Jül-4337

Institute of Energy and Climate Research (IEK)
Plasma Physics (IEK-4)

B2–B2.5 Code Benchmarking

W. Dekeyser, M. Baelmans,
S. Voskoboynikov, V. Rozhansky,
D. Reiter, S. Wiesen, V. Kotov, P. Börner

JÜLICH
FORSCHUNGSZENTRUM

B2–B2.5 Code Benchmarking

W. Dekeyser, M. Baelmans,
S. Voskoboynikov, V. Rozhansky,
D. Reiter, S. Wiesen, V. Kotov, P. Börner

Berichte des Forschungszentrums Jülich; 4337
ISSN 0944-2952
Institute of Energy and Climate Research (IEK)
Plasma Physics (IEK-4)
Jül-4337

Vollständig frei verfügbar im Internet auf dem Jülicher Open Access Server (JUWEL)
unter http://www.fz-juelich.de/zb/juwel

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek, Verlag
D-52425 Jülich · Bundesrepublik Deutschland
☎ 02461 61-5220 · Telefax: 02461 61-6103 · e-mail: zb-publikation@fz-juelich.de

Preface

ITER-IO currently (and since about 15 years) employs the SOLPS4.xxx code for its divertor design, currently version SOLPS4.3. SOLPS.xxx is a special variant of the B2-EIRENE code, which was originally developed by an European consortium (FZ Jülich, AEA Culham, ERM Belgium/KU Leuven) in the late eighties and early nineties of the last century under NET contracts. The particular code version SOLPS4.3 is at present jointly maintained and upgraded at ITER-IO and FZ Jülich. Other versions of B2-EIRENE (and SOLPS.xxx) have been advanced by other research groups, notably at IPP Garching, in various different directions. Most important amongst those for our present work is the SOLPS5.2 version with the probably most advanced model available today for classical drifts, currents and electric fields in the plasma fluid module, developed by V. Rozhansky and co-workers (St. Petersburg State Polytechnical University).

Surprisingly, merging the various versions of B2-EIRENE into one single, all-including code package requires significant effort. Until today even the very similar edge plasma codes within the SOLPS family, if run on a seemingly identical choice of physical parameters, still sometimes disagree significantly with each other.

It is obvious that in computational engineering applications, as they are carried out for the various ITER divertor aspects with SOLPS4.3 for more than a decade now, any transition from one to another code must be fully backward compatible, or, at least, the origin of differences in the results must be identified and fully understood quantitatively.

No computational engineer can allow his ability to solve an urgent design problem be limited by somebody else's decision to change a code.

Already in the past significant effort went into code-code benchmarking, even within the SOLPS family, but sometimes with limited success. We believe that one of the shortcomings of all previous code-code benchmarks was the application of the entire SOLPS model to highly integrated, multi-physics cases, i.e., to much too complex test problems, which has prevented the community from identifying all the details of code version differences on a fully quantitative level. And there are a number of ways by which different numerical results on the same problem can result. The most important of which probably are:

- unintentional choice of different code input options
- numerical effects, such as grid size, or convergence level
- not fully documented internal code changes, which affect the underlying physical model

The first and last point is a matter of code documentation and code operation. The second issue is unavoidable, but the effects can be quantified, e.g. by running physically identical cases on different grid sizes, or by measures of convergence levels, such as residuals (see Section 3.3 of the present report).

In this report we document efforts undertaken in 2010 to ultimately eliminate the third issue. For the kinetic EIRENE part within SOLPS this backward compatibility (back until 1996) was basically achieved (V. Kotov, 2004-2006) and SOLPS4.3 is now essentially up to date with the current EIRENE master maintained at FZ Juelich [1]. In order to achieve a similar level of reproducibility for the plasma fluid (B2, B2.5) part, we follow a similar strategy, which is quite distinct from the previous SOLPS benchmark attempts: the codes are *"disintegrated"* and pieces of it are run on smallest (i.e. simplest) problems. Only after full quantitative understanding is achieved, the code model is enlarged, integrated, piece by piece again, until, hopefully, a fully backward compatible B2 / B2.5 ITER edge plasma simulation will be achieved.

The status of this **code dis-integration** effort and its findings until now (Nov. 2010) are documented in the present technical note. This work was initiated in a small workshop by the three partner teams of KU Leuven, St. Petersburg State PU and FZ Jülich, held at FZ Jülich in February 2010, and it has been carried out as a preparatory step for a wider SOLPS re-unification activity foreseen in the coming year under a F4E (Fusion for Energy, Barcelona) ITER service contract.

The material of this report (with exception of appendix D) was made available to ITER IO Cadarache in November 2010, and, slightly later, presented at IPP Garching at a wider “SOLPS meeting" by one of the authors: S. Voskoboynikov.

Contents

1 Introduction **1**

2 Modeling Equations **4**
2.1 Continuity Equation 4
2.2 Momentum Equation 6
2.2.1 Viscosity coefficients η_x^i and η_y^i 6
2.2.2 Viscous flux limiters 8
2.2.3 Remarks 9
2.3 Electron Energy Equation 10
2.3.1 Electron heat conductivity coefficients κ_x^e and κ_y^e 11
2.3.2 Electron-ion energy equilibration term 12
2.3.3 Remarks 12
2.4 Ion Energy Equation 13
2.4.1 Ion heat conductivity coefficients κ_x^i and κ_y^i 14
2.4.2 The viscous energy terms 15
2.4.3 Remarks 15

3 Benchmark Test of Modeling Equations **17**
3.1 Geometry 17
3.2 Boundary Conditions 18
3.3 Results and Discussion 20

4 Conclusion **36**

A Derivation of the Internal Energy Equation **37**

B The Discretization Error **40**

C Input Parameters **42**
C.1 B2 Input Parameters 42
C.2 B2.5 Input Parameters 43

D Odd-even decoupling Oscillations in B2.5 **45**
D.1 Description of the test case 45
D.2 Results and Discussion 46

E B2 Bug Fixes **50**
E.1 Interpolation of radial viscous momentum flux 50
E.2 Correction to viscous energy terms 51

1 Introduction

In this report an overview is given of dedicated code-code benchmark results between the B2 code [2] (plasma fluid solver in SOLPS4.3) and B2.5 code [3] (plasma fluid solver in SOLPS5.2). For the purpose of this report both are operated in a mode isolated from the rest of SOLPS.

The goal of this work is to compare the performance and accuracy of the B2 and B2.5 code modules within SOLPS.

The main result of this preparatory study is that a reduced set of equations, boundary conditions, and configuration has been identified, on which both plasma fluid codes B2 and B2.5 now can be shown to give exactly the same solution, well within the numerical discretization errors, which have been quantified by grid refinement procedures. A number of input flags, physical parameters, coding details etc. had to be identified to achieve this common starting point for a wider code-code comparison study, which was launched under a F4E ITER service contract (now started, on Dec. 17th, 2010).

Distinct from earlier attempts [4, 5, 6] to identify differences between these two particular versions of B2-EIRENE codes from the SOLPS family, which apparently were based solely on comparing integrated and quite complex full edge modeling cases, our approach here is to first disintegrate the code packages and identify both physical and numerical differences, one by one. We then will, gradually, increase the complexity of the test cases until finally, hopefully, a fully backward-compatible ITER divertor modeling case is recovered.

This same strategy has already lead in the past (2004, V. Kotov et al.) to a successful upgrading of EIRENE from the 1996 version to the current master version (2009) maintained at FZJ, for the ITER-IO version of B2-EIRENE (SOLPS4.3), and similarly in 2007, when the EIRENE code replaced the older NIMBUS Monte Carlo solver in the JET suit of edge transport codes (S. Wiesen, 2007).

In this technical report we describe the status of corresponding work for the plasma fluid solvers B2 (Version: ITER-IO, Cadarache) and B2.5 (Version: St. Petersburg State Polytechnical University).

As starting point we check whether the 'basic physics' implemented in both codes is indeed the same, or, if not, what the differences are, quantitatively. Indeed, in the first documentation of the B2.5 code [3] a number of modifications as compared to B2 have been indicated by the code author (Bas Braams):

- "parallel heat and momentum transport is classical, following Braginskii [ll] **and Balescu [12]**, but flux-limited"
- "The charge conservation equation, which governs the electric potential...", Eq. (3) in [3], is added.
- The heat balance equation for ions is formulated for the internal energy (B2.5), i.e. in the frame moving with the plasma, rather than for total energy (B2) in the laboratory frame.

Our B2 vs. B2.5 comparisons are carried out by using a strongly simplified SOL test case for plasma fluid solvers alone, and then comparing the output of the codes in detail. Focus is solely on code benchmarking with respect to the modeling equations and their implementation. Ideally, for equal physical inputs, both codes should find the same steady state plasma solution up to the numerical precision determined solely by a given computational mesh size.

There are some further significant technical differences between both codes already identified in earlier benchmarking attempts [4, 5, 6, 7], both with respect to the physical model and with the numerical implementation. For example, in B2, velocities are defined on staggered locations with respect to the primary cells, while in B2.5 a collocated grid approach is used. Also the numerical schemes are somewhat different. The default recycling model in B2 is a "minimal kinetic description" of neutrals, while as default in B2.5 a fluid neutral gas model is available.

Our present study was mainly triggered by the work of [7], in which still significant (unresolved?) code discrepancies were identified when comparing SOLPS4.3 and SOLPS5.2 on rather well documented configurations (a) an ASDEX-U H-mode plasma, Chankin et al., PPCF, vol. 48 (2006) p. 839, and (b) a standard ITER (F12) case (both: single fluid (D), same transport coefficients, boundary conditions, grids, same (reduced) EIRENE neutral particle model, etc....).

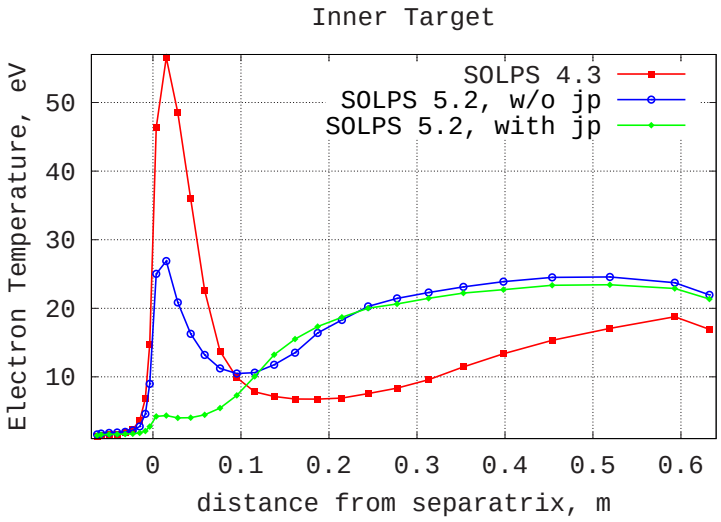


Figure 1: Comparison SOLPS4.3 vs. SOLPS5.2 [7], ITER F12 single fluid reference case. Electron temperature at inner target. x=0 corresponds to separatrix location. The two SOLPS5.2 runs are with and without parallel electrical currents taken into account. Other quantities at other locations in this code comparison agree comparatively much better, but some others still show large discrepancies like this one, for until then unidentified reasons

In addition to the issues already identified in the earlier code-code comparisons [4, 5, 6, 7] our present study, which was based upon inspection of the FORTRAN coding in B2.5 and B2 itself, has lead to identification of a few further such differences, as will be detailed below.

Already because of the purely numerical differences, the results will never be exactly equal. The goal then is to assure that the norm of the difference in the steady state plasma solution is smaller than the sum of the norms of the numerical discretization errors of the two codes.

In Section 2 we discuss the model equations as derived from inspecting the FORTRAN source codes of B2 and B2.5, as well as the adaptations needed to bring both models back together to an identical set of physics.

As we then clearly demonstrate in Section 3.3 we have been able to reduce the relative differences between solutions of B2 and B2.5 of the simple test case to a few percent, bringing them well within the range of the discretization error, which we have been able to quantify as well. Furthermore the issue of "Braginskii vs. Balescu" classical parallel

transport formulations in B2 and B2.5, resp., has been fully resolved at least for the reduced (but conventionally used) set of plasma equations, i.e. without drift flows and electric fields still.

Compared to the starting point, where solutions were even qualitatively very different, some important issues have been resolved. Thus far, extensive testing of boundary conditions is not yet performed. All our present tests employed pure von Neumann and/or Dirichlet boundary conditions. More complex boundary conditions, such as mixed boundary conditions or even decay lengths, or other boundary conditions involving fluxes, such as typical sheath boundary conditions are not considered here, because they are more strongly affected by the details of the numerical implementations. This will be done in the near future, and documented in a next status report. However, in appendix D we discuss first results obtained with conventional sheath outflow conditions to indicate already at this stage the issue of odd-even decoupling instabilities expected in the collocated grid approach in B2.5.

It should be noted that in order to achieve the present level of agreement both the B2 and B2.5 code versions had to be amended. The measures are described in the Section 2 of the present report for each plasma equation (continuity, momentum balance, electron and ion energy balance) separately.

Within the ITER service contract F4E-OPE-258 (starting date Dec. 17th, 2010), then the possibility of full backward compatibility for B2.5 towards B2, for a full ITER relevant model, will be examined in greater detail.

2 Modeling Equations

To be able to compare the B2 and B2.5 codes effectively, we start with a simplified geometry and a simplified version of the plasma model equations. This allows to verify whether the basic physics implemented in both codes is the same. In subsequent stages, the level of detail of the plasma model will be gradually increased.

Consider a single ion species with ion mass m_i and charge state Z_i. At this point there are no neutral particles included, e.g. in particular no recycling sources. Also drifts and currents are neglected. This leads to the "baseline" model equations for continuity, momentum and energy discussed in the next subsections.

Rather than a full ITER edge plasma configuration we use a strongly simplified geometry to first focus solely on the physics aspects. The model geometry is an inner double null SOL, in which the poloidal coordinates lines have been made straight lines (parallel to the symmetry axis of the torus) and the target surfaces are fully orthogonal. The geometry of the benchmark cases is described in more detail in Section 3.1 below.

The choices described there eliminate all possible issues related to metric coefficients, the purely geometric flux expansion effects, as well as effects related to grid distortion (cell shape, projection of fluxes onto cell faces, etc....). Because also drifts are neglected in these initial benchmarks, this configuration is fully up-down symmetric, hence the parallel electrical currents are strictly zero and the critical issue of different assumptions on ambipolarity between both codes is automatically eliminated at this stage.

2.1 Continuity Equation

A quasi-neutral plasma is assumed, with density $n = n_i = n_e/Z_i$. Because there is no recycling, there are no particle sources in the continuity equation:

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} n b_x V_\parallel - \frac{\sqrt{g}}{h_x^2} D^n \frac{\partial n}{\partial x}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y^2} D^n \frac{\partial n}{\partial y}\right) = 0. \tag{1}$$

Coordinates x and y denote the poloidal and radial direction, respectively, see Figure 2. $b_x = B_x/B$ is the pitch of the magnetic field, $h_x = 1/||\nabla x||$, $h_y = 1/||\nabla y||$ and $\sqrt{g} = 2\pi R h_x h_y$ are the metric coefficients, R is the local value of the major radius of the torus. For later use we introduce the third metric coefficient $h_z = 2\pi R$.

For convenience, the poloidal velocity V_x and the radial velocity V_y are introduced:

$$V_x = b_x V_\parallel - \frac{D^n}{n}\frac{1}{h_x}\frac{\partial n}{\partial x} \tag{2}$$

$$V_y = -\frac{D^n}{n}\frac{1}{h_y}\frac{\partial n}{\partial y} \tag{3}$$

Measures taken for code benchmarking

- Note the presence of an anomalous diffusion term in B2.5 both in the poloidal and radial direction (see remark below). The poloidal contribution (second term in Eq. (2)) is absent in the standard B2 code, but was now introduced for code comparison (using a flag VPFLAG). Its influence on the solution is very small despite adding a next higher derivative to the equation, however. This is because in this parallel (poloidal) direction the flow is strongly convection dominated (by the sonic or supersonic sheath outflow boundary conditions).
- The particle diffusion coefficient D^n is set to 1 $\mathrm{m^2 s^{-1}}$ in both codes (input parameter DIFNI for B2 and parm_dna in B2.5).

