eirene debugged version that now runs through)
 b5d4322 Linking to the Eirene module with closed contours
 53c67cb IMAS-75: trying to correct the whereami script to avoid multiple output
 0a55dcb Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 c423f69 Committing bug fix in Eirene about EIRENE.H.COLRAD calls
 f00ff4e Removing some apostrophes to avoid problems with quote mismatches
 de4b2af – Modified TAGS file to point to routines actually used when duplicates are present – Added treatment of graphite_ext.dat and mo_ext.dat files in run scripts and Makefile
 7d2e8d3 Doing a complete manual build on first compilation
 e085072 Merge branch 'develop' into bugfix/IMAS-92-Sonnet.FC
 4cac367 Added test target for building executable based on string_main
 50e7cd6 Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 7af875e Correction to the cpu script and adding missing chop_pwd
 b639c3f Since CXX is already hardcoded in to use g++ apparently without problem, I defined FC as gfortran. I could not get it to work with the system default of f77->g77. Since this string_main is just a test program anyway, I do not expect any serious issues with assuming gnu compilers
 ba870d5 IMAS-92: replacing hard-coded compiler names with FC environment variable
 55ec4a8 Bringing in IMAS-79 solution in B2.5 module
 0562005 Pointing to B2.5 modules after running standalone D-only benchmark case
 f175721 Changing the test in Uinp for the value of NADD
 3747f74 Adding the B2PLOT.DEV environment variable set to 'ps' in the setup files

b949fb6 Adding missing separator in Triang Makefile
 7bb8532 Also conditioned removing of .mod files
 0666194 made removal of .f file check SED setting
 6alee2a Remove mod files too when making clean
 59f2c3d 27 characters in Uinp version
 855b120 27 characters for git_version and another -C CPP flag removed
 59a295a Removing the -C CPP preprocessor flag for gfortran config files
 c5e76f8 Setting the SWIP device name to asdex
 77df087 Pointing to the updated modules
 a825de0 Merging config-SWIP branch
 66ba7f8 Adding the domain name for SWIP
 e239d97 Bring in bugfix in b2ai
 3fld1e1 Bring in SWIP config files
 30d02b3 Added separate checks for common and local directories
 9ala6d9 forgot one
 e48385e cleaned up and generalized pgf90 config files and added gfortran
 bcfc7f6 Bringing updated config files from ORNL for various submodules
 9cfb099 Bringing B2.5 changes suggested by various users
 6701f43 Changes required at ASIPP
 d0f0a91 IMAS-83: Better formatting of the git version number line from Uinp
 1c9e03f IMAS-83: Adding a GIT version number to Eirene input files produced by Uinp
 386cbaa Merge branch 'develop' of ssh://git.iter.org/bnd/solps-iter into develop
 feb3a2b Pointing to the Eirene submodule containing the IMAS-91 issue fix
 c7b465b Merge remote-tracking branch 'origin/develop' into bugfix/solps4-5_ank
 66a4d41 Formatting correction to the manual.
 96a6dda A fix to the manual spotted by Chaofeng
 f0a864b Add setup.csh file for SWIP
 e5eaa5f Add config files for SWIP
 8b4e612 Bring in format fix in Carre
 95a17ac Putting the echo messages in the right place
 c51d65d Making SOLPS4-5 links relative
 d172740 Remove non-gfortran flags from Triang config-files
 58d2d30 Bring in gfortran config changes in Eirene
 fde775c Corrected Eirene module
 9c5e043 Grabbing the Eirene submodule with the DN fixes
 efad7b0 Added messages about setting of debug and MPI flags and handles in Uinp for Eirene
 database files
 3be5deb Add instructions in README on how to update master/develop branches
 0a08673 Bring in bugfixes on B2.5 side
 5662597 Was pointing to the wrong B2.5 submodule
 3f6bec3 PPPL configuration files
 917bfce Initial configuration
 82a67b3 Pointing to the pruned and updated submodules
 b53b943 Bring in another gfortran-fix in B2.5
 7ac23fa Bring in IEEE fix (B2.5) and fix for gfortran in EFIT-translation (DivGeo)
 679fb79 Add config files for LEUVEN. Remove some error messages from config files.
 21fd6b8 Various fixes for gfortran compilation, too long lines, ...
 81f569b Dependencies files created for sonnet-compilation in .gitignore
 b75af02 Bring in changes in Carre
 f470a03 Only remove dg.*-links if not in DG-directory
 ac90294 adding PPPL config files
 7e83323 Update JET.pgf90 config
 bc79b85 Correct links in runs-Makefile
 f86517c IMAS-82 fix
 79382df Updated main config file and optimization level on Eirene
 18d4fd6 Proposed change to Makefile that uses libeirene.a instead of specific modules
 bdal166 Attempt to fix the problem of Uinp compilation on non-ITER systems
 050dd71 Changes from Aaron Sontag's branch for compilation at ORNL
 85d876a IMAS-84 issue resolved
 602dlac - Corrected issue IMAS-85 in Eirene - Made the grid mash for DG look for *.sno files
 abca916 Changing the .gitmodules file to point to the correct top SOLPS-ITER branches
 86066b2 Merge branch 'feature/IMAS-74-Triang-Makefile-include' of ssh://git.iter.org/bnd/solps-iter into feature/IMAS-74-Triang-Makefile-include
 1eeec6ad Modified the Triang compile file following Jeremy Lore's suggestion
 7fbdf8a Fixing repeat Triang compilation following suggestion from David Coster
 47fc079 Changes in B2.5 for IEEE arithmetic at JET
 75cee9f Importing some changes proposed by Sven Wiesen

4e48f84 Modified the Triang compile file following Jeremy Lore's suggestion
 634d1ae Merge config files for different uses
 19e2b17 changes for ITM Gateway; fix for Makefile deleting b2plot.commands
 8321d2a Add files needed for compilation at IPP
 e1458a1 Merge remote-tracking branch 'origin/bug-fix/uinp_compilation' into develop
 35394b2 Pointing back to Carre before mistakenly changing the input format reads
 301edfc Getting the Carre Makefile to be less verbose
 2d8b063 Making the Makefile less verbose
 12150f1 Various fixes pointed out by Jeremy Lore
 100c150 Undo change in some carre format statements
 46ba2b3 Bring in explicit format statements in carre
 def6f9e export dir-variables in config-files
 1a48e21 New link to the B2.5 module with the NO.CDF switches
 89e921b Grabbing the updated B2.5 module
 5e41c8f Bracketing some CDF.MOVIE code snippets with #ifndef NO.CDF statements
 268dc6e Removing some dangerous characters in comments
 f25b883 Some fixes for input.dat b2ytdr conversion
 c099966 Add LICENCE as agreed by rights holders (IMAS-69)
 e949d79 Updates to the manual for code update procedure and links to the ADAS submodule
 ccecb63 Various configuration bug fixes suggested by users
 2c4e637 Use fort.l instead of input.dat for Eirene input in runs/Makefile
 9c40c83 Adding a link between input.dat and fort.l
 8460223 Bug fixes before the workshop
 a527da6 Adding the stra.dat and weis.dat links to correct.b2yt_timestamps
 4dd789b The DivGeo LICENCE
 b814519 Final pre-release fixes
 a7d819c Removing b2upht..F as required by Serguey and associated changes
 3331625 Update Manual with descriptions of new build targets
 91a1789 Define new build targets for use by CI server
 7a38e34 Creating a Uinp MPI target
 dfd6247 More consequences to the change of naming of directories
 0756392 Change mpich module used to mpich2/3.1.3-intel
 4c31a9c Adapt Makefiles and scripts to new Eirene repository structure
 f98b12e Fix in makefile
 ed19e60 Update clean targets; add clean target for manual
 5033fec Added FZJ config files
 e5692e7 Changes associated with compiling without X libraries
 8ff6046 Ensure EIRENE does not try to link against X11 for nox target
 2dbb682 Added comments in manual about nox and nox_mpi Makefile targets
 a991291 Remove sonnet-light from nox targets
 5867dlf Ensure uinp is build after Eirene and B2.5 in nox and nox_mpi targets
 96de5fd Add Makefile targets that do not require graphics libraries
 f3d1a7b Added mention of first_setup script in README file and some small corrections to manual
 Makefile
 00c0b85 Added a first_setup script necessary for initial SOLPS-ITER installation
 6628ecf Last commit from B2.5
 e6fabef Adding a lot of new config files
 c0f93b2 load intel and makedepend modules when running at ITER
 59073d3 export B25_defines in config files
 9468355 Attempt to avoid a problem when compiling on Bamboo where NCAR is not present
 8c6f2f5 Have Uinp create a skeleton b2yt.dat
 32ad324 Some fixes to triang script
 823e6fc Rationalizing the compiler switches and updating the complete manual for SOLPS-ITER
 Release Workshop
 613923b Bringing in the needed contents of the data directory
 521813b Assigning ifort64 as default compiler for GA
 475a7e1 Getting b2sxdx to work, much better, with SOLPS4.3 from the IPP SVN server
 c61eea9 Adding a missing dependency
 468d85c Corrected a typo that led to an infinite loop
 37a54ad Merge branch 'feature/release' of ssh://git.iter.org/bnd/solps-iter into feature/release
 Merging Makefile with _debug suffix and b2sxdx fixes
 7f39503 Replacing SOLPS_LIB by SOLPSLIB
 acad9be Getting b2sxdx to compile in debug mode from /home/ITER/bonninx/SOLPS-ITER_work
 2751fbe Merge branch 'feature/release' of ssh://git.iter.org/bnd/solps-iter into feature/release
 2b52fb8 Bring in a number of fixes from different modules
 aa4a7a4 Merge branch 'feature/release' of ssh://git.iter.org/bnd/solps-iter into feature/release
 d38e00a Getting b2sxdx to compile

09e9dfc Add DivGeo/equitrn and DivGeo/convert to PATH
 4502793 All debug-targets switched to %_debug: allows for clean._debug as well
 a0acf7f Bring in fixes for coupled B25-Eirene runs
 7d987dc Replacing NAG with BLASLAPACK when unavailable
 a554576 Only change the link directive if not yet defined
 014016d Setup files for various user environments
 1d09aff Merge branch 'feature/release' of ssh://git.iter.org/bnd/solps-iter into feature/release
 d2afb0d Getting the conversion to templates to work
 51dd35e Update config-files to standard directory-names
 2b98d7d Second half of the previous comment
 a7287ae Added GLLHOME definition and corrected wall data initialization in B2.5
 237e62c Resolved an incompatibility between GLI and GKS modules
 790923d Getting triang to run up to the Tria step and Uinp to produce b2ag, b2ai and b2ar .dat files
 6ad30b8 Getting Uinp to run and adding NERSC to the list of hostnames
 0191281 Getting DivGeo and carre scripts to work within the new code structure, added config files for various users, updated manual
 67f3b8f Getting the manual to build
 7098dba Updates to the manual, importing back lots of scripts, and debugging B2.5
 e59e8d9 Added explanatory notes about compiler flags
 5624bb3 Additional ifort64 DIID config files
 1d7d718 Fixed manual Makefile for single run-through without interruptions when starting from scratch
 fbfba4 Adding environment variables settings for DIID
 4fdbd67 Fixing a syntax error in whereami
 b881372 Added error message in setup.csh if config file is not found
 f9d4a7d Removed compiled sonnet library file
 ba87408 Importing the correct new compiler config file into the manual and updates to the Makefile to avoid multiple compilations
 c750ad2 Added message about how to avoid uploading local changes to .gitmodules file
 d2e9063 Add initial set of setup files
 fa5460c Completing the README file and adding some informational messages in setup.csh
 fab97ac Updating the submodule references for the master
 9c98338 Finalizing code structure updates
 d3dc660 Finalizing codestructure updates
 f02de85 New code structure
 df4fb3d Additional debugging check and b2yt converter for Eirene_96 input files
 58f72ab Updating codestructure
 5801033 Added eirene_format switch for b2yt converter Got first pass at reading Eirene_96 formats through Changed USE.EIRENE switch in runs Makefiles and scripts to COUPLED.RUN
 9a8f462 Added link to DIMENSIONS.F file in documentation folder
 4ea7111 Continued manual updates
 125410e Improved build-structure
 b817cb5 Updated documentation about location of manuals and procedure for obtaining and compiling the code
 013ae96 Getting the conversion from Tria output to DG templates to work
 e6589f3 Changes required for the U|B|E|T steps in the triangulation script
 6c2d848 Bringing the Eirene submodule up to date
 f091553 Renaming B25Eirene in Uinp Makefile
 b8a37a3 Merge branch 'feature/codestructure' of ssh://git.iter.org/bnd/solps-iter into feature/codestructure
 3c9203a Adding new switches from Eirene into Uinp, corrected path form tags in Makefile
 770be18 Allow for running dg without local script
 a3876df Capitalize bin-directory B25Eirene
 ba57d13 No longer requiring Eirene stand-alone to be compiled
 92929c5 Various Makefile clean-ups
 264c258 Final tweaks to the Makefiles
 691772c Fixes to Uinp Makefile
 8e02232 Makefile changes to finalize run environment
 03a073e Move LISTOBJ to OBJDIR
 e410356 Merge branch 'feature/codestructure' of ssh://git.iter.org/bnd/solps-iter into feature/codestructure
 24dcf50 Makefile updates; moved uinp -> Uinp
 98e2b19 Updates to the manual concerning how to get the code
 2b1d21c Bring in fix in B2.5 Makefile
 9d54324 Update clean-* targets in Makefile; capitalize directory names in bin/
 21de514 Further updates directory structure on the Eirene side