Remarks

- In the B2.5 code, it turns out to be technically difficult to entirely eliminate the fluid neutrals. For all tests here, the continuity equation is not solved for the neutrals, and their density (= initial value) is kept very low everywhere in order to minimize their influence on the plasma. (Neutral particle densities of order of magnitude 1 m^{-3} compared to typical plasma densities $10^{19}m^{-3}$). Analogously, the parallel momentum equation is not solved for neutrals, and their parallel velocity is set to zero. In B2 all recycling coefficients are set to strictly zero.

- In B2, the default numerical scheme used for the computation of the convective terms in the residuals in the continuity equation is the central differencing scheme. This is not to be confused with the upwind scheme used in B2 for the stencil of the coefficients of the (pressure) correction equation. In B2.5 a hybrid scheme for the convection-diffusion type equation is used, but which reduces essentially to an upwind scheme for the comparatively extremely low diffusive components used here. As to be expected the central scheme in B2 tends to trigger spatial oscillations for large local Péclet numbers (in purely convective (original B2) or at least convection dominated regime (B2 with VPFLAG option), e.g. near the divertor targets). This had initially lead to oscillations appearing in the B2 density profiles for some tests, due to the relatively coarse grids used in typical edge simulations. These oscillations are avoided now by replacing the central scheme with an upwind scheme (new B2 flag LUPWINDCONT), which is used for all tests described below, and which behaves similar to the hybrid scheme implemented in B2.5 in the limit of high Péclet numbers, as those discussed here.

- The extra anomalous diffusion term in B2.5 (and now also in B2) introduces a second order derivative in the poloidal direction, making it, in principle possible to apply one more boundary condition in this direction. E.g. the densities at both targets could now be specified in purely subsonic flow regimes. In fusion edge plasma conditions (at least sonic outflow), however, this diffusion coefficient is too small to influence the upstream plasma solution significantly. At most, a sharp boundary layer is formed, which is not resolved by the typical grids used for edge simulations.

- We are not sure about the motivation of the original code author (Bas Braams, [3] to introduce such anomalous terms even in the poloidal direction nor about the default choices for these anomalous diffivities (the same is done in parallel momentum balance as well as for the parallel (poloidal) fluxes in the electron and ion energy equations).

  One reason for adding such anomalous fluxes might have been to account for a turbulent contribution to poloidal transport. One other reason might have been to add artificial diffusive components to cope with odd-even decoupling oscillations in the collocated grid discretisation scheme B2.5, although then we would have expected higher order terms for this purpose (see next bullet). Finally these terms may have been added to make the transport equations for (fluid) neutrals and charge particles formally identical. In this case all these terms should be turned off in B2.5 when the neutral transport will be dealt with by an external model (e.g. EIRENE), as it is already in place in B2-EIRENE.

- An important difference between the B2 and B2.5 codes is the fact that the B2 code calculates parallel velocities at staggered positions with respect to density- and temperature fields, while in B2.5 a collocated grid approach is used. For the collocated

approach, additional measures have to be taken to avoid numerical instabilities due to so called "odd-even decoupling". The diffusive term in the poloidal direction may be large enough to avoid odd-even decoupling oscillations, but this cannot be assured for all cell sizes and parameter ranges. From a numerical point of view, it would perhaps be better to add a higher order numerical diffusion term, which decreases with decreasing cell size, or to use appropriate interpolation schemes for the pressure gradient and cell face velocities in the discrete equations [12]. These kind of measures could be implemented in the future if required (see discussion in App. D).

2.2 Momentum Equation

The parallel momentum equation for ions solved in the codes is

$$\begin{aligned}\frac{\partial}{\partial t}\left(m_i n V_\parallel\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} m_i n V_x V_\parallel - \frac{\sqrt{g}}{h_x^2}\frac{4}{3}\eta_x^i \frac{\partial V_\parallel}{\partial x}\right) \\ &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} m_i n V_y V_\parallel - \frac{\sqrt{g}}{h_y^2}\eta_y^i \frac{\partial V_\parallel}{\partial y}\right) = -\frac{B_x}{B}\frac{1}{h_x}\frac{\partial p}{\partial x}\,. \end{aligned} \tag{4}$$

Again, all sources of parallel momentum due to interactions with neutrals are omitted for the tests described here (ionization, charge exchange, friction,...). As stated above, in B2.5, similar to the procedure for the continuity equation, for neutrals all velocities are set to zero and the momentum balance for neutrals is not solved in our test (i.e., these initial values are not altered).

Measures taken for code benchmarking

- The poloidal velocity V_x appearing in the momentum equation of B2.5 now also contains an anomalous diffusive part (see Eq. (2)). Its contribution to the momentum equation has been added in the B2 code for benchmarking purposes (same flag VPFLAG as for continuity equation).

2.2.1 Viscosity coefficients η_x^i and η_y^i

A critical issue for the momentum equations is the expression of the viscosities η_x^i and η_y^i.

In the B2 code, the poloidal viscosity is a projection of the classical parallel Braginskii [11] viscosity coefficient:

$$\eta^i_{x,B2} = \eta^i_{x,CL,B2} = b_x^2\, 0.96 n T_i \tau_{i,Brag}\,, \tag{5}$$

with $\tau_{i,Brag}$ the Braginskii collision time for ion-ion collisions:

$$\tau_{i,Brag} = \frac{3}{4}\frac{\sqrt{m_i}}{\sqrt{\pi}}\frac{T_i^{\frac{3}{2}}}{Z_i^4 n \ln\Lambda}\left(\frac{4\pi\epsilon_0}{e^2}\right)^2\,. \tag{6}$$

Expressions are given here for the case of a single ion species. In the radial direction, the anomalous viscosity coefficient is specified through an input parameter $c_{\eta_y^i}$:

$$\eta^i_{y,B2} = c_{\eta_y^i} m_i n\,. \tag{7}$$

In the B2.5 code, the poloidal viscosity coefficient is composed of a classical and an anomalous part:

$$\eta^i_{x,B2.5} = \eta^i_{x,CL,B2.5} + \eta^i_{x,AN,B2.5}\,. \tag{8}$$

The classical contribution is based on the Balescu [10] parallel viscosity:

$$\eta^i_{x,CL,B2.5} = b_x^2\, 1.357 n T_i \tau_{i,Bal}\,, \tag{9}$$

with $\tau_{i,Bal}$ the Balescu collision time for ion-ion collisions:

$$\tau_{i,Bal} = \frac{3}{4}\frac{\sqrt{m_i}}{\sqrt{2\pi}}\frac{T_i^{\frac{3}{2}}}{Z_i^4 n \ln\Lambda}\left(\frac{4\pi\epsilon_0}{e^2}\right)^2\,. \tag{10}$$

Again, expressions are given here for the case of a single ion species. For the radial direction, an anomalous viscosity coefficient is used. By default, this anomalous viscosity coefficient is linked to the anomalous diffusion coefficient by:

$$\eta^i_{y,B2.5} = m_i n D^n\,. \tag{11}$$

The extra anomalous contribution to the poloidal viscosity in B2.5 is the same as the radial viscosity coefficient $\eta^i_{y,B2.5}$:

$$\eta^i_{x,AN,B2.5} = \eta^i_{y,B2.5}\,. \tag{12}$$

For all tests described in this present report this anomalous term in the poloidal direction is turned off in B2.5, see third bullet below, and new input flag B2tral_visc_style.

Measures taken with respect to viscosity coefficients

- The classical viscosity coefficients used in the B2 and B2.5 codes are different. There is a factor of $1/\sqrt{2}$ difference between the Braginskii and Balescu collision times, Eqs. (6) and (10), respectively. However, there is also a difference in the numerical coefficients (dimensionless viscosities) appearing in Eqs. (5) and (9), which nearly cancels out the factor of $1/\sqrt{2}$ between collision times again. Indeed, the viscosities obtained by Balescu are very similar to those obtained by Braginskii [10]. For code comparison, the (small) remaining difference between the Braginskii and Balescu viscosity coefficients has been taken into account by multiplying the viscosity coefficient in B2 by 0.9924 (see also Sec. 3.3, input parameter PARVIS).

- The value of $\ln\Lambda$ in B2.5 is set to 12, which is the default value used in the B2 code. Usually, in B2.5 a more complicated expression for $\ln\Lambda$ is used, based on the local electron density and temperature. Note that use of the simplified $\ln\Lambda = 12$ might lead to noticeable differences in solutions compared to the more complicated expression, as was demonstrated earlier in B2.5 simulations with both forms of $\ln\Lambda$.

- To be able to compare the codes, the radial viscosity of B2.5 was changed to $\eta^i_{y,B2.5} = c_{\eta^i_y} m_i n$, see Eq. (7). Also, the anomalous contribution to the parallel viscosity coefficient in B2.5 is set to zero, so we have $\eta^i_{x,B2.5} = \eta^i_{x,CL,B2.5}$ for code benchmarking.

- A bug, apparently implemented into B2 at a later development stage, was identified in some versions, including the ITER-IO versions of B2 (but notably not in some other installations of B2, e.g. at FZJ / KUL), see appendix E. Due to an erroneous interpolation, the radial viscosity was incorrectly doubled. So e.g. all ITER B2-EIRENE simulations carried out so far have implicitly used a radial viscosity twice as large as specified in the input file (and other documents).

- For code comparison, $c_{\eta^i_y}$ equals 0.2 $\mathrm{m^2 s^{-1}}$, input parameter TRAVIS.

2.2.2 Viscous flux limiters

As already stated in [4] the treatment of kinetic corrections ("flux limiters") is different between B2 and B2.5.

The following considerations (and similarly for thermal conduction in the energy equations in the next Sections) show how these treatments can be reconciled.

The viscous momentum flux in the parallel momentum equation (4) is $\frac{4}{3}\eta_x^i \frac{\partial V_\parallel}{\partial x}$. In both codes, flux limiters are applied to the *classical* viscosity coefficients $\eta^i_{x,CL,B2}$ and $\eta^i_{x,CL,B2.5}$, respectively. However, there is a subtle but important difference in the treatment of flux limiters in both codes. In B2.5, the flux limiter is implemented as a coefficient $c_{\eta^i_{x,CL,B2.5}}$ multiplying the classical poloidal viscosity coefficient $\eta^i_{x,CL,B2.5}$, note: without the factor 4/3 appearing in the flux, and this factor 4/3 is then applied afterwards again to the limited flux:

$$c_{\eta^i_{x,CL,B2.5}} = \frac{1}{1 + \frac{\frac{\sqrt{g}}{h_x}\eta^i_{x,CL,B2.5}\,\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}}{c^{B2.5}_{lim}\frac{\sqrt{g}}{h_x} b_x n T_i}}, \tag{13}$$

($c^{B2.5}_{lim}$ being an input flag, which should be based on some flux limit theory) leading to a limited viscous flux in B2.5:

$$\frac{4}{3}\tilde{\eta}^i_x \frac{\partial V_\parallel}{\partial x} \quad \text{with} \quad \tilde{\eta}^i_x = c_{\eta^i_{x,CL,B2.5}}\eta^i_x \tag{14}$$

In B2, the flux limit is applied by harmonic averaging of the classical poloidal viscous momentum flux $\frac{4}{3}\eta_x^i \frac{\partial V_\parallel}{\partial x}$ itself (i.e. including the factor 4/3) and a limiting flux $c^{B2}_{lim}\frac{\sqrt{g}}{h_x} b_x n T_i$, leading to an effective multiplicative coefficient

$$c_{\eta^i_{x,CL,B2}} = \frac{1}{1 + \frac{\frac{\sqrt{g}}{h_x}\frac{4}{3}\eta^i_{x,CL,B2}\,\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}}{c^{B2}_{lim}\frac{\sqrt{g}}{h_x} b_x n T_i}}. \tag{15}$$

In both codes $c_{lim}\frac{\sqrt{g}}{h_x} b_x n T_i$ determines an upper limit to the classical viscous momentum flux, but the meaning of the numerical input values c_{lim} is slightly different between B2 and B2.5. Using equations (13) and (15), these two options can easily be reconciled:

Measures taken with respect to flux limiters

- In practice, the previous statement means the multiplication factor c_{lim} in the denominator of the flux limiter used in B2 (FLIMV) must be made a factor 4/3 larger than in B2.5 in order to have exactly the same expressions for the constrained fluxes.

  Note that depending on the particular version of B2.5, and the value of input flag 'b2tqca_model' in B2.5, this issue may be different. Our prescription here applies to the B2.5 version in SOLPS5.2 (St. Petersburg, 2010), option b2tqca_model=3.

  Also, if model = 3, the factor 4/3 will also be applied to the anomalous viscous coefficient in the poloidal direction. All benchmark tests are performed with b2tqca_model = 3, choosing correspondingly different values for c_{lim} for B2 and B2.5, and running without the anomalous contribution to the poloidal viscosity in B2.5.

- By default, the viscous flux limiter is *not* applied to the *closed* flux surfaces (inside the separatrix) in B2.5, whereas the default in B2 was to apply the flux limiter everywhere. For our tests, however, it has now been applied everywhere in B2.5 as well, by proper choices of input flags.

- In B2.5, some other corrections to the classical viscosity coefficients are available, e.g. so called "Luciani coefficients". For code comparison, all these other correction factors are switched off.

- For the tests described in Section 3.3 the coefficient c_{lim} is taken to be 2 in B2.5, and $4/3 \cdot 2$ in B2[1].

The expressions for the viscosity coefficients, and especially the flux limiter applied to them, have a very large influence on the simulation results, notably on the ion temperature (see below). They influence the parallel velocity and ion temperature profiles both quantitatively and qualitatively!

2.2.3 Remarks

- Due to the definition of the parallel velocities on staggered grid locations in B2, some difficulties arise around the X-point (as e.g. already noted in [6]). We believe, (and our tests seem to confirm this) however, that this is not solely an issue of direct or indirect cell addressing, but instead also of the staggered vs. collocated grid approaches in B2 and B2.5, respectively.

  Indeed, the 4 staggered cells located around the X-point each have *five* neighboring staggered cells instead of *four*. The usual forward and backward mapping procedure of the cells around the cuts to an auxiliary domain would lead to incorrect fluxes and coefficients. Therefore, up to now the momentum fluxes between the four cells wrapping the X-point are set to zero in the ITER IO version of B2 (also this is used here). As a result, the X-point seen by the B2 code has a finite size, leading to inaccuracies in the results also for certain test cases of our series. This is especially pronounced when the poloidal fluxes entering the balance equations of a cell are of the same order of magnitude as the radial fluxes. The inaccuracy can be reduced by locally reducing the aspect ratio l_x/l_y of the cells around the X-point, so that poloidal fluxes dominate the balance equations of the cell. l_x and l_y refer to the cell sizes in the poloidal and radial directions, respectively.

- In 2001, a new (according to [13] more complete) expression was suggested for the divergence of a tensor in B2.5 appearing in the parallel momentum equation, involving the third metric coefficient h_z (see discussion after continuity equation (1)):

$$\frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{h_z\sqrt{g}}{h_x}...\right)+\frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{h_z\sqrt{g}}{h_y}...\right). \tag{16}$$

  h_z is the norm of the tangent basis vector in the toroidal direction. In order to be able to compare the codes, however, the standard B2.5 expression for the divergence of a tensor was used in B2.5 (using flag b2mndr_hz). This form is also the one implemented in B2:

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}...\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}...\right). \tag{17}$$

[1] The more typical values of $c_{lim} = 0.5$ and $4/3 \cdot 0.5$ were used for the first test cases, but these values led to an inconsistency when studying the (isolated) 1D momentum equation for fixed density and temperatures. Therefore, they had to be raised for these particular tests, but their effect on the solution was confirmed to be still relevant.

2.3 Electron Energy Equation

Turning to the energy equations, matters become a bit more complicated. The electron energy equations solved by B2 and B2.5 are written in slightly different ways, requiring a careful study in order to be sure that the same physics is included. In this section, the electron energy equation solved in B2 is stated first, and is then converted to the equivalent electron energy equation used in B2.5. We will see that, with proper specifications of input flags, these two equations can indeed be made physically equivalent.

Neglecting electric currents and drifts, the electron energy equation in B2 is

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}Z_i n T_e\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{5}{2}Z_i n V_x T_e - \frac{\sqrt{g}}{h_x^2}\kappa_x^e\frac{\partial T_e}{\partial x}\right)\\ &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\frac{5}{2}Z_i n V_y T_e - \frac{\sqrt{g}}{h_y^2}\kappa_y^e\frac{\partial T_e}{\partial y}\right)\\ &= -k(T_e - T_i) + \frac{b_x V_\parallel}{h_x}\frac{\partial p_e}{\partial x} + \alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y} + \beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y}\,. \end{aligned} \tag{18}$$

In this equation, $p_e = Z_i n T_e$ and $p_i = n T_i$. The poloidal velocity V_x is the same as was defined in Eq. (2). The so-called 'vdp-terms' (work done by the electric field) are introduced with two factors α and β. These same terms appear with opposite signs in the ion energy equation.

There is an ongoing theoretical discussion on which form these energy transfer terms should have, because they mix anomalous cross field velocities into a classically derived term. As the anomalous transport stems from turbulent fluctuations and as such brings in a time averaged value for the transport, also for these 'vdp-terms' the correct closure expression still needs to be agreed upon. Below, we will see how these terms appear in B2.5.

Finally: As no interactions with neutrals are taken into account for the test case, there are no electron energy sources or sinks due to neutrals. Also other energy sources or sinks (e.g. due to Bremsstrahlung) are not taken into account.

In the B2.5 electron energy equation, the convective energy flux is an internal energy flux only (heat flux), whereas in B2 it is the total energy flux. For this the convective energy flux in the poloidal direction is split up in B2.5 into two contributions using the relation

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}Z_i n b_x V_\parallel T_e\right) - \frac{b_x V_\parallel}{h_x}\frac{\partial p_e}{\partial x} = \frac{p_e}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x V_\parallel\right)\,. \tag{19}$$

Combining this with expressions (2) and (3) for V_x and V_y, Eq. (18) becomes

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}Z_i n T_e\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{3}{2}Z_i n b_x V_\parallel T_e - \frac{\sqrt{g}}{h_x^2}\frac{5}{2}Z_i D^n\frac{\partial n}{\partial x}T_e - \frac{\sqrt{g}}{h_x^2}\kappa_x^e\frac{\partial T_e}{\partial x}\right)\\ &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(-\frac{\sqrt{g}}{h_y^2}\frac{5}{2}Z_i D^n\frac{\partial n}{\partial y}T_e - \frac{\sqrt{g}}{h_y^2}\kappa_y^e\frac{\partial T_e}{\partial y}\right) + \frac{p_e}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x V_\parallel\right)\\ &= -k(T_e - T_i) + \alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y} + \beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y}\,, \end{aligned} \tag{20}$$

This form of the electron energy equation, without the last two terms, i.e. with $\alpha = 0$ and $\beta = 0$, is implemented in B2.5, and hence, with these choices of α and β in B2, the equations are fully equivalent.

Measures taken for code benchmarking

- The vdp-source term in the energy equations is not implemented in B2.5, and hence this corresponds to the choice $\alpha = 0$ and $\beta = 0$ for B2. In the B2 code, there is an input flag NLRADE to switch these terms on or off:
  - NLRADE = -1: $\alpha = 0$ and $\beta = -1$ (default)
  - NLRADE = 0: $\alpha = 0$ and $\beta = 0$
  - NLRADE = 1: $\alpha = 1$ and $\beta = 0$

  For the benchmark, simulations are thus performed with NLRADE = 0. Note that the presence or absence of these terms in the energy equations has a very large influence on solutions, in particular the temperature profiles, of the modeling equations.
- In B2, electron energy sinks due to Bremsstrahlung are implemented by default. A flag has been added to switch this sink off for the benchmarking (BRMFLAG). Also B2.5 contains such an energy sink, (the exact expression still has to be identified from the coding) but it has been switched off for the tests described below.
- The poloidal velocity V_x in the poloidal convective term includes the anomalous diffusive term in B2.5, see Eq. (2). This term was absent in the standard B2 code, but has been included for the benchmark.

2.3.1 Electron heat conductivity coefficients κ_x^e and κ_y^e

A careful comparison of the electron heat conductivity coefficients is also required. As for the viscosities, B2 uses the Braginskii classical model, while in B2.5, the Balescu coefficients are implemented. Distinct from the B2 code, B2.5 also has an anomalous heat conduction term in the poloidal direction.

The B2 classical electron heat conductivity coefficient for a single ion species is

$$\kappa^e_{x,B2} = \kappa^e_{x,CL,B2} = b_x^2\, 3.2 \frac{n_e T_e \tau_e}{m_e}\,, \tag{21}$$

with

$$\tau_e = \frac{3}{4}\frac{\sqrt{m_e}}{\sqrt{2\pi}}\frac{T_e^{\frac{3}{2}}}{Z_i^2 n \ln\Lambda}\left(\frac{4\pi\epsilon_0}{e^2}\right)^2\,. \tag{22}$$

The corresponding coefficient in B2.5 is

$$\kappa^e_{x,CL,B2.5} = b_x^2\,\frac{5}{2}\frac{2.16 Z_i}{1+0.27 Z_i}\frac{n_e T_e \tau_e}{m_e}\,. \tag{23}$$

In this expression, the same collision time for electrons-ions τ_e is used (Eq. (22)). The difference compared to the Braginskii electron heat conductivity is in the factor multiplying $\frac{n_e T_e \tau_e}{m_e}$, i.e. ≈ 3.2 in B2 or the Z_i-dependent factor in B2.5 (being e.g. ≈ 4.2 for $Z_i = 1$).

In the radial direction, anomalous heat conductivities of the form

$$\kappa^e_y = c_{\kappa^e_y} Z_i n \tag{24}$$

are used in both codes. The electron heat conductivity in the poloidal direction in B2.5 also includes an anomalous contribution:

$$\kappa^e_{x,B2.5} = \kappa^e_{x,CL,B2.5} + \kappa^e_{x,AN,B2.5}\,, \tag{25}$$

with

$$\kappa^e_{x,AN,B2.5} = \kappa^e_y. \tag{26}$$

This anomalous contribution to the poloidal heat conductivity is thus the same as the radial electron heat conductivity. It is absent in the standard B2 code.

Measures taken with respect to electron heat conductivity coefficients

- For the case considered here, $Z_i = 1$, the same electron heat conductivity is obtained in both codes by multiplying $\kappa^e_{x,CL,B2}$ by 1.3228 in B2, (B2 input parameter kxe), see also Sec. 3.3.
- The anomalous contribution to the poloidal heat conductivity was absent in the standard B2 code, but has been added as an option for code comparison (new B2 input flag: LHCXANOM). The code comparison runs reported below are with this term turned on. Its influence is very small, however. It is quite negligible compared to the classical part of the coefficient.
- $\ln \Lambda = 12$ is enforced in both codes (see above).
- For the test case, *no* heat flux limiters are applied in either code.
- $c_{\kappa^e_y}$ is taken 1 $\mathrm{m^2 s^{-1}}$.

2.3.2 Electron-ion energy equilibration term

In the electron energy equation, Eq. (18), also the equilibration coefficient k has to be compared between the two codes. In the B2, the expression for k is

$$k = c_{eqp} \ln \Lambda e^{\frac{3}{2}} \left(\sum_\alpha \frac{Z^2_{i,\alpha} m_p}{m_{i,\alpha}} n_{i,\alpha} \right) \frac{n_e}{T_e^{\frac{3}{2}}}, \tag{27}$$

where the summation is taken over all ion species α, and m_p is the mass of a proton. $c_{eqp} = 4.8 \cdot 10^{-15}$. For one single ion species, and $n_e = Z_i n$, this reduces to

$$k = c_{eqp} \ln \Lambda e^{\frac{3}{2}} \frac{Z_i^2 m_p}{m_i} \frac{Z_i n^2}{T_e^{\frac{3}{2}}}. \tag{28}$$

In B2.5, the same expression is used for the equilibration term.

2.3.3 Remarks

- As can be seen from Eqs. (18) and (20), that despite the physical equivalence of these equations, there is a difference in the interpretation of electron energy fluxes computed by the two codes. In B2, the convective energy flux in the poloidal direction has a factor 5/2, while in B2.5, the part of this flux corresponding to $V_{||}$ has a factor 3/2. The rate of change in internal energy due to compressibility (which would provide another convective flux with factor 1) is taken as energy source term here. This difference has to be taken into account when specifying boundary conditions based on fluxes. Care should also be taken when comparing for example target loads obtained with the B2 and B2.5 codes. Note that this difference is not present in the radial convective electron energy fluxes.

- B2.5 includes a heat loss term proportional to the ion and electron density, apparently to account for electron energy losses/gains due to free-bound, bound-bound and or free-free inelastic collisions. Some care will be needed when coupling B2.5 to EIRENE, and to ADAS atomic data in order not to double count or omit contributions. For our test problems this term in B2.5 is switched off (input flag b2sqel_phm2=0).

2.4 Ion Energy Equation

The two codes treat the ion energy equation differently. In B2, the total energy equation for ions is solved. The total energy is the sum of the internal and kinetic energy of the ions:

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{3}{2}nT_i+\frac{1}{2}m_inV_\parallel^2\right)+\\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\left(\frac{5}{2}nV_xT_i+\frac{1}{2}m_inV_xV_\parallel^2\right)-\frac{\sqrt{g}}{h_x^2}\left(\kappa_x^i\frac{\partial T_i}{\partial x}+\frac{1}{2}\frac{4}{3}\eta_x^i\frac{\partial V_\parallel^2}{\partial x}\right)\right)+\\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\left(\frac{5}{2}nV_yT_i+\frac{1}{2}m_inV_yV_\parallel^2\right)-\frac{\sqrt{g}}{h_y^2}\left(\kappa_y^i\frac{\partial T_i}{\partial y}+\frac{1}{2}\eta_y^i\frac{\partial V_\parallel^2}{\partial y}\right)\right)\\
&=k(T_e-T_i)-\frac{b_xV_\parallel}{h_x}\frac{\partial p_e}{\partial x}-\alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y}-\beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y}\,.
\end{aligned}\tag{29}$$

Parameters α and β take the same values as in the electron energy equation. Note the presence of the factor 4/3 in the poloidal viscous contribution, needed for consistency with the momentum equation. In the B2 code version available to us this factor was (probably unintentionally) 2, rather than 4/3. See below and appendix E: Bug Fixes.

There are no energy source terms originating from interactions with neutrals in our tests, due to the choice of recycling = zero at all relevant surfaces.

In B2.5, an internal energy equation for the ions is solved:

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\frac{3}{2}nT_i\right) \;&+\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{3}{2}nb_xV_\parallel T_i-\frac{\sqrt{g}}{h_x^2}\frac{5}{2}D^n\frac{\partial n}{\partial x}T_i-\frac{\sqrt{g}}{h_x^2}\kappa_x^i\frac{\partial T_i}{\partial x}\right)\\
&+\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(-\frac{\sqrt{g}}{h_y^2}\frac{5}{2}D^n\frac{\partial n}{\partial y}T_i-\frac{\sqrt{g}}{h_y^2}\kappa_y^i\frac{\partial T_i}{\partial y}\right)+\frac{p_i}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_xV_\parallel\right)\\
&=\; k(T_e-T_i)-\alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y}-\beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y}\\
&\quad+\frac{4}{3}\eta_x^i\left(\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}\right)^2+\eta_y^i\left(\frac{1}{h_y}\frac{\partial V_\parallel}{\partial y}\right)^2\,.
\end{aligned}\tag{30}$$

In App. A, it is shown that this equation is equivalent to the total energy equation solved in B2, if both the momentum and continuity equations are satisfied simultaneously as well. In B2.5, the vdp-terms do not appear, so $\alpha = 0$ and $\beta = 0$. Any other energy terms, such as heat loss terms, are switched off for the benchmark.

Measures taken for code benchmarking

- The factor 4/3 multiplying the viscous coefficient in the energy equation was not correctly implemented in the original B2 ion energy equation. This has been corrected in the code and the tests are carried out with this factor (see appendix E: Bug Fixes).

- For the benchmark, simulations are performed with NLRADE = 0 (see above, vdp-terms).
- B2.5 includes a heat loss term proportional to the ion and electron density, presumably to account for some inelastic processes. For our test problems this term in B2.5 is switched off.

2.4.1 Ion heat conductivity coefficients κ_x^i and κ_y^i

We next consider the ion heat conductivity coefficients. In the B2 code, Braginskii's approach is followed. For one single ion species:

$$\kappa^i_{x,B2} = \kappa^i_{x,CL,B2} = b_x^2\, 3.9 \frac{nT_i\tau_{i,Brag}}{m_i}\,, \tag{31}$$

with $\tau_{i,Brag}$ from Eq. (6). An anomalous heat conductivity coefficient is used in the radial direction:

$$\kappa^i_{y,B2} = c_{\kappa^i_y} n\,. \tag{32}$$

In B2.5, the Balescu classical transport model is implemented. The classical ion heat conduction coefficient following Balescu is

$$\kappa^i_{x,CL,B2.5} = b_x^2\, 2.253 \cdot \frac{5}{2} \frac{nT_i\tau_{i,Bal}}{m_i}\,. \tag{33}$$

Taking into account the factor $1/\sqrt{2}$ between the Braginskii and Balescu ion-ion collision times $\tau_{i,Brag}, \tau_{i,Bal}$, respectively, it is seen that these heat conduction coefficients are in fact almost the same [10], Ch.5, Section 7B. The total ion heat conduction coefficient in the poloidal direction in B2.5 is composed of this classical term and an additional anomalous contribution:

$$\kappa^i_{x,B2.5} = \kappa^i_{x,CL,B2.5} + \kappa^i_{x,AN,B2.5}\,, \tag{34}$$

where by default the anomalous contribution is set equal to the radial heat conductivity,

$$\kappa^i_{y,B2.5} = \kappa^i_{x,AN,B2.5} = 1.2 D^n n\,. \tag{35}$$

Measures taken with respect to ion heat conductivity coefficients

- For the code comparison, a multiplication factor of 1.0140 is used for the classical ion heat conduction coefficient in B2 (B2 input parameter kxi), to compensate for remaining tiny differences in the numerical constants used in the codes (see also Sec. 3.3).
- Again $\ln\Lambda = 12$ is enforced in both codes (see above).
- The expression for the anomalous heat conductivity in B2.5 was modified to the non-default option (B2.5 parameter parm_hci)

  $$\kappa^i_{y,B2.5} = \kappa^i_{x,AN,B2.5} = c_{\kappa^i_y} n\,, \tag{36}$$

  in order to have the same form as in B2.

- Although the anomalous conduction coefficient in the poloidal direction is absent by default in B2, (new B2 flag lhcxanom) has been added for code comparison purposes. Note that in B2.5, there is also a contribution from the neutrals to the anomalous conduction coefficient. However, this contribution should be small for very low neutral densities described by the B2.5 neutral fluid model as was assured to be the case in all our tests described here. Note that with the neutral fluid model in B2.5 perfect equilibration is assumed between neutrals and ions, with possibly significant effects on the "true ion temperature" then.

- For the classical ion heat conduction coefficients, in both codes there is also the possibility to employ a (heat-) flux limiter. This is *not* done for the test cases. Other corrections available in B2.5, such as 'Luciniani factors', are *not* applied either.

- For the test cases, $c_{\kappa_y^i} = 1\mathrm{m}^2\mathrm{s}^{-1}$.

2.4.2 The viscous energy terms

The viscous coefficients appearing in Eqs. (29) and (30) are the same as in the parallel momentum equation. As a flux limiter has been applied there to the coefficients, see Sec. 2.2.2, the same flux limiter (and possibly Luciani restrictions,...) should also be applied in the energy balance for consistency. This was (probably unintentionally) missing in the original B2 code, and a corresponding correction to automatically enforce the same viscosity parameters in both the momentum and ion energy equation has now been carried out, see appendix E, Bug Fixes.