fcabced Bringing in new Eiren dir. structure
 5f6828a Create links for executables in Makefile; some fixes in uimp Makefile
 0d937b2 Corrections to Uinp and consequences on the Eirene side
 5d850e0 Package now fully compiles!
 99fff4b Change from Wouter for Eirene executable links with MPI
 9256a18 Compilation for all SOLPS with and without MPI
 9d6c908 Update Eirene ref
 520234d Update links for MPI-compiled Eirene
 03067b4 Changes towards B25-Eirene coupled compilation and introduced a GIT version number
 ca726b9 Changes to master Makefile and submodules
 ca6e674 Made array dimensions in Uinp consistent with those from Eirene and B2.5, modifications to master Makefile (to be merged), local determinations of NAG and NCARG environment variables
 d1dafd9 Bring in a number of fixes on Eirene side
 c4ba12f Bringing in earlier changes to if0prm.f and find_param.f in new Eirene directory structure
 2a0f290 Bringing in changes to Eirene directory structure
 65bfd0f Track SOLPS-ITER branch in Eirene submodule, and update to most recent commits of B2.5/Eirene
 8d9998a Merge in local changes into feature/codestructure
 1f044b8 Makefile/script updates
 da97c7f Introducing new block 1 logicals in Uinp, Eirene addition of nltrimesh switch, fix in find_param
 50eb29a Bug fix in find_param
 7204f13 Fixes for compilation standalone and coupled with and without MPI
 5392f16 Imported settings for various Eirene OBJECTCODE Makefiles
 7a3974a Eirene compiles both MPI and NO-MPI versions
 2702f5f Bringing in Eirene compilation with and without MPI
 b18cfc5 Using the .nompi option for Eirene in triang script
 94fe425 Modified scripts for grid creation tool chain
 f351393 Bringing in the manual and adapted changes to the Makefiles and setup files
 0e0ce12 Adding loading the MPI module and check for running at IFERC
 651b93a Getting Triang to compile and bringing in changes from submodules B2.5 and Carre
 cf95676 Rename lib/mscl submodule, all submodules track corresponding remote branches, updated submodules
 a2fac2c Various changes for compilation of Uinp and Triang and run environment settings
 45bc6d5 Putting corrected lnr and lns scripts
 b2bdfa3 Adding the 'count' script for use in setup.csh.local.linux.ifort64 files
 ba2d11a Correcting DG environment variable
 b4c45a9 merge script/e2d updates
 61b5d15 Update e2d script
 51b7271 Corrected path definitions
 2856a73 Changes to equtr Makefile in DivGeo submodule
 8b4a212 Makefile updates
 02c23a3 remove data from repository (eventually, data should come from separate database)
 b938516 Makefile updates
 958fb36 Move B2.5 driver routines into B2.5 submodule
 f8544fc further cleanup of repository
 68c2492 Fixing the e2d script
 c449c21 Merge
 47dfc63 Adding Makefile to list of files to keep track of
 926e576 Fixing the path for local install of sonnet library
 3aaefde Bringing in changes to msclMakefile
 94a9e1c remove scripts from repository
 e24a9df integrate code updates from S. Wiesen into SOLPS-ITER
 db06edf remove ma28 library
 246a96a small environment updates
 72cf81b move setup files
 108cd9c remove all binaries (bin and base); start Makefile
 be0bc8b remove .local directories
 730454d integrate B2.5 (solps5.2) makefiles into code structure
 072e008 add uinp, amds, fxd
 69f756f add include.local
 20d2673 ignore compiled code
 4a2a58f update/add some links in base
 a7fc936 ignore LISTOBJ
 6cb95e2 updates to setup files for new code structure


```

8fcbb98 renamed libs directory -> lib (compatibility with existing scripts)
dfedeb4 ignore user-specific setup files
e0275ae remove unnecessary links in top directory
cb994c9 added runs/.gitignore
c232d6c added libs/.gitignore
b35203e import libraries from solps5.1
10a3920 Merge branch 'feature/codestructure' of ssh://git.iter.org/bnd/solps-iter into feature/
codestructure
f240dc7 add .gitignore file inside bin.local, updates to .gitignore
884cbbc add .gitignore file inside bin.local
bbad27f import src/solps4-5 from solps5.1
69df51c import src/Triang from solps5.1
888dd8c added DivGeo as a submodule
0f3e8a6 added Carre as a submodule
df18b86 add setup-files for run environment
fc8e283 add .gitignore file
5bb8ab9 import runs directory from solps5.1
9a31a60 import include directory from solps5.1
b9f6247 import doc directory from solps5.1
3c376d8 import SETUP directory from solps5.1
e8eec6c import bin and bin.local directories from solps5.1
1d00bcc import base-directory from solps5.1
31757c8 added TRIANG data
006aaf3 Changes to standardise directory structure
7267b9f Update B2.5 reference to version with bug fix from b2mod_ppout
ad60d23 Update submodule references to B2.5 and libs/mscl
c5ef183 Remove B2.5 and mscl and replace with submodules referencing repositories.
702dd92 Moved Eirene_dummy to B2.5/src and removed Eirene_standalone.
c6ef026 Add Eirene as a submodule
ac22c2f remove setting of energy of chemically sputtered particles to 1 eV sputtered particles
should get their energy from the surface model
78d0467 import project status 20012014 from S. Wiesen
f562b04 introduce solps5.3 state 11102013 by S. Wiesen
1cb6c31 start Solps-ITER git repository from branch Eirene-ITER from Eirene SVN repository (rev.
202)

```

M.2 Changes in output for the AUG_16151_D test case

Whenever the code is changed, small changes can appear in the results of a simulation. In this section we look at some changes that have appeared as a result of the changes between **SOLPS 5.0** revisions r1360 and r1463.

The base case was stored as revision 1360 at 2008-01-30 13:39:40 GMT and saved in MDSplus as 21393. This forms the **REF** case for the following discussion.

The first variant, **000** looked at used the r1463 for all files except for `b2mod_constants.F`, where the r1403 version was used. (r1404 introduced a new more accurate set of physics constants.)

The second variant, **001**, used the r1463 revision for all files, but the run was performed with the previously generated `b2frates`, `b2fgmtry` and `b2fpardf` files.

The third variant, **002**, used the r1463 revision for all files, with the run performed with newly generated `b2frates`, `b2fgmtry` and `b2fpardf` files.

Two subvariants of **002** were then performed: **002a** with old `b2frates` but newly calculated `b2fgmtry` and `b2fpardf`; and **002b** with old `b2fgmtry` but newly calculated `b2frates` and `b2fpardf`.

The following observations were made (see figure [M.1](#)):

000 same as **REF**

001 small kick in the residuals (< 0.01)

002 larger kick in the residuals (between 100 and 1000)

002a very similar to **002**

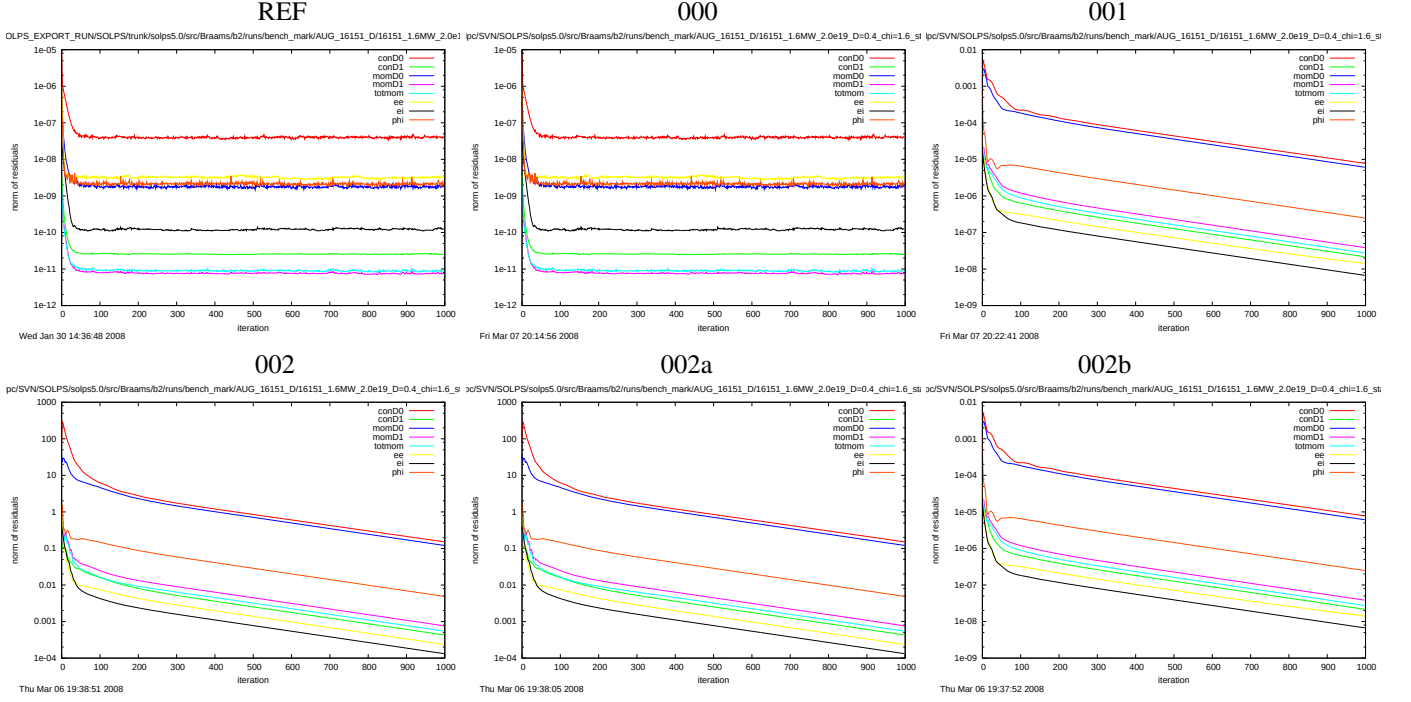


Figure M.1: Residuals of the AUG_16151_D standard case for different code versions and options.

002b very similar to **001**

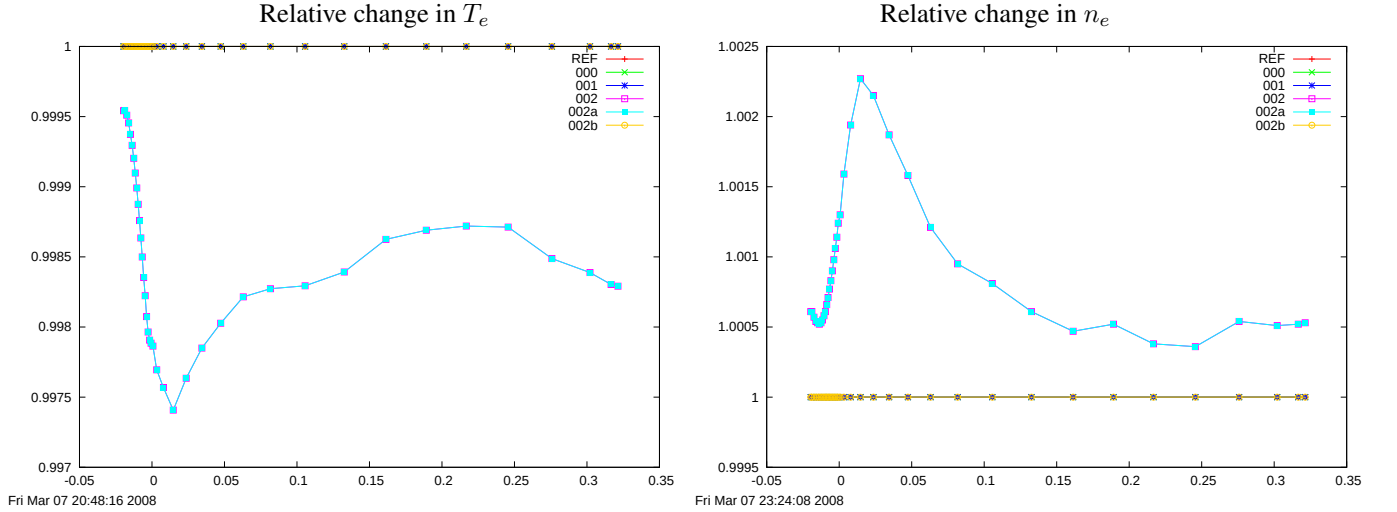


Figure M.2: Relative changes for the considered options.

Comparing (figure M.2) the converged values of T_e at the outer target, **000**, **001** and **002b** gave the same answer as **REF**. **001** and **002b** gave the same answer, and the maximum relative change was 0.997408. (For n_e at the outer target, the maximum relative error was 1.00227.)

The conclusion: the largest effect seems to have come from the correction in the cell volumes (causing a change in `b2fgmtry`), but this caused a small final change. The change of physics constants introduced a very small effect, seen only as a jump in the residuals.

Bibliography

- [1] WIESEN, S. et al., J. Nucl. Mater. **463** (2015) 480, 21st PSI, Kanazawa, Japan, 2014. [1](#)
- [2] BONNIN, X. et al., Plasma and Fusion Research **11** (2016) 1403102, PET-15 conference, 2015, Nara, Japan. [1](#)
- [3] HECHT, F., J. Numer. Math. **20** (2012) 251. [2.5](#), [3.18](#), [3.19](#)
- [4] HULSE, R. A., Nucl. Technol./Fusion **3** (1983) 259. [3.3.1](#)
- [5] BEHRINGER, K., Description of the impurity code STRAHL, Technical Report JET-R(87)08, JET Joint Undertaking, Culham, 1987. [3.3.1](#)
- [6] SUMMERS, H. P., Atomic data and analysis structure users manual, JET-IR 06 (Abingdon: JET Joint Undertaking) (1994). [3.3.1](#), [I.4.10](#)
- [7] SUMMERS, H. P. et al., Plasma Physics and Controlled Fusion **48** (2006) 263. [3.3.1](#)
- [8] WEISHEIT, J. C., J. Phys. B (Atom. Molec. Phys.) **8** (1975) 2556. [3.3.1](#)
- [9] JANEV, R. K. et al., J. Nuclear Mater. **121** (1984) 10. [3.3.1](#)
- [10] REITER, D. et al., Fus. Sci. Technol. **47** (2005) 172. [4](#)
- [11] GHOOS, K. et al., Contrib. Plasma Physics **56** (2016) 616. [5.4](#)
- [12] GHOOS, K. et al., J. Comp. Phys. **322** (2016) 162. [5.4](#)
- [13] GHOOS, K. et al., M&C 2017, Jeju, Korea, April 16-20, 2017. [5.5](#)
- [14] MARCHAND, R. et al., Computer Phys. Comm. **96** (1996) 232. [8](#)
- [15] SCHNEIDER, R. et al., Contrib. Plasma Phys. **46** (2006) 3, DOI 10.1002/ctpp.200610001. [B](#)
- [16] ROGNLIEN, T. et al., Contrib. Plasma Physics **38** (1998) 152. [B](#), [B](#), [B](#), [B](#), [B](#)
- [17] ROGNLIEN, T. et al., Calculation of 2-D profiles for the plasma and electric field near a tokamak separatrix, 1998. [B](#), [B](#), [B](#), [B](#)
- [18] ROGNLIEN, T. et al., Two-dimensional electric fields and drifts near the magnetic separatrix in divertor tokamaks, 1999. [B](#), [B](#), [B](#), [B](#), [B](#), [D.1](#)
- [19] HINTON, F. et al., Nucl. Fusion. (1994) 899. [B](#), [B](#), [B](#), [B](#)
- [20] MIHAILOVSKII, A. et al., Transport equations of plasma in a curvilinear magnetic field, 1984. [B](#)
- [21] HIRSHMAN, S. P. et al., Nucl. Fusion **21** (1981) 1079. [B](#)
- [22] ROZHANSKY, V. et al., Nucl. Fusion **39** (1999) 613. [B](#), [B](#)
- [23] RADFORD, G. J. et al., Contrib. Plasma Physics **36** (1996) 187, 5th Workshop on Plasma Edge Theory, December 1995, Asilomar, USA. [B](#)
- [24] MCDANIEL, E. et al., *The mobility and diffusion of ions in gases*, Wiley, New York, 1973. [B](#)