Measures taken with respect to the viscous energy terms

- In B2, the viscous flux limiter was originally not used in the ion energy equation, but has now been added in the code (flag FLIMVI).

- The viscous source terms on the right hand side of the B2.5 internal energy equation, Eq. (30), (two last terms in this equation) which represent a conversion from kinetic energy to internal energy, must be switched on for internal consistency.

2.4.3 Remarks

- As was the case for the electron energy equations, also for the ion energy equations the energy fluxes are defined differently in B2 and B2.5, see Eqs. (29) and (30). In B2, the convective energy flux in the poloidal direction has a factor 5/2, while B2.5 again has a factor 3/2 for the part of this flux corresponding to $V_{||}$. I.e., again the rate of change of energy due to compressibility is not regarded as being part of the energy flux in B2.5, but treated as a source term instead. Furthermore, the B2 energy flux also includes the kinetic energy of the ions. These differences will have significant consequences for formulating consistent boundary conditions for B2.5 and EIRENE as well as for exchanging the proper neutral-ion energy sources.

- An important further difference between the two codes has to be noted. Whereas the continuity and momentum equations are solved for each ion species separately, the ion temperature is defined as a common temperature for all species. This includes the neutral fluid in B2.5, but not in B2. As a result, neutrals are also taken into account in the 'ion' energy equation in B2.5. The equation solved in B2.5 is in fact a sum of the ion and neutral energy equations. This implies that also the

effective heat conduction coefficients may be modified by the presence of neutrals, quite distinct from B2, where the ion temperature is a common temperature only of all ions, excluding the neutrals. Care should also be taken when comparing boundary conditions, e.g. sheath conditions, as the boundary conditions in B2.5 must account for the presence of the neutrals. For very low neutral densities stored in the B2.5 arrays, as chosen in our test case, these changes should have minor effects in the weighted temperature results. Care will be needed *not* to transfer EIRENE neutral particle densities onto B2.5 arrays.

- From a theoretical point of view, further investigation is needed concerning the factor $4/3$ in the (total) energy equation for the ions (see detailed derivation Braginskii equations in toroidal-poloidal-radial coordinates in [8]).

3 Benchmark Test of Modeling Equations

In this Section, we show the results of simulations using the modeling equations described in this report. These modeling equations represent a 'common denominator' for the physics implemented in the B2 and B2.5 codes, as worked out in the previous sections.

On a very basic level, the implementation of all individual equations was checked by defining strongly simplified 1D and 2D test cases, where only a subset of one or two equations was solved (excluding the ion energy equation, which requires all three equations to be solved simultaneously, see above), taking prescribed profiles for the other plasma parameters. For 1D cases, analytical solutions can be found, allowing to compare the performance and accuracy of the codes on a very sound basis. During these tests, the differences between the codes described in the previous sections were identified one by one, and more insight was gained in which input parameters have to be specified in order to achieve code simulations that solve the same physical modeling equations.

Below, we present the results for a 2D test case with more representative tokamak plasma parameters. The boundary conditions and configuration are still kept very simple, however, in order to focus solely on the implementation of the modeling equations themselves.

3.1 Geometry

For the comparison tests, a very simple, 'slab' SOL geometry is used, see Fig. 2. The configuration is completely symmetric between two 'divertor targets' (AB) and (CD). The poloidal projections of the magnetic field lines are perfectly vertical. Toroidicity is maintained, however: cell volumes are computed by taking into account the local values of the major radius. This geometry can be interpreted as a simplified representation of the inboard half of a single separatrix double null divertor.

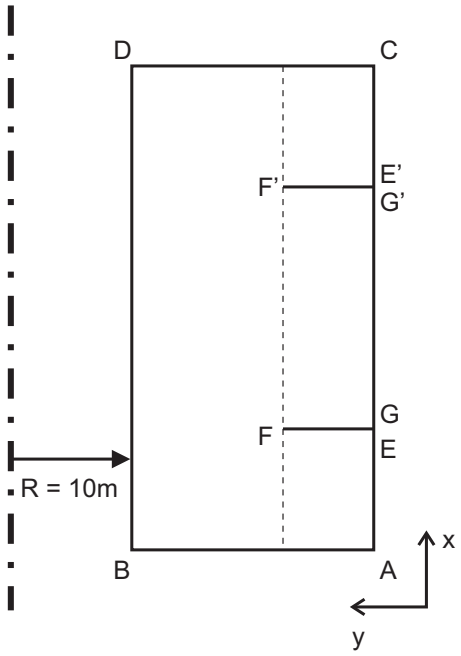


Figure 2: Test case geometry

The computational domain has a total length of 20 m in the poloidal x direction, and

0.1 m in the radial y direction. On the coarsest grid used, there are 100 interior cells (i.e. excluding the guard cells, see below) in the poloidal direction, each with equal length of 0.2 m, and 20 interior cells in the radial direction, each with width 0.005 m. The private flux region has a radial width of 0.04 m (8 interior cells), and a total poloidal length of 3.2 m (16 interior cells – 8 cells below the lower cut, and 8 cells above the upper cut). To carry out a grid sensitivity study, two finer grids are also constructed by dividing the cells of the coarsest grid in 4 resp. 16 equal parts. Thus we obtain grids with 200 by 40 cells (middle grid), and 400 by 80 cells (fine grid).

The left boundary of the domain is located at a major radius of 10 m. The magnetic field has a constant pitch of 0.1. The product $R \cdot B_z = 10$ m·T is constant (B_z is the toroidal component of the magnetic field).

For this geometry, the metric coefficients have simple, analytical expressions (following B2 and B2.5 notation: x: poloidal coordinate, y: radial coordinate, see Fig. 2)

$$\begin{aligned} \sqrt{g} &= 2\pi(R_0 - y) \\ h_x &= 1 \\ h_y &= 1 \\ h_z &= 2\pi(R_0 - y) \end{aligned} \tag{37}$$

where $R_0 = 10,05$ m is the major radius at $y = 0$ m, halfway between (AC) and (BD).

The choice of this simple up-down symmetric geometry provides a key code-diagnostic feature, because the plasma fluid code (and later the entire SOLPS suite) has to fully preserve this model symmetry. For example, if symmetric boundary conditions are applied in the poloidal direction for density and temperatures, and antisymmetric conditions for the velocities, and if magnetic drifts are turned off, then density and temperature profiles have to be symmetric, and poloidal velocities antisymmetric. This choice of symmetry also eliminates parallel electric currents entirely by the physical boundary conditions. When parallel electric currents are included in the simulations, the resulting current should turn out to be identically zero.

Remarks

- Some minor changes have been made to the grid generation program for B2, in order to have the same size of the 'guard cells'[2] for both codes. By default, guard cells in B2 have relative dimensions of the order of $\sqrt{\varepsilon}$, compared to "interior cells", with ε the machine precision. For the benchmark, we use guard cell sizes of $1 \cdot 10^{-6}$ m both in B2 and in B2.5.

- As already noted in [6], by comparing metric coefficients of the codes, it was found that there is a slight difference in the way cell volumes are calculated from the grid coordinates, even in strictly orthogonal grids. The B2 grid generation program uses the coordinates of the cell centers specified by the input file (in addition to the coordinates of the nodes), while in B2.5 these coordinates are computed by averaging of the cell node coordinates.

3.2 Boundary Conditions

Boundary conditions in B2 and B2.5 are a very subtle issue, resulting from the different formulations of equations, even without the additional electric field, drifts and electric current

[2]Guard cells are additional cells outside the computational domain used for imposing boundary conditions.

options in B2.5. We therefore perform the benchmark exercise with 'safe' boundary conditions, which strongly constrain the plasma at the edges, allowing to focus the benchmark on the modeling equations first. By this we mean: we only test Dirichlet and v. Neumann type boundary conditions, leaving more complex (e.g. sheath boundary) conditions for a later stage.

In future work, we will extend the benchmark by considering more realistic, yet numerically more challenging, boundary conditions.

For the test case discussed below, the following boundary conditions were imposed[3].

Core plasma (GG') At the core plasma , fixed values are prescribed for density and temperatures, while the radial gradient of the parallel velocity is set to zero:

- Density: $n = 3 \cdot 10^{19}$ ions/m^3
- Parallel velocity: $\frac{\partial V_\parallel}{\partial y} = 0$
- Electron temperature: $T_e = 400$ eV
- Ion temperature: $T_i = 400$ eV

Target conditions (AB and CD) Instead of imposing the usual sheath conditions at the targets, we take more basic boundary conditions here, which somewhat resemble the sheath conditions. At the east and west targets, fixed values are prescribed for the parallel velocity and ion and electron temperatures. Density gradients are taken to be zero. For the east target (CD), we have

- Density: $\frac{\partial n}{\partial x} = 0$
- Parallel velocity: $V_\parallel = 6.1899383 \cdot 10^4$ m/s
- Electron temperature: $T_e = 40$ eV
- Ion temperature: $T_i = 40$ eV

At the west target, the same conditions are applied, but with a negative sign for the parallel velocity. The parallel velocity imposed at the targets corresponds to a ion Mach number of 1 (one).

Private flux (AE and E'C) and Wall (BD) At all interfaces with 'walls', we apply the following conditions:

- Density: $n = 3 \cdot 10^{18}$ ions/m^3
- Parallel velocity: $\frac{\partial V_\parallel}{\partial y} = 0$
- Electron temperature: $T_e = 40$ eV
- Ion temperature: $T_i = 40$ eV

[3]Letters in between parentheses refer to line segments in Fig. 2.

3.3 Results and Discussion

Here, we present and compare the solutions to the test case problem obtained with the B2 and B2.5 codes. A complete list with input parameters used to obtain these solutions with the B2 and B2.5 codes is given in App. C.

As both codes use different numerical schemes to solve the modeling equations on grids with a finite number of cells, the output of B2 and B2.5 will never be exactly the same. A comparison of the results should therefore also account for the discretization errors of the codes. The discretization error, which is the difference between the exact solution of the continuous modeling equations and their discrete approximation on a finite grid, can be estimated by solving the (discrete) modeling equations on three successively finer grids, refined with a factor 2 in all directions. By analyzing the difference between the numerical solutions on these three grids, and assuming monotonous convergence towards the exact solution, the order of the discretization scheme can be estimated as

$$p \approx \frac{\log\left(\frac{\phi_{2\,\Delta x}-\phi_{4\,\Delta x}}{\phi_{\Delta x}-\phi_{2\,\Delta x}}\right)}{\log 2}\,, \tag{38}$$

with ϕ_h the solution for a plasma parameter ϕ on a grid with characteristic size h. This order can be used to estimate the relative discretization error as

$$\epsilon^{\mathrm{rel}}_{\Delta x} \approx \frac{1-\frac{\phi_{2\,\Delta x}}{\phi_{\Delta x}}}{2^p-1}\,. \tag{39}$$

For a derivation of these expressions, see App. B. The discretization error obtained via these expressions gives an estimate of the error on the computed solutions *on the finest grids*. We can then argue that the two codes have found the "same" solution for the test case problem if the discrete solutions is at most of the order of the estimated discretization error.

Before proceeding to the solutions found with the B2 and B2.5 codes, some numerical checks are performed. By comparing the transport coefficients calculated by both codes for a run with fixed density, velocity and temperature profiles, the precise multiplication factors can be identified which 'convert' the Braginskii classical tranport model of B2 into the Balescu model implemented in B2.5. By calculating these multiplication factors from the output of the codes, also differences in the number of digits used for numerical constants (numerical rounding effects) are taken into account. In this way, the multiplication factor 0.9924 for the parallel viscosity (see Sec. 2.2.1), as well as the factors 1.0140 and 1.3228 for the ion and electron heat conductivity, respectively, are determined (see Secs. 2.4.1 and 2.3.1). I.e. for the "standard" application of B2 and B2.5 codes (without electric currents, drifts, etc.) the only significant difference between Braginskii and Balescu classical transport coefficients is in the electron heat conductivity (i.e. the factor 1.3228, or approximately 30% difference).

B2 and B2.5 solutions of n, $V_{||}$, T_i and T_e for the benchmarking test case are shown in Figs. 3 and 4. These are the solutions on the finest grid (400x80 cells). The difference between the solutions can be studied most accurately by looking at the relative error between the profiles. These are given in Fig. 5. Absolute differences are also given in Fig. 6. The large relative difference in parallel velocity is due to a division by (nearly) zero at the stagnation region, and is therefore meaningless. In this domain, the absolute difference gives a better indication. As expected, the relative differences between the two solutions are largest around the X-point (see also earlier remark in Section 2.2.3), and also in regions where the gradients in the plasma profiles are large, i.e. close to the targets.

Remark All these quantities are compared in cell centers. Because the B2 code uses a staggered approach for the parallel velocity, the interpolated value UPC (as used in the B2 code for the ion energy equation) is compared to the parallel velocity of B2.5.

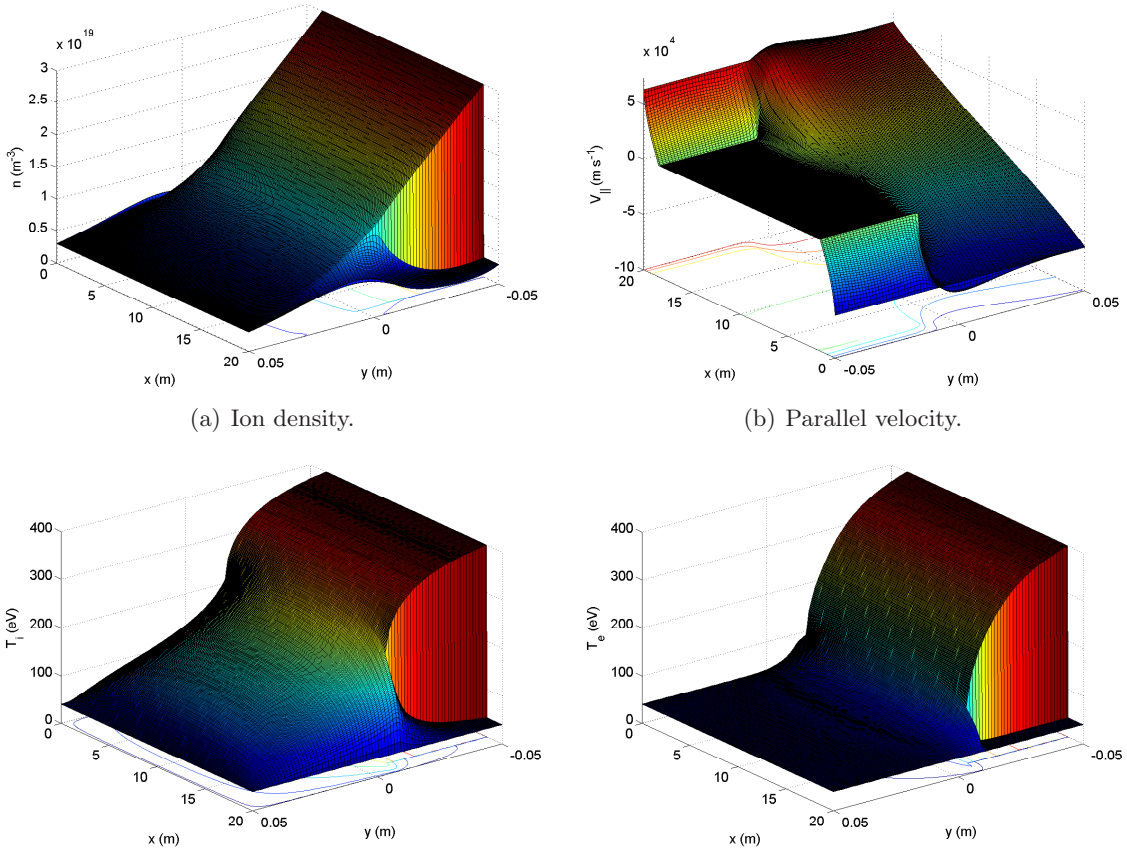
Figs. 7 and 8 show estimates of the discretization errors of these codes. These are the discretization errors *on the finest grids*. Both codes have discretization errors of the same order of magnitude. It is clear that these discretization errors are considerably larger than the difference in solutions between the codes, often up to a factor of 10.

This proofs the key result of the present report, namely that for the reduced model described above (our "common denominator") the two codes now perfectly agree with each other, well within the discretisation error.

Some profiles of the plasma parameters are shown in Figs. 9 through 12. The profiles are taken in two locations: 1) poloidal profiles from west target to east target at a radial location of 0.0025 m outside of the separatrix (i.e. at the cell center of the first cell outside of the separatrix on the 100x20 grid) and 2) radial profiles from core plasma to outer wall at approximately halfway between the divertor targets. The profiles as computed on the three different grids are shown, and also the solutions on the finest grid corrected for the discretization error (referred to as 'extrapolated' solutions)[4]. These figures show that the solutions are converging quite monotonically towards the extrapolated solutions when the grid is refined. Figs. 13 and 14 show profiles of the plasma parameters on the finest grid as well as error bars corresponding to the discretization error. These figures again clearly confirm that the solutions found with both codes agree very well, and that the difference between the solutions is well below the discretization error of the codes.

From this comparison, we can conclude that a common physical model implemented in both codes has been correctly identified. For this common physical basis, the both codes B2 and B2.5 find solutions which differ only due to discretization effects. We would like to stress that the small relative differences shown here are only obtained on the finest grids, which are considerably less coarse than the typical grids used for present tokamak edge plasma modeling. For completeness, also the errors on the coarsest grid, with 100x20 cells, are shown in Figs. 15 and 16. This error can be found by adding the discretization error on the finest grid to the difference between the solutions on the fine and coarse grids.

[4]For the finer grids (200x40 and 400x80 cells), solutions are interpolated to the cell centers of the coarsest grid (100x20 cells).



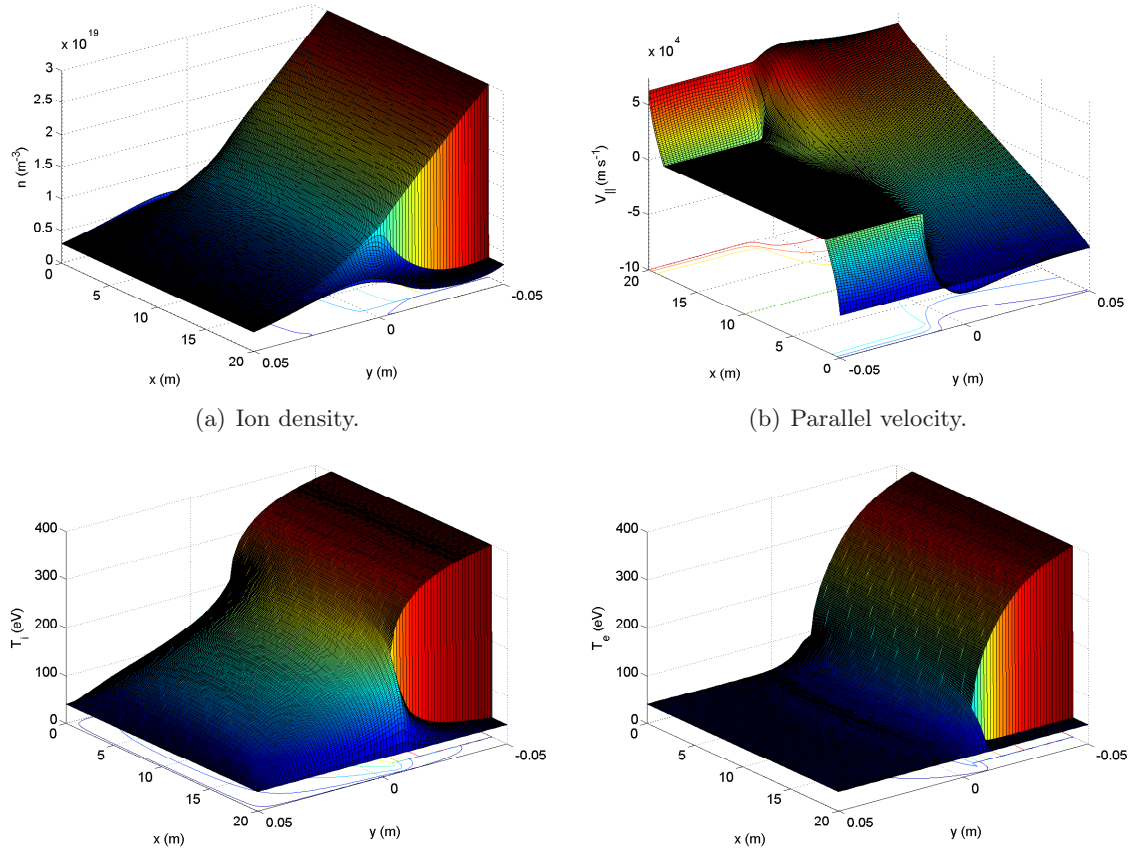
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 3: Test case solution obtained with B2.



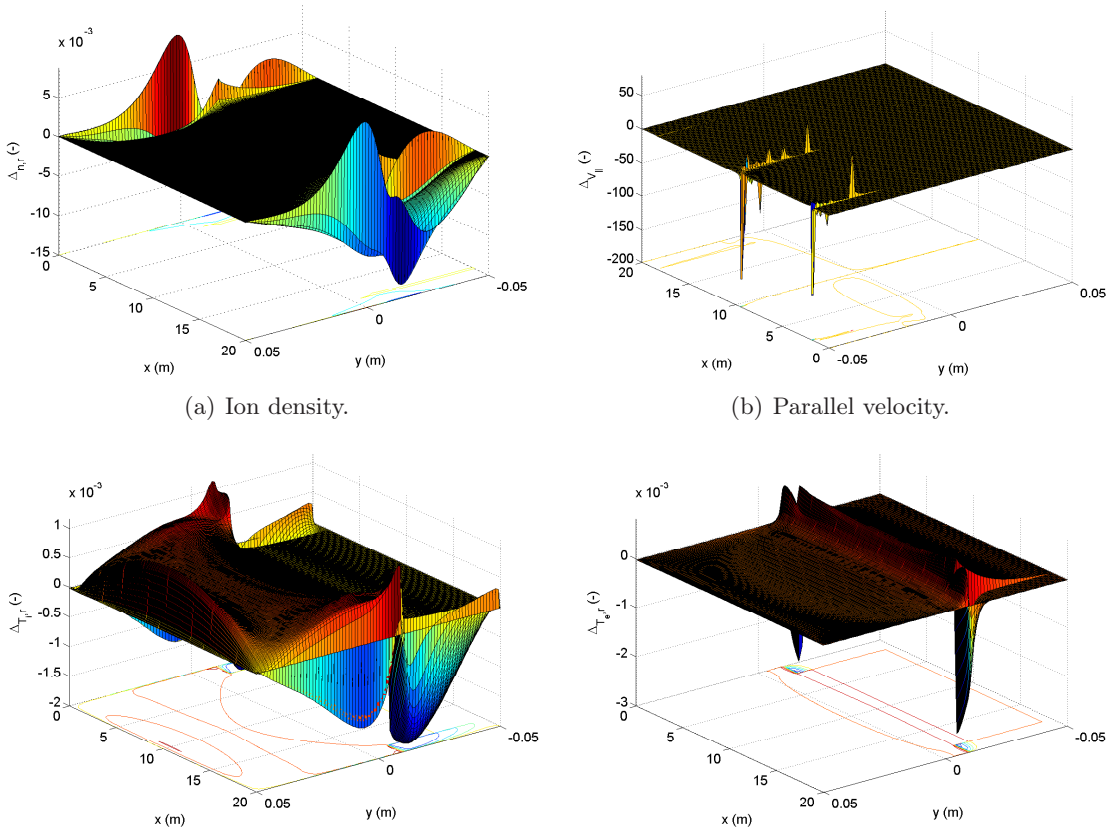
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 4: Test case solution obtained with B2.5.



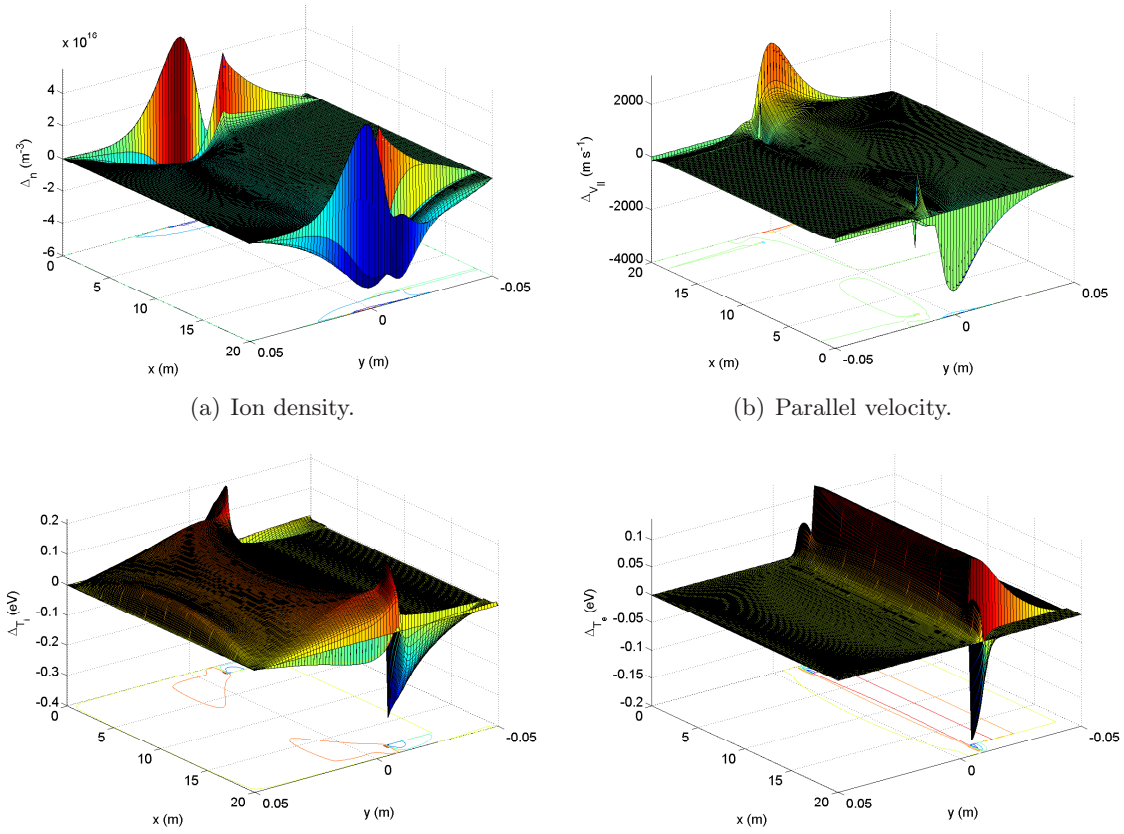
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 5: Relative difference between B2 and B2.5 solutions.



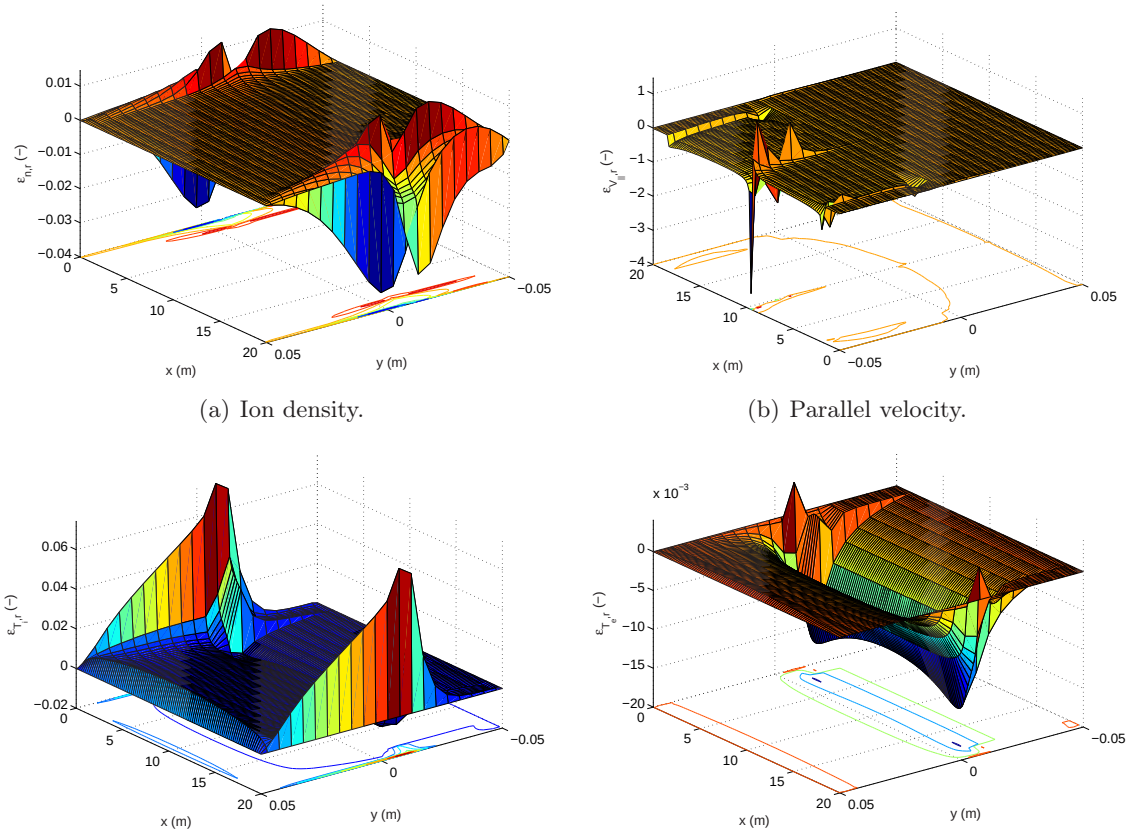
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 6: Absolute difference between B2 and B2.5 solutions.



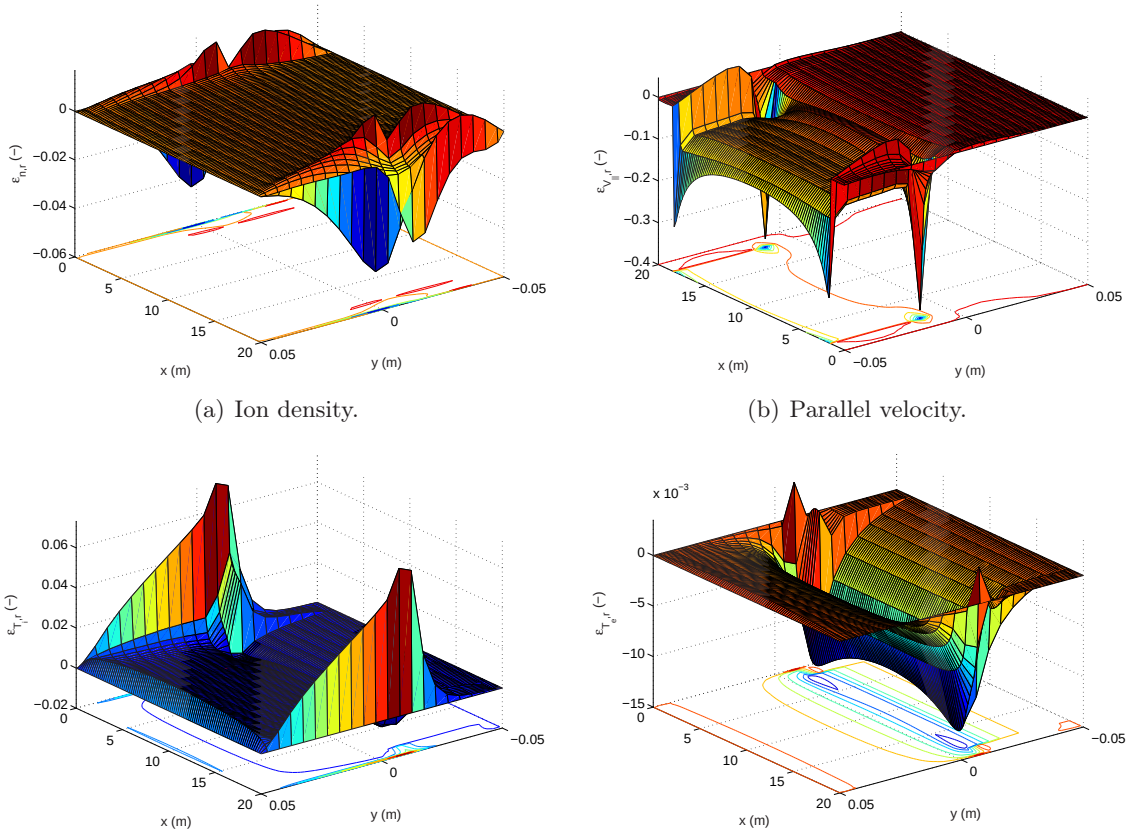
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 7: Relative discretization errors B2. The errors are for the finest grid (400 by 80 cells), but are evaluated at the cells of the coarse grid (100 by 20 cells).