- [25] ROZHANSKY, V. et al., *Collisional transport in partially ionised plasma*, Energoatomizdat, Moscow, 1988. **B**
- [26] BAELMANS, M., *Code Improvements and Applications of a Two-dimensional Edge Plasma Model for Toroidal Devices*, PhD thesis, Universiteit Leuven, Belgium, 1993, published as Report No. Jül-2891 and as Report LPP-ERM/KMS No. 100 in 1994. **B, D.1**
- [27] ROSENBLUTH, M. et al., Plasma transport in toroidal confinement systems, 1972. **B**
- [28] SCHNEIDER, R. et al., Contrib. Plasma Physics **46** (2006) 3. **B**
- [29] BRAAMS, B. et al., Contrib. Plasma Phys. **36** (1996) 276. **B**
- [30] BOOK, D. L., *NRL Plasma Formulary*, Office of Naval Research, Naval Research Laboratory, Washington, D.C. 20375, 1980 (Revised). **B**
- [31] ZAGORSKI, R. et al., Contrib. Plasma Physics **38** (1998) 61. **B**
- [32] STANGEBY, P. et al., The ion velocity (Bohm-Chodura) boundary condition at the entrance to the magnetic presheath in the presence of diamagnetic and $e \times b$ drifts in the scrape-off layer, 1995. **B.1, B.1, D.1**
- [33] FENG, Y. et al., Journal of Nuclear Materials **266–269** (1999) 812. **B.1**
- [34] CHANKIN, A. V. et al., Plasma Physics and Controlled Fusion **36** (1994) 1485, Preprint in JET-P(94)18, 1994. **B.1, D.1**
- [35] NEUHAUSER, J. et al., Nucl. Fusion **26** (1986) 1679. **B.2**
- [36] CHODURA, R., Plasma flow in the sheath and presheath of a scrape-off layer, in *Physics of Plasma Wall Interaction in Controlled Fusion (Proc. NATO Workshop Val-Morin, Quebec, Canada)*, edited by POST, D. E. et al., pages 99–134, New York, 1984, Plenum Press. **B.2**
- [37] LLE 48 (1991), 178. **B.2**
- [38] KRASHENINIKOV, S. et al., Contrib. Plasma Physics **32** (1992) 255, 3rd Workshop on Plasma Edge Theory, June 1992, Bad Honnef, FRG. **B.2**
- [39] IGITKHANOV, Y. et al., Contrib. Plasma Physics **32** (1992) 308, 3rd Workshop on Plasma Edge Theory, June 1992, Bad Honnef, FRG. **B.2**
- [40] ABOU-ASSALEH, Z. et al., Contrib. Plasma Phys. **32** (1992) 268. **B.2**
- [41] PRASAD, M. et al., Stable solutions of nonlocal heat transport equations, 1991. **B.2**
- [42] COSTER, D. et al., Physica Scripta **T108** (2004) 7, Hydrogen in Condensed Matter 2003 Workshop, Helsinki. **B.2, B.3.1, B.3.2**
- [43] WARRIER, M. et al., Computer Physics Communications **160** (2004) 46. **B.3.1**
- [44] UMANSKY, M. et al., Journal of Nuclear Materials **313–316** (2003) 559, 15th PSI, Gifu, Japan, 27.–31. May 2002. **B.3.1**
- [45] KUKUSHKIN, A. S. et al., Plasma Phys. Controlled Fusion **44** (2002) 931. **B.3.3**
- [46] BRAGINSKII, S. I., Transport processes in a plasma, in *Reviews of Plasma Physics*, edited by LEONTOVICH, M. A., volume 1, pages 205–311, Consultants Bureau, New York, 1965. **B.4, C.8**
- [47] BALESCU, R., *Transport processes in plasmas I*, North Holland, New York, 1998. **B.4**
- [48] ZHDANOV, V., *Transport Processes in Multicomponent Plasma*, Taylor and Francis, New York and London, 2002. **B.4**
- [49] ROZHANSKY, V. et al., Nuclear Fusion **49** (2009) 025007. **C**
- [50] SAUTER, O. et al., Computer Physics Communications **184** (2013) 293. **C.1**
- [51] STANGEBY, P., The Bohm-Chodura plasma sheath criterion, 1995. **D.1**

- [52] RIEMANN, K.-U., Journal of Physics D: Applied Physics **24** (1991) 493. [D.1](#)
- [53] RIEMANN, K.-U., Phys. Plasmas **4** (1997) 4158. [D.1](#)
- [54] DAUBE, T. et al., Phys. Plasmas **6** (1999) 2409. [D.1](#)
- [55] ROGNLIEN, T. D. et al., J. Nucl. Mater. **266–269** (1999) 654. [D.1](#)
- [56] SALA, M. et al., Plasma Phys. Control. Fusion **62** (2020) 055005. [H.1](#)
- [57] BONNIN, X. et al., J. Nucl. Mater. **290–293** (2001) 829, PSI-14 conference, 2000, Rosenheim. [H.6](#)

Index

- [rranareg_per_rsanareg.py, 53
- _ual_write, 573
- 2d, 46
- 2d, tesepm, 89
- 2d.summarize, 46
- 2d.plots, 46
- 2d.plots_av, 46
- 2d.profiles, 46
- 2d.profiles_extended, 46
- 2da, 46
- 2dt, 46
- 2dt_av, 46, 395
- 2dt_tallies.py, 46
- 2dvi, 46

- ADAS, 36, 39, 66, 488, 496
- adas, 39
- add_central_device, 27, 49
- ADD_TE_CORR_TO_PO, 80
- addspec, 82
- ADPAK, 66
- ALLOCATE_AND_NAMELIST, 529
- AMDS, 39
- AMJUEL, 38
- AMNS, 66, 530
- an3da, 393
- an3di, 393
- an3dl, 392
- an3dr, 393
- an3dtl, 394
- an3dtr, 394
- ASTRA, 530
- ave.awk, 49
- Averaging, 153
- awk.derivative, 53
- awk.first+even, 49
- awk.integrate, 49
- awk.mesh, 49
- awk.notfirst, 49
- awk.plot_mesh, 49
- awk.reverse, 49
- awk.transpose, 49

- b2, 49
- B2.5, 33
- B2.5, local, 33
- B2.5, builds, 33
- B2.5, code documentation, 198
- B2.5, communication files, 575
- B2.5, compiling, 24, 506
- B2.5, compiling options, 525
- B2.5, equations, 334
- B2.5, input/output files, 574
- B2.5, list of namelists, 290
- B2.5, list of routines, 201
- B2.5, list of switches, 243
- B2.5, list of variables, 216
- B2.5, Makefile, 506
- B2.5, model, 334
- B2.5, modifying the source, 33
- B2.5, obtaining, 18
- B2.5, programs, 572
- B2.5, recover an archived run, 123–125, 127
- B2.5, run recovery, 121
- B2.5, runs, 40
- B2.5, src, 33
- B2.5, Updating, 18
- b2.atomic_physics_rescale.parameters, 88
- b2.boundary.parameters, 70, 118
- b2.boundary.parameters, D+C+He example, 145
- b2.boundary.parameters.2, 118
- b2.neutrals.parameters, 88, 118
- b2.neutrals.parameters, D+C+He example, 147
- b2.neutrals.parameters.2, 118
- b2.numerics.parameters, 79
- b2.numerics.parameters, D+C+He example, 146
- b2.sources.profile, 83
- b2.transport.inputfile, 80, 447
- b2.transport.parameters, 77, 80, 118
- b2.transport.parameters, D+C+He example, 146
- b2.transport.parameters.2, 118
- b2.transport.profile, output by b2plot, 487
- b2.user.parameters, 88
- b2.wall_save.parameters, 118
- b2.wall_save.parameters.2, 118
- B25_DEFINES, 30, 530
- B25_EIRENE, 529
- b2_ual_write, 128
- b2ag, 60, 572

b2ag.dat, 60, 574
 b2ag.dat, D+C+He example, 135
 b2ag.prt, 574
 b2ah, 61, 109, 572
 b2ah.dat, 80, 118, 574
 b2ah.dat, D example, 61
 b2ah.dat, D+C+He example, 136
 b2ah.dat.2, 118
 b2ah.prt, 574
 b2ai, 67, 110, 572
 b2ai.dat, 574
 b2ai.dat, D example, 67
 b2ai.dat, D+C+He example, 135
 b2ai.prt, 574
 b2ar, 36, 66, 572
 b2ar.dat, 574
 b2ar.dat, D example, 66
 b2ar.dat, D+C+He example, 136
 b2ar.prt, 574
 b2bremreg, 397
 b2cdca.F, 198
 b2cdci.F, 243
 b2cdcn.F, 290
 b2cdcr.F, 201
 b2cdcv.F, 216
 b2cmpa, 64
 b2cmpb, 64
 b2cmpt, 65
 b2co, 50–53, 57, 121, 572
 b2co.prt, 574
 b2divua, 397
 b2divue, 397
 b2exba, 397
 b2exbe, 397
 b2favere, 153, 154, 575
 b2favere.2, 118
 b2favere_OK, 49
 b2faveri, 153, 575
 b2fgmtrt, 123–125, 127, 575
 b2fgmtry, 60, 107, 118, 123–125, 127, 575
 b2fgmtry.old, 118
 b2fmovie, 69, 114, 575
 b2fparam, 69, 109, 575
 b2fpardf, 61, 109, 118, 575
 b2fplasma, 69, 122, 575
 b2fplasmf, 122, 575
 b2fraa, 397
 b2frates, 66, 113, 118, 575
 b2fstate, 50, 57, 69, 110, 121, 123–125, 127, 153, 575
 b2fstate_OK, 50
 b2fstate_OK_bool, 50
 b2fstatl, 67, 110, 153, 575
 b2fstatt, 118, 575
 b2ftrace, 69, 116, 575
 b2ftrack, 69, 115, 575
 b2fu, 122, 572
 b2fu.prt, 574
 b2joule, 397
 b2md, 158, 572
 b2md.dat, 574
 b2md.prt, 574
 b2mn, 69, 110, 572
 b2mn.dat, 80, 118, 574
 b2mn.dat, D example, 69
 b2mn.dat, D+C+He example, 144
 b2mn.dat.2, 118
 b2mn.prt, 574
 b2movies.nc, 121
 b2p, 50
 b2pc, 50
 b2pl, 50, 90, 572
 b2plot, 50, 90
 b2plot, a0, 435
 b2plot, a4l, 432, 472
 b2plot, a4p, 432, 472
 b2plot, abs, 431, 452
 b2plot, acol, 432, 479
 b2plot, adas, 436, 496
 b2plot, additional commands, 446
 b2plot, ahal, 423
 b2plot, alf, 423
 b2plot, alph, 423
 b2plot, amju, 436, 496
 b2plot, anjuel, 436
 b2plot, arithmetic commands, 458
 b2plot, arow, 432, 479
 b2plot, arox, 431, 479, 485
 b2plot, aroy, 431, 479, 485
 b2plot, arww, 432, 479
 b2plot, aspc, 433, 473
 b2plot, ave, 431, 488
 b2plot, avogr, 446
 b2plot, b, 423
 b2plot, b0, 435
 b2plot, b2br, 423
 b2plot, b2eatcr, 423
 b2plot, b2espcr, 423
 b2plot, b2im, 423
 b2plot, b2in, 423
 b2plot, b2io, 423
 b2plot, b2ne, 423
 b2plot, b2ni, 423
 b2plot, b2nm, 423
 b2plot, b2nn, 423
 b2plot, b2ra, 423
 b2plot, backscae, 440

b2plot, backscat, 440
 b2plot, bala, 431, 490
 b2plot, bb, 423
 b2plot, bbdyemis, 440
 b2plot, bc, 423
 b2plot, bcch, 423
 b2plot, bche, 423
 b2plot, bchi, 423
 b2plot, bcmo, 423
 b2plot, bcna, 423
 b2plot, bcol, 433, 478
 b2plot, bix, 423
 b2plot, biy, 423
 b2plot, bold, 433, 474
 b2plot, both, 433, 472
 b2plot, bpol, 423
 b2plot, brad, 423
 b2plot, brch, 423
 b2plot, brhe, 423
 b2plot, brhi, 423
 b2plot, brmo, 423
 b2plot, brna, 423
 b2plot, btor, 423
 b2plot, bvis, 423
 b2plot, bw, 433, 474
 b2plot, bx, 423
 b2plot, by, 423
 b2plot, bz, 423
 b2plot, c, 446
 b2plot, c_cp_b1, 440
 b2plot, c_cp_b2, 440
 b2plot, c_cp_b3, 440
 b2plot, c_cp_b4, 440
 b2plot, cam, 436, 495
 b2plot, cd, 423
 b2plot, ceqp, 423
 b2plot, changes from SOLPS4.0 to SOLPS5.0, 106
 b2plot, character commands, 436
 b2plot, chbx, 436, 495
 b2plot, chhichii, 423
 b2plot, chineo, 423
 b2plot, chmp, 436, 495
 b2plot, chor, 436, 492
 b2plot, chord plots, 492, 497
 b2plot, chpl, 436, 492
 b2plot, chs, 431, 452
 b2plot, chvl, 436, 495
 b2plot, ckappa_a, 440
 b2plot, ckappa_b, 440
 b2plot, ckappaa1, 440
 b2plot, ckappaa2, 440
 b2plot, ckappaa3, 440
 b2plot, ckappaa4, 440
 b2plot, ckappaa5, 440
 b2plot, ckappa1, 440
 b2plot, ckappa2, 440
 b2plot, ckappa3, 440
 b2plot, ckappa4, 440
 b2plot, ckappa5, 440
 b2plot, cltb, 436, 491
 b2plot, clvd, 433, 482
 b2plot, clvl, 433, 483
 b2plot, clvn, 433, 482
 b2plot, clvp, 433, 482
 b2plot, clvs, 433, 482
 b2plot, cmd, 436, 494
 b2plot, cmdcdf, 436, 494
 b2plot, coathick, 440
 b2plot, col, 433, 474
 b2plot, colour table, 477
 b2plot, comp, 433, 473
 b2plot, coni, 423
 b2plot, conn, 423
 b2plot, constants, 446
 b2plot, cont, 433, 449
 b2plot, contour overlays, 477
 b2plot, contour parameters, 482
 b2plot, cooltemp, 440
 b2plot, cora, 423
 b2plot, coro, 446
 b2plot, cp, 443
 b2plot, cp_b1, 440
 b2plot, cp_b2, 440
 b2plot, cp_b3, 440
 b2plot, cp_b4, 440
 b2plot, cpa, 423
 b2plot, cpo, 424
 b2plot, cr, 424
 b2plot, crx, 424
 b2plot, cry, 424
 b2plot, cte, 424
 b2plot, ctgdns, 440
 b2plot, cti, 424
 b2plot, cua, 424
 b2plot, cut, 424
 b2plot, cvsx, 424
 b2plot, cvsy, 424
 b2plot, cvv, 424
 b2plot, cz, 424
 b2plot, d, 424
 b2plot, dab2, 424, 485
 b2plot, dase, 431, 485
 b2plot, data manipulation, 452
 b2plot, data plotting commands, 448
 b2plot, dcol, 433, 478
 b2plot, dd, 424

b2plot, ddw, 442, 453
 b2plot, ddx, 431, 453
 b2plot, ddy, 431, 453
 b2plot, debu, 446
 b2plot, debug, 446
 b2plot, deltab, 424
 b2plot, deltal, 424
 b2plot, deltapar, 424
 b2plot, deltaw, 424
 b2plot, deltawn, 424
 b2plot, deltawt, 424
 b2plot, depo_ml, 440
 b2plot, deposit, 440
 b2plot, dfhe, 424
 b2plot, dfhet, 424
 b2plot, dfhi, 424
 b2plot, dfhit, 424
 b2plot, dfhj, 424
 b2plot, dfhm, 424
 b2plot, dfhp, 424
 b2plot, dfmo, 424
 b2plot, dfna, 424
 b2plot, dhet, 424
 b2plot, dhit, 424
 b2plot, dib2, 424
 b2plot, dkappadt, 443
 b2plot, dmb2, 424
 b2plot, dnneo, 424
 b2plot, dp, 424
 b2plot, drop, 431, 466
 b2plot, dup, 431, 466
 b2plot, dvua, 424
 b2plot, dvue, 424
 b2plot, dx, 424
 b2plot, dy, 424
 b2plot, eabs, 438, 452
 b2plot, earow, 438, 479
 b2plot, earox, 438, 479
 b2plot, earoy, 438, 479
 b2plot, east, 440
 b2plot, eavg, 438, 486
 b2plot, ebal, 424
 b2plot, ebld, 424
 b2plot, ebls, 424
 b2plot, ecam, 436, 495
 b2plot, echbx, 436, 495
 b2plot, echmp, 437, 495
 b2plot, echor, 437, 492
 b2plot, echs, 438, 452
 b2plot, echvl, 437, 495
 b2plot, eclv, 433, 483
 b2plot, eclvd, 438, 482
 b2plot, eclvn, 439, 482
 b2plot, eclvp, 439, 483
 b2plot, eclvs, 439, 482
 b2plot, ecol, 433, 479
 b2plot, edena, 437
 b2plot, edeni, 437
 b2plot, edenm, 437
 b2plot, edims, 439, 470
 b2plot, edissml, 424
 b2plot, edrop, 439, 470
 b2plot, edup, 439, 470
 b2plot, eexp, 439, 457
 b2plot, ehx, 424
 b2plot, ehy, 424
 b2plot, ei, 440
 b2plot, eirc, 431, 485
 b2plot, Eirene quantities, 437
 b2plot, Eirene stack commands, 438
 b2plot, elbl, 439, 475
 b2plot, elist, 439, 470
 b2plot, eln, 439, 457
 b2plot, elog, 439, 457
 b2plot, em*, 439, 459
 b2plot, em+, 439, 459
 b2plot, em-, 439, 459
 b2plot, em/, 439, 459
 b2plot, emax, 435
 b2plot, emeq, 439, 463
 b2plot, emge, 439, 463
 b2plot, emgt, 439, 463
 b2plot, emin, 435
 b2plot, emle, 439, 463
 b2plot, emlt, 439, 463
 b2plot, emmax, 439, 459
 b2plot, emmin, 439, 459
 b2plot, emne, 439, 463
 b2plot, end, 447, 490
 b2plot, eof, 447, 490
 b2plot, eoutp, 439, 487
 b2plot, eover, 439, 470
 b2plot, epos, 439, 463
 b2plot, epvt, 439, 486
 b2plot, eps0, 446
 b2plot, erel, 440
 b2plot, ero_ml, 440
 b2plot, erosion, 440
 b2plot, esel, 439, 456
 b2plot, esqrt, 439, 457
 b2plot, esumz, 439, 455
 b2plot, esurf, 439, 450
 b2plot, eswap, 439, 470
 b2plot, etal, 439, 485
 b2plot, etos, 439, 471
 b2plot, ev, 446

b2plot, evac, 439, 486
 b2plot, ewid, 433, 479
 b2plot, exba, 424
 b2plot, exbe, 424
 b2plot, exch, 83, 424
 b2plot, exhe, 83, 424
 b2plot, exhi, 83, 424
 b2plot, exit, 447, 490
 b2plot, exmo, 83, 424
 b2plot, exna, 83, 425
 b2plot, exne, 83
 b2plot, exp, 457
 b2plot, expf, 437, 479
 b2plot, expl, 433, 479
 b2plot, exps, 433, 479
 b2plot, extl, 437, 490
 b2plot, ezrng, 439, 455
 b2plot, eysel, 439, 456
 b2plot, f.lx, 431, 449
 b2plot, f.ly, 431, 449
 b2plot, f.x, 431, 449
 b2plot, f.y, 431, 449
 b2plot, f30, 447, 487
 b2plot, f31, 447, 487
 b2plot, fac_exb, 425
 b2plot, fac_vis, 425
 b2plot, facdrift, 425
 b2plot, fax_ext, 425
 b2plot, fay_ext, 425
 b2plot, fchp, 425
 b2plot, fchx, 425
 b2plot, fchx32, 425
 b2plot, fchx52, 425
 b2plot, fchxn, 425
 b2plot, fchxvspa, 425
 b2plot, fchxvspe, 425
 b2plot, fchy, 425
 b2plot, fchy32, 425
 b2plot, fchy52, 425
 b2plot, fchyn, 425
 b2plot, fchyvspa, 425
 b2plot, fchyvspe, 425
 b2plot, fclv, 433, 484
 b2plot, fcol, 433, 478
 b2plot, fetx, 425
 b2plot, fety, 425
 b2plot, ffbz, 425
 b2plot, fhctx, 425
 b2plot, fhety, 425
 b2plot, fhcx, 425
 b2plot, fhcy, 425
 b2plot, fhctx, 425
 b2plot, fhity, 425

b2plot, fhix, 425
 b2plot, fhiy, 425
 b2plot, fhjx, 425
 b2plot, fhjy, 425
 b2plot, fhmx, 425
 b2plot, fhmy, 425
 b2plot, fhpx, 425
 b2plot, fhpy, 425
 b2plot, fhtx, 425
 b2plot, fhty, 425
 b2plot, filo, 433, 484
 b2plot, flal, 425
 b2plot, flalc, 425
 b2plot, fle, 425
 b2plot, fli, 425
 b2plot, flke, 425
 b2plot, flki, 425
 b2plot, fluca, 425
 b2plot, fluce, 425
 b2plot, fluci, 425
 b2plot, flucs, 425
 b2plot, flucv, 425
 b2plot, flux, 437, 495
 b2plot, flvisc, 425
 b2plot, fmax, 435
 b2plot, fmin, 435
 b2plot, fmox, 426
 b2plot, fmoy, 426
 b2plot, fmx, 426
 b2plot, fmy, 426
 b2plot, fnax, 426, 485
 b2plot, fnax32, 426
 b2plot, fnax52, 426
 b2plot, fnaxnodr, 426
 b2plot, fnay, 426, 485
 b2plot, fnay32, 426
 b2plot, fnay52, 426
 b2plot, fnaynodr, 426
 b2plot, fnex, 426
 b2plot, fnex32, 426
 b2plot, fnex52, 426
 b2plot, fney, 426
 b2plot, fney32, 426
 b2plot, fney52, 426
 b2plot, fnix32, 426
 b2plot, fnix52, 426
 b2plot, fniy32, 426
 b2plot, fniy52, 426
 b2plot, font, 433, 475
 b2plot, fps, 431, 474
 b2plot, fraa, 426
 b2plot, gcol, 433, 478
 b2plot, gdcl, 426

b2plot, glbl, 437, 475
 b2plot, glob, 433, 476
 b2plot, glsz, 433, 476
 b2plot, graph appearance commands, 474
 b2plot, graph format commands, 472
 b2plot, graphformat, 472
 b2plot, grid, 433, 480
 b2plot, hcol, 433, 478
 b2plot, help, 447
 b2plot, hx, 426
 b2plot, hy, 426
 b2plot, hy1, 426
 b2plot, i*, 433, 464
 b2plot, i**, 433, 464
 b2plot, i+, 433, 463
 b2plot, i-, 433, 464
 b2plot, i/, 433, 464
 b2plot, ibnd, 440
 b2plot, idrop, 433, 468
 b2plot, idup, 433, 468
 b2plot, iix, 426
 b2plot, iiy, 426
 b2plot, imapx, 440
 b2plot, imapy, 440
 b2plot, imp, 433, 488
 b2plot, intrgw, 444, 453
 b2plot, intrgw*, 444, 453
 b2plot, intrgz, 445, 453
 b2plot, intw, 442, 453
 b2plot, intw*, 442, 453
 b2plot, intx, 431, 454
 b2plot, inty, 431, 454
 b2plot, ionrad, 426
 b2plot, iover, 433, 468
 b2plot, iprt, 433, 469
 b2plot, iseq, 426
 b2plot, isge, 426
 b2plot, isgt, 426
 b2plot, isle, 426
 b2plot, islt, 426
 b2plot, ismain, 446
 b2plot, isne, 426
 b2plot, iswap, 433, 468
 b2plot, isweq, 440
 b2plot, iswgt, 440
 b2plot, iswle, 441
 b2plot, iswlt, 441
 b2plot, iswne, 441
 b2plot, itoe, 433, 468
 b2plot, itom, 433, 468
 b2plot, itor, 433, 468
 b2plot, itostr, 433, 468
 b2plot, itot, 433, 469
 b2plot, itow, 433, 468
 b2plot, itrgeq, 443
 b2plot, itrgge, 443
 b2plot, itrsgt, 444
 b2plot, itrgle, 444
 b2plot, itrslt, 444
 b2plot, itrgne, 444
 b2plot, iweq, 441
 b2plot, iwge, 441
 b2plot, iwgt, 441
 b2plot, iwle, 441
 b2plot, iwlt, 441
 b2plot, iwne, 441
 b2plot, ix, 426
 b2plot, ixeq, 426
 b2plot, ixge, 426
 b2plot, ixgt, 426
 b2plot, ixle, 426
 b2plot, ixlt, 426
 b2plot, ixne, 426
 b2plot, ixref, 446
 b2plot, iy, 426
 b2plot, iyeq, 426
 b2plot, iyge, 426
 b2plot, iygt, 426
 b2plot, iyle, 426
 b2plot, iylt, 426
 b2plot, iyne, 426
 b2plot, iyref, 446
 b2plot, joul, 427
 b2plot, jsep, 446
 b2plot, jxa, 446
 b2plot, jxi, 446
 b2plot, kappa, 444
 b2plot, kappa_a, 441
 b2plot, kappa_a1, 441
 b2plot, kappa_a2, 441
 b2plot, kappa_a3, 441
 b2plot, kappa_a4, 441
 b2plot, kappa_a5, 441
 b2plot, kappa_ab, 444
 b2plot, kappa_b, 441
 b2plot, kappap, 444
 b2plot, kappapa1, 441
 b2plot, kappapa2, 441
 b2plot, kappapa3, 441
 b2plot, kappapa4, 441
 b2plot, kappapa5, 441
 b2plot, kappapdt, 444
 b2plot, kbolt, 446
 b2plot, ke, 427
 b2plot, kye, 427
 b2plot, ky, 427

b2plot, kyi0, 427
 b2plot, lacl, 433, 484
 b2plot, lbof, 433, 484
 b2plot, lbon, 433, 484
 b2plot, lbsa, 433, 484
 b2plot, lbsz, 434, 476
 b2plot, lcp, 441
 b2plot, ldepth, 441
 b2plot, lemis, 441
 b2plot, lfsz, 434, 476
 b2plot, lhor, 434, 484
 b2plot, line, 431, 449
 b2plot, linf, 434, 473
 b2plot, lint, 434, 450
 b2plot, lintrg, 445, 473
 b2plot, linw, 442, 481
 b2plot, linx, 434, 473
 b2plot, linxtrg, 445, 473
 b2plot, linxw, 442, 481
 b2plot, list, 431, 467
 b2plot, lix, 427
 b2plot, liy, 427
 b2plot, lkappa, 441
 b2plot, ln, 431, 457
 b2plot, lnlam, 427
 b2plot, lnn, 427
 b2plot, log, 431, 457
 b2plot, logf, 434, 473
 b2plot, logt, 434, 450
 b2plot, logtrg, 445, 473
 b2plot, logw, 442, 481
 b2plot, logx, 434, 473
 b2plot, logxtrg, 445, 473
 b2plot, logxw, 442, 481
 b2plot, lrho, 441
 b2plot, lthick, 441
 b2plot, ltt, 427
 b2plot, lwid, 434, 477
 b2plot, m*, 431, 458
 b2plot, m+, 431, 458
 b2plot, m-, 431, 458
 b2plot, m/, 431, 458
 b2plot, mach, 427
 b2plot, mach4, 427
 b2plot, massa, 438
 b2plot, massi, 438
 b2plot, massm, 438
 b2plot, mch2, 427
 b2plot, mcol, 434, 480
 b2plot, mdims, 431, 467
 b2plot, me, 446
 b2plot, meq, 431, 462
 b2plot, mesh, 434, 450
 b2plot, mesh plot, 497
 b2plot, mge, 431, 462
 b2plot, mgt, 431, 462
 b2plot, mhal, 427
 b2plot, mi, 427
 b2plot, mi_ext, 427
 b2plot, mle, 431, 463
 b2plot, mlt, 431, 463
 b2plot, mmax, 431, 458
 b2plot, mmid, 431, 474
 b2plot, mmin, 431, 458
 b2plot, mne, 431, 462
 b2plot, molrad, 427
 b2plot, movi, 437, 495
 b2plot, movie, 437, 496
 b2plot, mp, 446
 b2plot, mprr, 437, 486
 b2plot, mpsi, 431, 474
 b2plot, mtab, 437, 486
 b2plot, mu0, 446
 b2plot, mvsk, 434, 488
 b2plot, na, 427, 485
 b2plot, na0, 427
 b2plot, na_ext, 427
 b2plot, natm, 446
 b2plot, nbnd, 446
 b2plot, ndec, 434, 474
 b2plot, ndepth, 446
 b2plot, ne, 427
 b2plot, ne2, 427
 b2plot, ne_ext, 427
 b2plot, nei_ic, 484
 b2plot, neo_ic, 434
 b2plot, nese, 431, 488
 b2plot, neutrad, 427
 b2plot, new, 434, 477
 b2plot, nflfhix, 427
 b2plot, nflfhiy, 427
 b2plot, nflfnax, 427
 b2plot, nflfnay, 427
 b2plot, nh, 427
 b2plot, ni, 427
 b2plot, ni0, 427
 b2plot, nice, 434, 476
 b2plot, nion, 446
 b2plot, niso, 427
 b2plot, nmol, 446
 b2plot, nmovie, 437
 b2plot, noax, 434, 477
 b2plot, nocp, 434, 482
 b2plot, north, 441
 b2plot, nreg, 427
 b2plot, ns, 446

b2plot, nseg, 441
 b2plot, nspe, 431, 488
 b2plot, nspec, 431
 b2plot, nspecies, 446
 b2plot, nua, 427
 b2plot, nuab, 427
 b2plot, nustar, 427
 b2plot, nx, 446
 b2plot, nxl, 446
 b2plot, nxtr, 446
 b2plot, ny, 446
 b2plot, ocol, 434, 481
 b2plot, omax, 435
 b2plot, omega, 427
 b2plot, omin, 435
 b2plot, omovie, 437, 496
 b2plot, omp, 434, 488
 b2plot, oson, 437, 496
 b2plot, outl, 434, 481
 b2plot, outp, 437, 487
 b2plot, outptrg, 445, 487
 b2plot, outpw, 442, 487
 b2plot, over, 431, 467
 b2plot, owid, 434, 481
 b2plot, p*, 427
 b2plot, page, 434, 472
 b2plot, page format commands, 472
 b2plot, part, 434, 490
 b2plot, pbal, 427
 b2plot, pdena, 438
 b2plot, pdeni, 438
 b2plot, pdenm, 438
 b2plot, pdev, 437, 488
 b2plot, pefa, 427
 b2plot, pefm, 427
 b2plot, pfla, 427, 485
 b2plot, pflm, 427
 b2plot, phi, 427
 b2plot, phys, 434, 473
 b2plot, pi, 446
 b2plot, pit, 427
 b2plot, pitch, 427
 b2plot, plasma quantities, 423
 b2plot, plathick, 441
 b2plot, plot commands, 432
 b2plot, plotting plasma quantities, 497
 b2plot, plotting target quantities, 499
 b2plot, plotting wall quantities, 498
 b2plot, plwl, 442, 449
 b2plot, pne, 431, 454
 b2plot, po, 427
 b2plot, po0, 427
 b2plot, pos, 431, 463
 b2plot, pot, 427
 b2plot, pot*, 427
 b2plot, poti, 428
 b2plot, pr, 428
 b2plot, psi, 428
 b2plot, psta, 439, 454
 b2plot, pstv, 439, 454
 b2plot, pswa, 442, 454
 b2plot, psx, 432, 454
 b2plot, psxperp, 432, 454
 b2plot, psy, 432, 454
 b2plot, pux, 438
 b2plot, puy, 438
 b2plot, pvol, 432, 454
 b2plot, pvx, 438
 b2plot, pvy, 438
 b2plot, pwne, 442, 454
 b2plot, pz, 428
 b2plot, q, 428
 b2plot, qc, 428
 b2plot, qe, 446
 b2plot, qevap, 441
 b2plot, qrad, 441
 b2plot, quit, 447, 490
 b2plot, qz0, 428
 b2plot, qz1, 428
 b2plot, r*, 434, 459
 b2plot, r**, 434, 459
 b2plot, r+, 434, 459
 b2plot, r-, 434, 459
 b2plot, r/, 434, 459
 b2plot, r0, 435
 b2plot, r1, 441
 b2plot, r2, 441
 b2plot, radiation plots, 500
 b2plot, rc, 428
 b2plot, rco, 428
 b2plot, rcxi, 428
 b2plot, rcxm, 428
 b2plot, rcxn, 428
 b2plot, rdi, 428
 b2plot, rdi1, 428
 b2plot, rdop, 434, 467
 b2plot, rdup, 434, 467
 b2plot, real variables, 435
 b2plot, ree, 428
 b2plot, refa, 428
 b2plot, refm, 428
 b2plot, rei, 428
 b2plot, rem*, 439, 460
 b2plot, rem**, 439, 461
 b2plot, rem+, 439, 460
 b2plot, rem-, 439, 460