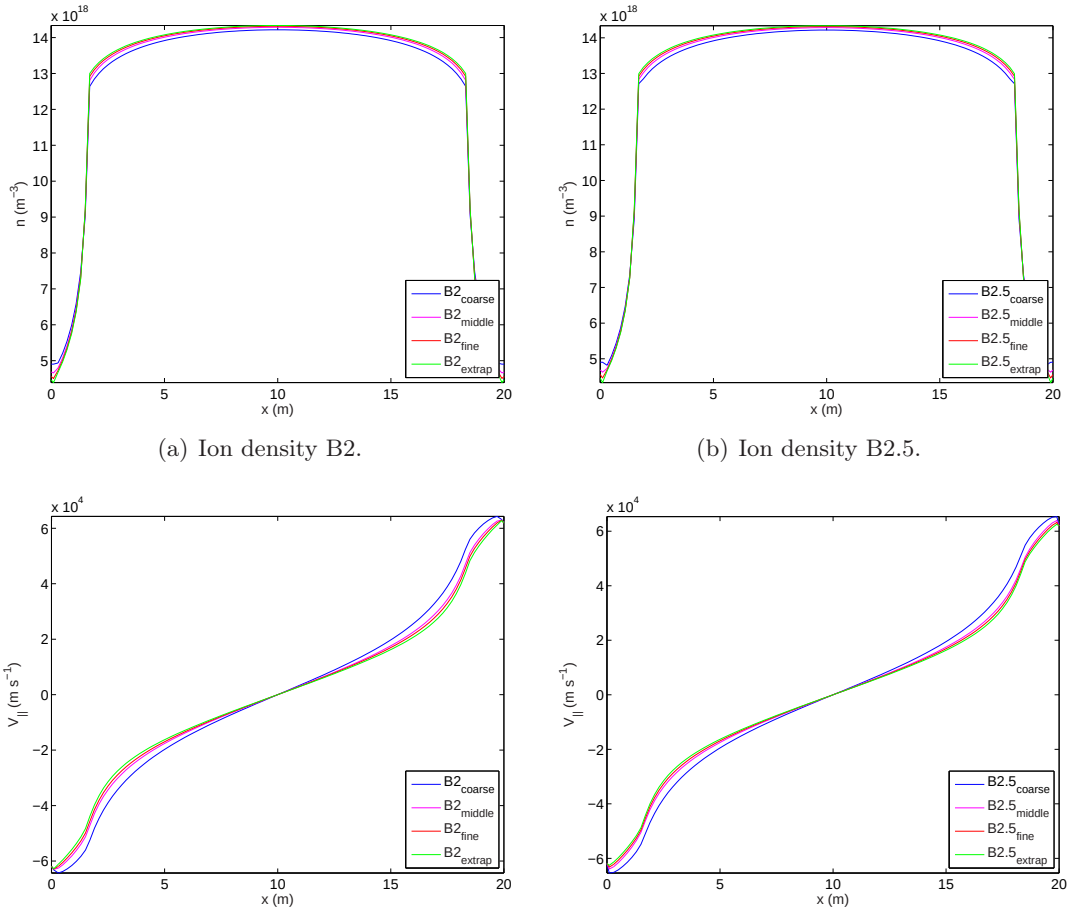
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 8: Relative discretization errors B2.5. The errors are for the finest grid (400 by 80 cells), but are evaluated at the cells of the coarse grid (100 by 20 cells).



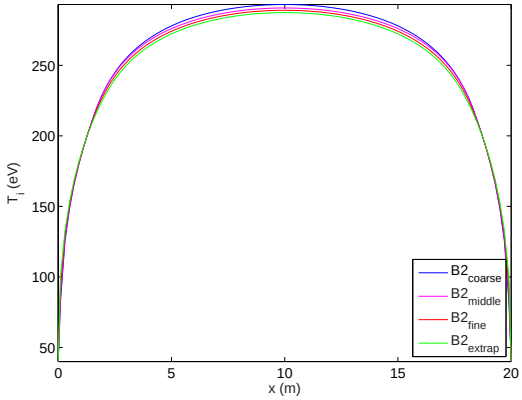
(a) Ion density B2.

(b) Ion density B2.5.

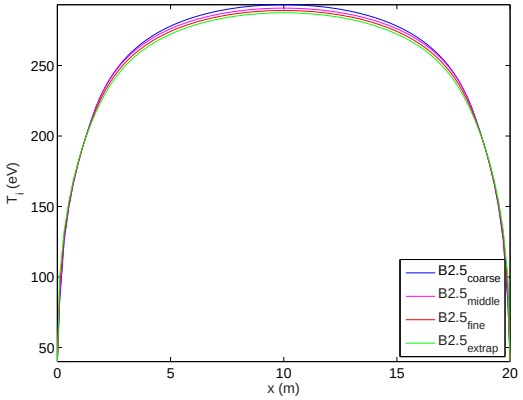
(c) Parallel velocity B2.

(d) Parallel velocity B2.5.

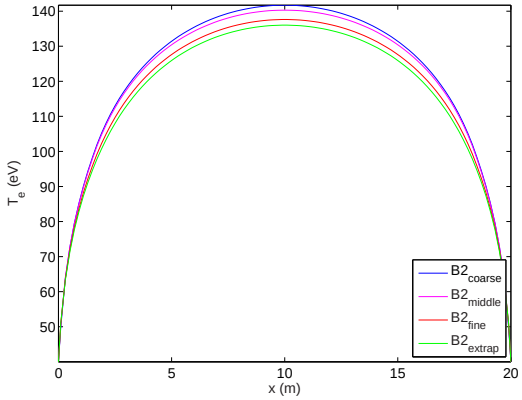
Figure 9: Poloidal profiles of n and $V_{||}$, at $y = -0.0075$m (just outside of the separatrix).



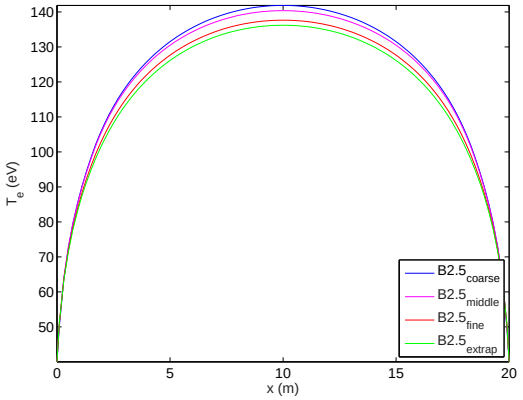
(a) Ion temperature B2.



(b) Ion temperature B2.5.

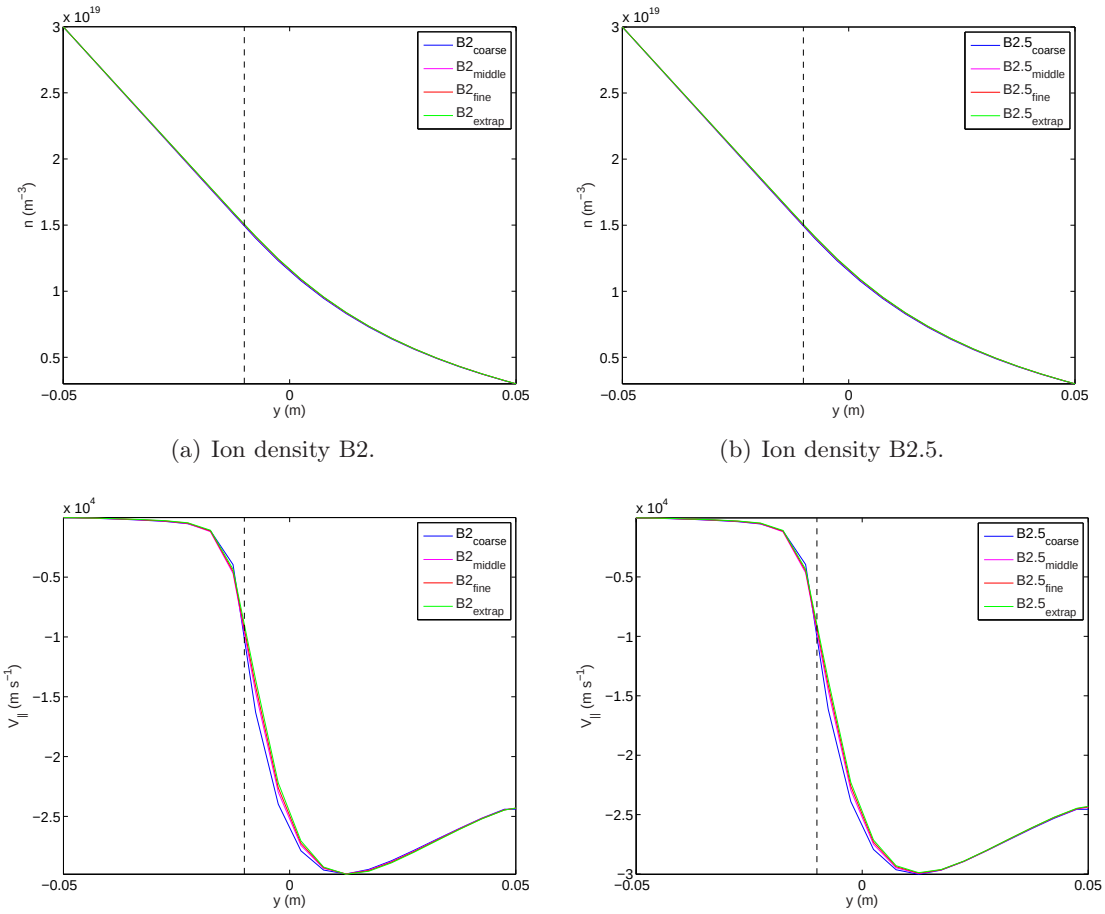


(c) Electron temperature B2.



(d) Electron temperature B2.5.

Figure 10: Poloidal profiles of T_i and T_e, at $y = -0.0075$m.



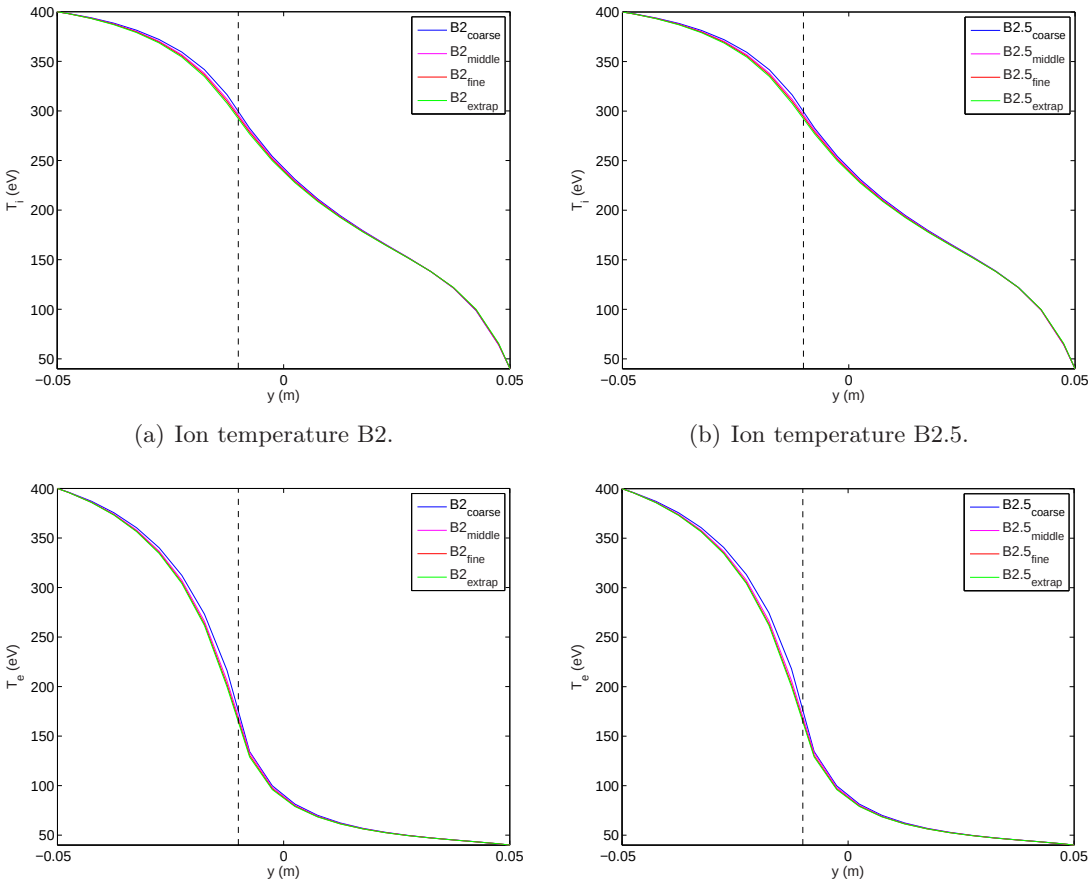
(a) Ion density B2.

(b) Ion density B2.5.

(c) Parallel velocity B2.

(d) Parallel velocity B2.5.

Figure 11: Radial profiles of n and $V_{||}$, at $x = 5.7$m. The dashed, vertical line indicates the location of the separatrix.



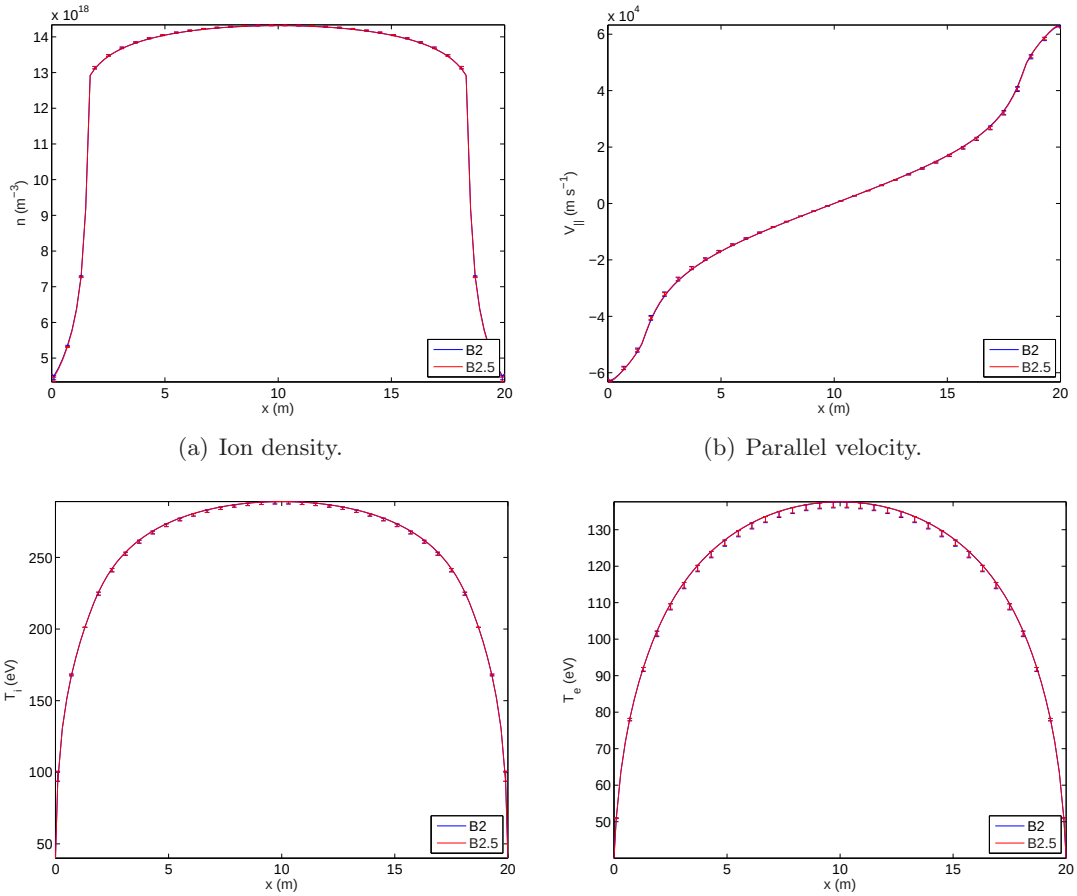
(a) Ion temperature B2.

(b) Ion temperature B2.5.

(c) Electron temperature B2.

(d) Electron temperature B2.5.

Figure 12: Radial profiles of T_i and T_e, at $x = 5.7$m. The dashed, vertical line indicates the location of the separatrix.



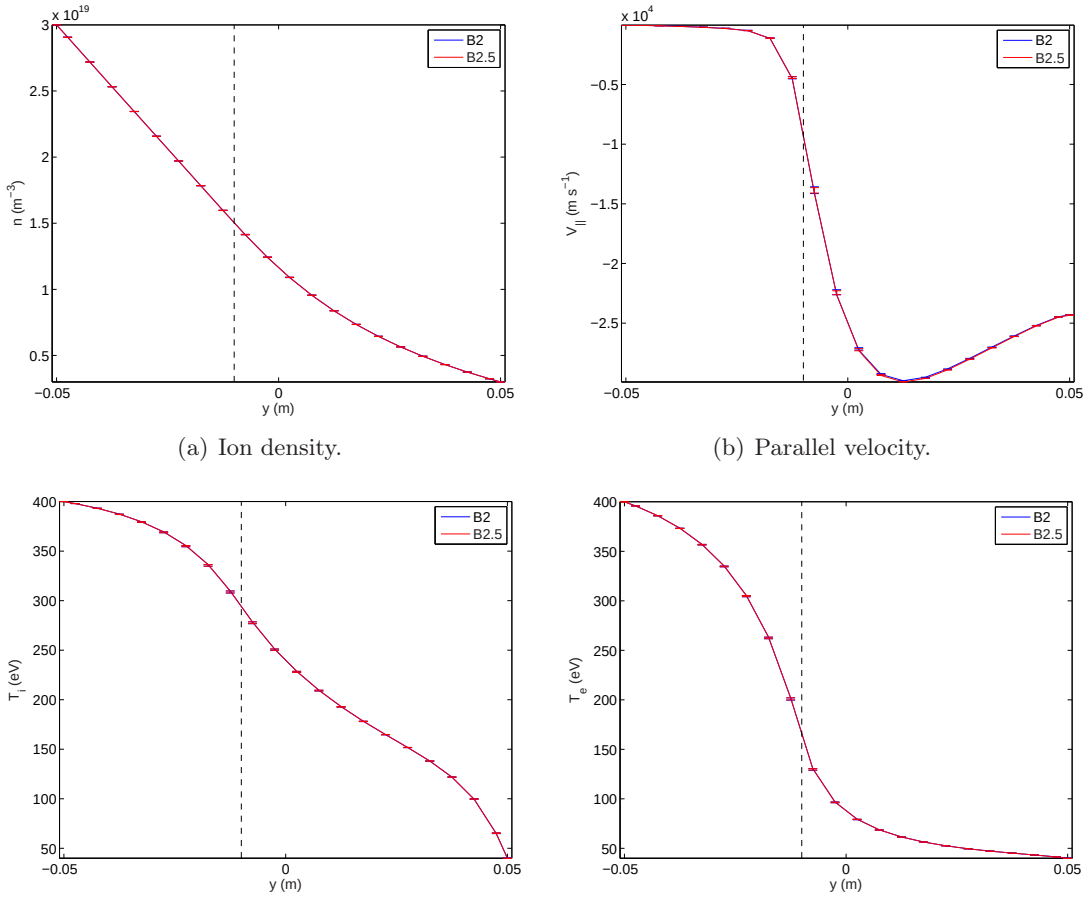
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 13: Fine grid solutions of B2 and B2.5. Error bars indicate the discretization error of the codes. Profiles are taken at $y = -0.0075$m.



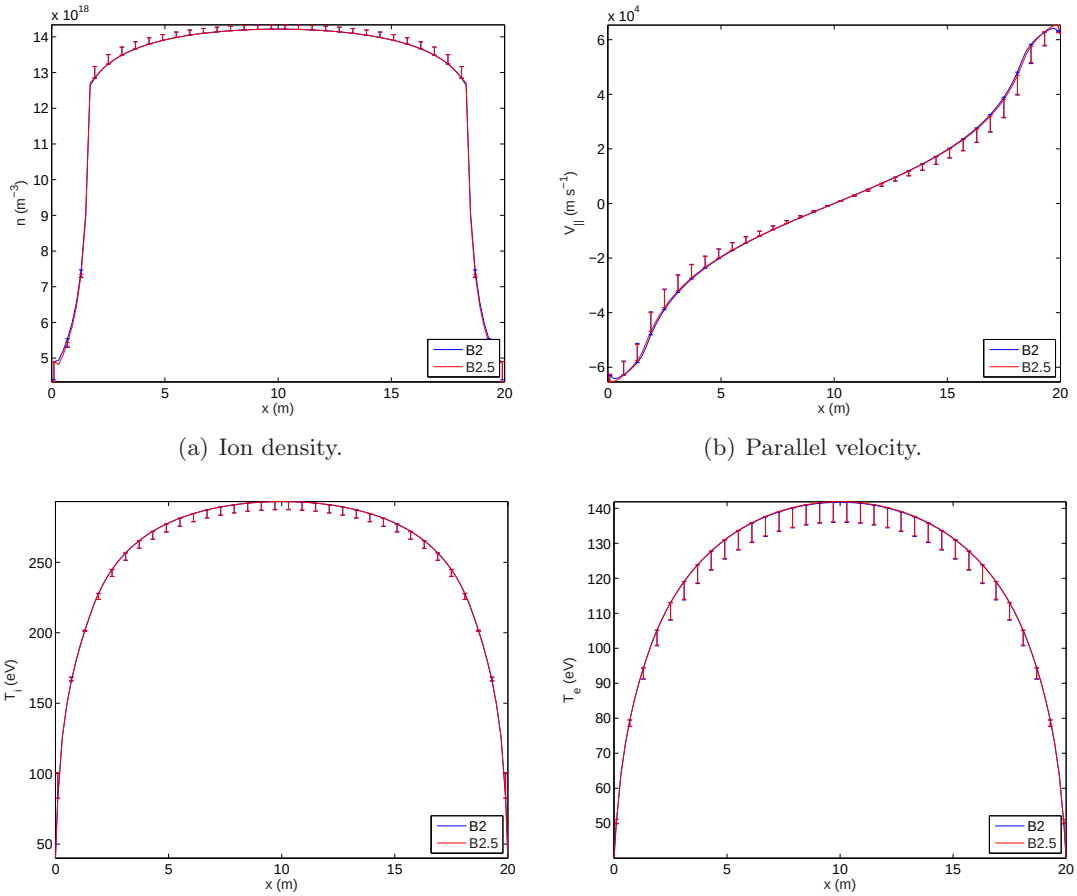
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 14: Fine grid solutions of B2 and B2.5. Error bars indicate the discretization error of the codes. Profiles are taken at $x = 5.7$m. The dashed, vertical line indicates the location of the separatrix.



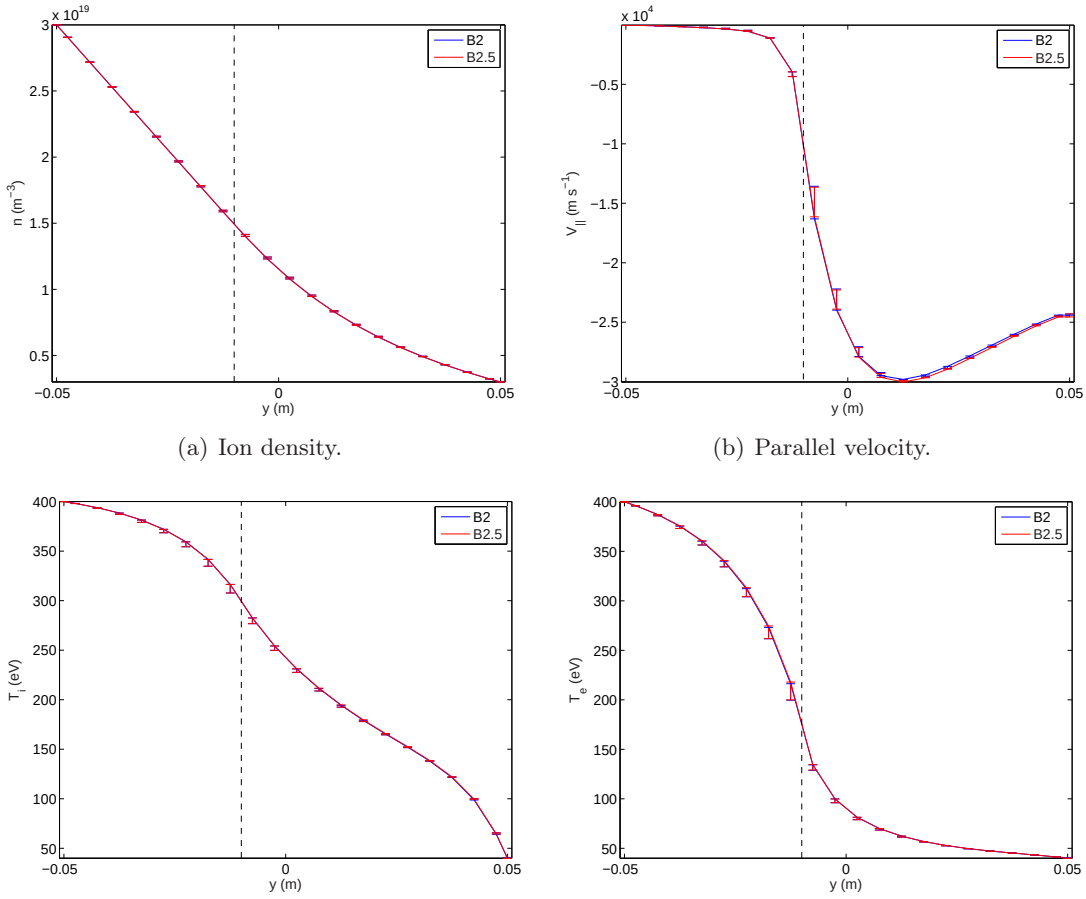
(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 15: Coarse grid solutions of B2 and B2.5. Error bars indicate the discretization error of the codes. Profiles are taken at $y = -0.0075$m.



(a) Ion density.

(b) Parallel velocity.

(c) Ion temperature.

(d) Electron temperature.

Figure 16: Coarse grid solutions of B2 and B2.5. Error bars indicate the discretization error of the codes. Profiles are taken at $x = 5.7$m. The dashed, vertical line indicates the location of the separatrix.

4 Conclusion

In this report, code benchmarking between B2 and B2.5 is presented. Differences in modeling equations and their related transport coefficients are assessed and subsequently removed by either using readily available flags in one of the codes or by small code adaptations. The few (small) differences due to Braginskii vs. Balescu classical transport formulations could be fully removed, for the standard set of equations (without drifts and currents). For a simple set of boundary conditions on a test case without recycling, both codes converge to the same solution, properly taking into account the discretization errors. Further benchmarking is needed in particular for i) boundary conditions, ii) for further physics components (drifts, electrical currents, electric fields) and iii) for the coupling with EIRENE.

For i): Some preliminary results already indicate that especially for more advanced choices of boundary conditions numerical issues might play an even more crucial role than it was the case in the present initial study

For ii) Differences between Braginsikii and Balescu classical transport are expected to become much more relevant then.

For iii) in particular the issue of providing an internal energy source term for B2.5, as compared to the total energy term for B2, may become more challenging due to inherent Monte Carlo noise on moments estimated from contributions which can change sign ("Monte Carlo cancelation issue").

Based on this study and optimistically assuming these further issues can also be resolved satisfactorily, we expect that full backward compatibility can then be established. Nevertheless, certain theoretical questions with respect to particular choices in the modeling equations (in particular under ii) should be further clarified. We note that also the B2 code (version KUL-FZJ) is equipped with options for drifts and currents [8, 9]. Although these options have not been actively used since more than 10 years, this should also serve as a very powerful benchmark case in this respect.

A Derivation of the Internal Energy Equation

In this section, the internal energy equation for ions is derived. We start from the equation for the total energy of the ions. Using the momentum and continuity equations, an equation for the kinetic ion energy is obtained. The internal energy equation is then found by subtracting the kinetic energy equation from the equation for the total energy.

The total energy is the sum of the internal and kinetic energy of the ions:

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{3}{2}nT_i + \frac{1}{2}m_i n V_\parallel^2\right) + \\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\left(\frac{5}{2}nV_xT_i + \frac{1}{2}m_i n V_x V_\parallel^2\right) - \frac{\sqrt{g}}{h_x^2}\left(\kappa_x^i\frac{\partial T_i}{\partial x} + \frac{1}{2}\frac{4}{3}\eta_x^i\frac{\partial V_\parallel^2}{\partial x}\right)\right) + \\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\left(\frac{5}{2}nV_yT_i + \frac{1}{2}m_i n V_y V_\parallel^2\right) - \frac{\sqrt{g}}{h_y^2}\left(\kappa_y^i\frac{\partial T_i}{\partial y} + \frac{1}{2}\eta_y^i\frac{\partial V_\parallel^2}{\partial y}\right)\right) \\
&= S_E + k(T_e - T_i) - \frac{b_x V_\parallel}{h_x}\frac{\partial p_e}{\partial x} - \alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y} - \beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y}\,.
\end{aligned} \tag{40}$$

The vdp-sources are included with parameters α and β, which take the same values as in the electron energy equation. Note the presence of the factor $4/3$ in the poloidal viscous contribution, needed for consistency with the momentum equation. Here, the energy source term originating from interactions with neutrals S_E is included for generality of the derivation of the internal energy equation.