b2plot, rem/, 439, 461
 b2plot, remma, 439, 461
 b2plot, remmi, 439, 461
 b2plot, reso, 434, 487
 b2plot, rfla, 428, 485
 b2plot, rflm, 428
 b2plot, rhol, 428
 b2plot, rigz, 432, 434, 456
 b2plot, rix, 428
 b2plot, riy, 428
 b2plot, rlcl, 428
 b2plot, rm*, 432, 460
 b2plot, rm**, 432, 460
 b2plot, rm+, 432, 460
 b2plot, rm-, 432, 460
 b2plot, rm/, 432, 460
 b2plot, rmax, 435
 b2plot, rmin, 435
 b2plot, rmma, 432, 460
 b2plot, rmmx, 432
 b2plot, rmmi, 432, 460
 b2plot, rmmmin, 432
 b2plot, rmo, 428
 b2plot, rmt, 428
 b2plot, ro, 428, 485
 b2plot, rover, 434, 467
 b2plot, rpo, 428
 b2plot, rprr, 434, 468
 b2plot, rpsi, 432, 474
 b2plot, rqa, 428
 b2plot, rrai, 428
 b2plot, rram, 428
 b2plot, rran, 428
 b2plot, rsai, 428
 b2plot, rsam, 428
 b2plot, rsan, 428
 b2plot, rshi, 435
 b2plot, rswap, 434, 467
 b2plot, rtoc, 434, 467
 b2plot, rtom, 434, 467
 b2plot, rtotr, 434, 467
 b2plot, rtot, 434, 468
 b2plot, rtow, 434, 467
 b2plot, rtrgm*, 445, 461
 b2plot, rtrgm**, 445, 461
 b2plot, rtrgm+, 445, 461
 b2plot, rtrgm-, 445, 461
 b2plot, rtrgm/, 445, 461
 b2plot, rtrgmma, 445, 461
 b2plot, rtrgmmi, 445, 461
 b2plot, rud, 428
 b2plot, rudc, 428
 b2plot, rvd, 428
 b2plot, rwm*, 442, 462
 b2plot, rwm**, 442, 462
 b2plot, rwm+, 442, 462
 b2plot, rwm-, 442, 462
 b2plot, rwm/, 442, 462
 b2plot, rwmma, 442, 462
 b2plot, rwmmi, 442, 462
 b2plot, s, 428
 b2plot, same, 434, 477
 b2plot, saw, 437, 469
 b2plot, saws, 437, 469
 b2plot, sawx, 437, 470
 b2plot, scat, 437, 469
 b2plot, sch, 428
 b2plot, scol, 434, 480
 b2plot, sdop, 437, 469
 b2plot, sdup, 437, 469
 b2plot, sep, 434, 480
 b2plot, sepw, 434, 480
 b2plot, shco, 434, 490
 b2plot, she, 428
 b2plot, shi, 428
 b2plot, sig, 428
 b2plot, slbl, 437, 476
 b2plot, smaf, 428
 b2plot, smag, 428
 b2plot, smav, 428
 b2plot, smax, 435
 b2plot, smfr, 428
 b2plot, smin, 436
 b2plot, smo, 428
 b2plot, smp, 428
 b2plot, smpr, 429
 b2plot, smpt, 429
 b2plot, smq, 429
 b2plot, smth, 429
 b2plot, sna, 429
 b2plot, sonn, 437, 496
 b2plot, south, 441
 b2plot, sover, 437, 469
 b2plot, spec, 432, 487
 b2plot, spln, 434, 484
 b2plot, sprt, 437, 469
 b2plot, sput, 441
 b2plot, sput2, 441
 b2plot, sputchem, 441
 b2plot, sputfrac, 441
 b2plot, sputres, 441
 b2plot, sq, 435, 472
 b2plot, sqrt, 432, 457
 b2plot, srcml, 429
 b2plot, sswap, 437, 469
 b2plot, stack commands, 431

b2plot, stack manipulation, 466
 b2plot, stacks, 422
 b2plot, stckcoef, 441
 b2plot, stdl, 435, 475
 b2plot, stoe, 432, 470
 b2plot, stow, 432, 470
 b2plot, str, 429
 b2plot, sublheat, 441
 b2plot, sumt, 439, 455
 b2plot, sumtrgw, 445, 455
 b2plot, sumtrgz, 445, 455
 b2plot, sumw, 442, 455
 b2plot, sumx, 432, 455
 b2plot, sumy, 432, 455
 b2plot, sumz, 432, 455
 b2plot, surf, 435, 450
 b2plot, swap, 432, 466
 b2plot, swid, 435, 489
 b2plot, sx, 429
 b2plot, sxperp, 429
 b2plot, sy, 429
 b2plot, sz, 429
 b2plot, t.#p, 432, 450
 b2plot, t.pt, 436
 b2plot, t.r+, 435, 490
 b2plot, t.r-, 435, 490
 b2plot, ta_ext, 429
 b2plot, tab2, 429
 b2plot, target, 435, 481
 b2plot, target commands, 444
 b2plot, target quantities, 443
 b2plot, tclvd, 445, 483
 b2plot, tclvn, 445, 483
 b2plot, tclvp, 445, 483
 b2plot, tclvs, 445, 483
 b2plot, tcol, 435, 478
 b2plot, te, 429
 b2plot, te0, 429
 b2plot, tese, 432, 488
 b2plot, tgdns, 441
 b2plot, thermev, 441
 b2plot, ti, 429
 b2plot, ti0, 429
 b2plot, tib2, 429
 b2plot, time, 441
 b2plot, tindex, 438
 b2plot, tix, 429
 b2plot, tiy, 429
 b2plot, tlbl, 437, 476
 b2plot, tmb2, 429
 b2plot, traj, 432, 450
 b2plot, trajectories plot, 450
 b2plot, tranprof, 447, 487
 b2plot, trg.w, 445, 449
 b2plot, trg.z, 445, 449
 b2plot, trgabs, 445, 452
 b2plot, trgbll, 437, 476
 b2plot, trgchs, 445, 452
 b2plot, trgddp, 445, 453
 b2plot, trgddw, 445, 453
 b2plot, trgddz, 445, 453
 b2plot, trgdepth, 444
 b2plot, trgdrop, 445, 471
 b2plot, trgdup, 445, 471
 b2plot, trgeast, 444
 b2plot, trgeq, 445, 464
 b2plot, trgexp, 445, 458
 b2plot, trgfmax, 436
 b2plot, trgfmin, 436
 b2plot, trgge, 445, 464
 b2plot, trgg, 445, 464
 b2plot, trgibnd, 444
 b2plot, trigiz, 444
 b2plot, trgl, 444, 488
 b2plot, trgle, 445, 464
 b2plot, trglist, 445, 471
 b2plot, trgln, 445, 457
 b2plot, trglog, 445, 457
 b2plot, trgl, 445, 464
 b2plot, trgm*, 445, 465
 b2plot, trgm+, 445, 465
 b2plot, trgm-, 445, 465
 b2plot, trgm/, 445, 465
 b2plot, trgmmax, 445, 465
 b2plot, trgmmin, 445, 464
 b2plot, trgne, 445, 464
 b2plot, trgnorth, 444
 b2plot, trgoff, 435, 482
 b2plot, trgomax, 436
 b2plot, trgomin, 436
 b2plot, trgover, 445, 471
 b2plot, trgpos, 445, 465
 b2plot, trgprt, 437, 486
 b2plot, trgr1, 444
 b2plot, trgr2, 444
 b2plot, trgrmax, 436
 b2plot, trgrmin, 436
 b2plot, trgseg, 444
 b2plot, trgsouth, 444
 b2plot, trgsqrt, 445, 457
 b2plot, trgsurf, 445, 450
 b2plot, trgswap, 445, 471
 b2plot, trgtab, 437, 487
 b2plot, trgtemp, 444
 b2plot, trgtow, 446, 471
 b2plot, trgv, 444

b2plot, trgwest, 444
 b2plot, trgwidth, 444
 b2plot, trgwsel, 446, 456
 b2plot, trgxmax, 436
 b2plot, trgxmin, 436
 b2plot, trgymin, 436
 b2plot, trgz1, 444
 b2plot, trgz2, 444
 b2plot, trgzmax, 436
 b2plot, trgzmin, 436
 b2plot, trgzsel, 446, 456
 b2plot, triarea, 438
 b2plot, triavol, 438
 b2plot, trig, 435, 480
 b2plot, trigmesh, 435, 450
 b2plot, triple plot, 499
 b2plot, troubleshooting, 447
 b2plot, tt0, 436
 b2plot, ua, 429, 485
 b2plot, ua0, 429
 b2plot, ua_ext, 429
 b2plot, ud, 429
 b2plot, ue, 429
 b2plot, uedgegrd, 437, 496
 b2plot, up, 429, 485
 b2plot, user, 447, 489
 b2plot, uu, 429
 b2plot, uuse, 432, 450, 485
 b2plot, vadia.x, 429
 b2plot, vadia.y, 429
 b2plot, valfven, 429
 b2plot, vappr, 441
 b2plot, vappr_a, 441
 b2plot, vappr_b, 441
 b2plot, vaprtmp, 441
 b2plot, vcol, 435, 480
 b2plot, vcon, 429
 b2plot, vd, 429
 b2plot, vedia.x, 429
 b2plot, vedia.y, 429
 b2plot, veexb.x, 429
 b2plot, veexb.y, 429
 b2plot, vesl, 435, 480
 b2plot, vexb.x, 429
 b2plot, vexb.y, 429
 b2plot, visa, 429
 b2plot, visc, 429
 b2plot, vlax, 429
 b2plot, vlay, 429
 b2plot, vnneo, 429, 430
 b2plot, vol, 429
 b2plot, vth, 429
 b2plot, vthe, 429
 b2plot, vthi, 429
 b2plot, vv, 429, 485
 b2plot, vv0, 430
 b2plot, vvse, 432, 450, 485
 b2plot, vwid, 435, 480
 b2plot, vxa, 438
 b2plot, vxdena, 438
 b2plot, vxdeni, 438
 b2plot, vxdenm, 438
 b2plot, vxi, 438
 b2plot, vxm, 438
 b2plot, vya, 438
 b2plot, vydena, 438
 b2plot, vydeni, 438
 b2plot, vydenm, 438
 b2plot, vyi, 438
 b2plot, vym, 438
 b2plot, vza, 438
 b2plot, vzdena, 438
 b2plot, vzdeni, 438
 b2plot, vzdenm, 438
 b2plot, vzi, 438
 b2plot, vzm, 438
 b2plot, wabs, 442, 452
 b2plot, wadia.x, 430
 b2plot, wadia.y, 430
 b2plot, wall, 435, 442, 481
 b2plot, wall commands, 442
 b2plot, wall quantities, 440
 b2plot, wall_ank, 442, 490
 b2plot, wallarea, 441
 b2plot, wallblbl, 437, 476
 b2plot, walldist, 442
 b2plot, wallf, 442, 479
 b2plot, wallflux, 442
 b2plot, wallhflx, 442
 b2plot, wallngth, 442
 b2plot, walltemp, 442
 b2plot, wb, 430
 b2plot, wce, 430
 b2plot, wchs, 442, 452
 b2plot, wci, 430
 b2plot, wclvd, 442, 483
 b2plot, wclvn, 443, 483
 b2plot, wclvp, 443, 483
 b2plot, wclvs, 443, 483
 b2plot, wcol, 435, 478
 b2plot, wdrop, 443, 471
 b2plot, wdup, 443, 471
 b2plot, west, 442
 b2plot, wexp, 443, 458
 b2plot, wlbl, 437, 476

b2plot, wlist, 443, 472
 b2plot, wlld, 447
 b2plot, wln, 443, 458
 b2plot, wlog, 443, 458
 b2plot, wm*, 443, 465
 b2plot, wm+, 443, 465
 b2plot, wm-, 443, 465
 b2plot, wm/, 443, 465
 b2plot, wmax, 436
 b2plot, wmeq, 443, 466
 b2plot, wmge, 443, 466
 b2plot, wmg, 443, 466
 b2plot, wmin, 436
 b2plot, wmle, 443, 466
 b2plot, wmlt, 443, 466
 b2plot, wmmx, 443, 466
 b2plot, wmmn, 443, 465
 b2plot, wmne, 443, 466
 b2plot, woff, 443, 481
 b2plot, wover, 443, 471
 b2plot, wpe, 430
 b2plot, wpos, 443, 466
 b2plot, wp, 437, 443, 486
 b2plot, writ, 435, 486
 b2plot, write, 435
 b2plot, wsaw, 443, 470
 b2plot, wsel, 443, 456
 b2plot, wslbl, 437, 476
 b2plot, wsqrt, 443, 458
 b2plot, wsumz, 443, 455
 b2plot, wswap, 443, 471
 b2plot, wtab, 437, 486
 b2plot, wtos, 432, 443, 472
 b2plot, wtot, 443, 472
 b2plot, ww, 430
 b2plot, ww, 443, 481
 b2plot, wzrng, 443, 456
 b2plot, wzsel, 443, 456
 b2plot, xcol, 435, 478
 b2plot, xmax, 436
 b2plot, xmin, 436
 b2plot, xpar, 435, 474
 b2plot, xpol, 435, 474
 b2plot, xref, 436
 b2plot, xreg, 430
 b2plot, xsel, 432, 456
 b2plot, xtow, 432, 472
 b2plot, xymap, 430
 b2plot, xyrz, 435, 489
 b2plot, ymax, 436
 b2plot, ymin, 436
 b2plot, yref, 435, 488
 b2plot, yreg, 430
 b2plot, ysel, 432, 456
 b2plot, ytow, 432, 472
 b2plot, ytra, 435, 480
 b2plot, z0, 436
 b2plot, z1, 442
 b2plot, z2, 442
 b2plot, za, 430
 b2plot, za2_ext, 430
 b2plot, za_ext, 430
 b2plot, zcol, 435, 477
 b2plot, zeff, 430
 b2plot, zero, 435, 477
 b2plot, zi, 430
 b2plot, zmax, 436
 b2plot, zmin, 436
 b2plot, zn, 430
 b2plot, zn_ext, 430
 b2plot, zrng, 432, 455
 b2plot, zsel, 432, 456
 b2plot, zshi, 436
 b2plot, ztrgeq, 444
 b2plot, ztrgge, 444
 b2plot, ztrggt, 444
 b2plot, ztrgle, 444
 b2plot, ztrgl, 444
 b2plot, ztrgne, 444
 b2plot, zwid, 435, 477
 b2plot, tr, 438
 b2plot, tz, 438
 b2pp, 50
 b2pt, 50
 b2qie, 397
 b2radreg, 397
 b2rd, 123, 158, 572
 b2rd.prt, 574
 b2run, 50, 56, 60, 61, 66, 67, 69, 107, 109, 110, 113–116, 118, 121–123, 154
 b2run.hgw_clus, 50
 b2sext.sch_reg, 398
 b2sext.sch_reg_region, 43
 b2sext.she_reg, 398
 b2sext.she_reg_region, 43
 b2sext.shi_reg, 398
 b2sext.shi_reg_region, 43
 b2sext.smo_reg, 42, 398
 b2sext.smo_reg_region, 43
 b2sext.sna_reg, 42, 398
 b2sext.sna_reg_region, 43
 b2sext.sne_reg, 398
 b2sext.sne_reg_region, 43
 b2she, 397
 b2she0, 397
 b2shi, 397

b2shi0, 397
 b2specp, 110, 575
 b2sral, 83
 b2sral_inputfile, 83
 b2stbc, 70
 b2stbc_boundary_namelist, 70
 b2stbc_sch_reg, 397
 b2stbc_sch_reg_region, 42
 b2stbc_she_reg, 397
 b2stbc_she_reg_region, 42
 b2stbc_shi_reg, 397
 b2stbc_shi_reg_region, 42
 b2stbc_smo_reg, 42, 397
 b2stbc_smo_reg_region, 42
 b2stbc_sna_reg, 42, 397
 b2stbc_sna_reg_region, 42
 b2stbc_sne_reg, 397
 b2stbc_sne_reg_region, 42
 b2stbm_sch_reg, 398
 b2stbm_sch_reg_region, 43
 b2stbm_she_reg, 398
 b2stbm_she_reg_region, 43
 b2stbm_shi_reg, 398
 b2stbm_shi_reg_region, 43
 b2stbm_smo_reg, 42, 398
 b2stbm_smo_reg_region, 43
 b2stbm_sna_reg, 42, 398
 b2stbm_sna_reg_region, 43
 b2stbm_sne_reg, 398
 b2stbm_sne_reg_region, 43
 b2stbr_sch_reg, 397
 b2stbr_sch_reg_region, 43
 b2stbr_she_reg, 397
 b2stbr_she_reg_region, 42
 b2stbr_shi_reg, 398
 b2stbr_shi_reg_region, 43
 b2stbr_smo_reg, 42, 398
 b2stbr_smo_reg_region, 43
 b2stbr_sna_reg, 42, 398
 b2stbr_sna_reg_region, 43
 b2stbr_sne_reg, 398
 b2stbr_sne_reg_region, 43
 b2str, 398
 b2sxdr, 17, 50
 b2tallies.nc, 121, 397
 b2tallies.nc, b2bremreg, 397
 b2tallies.nc, b2divua, 397
 b2tallies.nc, b2divue, 397
 b2tallies.nc, b2exba, 397
 b2tallies.nc, b2exbe, 397
 b2tallies.nc, b2fraa, 397
 b2tallies.nc, b2joule, 397
 b2tallies.nc, b2qie, 397

b2tallies.nc, b2radreg, 397
 b2tallies.nc, b2sext_sch_reg, 398
 b2tallies.nc, b2sext_she_reg, 398
 b2tallies.nc, b2sext_shi_reg, 398
 b2tallies.nc, b2sext_smo_reg, 398
 b2tallies.nc, b2sext_sna_reg, 398
 b2tallies.nc, b2sext_sne_reg, 398
 b2tallies.nc, b2she, 397
 b2tallies.nc, b2she0, 397
 b2tallies.nc, b2shi, 397
 b2tallies.nc, b2shi0, 397
 b2tallies.nc, b2stbc_sch_reg, 397
 b2tallies.nc, b2stbc_she_reg, 397
 b2tallies.nc, b2stbc_shi_reg, 397
 b2tallies.nc, b2stbc_smo_reg, 397
 b2tallies.nc, b2stbc_sna_reg, 397
 b2tallies.nc, b2stbc_sne_reg, 397
 b2tallies.nc, b2stbm_sch_reg, 398
 b2tallies.nc, b2stbm_she_reg, 398
 b2tallies.nc, b2stbm_shi_reg, 398
 b2tallies.nc, b2stbm_smo_reg, 398
 b2tallies.nc, b2stbm_sna_reg, 398
 b2tallies.nc, b2stbm_sne_reg, 398
 b2tallies.nc, b2stbr_sch_reg, 397
 b2tallies.nc, b2stbr_she_reg, 397
 b2tallies.nc, b2stbr_shi_reg, 398
 b2tallies.nc, b2stbr_smo_reg, 398
 b2tallies.nc, b2stbr_sna_reg, 398
 b2tallies.nc, b2stbr_sne_reg, 398
 b2tallies.nc, b2str, 398
 b2tallies.nc, b2visa, 398
 b2tallies.nc, b2wrong1, 398
 b2tallies.nc, b2wrong2, 398
 b2tallies.nc, b2wrong3, 398
 b2tallies.nc, fchxreg, 398
 b2tallies.nc, fchyreg, 398
 b2tallies.nc, fhexreg, 398
 b2tallies.nc, fheyreg, 398
 b2tallies.nc, fhixreg, 398
 b2tallies.nc, fhiyreg, 398
 b2tallies.nc, fhjxreg, 398
 b2tallies.nc, fhjyreg, 398
 b2tallies.nc, fhmxreg, 398
 b2tallies.nc, fhmyreg, 398
 b2tallies.nc, fhpxreg, 398
 b2tallies.nc, fhpyreg, 398
 b2tallies.nc, fhtxreg, 398
 b2tallies.nc, fhtyreg, 398
 b2tallies.nc, fnaxreg, 398
 b2tallies.nc, fnayreg, 398
 b2tallies.nc, nareg, 398
 b2tallies.nc, ne2reg, 398
 b2tallies.nc, nereg, 398

b2tallies.nc, nireg, 398
 b2tallies.nc, poreg, 398
 b2tallies.nc, qconvexreg, 398
 b2tallies.nc, qconveyreg, 398
 b2tallies.nc, qconvixreg, 398
 b2tallies.nc, qconviyreg, 398
 b2tallies.nc, quantities stored in, 397
 b2tallies.nc, rcxhireg, 398
 b2tallies.nc, rcxmoreg, 398
 b2tallies.nc, rcxnareg, 399
 b2tallies.nc, rdneureg, 399
 b2tallies.nc, resco_reg, 399
 b2tallies.nc, reshe_reg, 399
 b2tallies.nc, reshi_reg, 399
 b2tallies.nc, resmo_reg, 399
 b2tallies.nc, resmt_reg, 399
 b2tallies.nc, respo_reg, 399
 b2tallies.nc, rqahereg, 399
 b2tallies.nc, rqbrmreg, 399
 b2tallies.nc, rquadreg, 399
 b2tallies.nc, rrahireg, 399
 b2tallies.nc, rramoreg, 399
 b2tallies.nc, rranareg, 399
 b2tallies.nc, rsahireg, 399
 b2tallies.nc, rsamoreg, 399
 b2tallies.nc, rsanareg, 399
 b2tallies.nc, tereg, 399
 b2tallies.nc, tireg, 399
 b2tallies.nc, volreg, 399
 b2time.nc, 121, 154, 390
 b2time.nc, an3da, 393
 b2time.nc, an3di, 393
 b2time.nc, an3dl, 392
 b2time.nc, an3dr, 393
 b2time.nc, an3dtl, 394
 b2time.nc, an3dtr, 394
 b2time.nc, batchsa, 395
 b2time.nc, dn3da, 393
 b2time.nc, dn3di, 393
 b2time.nc, dnsepm, 391
 b2time.nc, dp3da, 393
 b2time.nc, dp3di, 393
 b2time.nc, dpsepm, 391
 b2time.nc, fc3dl, 392
 b2time.nc, fc3dr, 393
 b2time.nc, fc3dtl, 394
 b2time.nc, fc3dtr, 394
 b2time.nc, fch2d, 395
 b2time.nc, fchsap, 392
 b2time.nc, fchsapp, 392
 b2time.nc, fchsip, 392
 b2time.nc, fchsipp, 392
 b2time.nc, fchxap, 390

b2time.nc, fchxip, 390
 b2time.nc, fchyap, 391
 b2time.nc, fchyip, 391
 b2time.nc, fe3dl, 392
 b2time.nc, fe3dr, 393
 b2time.nc, fe3dtl, 394
 b2time.nc, fe3dtr, 394
 b2time.nc, feesap, 392
 b2time.nc, feesapp, 392
 b2time.nc, feesip, 392
 b2time.nc, feesipp, 392
 b2time.nc, feexap, 390
 b2time.nc, feexip, 390
 b2time.nc, feeyap, 391
 b2time.nc, feeyip, 391
 b2time.nc, feisap, 392
 b2time.nc, feisapp, 392
 b2time.nc, feisip, 392
 b2time.nc, feisipp, 392
 b2time.nc, feixap, 390
 b2time.nc, feixip, 390
 b2time.nc, feiyap, 391
 b2time.nc, feiyip, 391
 b2time.nc, fetsap, 392
 b2time.nc, fetsapp, 392
 b2time.nc, fetsip, 392
 b2time.nc, fetsipp, 392
 b2time.nc, fetxap, 390
 b2time.nc, fetxip, 390
 b2time.nc, fetyap, 391
 b2time.nc, fetyip, 391
 b2time.nc, fhe2d, 395
 b2time.nc, fhi2d, 395
 b2time.nc, fi3dl, 392
 b2time.nc, fi3dr, 393
 b2time.nc, fi3dtl, 394
 b2time.nc, fi3dtr, 394
 b2time.nc, fl3dl, 392
 b2time.nc, fl3dr, 393
 b2time.nc, fl3dtl, 394
 b2time.nc, fl3dtr, 394
 b2time.nc, fn3dl, 392
 b2time.nc, fn3dr, 393
 b2time.nc, fn3dtl, 394
 b2time.nc, fn3dtr, 394
 b2time.nc, fna2d, 395
 b2time.nc, fnisap, 392
 b2time.nc, fnisapp, 392
 b2time.nc, fnisip, 391
 b2time.nc, fnisipp, 392
 b2time.nc, fnixap, 390
 b2time.nc, fnixip, 390
 b2time.nc, fniyap, 391

b2time.nc, fniyip, 391
 b2time.nc, fo3dl, 392
 b2time.nc, fo3dr, 393
 b2time.nc, fo3dtl, 394
 b2time.nc, fo3dtr, 394
 b2time.nc, ft3dl, 392
 b2time.nc, ft3dr, 393
 b2time.nc, ft3dtl, 394
 b2time.nc, ft3dtr, 394
 b2time.nc, ke3da, 393
 b2time.nc, ke3di, 393
 b2time.nc, kesepm, 391
 b2time.nc, ki3da, 393
 b2time.nc, ki3di, 393
 b2time.nc, kin2d, 394
 b2time.nc, ksepm, 391
 b2time.nc, mn3da, 393
 b2time.nc, mn3di, 393
 b2time.nc, mn3dl, 392
 b2time.nc, mn3dr, 393
 b2time.nc, mn3dtl, 394
 b2time.nc, mn3dtr, 394
 b2time.nc, nastep, 395
 b2time.nc, ne2d, 394
 b2time.nc, ne3da, 393
 b2time.nc, ne3di, 392
 b2time.nc, ne3dl, 392
 b2time.nc, ne3dr, 393
 b2time.nc, ne3dtl, 394
 b2time.nc, ne3dtr, 394
 b2time.nc, nemxap, 391
 b2time.nc, nemxap_av, 395
 b2time.nc, nemxap_std, 396
 b2time.nc, nemxip, 391
 b2time.nc, nemxip_av, 395
 b2time.nc, nemxip_std, 396
 b2time.nc, nesepa, 391
 b2time.nc, nesepa_av, 395
 b2time.nc, nesepa_std, 395
 b2time.nc, nesepi, 391
 b2time.nc, nesepi_av, 395
 b2time.nc, nesepi_std, 395
 b2time.nc, nesepm, 391
 b2time.nc, nesepm_av, 395
 b2time.nc, nesepm_std, 395
 b2time.nc, ntstep, 390
 b2time.nc, po2d, 394
 b2time.nc, po3da, 393
 b2time.nc, po3di, 393
 b2time.nc, po3dl, 392
 b2time.nc, po3dr, 393
 b2time.nc, po3dtl, 394
 b2time.nc, po3dtr, 394
 b2time.nc, pomxap, 391
 b2time.nc, pomxap_av, 395
 b2time.nc, pomxap_std, 396
 b2time.nc, pomxip, 391
 b2time.nc, pomxip_av, 395
 b2time.nc, pomxip_std, 396
 b2time.nc, posepa, 391
 b2time.nc, posepa_av, 395
 b2time.nc, posepa_std, 396
 b2time.nc, posepi, 391
 b2time.nc, posepi_av, 395
 b2time.nc, posepi_std, 395
 b2time.nc, posepm, 391
 b2time.nc, posepm_av, 395
 b2time.nc, posepm_std, 395
 b2time.nc, pwmxap, 391
 b2time.nc, pwmxip, 391
 b2time.nc, quantities stored in, 390
 b2time.nc, rcxhi2d, 394
 b2time.nc, rcxna2d, 395
 b2time.nc, rqahe2d, 395
 b2time.nc, rgrad2d, 395
 b2time.nc, rrahi2d, 394
 b2time.nc, rrana2d, 394
 b2time.nc, rsahi2d, 394
 b2time.nc, rsana2d, 394
 b2time.nc, te2d, 394
 b2time.nc, te3da, 393
 b2time.nc, te3di, 393
 b2time.nc, te3dl, 392
 b2time.nc, te3dr, 393
 b2time.nc, te3dtl, 394
 b2time.nc, te3dtr, 394
 b2time.nc, temxap, 391
 b2time.nc, temxap_av, 395
 b2time.nc, temxap_std, 396
 b2time.nc, temxip, 391
 b2time.nc, temxip_av, 395
 b2time.nc, temxip_std, 396
 b2time.nc, tesepa, 391
 b2time.nc, tesepa_av, 395
 b2time.nc, tesepa_std, 395
 b2time.nc, tesepi, 391
 b2time.nc, tesepi_av, 395
 b2time.nc, tesepi_std, 395
 b2time.nc, tesepm, 391
 b2time.nc, tesepm_av, 395
 b2time.nc, tesepm_std, 395
 b2time.nc, ti2d, 394
 b2time.nc, ti3da, 393
 b2time.nc, ti3di, 393
 b2time.nc, ti3dl, 392
 b2time.nc, ti3dr, 393

- b2time.nc, ti3dtl, 394
- b2time.nc, ti3dtr, 394
- b2time.nc, timesa, 390
- b2time.nc, timxap, 391
- b2time.nc, timxap_av, 395
- b2time.nc, timxap_std, 396
- b2time.nc, timxip, 391
- b2time.nc, timxip_av, 395
- b2time.nc, timxip_std, 396
- b2time.nc, tisepa, 391
- b2time.nc, tisepa_av, 395
- b2time.nc, tisepa_std, 395
- b2time.nc, tisepi, 391
- b2time.nc, tisepi_av, 395
- b2time.nc, tisepi_std, 395
- b2time.nc, tisepm, 391
- b2time.nc, tisepm_av, 395
- b2time.nc, tisepm_std, 395
- b2time.nc, tmhacore, 391
- b2time.nc, tmhadiv, 391
- b2time.nc, tmhasol, 391
- b2time.nc, tmne, 391
- b2time.nc, tmte, 391
- b2time.nc, tmti, 391
- b2time.nc, tp3dl, 392
- b2time.nc, tp3dr, 393
- b2time.nc, tp3dtl, 394
- b2time.nc, tpmxap, 391
- b2time.nc, tpmxip, 391
- b2time.nc, tpsepa, 391
- b2time.nc, tpsepi, 391
- b2time.nc, vs3da, 393
- b2time.nc, vs3di, 393
- b2time.nc, vssepm, 391
- b2time.nc, vx3da, 393
- b2time.nc, vx3di, 393
- b2time.nc, vxsepm, 391
- b2time.nc, vy3da, 393
- b2time.nc, vy3di, 393
- b2time.nc, vysepm, 391
- b2tqna, 77, 80
- b2tqna_inputfile, 80
- b2tqna_transport_namelist, 77
- b2ts, 572
- b2uf, 122, 572
- b2uf.prt, 574
- b2visa, 398
- b2wall.nc, 121
- b2wrong1, 398
- b2wrong2, 398
- b2wrong3, 398
- b2xd, 18, 572
- b2ydm, 127, 572

- b2ydm.dat, 127, 574
- b2ydm.prt, 574
- b2ye, 154, 572
- b2ye.prt, 574
- b2yg, 107, 573
- b2yg.dat, 574
- b2yg.prt, 574
- b2yh, 109, 573
- b2yh.dat, 574
- b2yh.prt, 574
- b2yi, 110, 573
- b2yi.dat, 574
- b2yi.prt, 574
- b2yi_gnuplot, 573
- b2ym, 114, 573
- b2ym.dat, 574
- b2ym.prt, 574
- b2ymb, 124, 573
- b2ymb.dat, 124, 574
- b2ymb.prt, 574
- b2yn, 115, 573
- b2yn.dat, 574
- b2yn.prt, 574
- b2yp, 110, 573
- b2yp.dat, 574
- b2yp.prt, 574
- b2yq, 116, 573
- b2yq.dat, 574
- b2yq.prt, 574
- b2yr, 113, 573
- b2yr.dat, 113, 574
- b2yr.prt, 574
- b2yrp, 125, 573
- b2yrp.dat, 125, 574
- b2yrp.prt, 574
- b2yt, 18, 50, 51, 118, 573
- b2yt.dat, 574
- b2yt.prt, 574
- balance, 54
- Balescu, 331, 349
- baserun, 36
- basic parameters, 64
- batchsa, 395
- BCCHAR, 70
- BCCON, 71
- BCEND, 71
- BCENE, 73
- BCENI, 74
- BCMOM, 72
- BCPOS, 71
- BCPOT, 76
- BCSTART, 71
- blank, 53

boundary conditions, 64, 70, 380
 boundary conditions, drifts, 77
 boundary conditions, drifts and currents, 323
 BOUNDARY_FILENAME, 77
 BOUNDARY_TIME_MOD, 77
 BOUNDARY_TIME_SWITCH, 77
 Braginskii, 331
 bremsstrahlung, 46
 BUILDING_CARRE, 528
 builds, 33
 Bulk_properties, 36

 calculate_sputter_inner_target.py, 53
 calculate_sputter_outer_target.py, 53
 canonical, 54
 Carre, compiling, 24, 190
 Carre, obtaining, 18
 Carre, running, 190
 Carre, Updating, 18
 CATALYST, 527
 CDF_MOVIE, 530
 cell corner indices, 430
 CFLAB, 79
 CFLAL, 79
 CFLME, 79
 CFLMI, 79
 CFLMV, 79
 check_b2_output, 47
 chop_pwd, 50
 Classical transport, 331
 CLASSICAL_CURRENT, 530
 climb_and_look_for, 50
 cnvtria, 54
 COCOS, 334
 commands, 54
 compare_2_solps.py, 53
 COMPILER_MAJOR_VERSION, 527
 COMPILER_MINOR_VERSION, 527
 Compiling the code, 24
 COMPLICATED, 530
 config, 33
 CONPAR, 76
 convert_from_solps4.3, 49
 Converting from SOLPS4 to SOLPS5, 17
 Converting from SOLPS5 to SOLPS4, 17
 copy_tracing_files, 50
 CORE_DT_FACTOR, 80
 CORE_DT_SUPPRESSION, 80
 CORR_CORE_DN, 80
 CORR_CORE_DT, 80
 correct_b2yt_timestamps, 50, 120
 correct_baserun_timestamps, 50, 51, 120
 count, 50
 Coupling to other codes via MDSplus, 183

 cpu, 47, 89
 CRAY, 527
 cropequ, 554
 currents.py, 47
 cview, 172

 D+_fluid_fluxes, 46
 D0_fluid_fluxes, 46
 d2d, 50, 552
 d2e, 50, 553
 d2v, 50, 553
 D_coupled_conservation, 46
 D_coupled_conservation_core, 46
 D_fluid_conservation, 46
 D_fluid_conservation_core, 46
 D_fluid_fluxes, 46
 data, 55
 database, 181
 database query via IDL, 181
 database query via python, 182
 datestamp, 50
 DBG, 527
 DBX, 30
 DEF_ISOEXTRA, 504
 DEF_MAXCHORDS, 504
 DEF_NADD, 504
 DEF_NATM, 504
 DEF_NBC, 504
 DEF_NCHEN, 504
 DEF_NCUT, 504
 DEF_NFL, 504
 DEF_NGSTAL, 504
 DEF_NION, 504
 DEF_NISO, 504
 DEF_NLIM, 505
 DEF_NMOL, 505
 DEF_NPHID, 505
 DEF_NPLS, 505
 DEF_NPRNL, 505
 DEF_NREAC, 505
 DEF_NSD, 505
 DEF_NSPCRFG, 505
 DEF_NSRFS, 505
 DEF_NSTRA, 505
 DEF_NSTS, 505
 DEF_NTOR, 505
 DEF_NXD, 505
 DEF_NYD, 505
 DEFINES, used in compiling B2.5, 530
 DEFINES, used in compiling B2.5-Eirene, 529
 DEFINES, used in compiling Carre, 528
 DEFINES, used in compiling DivGeo, 529
 DEFINES, used in compiling Eirene, 538
 DEFINES, used throughout SOLPS-ITER, 527

- delta, 47
- deltpl, 191
- deltpn, 191
- deltr1, 191, 192
- deltrn, 191, 192
- density, 47
- DEVICE, 30, 188
- DG, adding elements, 548
- DG, adding grid points, 559
- DG, adding surfaces, 557
- DG, changing external normals, 549
- DG, chop equilibrium, 554
- DG, chords, 560
- DG, configuration file, 567
- DG, connecting points, 548
- DG, convert to Carre equilibrium, 553
- DG, convert to EFIT equilibrium, 553
- DG, converting a template, 548
- DG, converting from Carre equilibrium, 554
- DG, converting from EFIT equilibrium, 553
- DG, converting from Naka equilibrium, 553
- DG, creating a source, 561
- DG, creating chords, 560
- DG, creating output files, 566
- DG, crop equilibrium, 554
- DG, customizing, 566
- DG, defining cells, 562
- DG, deleting a source, 561
- DG, deleting chords, 560
- DG, deleting elements, 549
- DG, deleting grid points, 559
- DG, deleting surfaces, 557
- DG, displaying numbers, 549
- DG, directory masks, 567
- DG, displaying objects, 550
- DG, doubling resolution, 552
- DG, eliminating unused numbers, 549
- DG, equilibrium, 554
- DG, extending chords, 560
- DG, file formats, 569
- DG, files needed by, 566
- DG, grid points, 559
- DG, hiding numbers, 549
- DG, hiding objects, 550
- DG, importing a SONNET grid, 561
- DG, importing an equilibrium, 554
- DG, installing, 566
- DG, joining elements, 549
- DG, limiter configurations, 558
- DG, menu bar, 546
- DG, meshes, 561
- DG, message line, 546
- DG, modifying a mesh, 561
- DG, modifying chords, 560
- DG, moving a source, 561
- DG, moving elements, 549
- DG, moving grid points, 559
- DG, moving surfaces, 557
- DG, new, 547
- DG, obtaining, 18
- dg, obtaining, 18
- DG, open, 547
- DG, output mode, 555
- DG, print equilibrium, 554
- DG, quit, 548
- DG, renumbering elements, 549
- DG, repairing cells, 562
- DG, repositioning elements, 549
- DG, rotating view, 550
- DG, running, 188
- DG, save, 547
- DG, shifting flux values, 553
- DG, sources, 561
- DG, splitting elements, 548
- DG, starting, 545
- DG, stretching and shrinking view, 550
- DG, surfaces, 555
- DG, template, 548
- DG, toolbar, 546
- DG, top-down view, 551
- DG, undo, 547
- DG, updating, 18
- dg, Updating, 18
- DG, using tools, 546
- DG, variable attributes, 563
- DG, variables, 563
- DG, variables, default value, 564
- DG, variables, export/no export, 564
- DG, variables, help text, 564
- DG, variables, name, 563, 564
- DG, variables, scope, 564
- DG, variables, setting, 564
- DG, variables, type, 564
- DG, work area, 546
- DG, X-point, 558
- DG, zoom, 550
- dg2dg, 552
- dg2ef, 553
- dg2vr, 553
- dgcut, 554
- DIMENSIONS.F, 23, 26, 47, 48, 504
- display_tallies, 47
- DivGeo, compiling, 24, 189
- DivGeo, obtaining, 18
- DivGeo, Updating, 18
- divheat, 84

dn3da, 393
 dn3di, 393
 dnsepm, 391
 dp3da, 393
 dp3di, 393
 dpmax, 191
 dpmin, 191
 dpsepm, 391
 drifts and currents, boundary conditions, 323
 drifts, boundary conditions, 77
 drmax, 191, 192
 drmin, 191, 192
 DTCO, 79
 DTEE, 80
 DTEL, 80
 DTMO, 79

 e2d, 50, 553
 EASY, 530
 ef2dg, 553
 Eirene, 37, 131
 Eirene, compilation, 38
 Eirene, compiling, 24
 Eirene, obtaining, 18
 Eirene, Updating, 18
 EIRENE_DEFINES, 30, 539, 540
 eiobj, 50
 eiobjx, 50
 ENEPAR, 76
 energy_analysis, 47, 89
 energy_analysis.py, 53, 89
 energy_analysis.fht, 47
 energy_analysis.total, 47
 energy_balance, 47
 energy_balance_core, 47
 energy_balance_core.extended, 47
 energy_balance_coupled, 47
 energy_balance_extended, 47
 ENIPAR, 76
 equations, 334
 equtrn, 552
 error assessment, 154
 EXTRA_GRAPHS, 528

 F2003, 527
 F77, 527
 fc3dl, 392
 fc3dr, 393
 fc3dtl, 394
 fc3dtr, 394
 fch2d, 395
 fchsap, 392
 fchsapp, 392
 fchsip, 392
 fchsipp, 392
 fchxap, 390
 fchxip, 390
 fchxreg, 47, 398
 fchyap, 391
 fchyip, 391
 fchyreg, 47, 398
 fe3dl, 392
 fe3dr, 393
 fe3dtl, 394
 fe3dtr, 394
 feesap, 392
 feesapp, 392
 feesip, 392
 feesipp, 392
 feexap, 390
 feexip, 390
 feeyap, 391
 feeyip, 391
 feisap, 392
 feisapp, 392
 feisip, 392
 feisipp, 392
 feixap, 390
 feixip, 390
 feiyap, 391
 feiyip, 391
 fetsap, 392
 fetsapp, 392
 fetsip, 392
 fetsipp, 392
 fetxap, 390
 fetxip, 390
 fetyap, 391
 fetyip, 391
 FFLAGSEXTRA, 30
 fhe2d, 395
 fhexreg, 47, 398
 fheyreg, 47, 398
 fhi2d, 395
 fhixreg, 47, 398
 fhiyreg, 47, 398
 fhjxreg, 47, 398
 fhjyreg, 47, 398
 fhmxreg, 47, 398
 fhmyreg, 47, 398
 fhpxreg, 47, 398
 fhpyreg, 47, 398
 fhtxreg, 47, 398
 fhtyreg, 47, 398
 fi3dl, 392
 fi3dr, 393
 fi3dtl, 394

- fi3dtr, 394
- first_setup, 50
- fixbb, 49
- fl3dl, 392
- fl3dr, 393
- fl3dtl, 394
- fl3dtr, 394
- FLAG_ALF, 78
- FLAG_DNA, 77
- FLAG_DPA, 77
- FLAG_HCE, 78
- FLAG_HCI, 78
- FLAG_SIG, 78
- FLAG_VLA, 78
- FLAG_VSA, 78
- fluid_core_fluxes.py, 53
- fluid_fluxes, 47
- fluid_fluxes.py, 53
- flux limits, 324
- fluxt, 47
- fn3dl, 392
- fn3dr, 393
- fn3dtl, 394
- fn3dtr, 394
- fna2d, 395
- fnaxreg, 398
- fnayreg, 398
- fnisap, 392
- fnisapp, 392
- fnisip, 391
- fnisipp, 392
- fnixap, 390
- fnixip, 390
- fniyap, 391
- fniyip, 391
- fo3dl, 392
- fo3dr, 393
- fo3dtl, 394
- fo3dtr, 394
- fort.30, 60
- fort.44, 148
- fort.46, 148
- FreeGS, 553
- ft3dl, 392
- ft3dr, 393
- ft3dtl, 394
- ft3dtr, 394
- G95, 527
- GAMMAE, 76
- GAMMAI, 76
- get_git_email, 51
- get_git_name, 51
- get_parm.bpl, 51
- get_solps_fluids, 49
- GFORTRAN, 527
- GGD_MINOR_VERSION, 527
- GIT repository log, SOLPS-ITER, 576
- GKS_ROOT, 30
- GLI_HOME, 30
- GNUPLOT_CMD, 44
- GNUPLOT_COUNT, 44
- GNUPLOT_DATA, 44
- GNUPLOT_DELAY, 44
- GNUPLOT_EXTRA, 44
- GNUPLOT_LINEWIDTH, 44
- GNUPLOT_MAX_POINTS, 44
- GNUPLOT_MAX_POINTS_FACTOR, 44
- GNUPLOT_NOTIMESTAMP, 44
- GNUPLOT_POINTSIZE, 44
- GNUPLOT_POSTSCRIPT_OPTIONS, 44
- GNUPLOT_TERM, 44
- GNUPLOT_TMP, 44
- GOT_AMNSPROTO, 530
- GR_ROOT, 30
- grf, 49
- gtdr, 51
- gtfl, 51
- H2VIBR, 38
- H5DIR, 30
- He_coupled_conservation_core, 46
- He_fluid_conservation, 46
- He_fluid_fluxes, 46
- HITS_PER_CHORD, 505
- Hostnames, 21
- HPUX, 527
- HYDHEL, 38
- IBMaix, 527
- idl, 173
- IDS, Interface Data Structures, 128
- IMAS, 21, 30, 527
- IMAS_MINOR_VERSION, 527
- import_files_from_convert, 51, 120
- incatm, 36
- include, 32
- Initial set-up, 21
- input.dat, 51, 118
- input.dat.2, 118
- input.dat_to_mesh.extra, 51
- inspect_plasmastate, 49, 121
- internal_energy_balance, 48
- internal_energy_balance_coupled, 48
- ionization_potentials, 35
- ippsubmit, 51, 57
- ISOEXTRA, 504
- ITER_divertor_HTM, 51

- itersubmit, 51, 57
- ITM, 528
- ITM_ENVIRONMENT_LOADED, 528
- itmsubmit, 51, 57
- JET, 530
- jetdsp, 172
- jetsubmit, 51, 57
- jfrssubmit, 51, 57
- joule_heating, 48
- jxa, 390
- jxi, 390
- ke3da, 393
- ke3di, 393
- keseqm, 391
- ki3da, 393
- ki3di, 393
- kin2d, 394
- kind_coeff, 82
- kisepm, 391
- last10, 51
- LBNDUSR, 77
- LCBS, 77
- LFEEDBACK, 77
- lg, 191
- lln, 51
- lnr, 51
- lns, 189, 559
- localsubmit, 51, 57
- lxlyplot, 45
- lxlyplot2, 45
- lxlyplot3, 45
- lxlyplot4, 46
- lxlyplot5, 46
- lxlyplot6, 46
- lxlyplot7, 46
- lxlyplot8, 46
- lxlyplot9, 46
- lxyplot, 45
- lxyplot2, 45
- lxyplot3, 45
- lxyplot4, 45
- lxyplot5, 45
- lxyplot6, 45
- lxyplot7, 45
- lxyplot8, 45
- lxyplot9, 45
- Makefile, 24
- marconisubmit, 51, 57
- Matlab, 54
- matlab, 174

- mds_id, 51
- MDSPLUS, 158, 528
- MDSplus SOLPS tree structure, 159
- MDSPLUS_DIR, 30
- mesh.extra, 51
- METHANE, 38
- min_ave_max.awk, 54
- mn3da, 393
- mn3di, 393
- mn3dl, 392
- mn3dr, 393
- mn3dtl, 394
- mn3dtr, 394
- model, 334
- MODIF, 528
- modifying the source of B2.5, 33
- modules, 32
- MOMPAR, 76
- move_optional_files, 51
- move_save_files, 51
- movie, python, 177
- MPI_VERSION, 538
- MSCL_ROOT, 30
- mtv_3d, 48
- mtv_3d_f, 48
- mtv_3d_nf, 48
- mtv_3d_t, 48
- MySQL, 181
- n2d, 51, 553
- NADD, 504
- NAG, 20
- NAG calls in B2.5, 540
- NAG calls in EIRENE, 541
- NAG calls in SOLPS, 540
- NAG22, 528
- NAGDIR, 30
- NAGF95, 528
- nareg, 48, 398
- nareg.py, 53
- nareg_species, 48
- nareg_species.py, 53
- nastep, 395
- NATM, 504
- NBC, 70, 504
- nc2text, 21, 51, 54
- ncar.cfg, 107, 110, 113–115, 191
- NCAR4, 528
- NCARG_ROOT, 30
- NCDIR, 30
- NCHEN, 504
- NCUT, 504
- ndata, 82
- ne2d, 394

- ne2reg, 398
- ne3da, 393
- ne3di, 392
- ne3dl, 392
- ne3dr, 393
- ne3dtl, 394
- ne3dtr, 394
- nemxap, 391
- nemxap_av, 395
- nemxap_std, 396
- nemxip, 391
- nemxip_av, 395
- nemxip_std, 396
- neoart, 484
- neoclassical, 484
- nereg, 48, 398
- nereg.py, 53
- nesepa, 391
- nesepa_av, 395
- nesepa_std, 395
- nesepi, 391
- nesepi_av, 395
- nesepi_std, 395
- nesepm, 391
- nesepm_av, 395
- nesepm_feedback, 48
- nesepm_feedback_by_core, 48
- nesepm_feedback_by_pfr, 48
- nesepm_feedback_fluid, 48
- nesepm_std, 395
- netcdf, 20
- neutral sources, coupling, 328
- neutrals, first-flight model, 328
- neutrals, fluid model, 326
- neutrals, flux limits, 327
- neutrals, transport coefficients, 327
- NFL, 504
- NGSTAL, 504
- NIISO, 77
- NION, 504
- nireg, 48, 398
- NISO, 504
- nk2dg, 553
- NLIM, 505
- NMOL, 505
- nn2d, 51, 553
- NNISO, 77
- NO_21, 530
- NO_B2_CHEM_SPUT, 529
- NO_CDF, 528
- no_div, 82
- NO_F01AAF, 530
- NO_GETENV, 528

- NO_IEEE, 530
- NO_LONG_LINES, 528
- NO_NAG, 528
- NO_OPENMP, 530
- NO_OPT, 530
- no_pflux, 82
- NO_SAVE, 529
- noquote_pwd, 52
- NPHID, 505
- NPLS, 505
- npr, 191, 192
- NPRNL, 505
- nptseg, 191
- NREAC, 505
- NREG, 43, 49, 79, 411
- nrelax, 192
- NSD, 505
- nsdata, 84
- NSPCSRFG, 505
- NSRFS, 505
- NSTRA, 505
- NSTS, 505
- NTOR, 505
- ntstep, 390
- NXD, 505
- nxdata, 84
- NXISO1, 77
- NXISO2, 77
- NYD, 505
- NYISO1, 77
- NYISO2, 77
- obtain_tracing_files, 52
- Obtaining the code, 18
- palettes, 54
- PAR_OPT, 530
- parallel viscosity, 322
- PARM_ALF, 79
- PARM_DNA, 79
- PARM_DPA, 79
- PARM_HCE, 79
- PARM_HCI, 79
- PARM_SIG, 79
- PARM_VLA, 79
- PARM_VSA, 79
- particle_conservation_coupled, 48
- pasmin, 191, 192
- PERFMON, 30, 530, 541
- performance monitoring, 541
- PGF90, 528
- PHIISO, 77
- pipe_b2.log, 52
- pipe_run.log, 52

plasmastate, 52, 121
 Platforms, 22
 plot, 43
 plot_bc_conpar13, 54
 plot_cps, 49
 plot_it, 49
 plot_mesh, 52
 plot_png, 45
 plot_ps, 49
 plot_rzpsi.dat, 49
 plot_set, 49
 plotdg, 54
 PLOTPOINTS, 44
 PLPLOT, 530
 pntrat, 192
 po2d, 394
 po3da, 393
 po3di, 393
 po3dl, 392
 po3dr, 393
 po3dtl, 394
 po3dtr, 394
 pomxap, 391
 pomxap_av, 395
 pomxap_std, 396
 pomxip, 391
 pomxip_av, 395
 pomxip_std, 396
 poreg, 398
 posepa, 391
 posepa_av, 395
 posepa_std, 396
 posepi, 391
 posepi_av, 395
 posepi_std, 395
 posepm, 391
 posepm_av, 395
 posepm_std, 395
 postprocessing, 54
 POTPAR, 76
 preserve_scratch_to_work, 52
 prinequ, 554
 PRINT_PLOT, 45
 print_species, 48
 print_versions, 52
 PSR, 528
 pwmzap, 391
 pwmzap, 391
 python, 174
 Python scripts, 53
 python, database query, 182
 python, movie, 177
 python, plots to compare profiles, 174
 qconvexreg, 398
 qconveyreg, 398
 qconvixreg, 398
 qconviyreg, 398
 QSUB files, 57
 QSUB.postprocess, 52
 QSUB.postprocess_coupled, 52
 querying database, 181
 quote_pwd, 52
 ratadas.filelist, 36
 rcxhi2d, 394
 rcxhireg, 42, 398
 rcxhireg_region, 43
 rcxmoreg, 42, 398
 rcxmoreg_region, 43
 rcxna2d, 395
 rcxnareg, 42, 399
 rcxnareg_region, 43
 rdneureg, 42, 399
 READOPT, 529
 recover_initial_files, 52, 121
 reflection, 48
 reflection_energy, 48
 regions, 80, 411
 regions, 1-D cases, 412
 regions, CDN cases, 416
 regions, DDN cases, 418
 regions, limiter cases, 413
 regions, periodic BC stellarator island cases, 420
 regions, SN cases, 414
 regression_test, 52
 regxy_sum_over_species, 54
 relax, 192
 repart, 191
 Requirements, 20
 res_av, 52, 153
 res_D+C, 41
 res_D+C+Ar+Ne+He, 41
 res_D+C+He, 41
 res_D+O, 41
 resall_D, 41, 89
 resall_D+_only, 41
 resall_D+D, 41
 resall_D+T, 41
 resall_H, 41
 resave_mds, 52
 RESCALE_BR, 88
 RESCALE_CX, 88
 RESCALE_QA, 88
 RESCALE_RA, 88
 RESCALE_RD, 88
 RESCALE_SA, 88
 resco, 41

resco_D+C, 41
 resco_D+C+He, 41
 resco_D+O, 41
 resco_reg, 41, 399
 resco_reg_region, 41
 reshe_reg, 41, 399
 reshi_reg, 41, 399
 resmo, 41
 resmo_D+C, 41
 resmo_D+C+He, 41
 resmo_D+O, 41
 resmo_reg, 41, 399
 resmo_reg_region, 41
 resmod, 42
 resmt_reg, 41, 399
 resopt, 42
 respo, 41
 respo_num, 42
 respo_reg, 42, 399
 resrest, 41
 reviewplus, 173
 risepsi, 553
 rlcept, 192
 rlndr, 52
 rps, 52, 553
 rqahe2d, 395
 rqahereg, 42, 399
 rqahereg_region, 43
 rqbrmreg, 42, 399
 rqbrmreg_region, 43
 rqr2d, 395
 rqr2dreg, 42, 399
 rqr2dreg.py, 53
 rqr2dreg_region, 43
 rrahi2d, 394
 rrahireg, 42, 399
 rrahireg_region, 43
 rramoreg, 42, 399
 rramoreg_region, 43
 rrana2d, 394
 rranareg, 42, 399
 rranareg.py, 53
 rranareg_region, 43
 rsahi2d, 394
 rsahireg, 42, 399
 rsahireg_region, 43
 rsamoreg, 42, 399
 rsamoreg_region, 43
 rsana2d, 394
 rsanareg, 42, 399
 rsanareg.py, 53
 rsanareg_region, 43
 Run set-up, 26
 run_av, 52, 153
 save_initial_files, 52
 save_mds, 52
 sb2, 33
 sbr, 40
 scr, 41
 sdata, 84
 sei, 37
 sellan, 192
 setup.csh, 26, 27, 30
 setup_baserun_eirene_links, 52, 135
 sgesubmit, 52, 57
 shotnumber.history, 123–125, 127
 sign conventions, 334
 slib, 40
 SLURM files, 57
 SLURM.postprocess, 52
 SLURM.postprocess_coupled, 52
 SOLPS review, 315
 SOLPS tree structure in MDSplus, 159
 SOLPS-GUI, 18
 SOLPS-ITER, compiling, 24
 SOLPS-ITER, obtaining, 18
 SOLPS-ITER, Updating, 18
 solps-iter_update, 52
 solps-iter_update_develop, 52
 SOLPS4_3, 529
 SOLPS4to5, 52
 SOLPS_CPP, 30, 528, 530, 539, 540
 SOLPS_FLUIDS, 43, 49, 118
 SOLPSTOP file, 57
 SOLVECO, 80
 SOLVEEEE, 80
 SOLVEEI, 80
 SOLVEMO, 80
 SOLVEPO, 80
 sources, 83
 species, 49
 SPUTER, 38
 sputter, 48
 sputter_chemical, 48
 sputter_chemical_energy, 48
 sputter_energy, 48
 sputter_physical, 48
 sputter_physical_energy, 48
 sputter_RES, 48
 sputter_RES_energy, 48
 src, 33
 src.local, 33
 ssb, 33
 stop, 31
 str2ogr, 52
 stra.dat, 36, 66, 574

STRAHL, 36, 66
 subversion repository logs, 576
 summarize_run, 53
 Surface_properties, 36
 switch_to_master, 53

 tdata, 82
 te2d, 394
 te3da, 393
 te3di, 393
 te3dl, 392
 te3dr, 393
 te3dtl, 394
 te3dtr, 394
 TEISO, 77
 temxap, 391
 temxap_av, 395
 temxap_std, 396
 temxip, 391
 temxip_av, 395
 temxip_std, 396
 tereg, 48, 399
 tereg.py, 53
 tesepa, 391
 tesepa_av, 395
 tesepa_std, 395
 tesepi, 391
 tesepi_av, 395
 tesepi_std, 395
 tesepm, 391
 tesepm_av, 395
 tesepm_std, 395
 tflux, 48
 thermal_evaporation, 48
 thermal_evaporation_energy, 48
 ti2d, 394
 ti3da, 393
 ti3di, 393
 ti3dl, 392
 ti3dr, 393
 ti3dtl, 394
 ti3dtr, 394
 TIISO, 77
 time-dependent quantities, b2tallies.nc, 397
 time-dependent quantities, b2time.nc, 390
 time_dep, 48
 TIME_FACTOR_REQUIRED, 80
 timesa, 390
 timxap, 391
 timxap_av, 395
 timxap_std, 396
 timxip, 391
 timxip_av, 395
 timxip_std, 396

 tireg, 49, 399
 tireg.py, 53
 tisepa, 391
 tisepa_av, 395
 tisepa_std, 395
 tisepi, 391
 tisepi_av, 395
 tisepi_std, 395
 tisepm, 391
 tisepm_av, 395
 tisepm_std, 395
 tmhacore, 391
 tmhadv, 391
 tmhasol, 391
 tmne, 391
 tmte, 391
 tmti, 391
 tp3dl, 392
 tp3dr, 393
 tp3dtl, 394
 tpmxap, 391
 tpmxip, 391
 tpsepa, 391
 tpsepi, 391
 TRACE, 539
 tracing, 121
 tran, 575
 transfer_solps-iter_runs, 53
 transport coefficients, 65, 77, 80
 TRANSPORT_FILENAME, 79
 TRANSPORT_TIME_MOD, 79
 TRANSPORT_TIME_SWITCH, 79
 tree structure in MDSplus, 159
 triang, 53, 120
 triang.conf, 53
 TRIM, 39
 trim.data, 36
 truncate_tracing_files, 53

 UAL, 30
 UAL_MAJOR_VERSION, 528, 530
 Updating the code, 18
 USE_CDF_B2STIME, 530
 USE_IMPGYRO, 529
 USE_MPI, 528
 USE_PXFGETENV, 528
 ustcsbmit, 53, 57

 v2d, 53, 554
 viscous_heating, 49
 volreg, 399
 vr2dg, 554
 vreg, 411
 vs3da, 393

vs3di, 393
vssepm, 391
vx3da, 393
vx3di, 393
vxsepm, 391
vy3da, 393
vy3di, 393
vysepm, 391

wall power loading, 447
wall segments, 498
WANT_PROCESS, 530
wavelength.to_energy, 49
weis.dat, 36, 66, 574
Weisheit, 66
width, 191, 192
WINDOWS, 528
Windows, 23
WOS, 530

xdata, 84
xlyplot, 45
xlyplot2, 45
xlyplot3, 45
xlyplot4, 45
xlyplot5, 45
xlyplot6, 45
xlyplot7, 45
xlyplot8, 45
xlyplot9, 45
xreg, 411
xyplot, 45
xyplot2, 45
xyplot3, 45
xyplot4, 45
xyplot5, 45
xyplot6, 45
xyplot7, 45
xyplot8, 45
xyplot9, 45

yreg, 411

zeffreg, 49
zeffreg.py, 53
Zhdanov, 331, 349
Zope, 181