In order to derive the equation for the parallel (macroscopic) kinetic energy of the ions, we multiply the parallel momentum equation by $V_\parallel$:

$$\begin{aligned}
\frac{\partial}{\partial t}\left(m_i n V_\parallel\right)V_\parallel \;&+\; \frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel - \frac{\sqrt{g}}{h_x^2}2\eta_x^i\left(\frac{2}{3}\frac{\partial V_\parallel}{\partial x}\right)\right) \\
&+\; \frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel - \frac{\sqrt{g}}{h_y^2}\eta_y^i\frac{\partial V_\parallel}{\partial y}\right) = V_\parallel S_{m_i V_\parallel} - V_\parallel\frac{B_x}{B}\frac{1}{h_x}\frac{\partial p}{\partial x}\,.
\end{aligned} \tag{41}$$

The momentum source $S_{m_i V_\parallel}$ due to neutrals is included for generality. The time derivative can be rewritten as

$$\frac{\partial}{\partial t}\left(m_i n V_\parallel\right)V_\parallel = \frac{\partial}{\partial t}\left(m_i n V_\parallel V_\parallel\right) - \frac{\partial}{\partial t}\left(V_\parallel\right)m_i n V_\parallel = \frac{\partial}{\partial t}\left(m_i n V_\parallel^2\right) - m_i n\frac{\partial}{\partial t}\left(\frac{V_\parallel^2}{2}\right)\,. \tag{42}$$

An expression for the time derivative of the ion flow kinetic energy can be found by subtracting the continuity equation multiplied by $m_i V_\parallel^2/2$ from the left hand side of (41):

$$m_i\frac{V_\parallel^2}{2}\left(\frac{\partial n}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}nV_x\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}nV_y\right) - S_n\right) = 0\,. \tag{43}$$

Again, the particle source term S_n due to neutrals has been added. Combining time derivative terms leads to

$$\frac{\partial}{\partial t}\left(m_i n V_\parallel\right)V_\parallel - m_i\frac{V_\parallel^2}{2}\frac{\partial n}{\partial t} = \frac{\partial}{\partial t}\left(m_i n V_\parallel^2\right) - m_i n\frac{\partial}{\partial t}\left(\frac{V_\parallel^2}{2}\right) - \frac{V_\parallel^2}{2}\frac{\partial m_i n}{\partial t} = \frac{\partial}{\partial t}\left(m_i n\frac{V_\parallel^2}{2}\right)\,. \tag{44}$$

An analogous procedure is followed for the convective parts:

$$\begin{aligned}
\frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel\right) &+ \frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel\right) \qquad (45)\\
&= \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel^2\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel^2\right)\\
&\quad -m_i n V_x V_\parallel \frac{1}{h_x}\frac{\partial}{\partial x}\left(V_\parallel\right) - m_i n V_y V_\parallel \frac{1}{h_y}\frac{\partial}{\partial y}\left(V_\parallel\right)\\
&= \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel^2\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel^2\right)\\
&\quad -m_i n V_x \frac{1}{h_x}\frac{\partial}{\partial x}\left(\frac{V_\parallel^2}{2}\right) - m_i n V_y \frac{1}{h_y}\frac{\partial}{\partial y}\left(\frac{V_\parallel^2}{2}\right)
\end{aligned}$$

Combining (45) with the convective terms in (43) then gives:

$$\begin{aligned}
\frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel\right) &+ \frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel\right)\\
&\quad -m_i\frac{V_\parallel^2}{2}\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}n V_x\right) - m_i\frac{V_\parallel^2}{2}\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}n V_y\right)\\
&= \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x V_\parallel^2\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y V_\parallel^2\right)\\
&\quad -m_i n V_x \frac{1}{h_x}\frac{\partial}{\partial x}\left(\frac{V_\parallel^2}{2}\right) - m_i n V_y \frac{1}{h_y}\frac{\partial}{\partial y}\left(\frac{V_\parallel^2}{2}\right)\\
&\quad -\frac{V_\parallel^2}{2}\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x\right) - \frac{V_\parallel^2}{2}\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y\right)\\
&= \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x \frac{V_\parallel^2}{2}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y \frac{V_\parallel^2}{2}\right) \qquad (46)
\end{aligned}$$

Also source terms due to neutrals in (41) and (43) can be combined to give the kinetic energy source term:

$$S_{m_i V_\parallel^2} = V_\parallel S_{m_i V_\parallel} - m_i \frac{V_\parallel^2}{2} S_n \,. \qquad (47)$$

Finally, the viscous contribution to Eq. (41) is rewritten as

$$\begin{aligned}
-\frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x^2}\frac{4}{3}\eta_x^i\frac{\partial V_\parallel}{\partial x}\right) &- \frac{V_\parallel}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y^2}\eta_y^i\frac{\partial V_\parallel}{\partial y}\right) \qquad (48)\\
&= -\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x^2}\frac{4}{3}\eta_x^i V_\parallel\frac{\partial V_\parallel}{\partial x}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y^2}\eta_y^i V_\parallel\frac{\partial V_\parallel}{\partial y}\right)\\
&\quad +\frac{4}{3}\eta_x^i\frac{1}{h_x^2}\frac{\partial V_\parallel}{\partial x}\frac{\partial V_\parallel}{\partial x} + \eta_y^i\frac{1}{h_y^2}\frac{\partial V_\parallel}{\partial y}\frac{\partial V_\parallel}{\partial y}\\
&= -\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x^2}\frac{1}{2}\frac{4}{3}\eta_x^i\frac{\partial V_\parallel^2}{\partial x}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y^2}\frac{1}{2}\eta_y^i\frac{\partial V_\parallel^2}{\partial y}\right)\\
&\quad +\frac{4}{3}\eta_x^i\left(\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}\right)^2 + \eta_y^i\left(\frac{1}{h_y}\frac{\partial V_\parallel}{\partial y}\right)^2
\end{aligned}$$

Combining results, the equation for the kinetic energy of the ions is finally:

$$\begin{aligned}\frac{\partial}{\partial t}\left(m_i n\frac{V_\parallel^2}{2}\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}m_i n V_x\frac{V_\parallel^2}{2} - \frac{\sqrt{g}}{h_x^2}\frac{1}{2}\frac{4}{3}\eta_x^i\frac{\partial V_\parallel^2}{\partial x}\right) \\ &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}m_i n V_y\frac{V_\parallel^2}{2} - \frac{\sqrt{g}}{h_y^2}\frac{1}{2}\eta_y^i\frac{\partial V_\parallel^2}{\partial y}\right) \\ &= S_{m_i V_\parallel^2} - \frac{b_x V_\parallel}{h_x}\frac{\partial p}{\partial x} - \frac{4}{3}\eta_x^i\left(\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}\right)^2 - \eta_y^i\left(\frac{1}{h_y}\frac{\partial V_\parallel}{\partial y}\right)^2.\end{aligned} \tag{49}$$

The equation for the kinetic energy (49) is now subtracted from the total ion energy equation (40) to yield the internal ion energy equation:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}nT_i\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{5}{2}nV_xT_i - \frac{\sqrt{g}}{h_x^2}\kappa_x^i\frac{\partial T_i}{\partial x}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\frac{5}{2}nV_yT_i - \frac{\sqrt{g}}{h_y^2}\kappa_y^i\frac{\partial T_i}{\partial y}\right) \\ &= S_E - S_{m_i V_\parallel^2} + k(T_e - T_i) + \frac{b_x V_\parallel}{h_x}\frac{\partial p_i}{\partial x} - \alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y} - \beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y} \\ &\quad + \frac{4}{3}\eta_x^i\left(\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}\right)^2 + \eta_y^i\left(\frac{1}{h_y}\frac{\partial V_\parallel}{\partial y}\right)^2.\end{aligned} \tag{50}$$

Here $p_i = nT_i$. As was done for the electron energy equation, the relation

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}nb_xV_\parallel T_i\right) - \frac{b_xV_\parallel}{h_x}\frac{\partial p_i}{\partial x} = \frac{p_i}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_xV_\parallel\right) \tag{51}$$

is used to rewrite (50):

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}nT_i\right) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{3}{2}nb_xV_\parallel T_i - \frac{\sqrt{g}}{h_x^2}\frac{5}{2}D^n\frac{\partial n}{\partial x}T_i - \frac{\sqrt{g}}{h_x^2}\kappa_x^i\frac{\partial T_i}{\partial x}\right) \\ &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(-\frac{\sqrt{g}}{h_y^2}\frac{5}{2}D^n\frac{\partial n}{\partial y}T_i - \frac{\sqrt{g}}{h_y^2}\kappa_y^i\frac{\partial T_i}{\partial y}\right) + \frac{p_i}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_xV_\parallel\right) \\ &= S_E - S_{m_iV_\parallel^2} + k(T_e - T_i) - \alpha\frac{V_y}{h_y}\frac{\partial p_e}{\partial y} - \beta\frac{V_y}{h_y}\frac{\partial p_i}{\partial y} \\ &\quad + \frac{4}{3}\eta_x^i\left(\frac{1}{h_x}\frac{\partial V_\parallel}{\partial x}\right)^2 + \eta_y^i\left(\frac{1}{h_y}\frac{\partial V_\parallel}{\partial y}\right)^2.\end{aligned} \tag{52}$$

This last equation is the internal energy equation. Solving this equation is equivalent to solving the total energy equation (40) if both the momentum and continuity equations are solved along with them.

B The Discretization Error

(Taken from: W. Dekeyser, "Effect of main chamber recycling on the plasma transport in the ITER fusion reactor", Master thesis, KU Leuven, 2009, available from the author.)

In a numerical code, the exact differential equations can't be solved. Rather, these equations are discretized in space and time, resulting in a set of algebraic difference equations. The choice of the grid on which the equations are discretized, will have an influence on the accuracy of the solution. It is convenient to have an idea of the discretization error that accompanies the solution on a certain grid. This error can be estimated by a method described in [12].

The exact solution to the differential equations, $\overline{\phi}$, does not satisfy the difference equations. Due to the truncation of the Taylor series during discretization, there is a truncation error $\tau_{\Delta x}$, where Δx refers to the size of the grid cells:

$$L(\overline{\phi}) = L_{\Delta x}(\overline{\phi}) + \tau_{\Delta x}\,. \tag{53}$$

Here, the operator L denotes the differential equations, while $L_{\Delta x}$ is the operator representing the algebraic equations obtained after discretization on grid Δx. The solution to the difference equations on grid Δx, $\phi_{\Delta x}$, differs from the exact solution $\overline{\phi}$ by the discretization error $\epsilon_{\Delta x}$:

$$\overline{\phi} = \phi_{\Delta x} + \epsilon_{\Delta x}\,. \tag{54}$$

Substituting into equation (53), using the obvious fact that $L_{\Delta x}(\phi_{\Delta x}) = 0$, and assuming a linear operator $L_{\Delta x}$, it is found:

$$L_{\Delta x}(\epsilon_{\Delta x}) = -\tau_{\Delta x}\,. \tag{55}$$

Although this equation was derived for linear operators, it is assumed that the result can be extrapolated to nonlinear operators as well [12].

An estimate of the discretization error can now be obtained by solving the difference equations on three different grids, with sizes $4\,\Delta x$, $2\,\Delta x$ and Δx. For sufficiently fine grids, the truncation and discretization errors will both be proportional to the leading term in the Taylor series. If the order of the discretization scheme is p, the error is approximately:

$$\epsilon_{\Delta x} \approx a\,(\Delta x)^p\,, \tag{56}$$

with a a constant. For the three grids, this means that

$$\overline{\phi} \approx \phi_{\Delta x} + a\,(\Delta x)^p \approx \phi_{2\,\Delta x} + a\,(2\,\Delta x)^p \approx \phi_{4\,\Delta x} + a\,(4\,\Delta x)^p\,. \tag{57}$$

With this relation, the order of the discretization scheme can be estimated as

$$p \approx \frac{\log\left(\frac{\phi_{2\,\Delta x} - \phi_{4\,\Delta x}}{\phi_{\Delta x} - \phi_{2\,\Delta x}}\right)}{\log 2}\,, \tag{58}$$

while the discretization error *on the finest grid* is approximately

$$\epsilon_{\Delta x} \approx \frac{\phi_{\Delta x} - \phi_{2\,\Delta x}}{2^p - 1}\,. \tag{59}$$

This expression gives an absolute error. Relative errors are often more useful in evaluating the accuracy of a solution. Therefore, a relative discretization error is defined by dividing $\epsilon_{\Delta x}$ by $\phi_{\Delta x}$:

$$\epsilon^{\text{rel}}_{\Delta x} \approx \frac{1 - \frac{\phi_{2\,\Delta x}}{\phi_{\Delta x}}}{2^p - 1}\,. \tag{60}$$

Referring to equation (54), the exact solution to the partial differential equations can be found by adding the discretization error to the numerical solution obtained with the codes (Richardson extrapolation). This method is used in this work in order to compare the B2 and B2.5 solutions on a sound basis. Indeed, by calculating the discretization error for both codes and adding these errors to the respective solutions, in principle the same, exact solution to the system of PDEs is obtained. We argue then that the codes have truly found 'identical, exact' solutions if the remaining difference between the extrapolated numerical solutions is sufficiently small compared to the discretization error (e.g. an order of magnitude smaller).

C Input Parameters

Here, we present an overview of the relevant settings and input parameters used for the B2 and B2.5 codes in order to obtain the results described in Sec. 3.

C.1 B2 Input Parameters

EQUS-namelist

Variable	Value	Description
NLRADE	0	exclude vdp-terms
VPFLAG	1	use anomalous particle diffusion term in poloidal direction
BRMFLAG	0	no sources due to Brehmsstrahlung

PARA-namelist

Variable	Value	Description
DIFNI	1.0	particle diffusion coefficient
DIFPR	0.0	
DIFBO	0.0	
DIFMX	10.0	
VCONV	0	no additional radial convective velocity (anomalous)
FLIM	1.0E30	no electron heat conductivity flux limiter
FLIMI	1.0E30	no ion heat conductivity flux limiter
FLIMV	2.666667	viscous flux limiter
FLIMVI	1	consistent viscous flux limiter in ion energy equation
KYE	1.0	radial (anomalous) electron heat conduction coefficient
KYI	1.0	radial (anomalous) ion heat conduction coefficient
KXE	1.3228	factor for classical electron heat conduction coefficient
KXI	1.0140	factor for classical ion heat conduction coefficient
TRAVIS	0.2	radial (anomalous) viscosity
PARVIS	0.9924	factor for classical parallel viscosity
LHCXANOM	1	anom. heat conduction coefficients in poloidal direction

BOUND-namelist

Variable	Value	Description
NCORN	0	no special treatment for corner cells

NUMS-namelist

Variable	Value	Description
LUPWINDCONT	1	use upwind scheme for continuity equation

C.2 B2.5 Input Parameters

b2.numerics.parameters-namelist

Variable	Value	Description
SOLVECO(0,0)	.false.	Indicates whether the continuity equation for species (0-neutrals) is to be solved in region (ireg=0,1,2,3,4).
SOLVECO(0,1)	.false.	
SOLVECO(0,2)	.false.	
SOLVECO(0,3)	.false.	
SOLVECO(0,4)	.false.	
SOLVEMO(0,0)	.false.	Indicates whether the momentum equation for species (0-neutrals) is to be solved in region (ireg=0,1,2,3,4).
SOLVEMO(0,1)	.false.	
SOLVEMO(0,2)	.false.	
SOLVEMO(0,3)	.false.	
SOLVEMO(0,4)	.false.	

b2.transport.parameters-namelist

Variable	Value	Description
flag_dna	1	set up the value of the ion anomalous diffusion coefficient
flag_vsa	1	set up the value of the ion anomalous viscosity coefficient
flag_hci	1	set up the value of the ion anomalous heat coefficient
flag_hce	1	set up the value of the electron anomalous heat coefficient
parm_dna	1.0	the value of the ion anomalous diffusion coefficient
parm_vsa	0.2	the value of the ion anomalous viscosity coefficient
parm_hci	1.0	the value of the ion anomalous heat coefficient
parm_hce	1.0	the value of the electron anomalous heat coefficient

b2mn.dat

Internal Key	Value	Description
b2trno_dpa_x_contr	0	no contribution from neutrals
b2trno_dpa_y_contr	0	no contribution from neutrals
b2tral_visc_style	1	no anomalous contribution in poloidal direction
b2sqel_phm0	0.0	set up the rate coefficient for ionization
b2sqel_phm1	0.0	set up the rate coefficient for recombination
b2sqel_phm2	0.0	set up the rate coefficient for electron heat loss
b2sqcx_phm0	0.0	set up the charge exchange rate coefficient
b2tqna_neutral_contr_to_hci'	0	no neutrauls contributions to ion heat coefficient
b2trcl_lambda	12.0	set up the value of Coulomb logarithm
b2tfnb_xfrhie	0.0	no smooth term in x-particle flux
b2tfnb_xfrhiehz	0.0	no smooth term in x-particle flux in parallel balance momentum equation
b2tfhe_fch_pTe	0.0	no current
b2tfhe_alfTeEh	0.0	no electrical field
b2npmo_modvis	0	set up old expression for viscocity
b2sicf_phm0	0.0	no centrifugal force
b2sicf_phm1	0.0	no centrifugal force
b2mndr_hz	0.0	use old form the parallel balance momentum equation
b2tlmv_style	0	apply viscous flux limit for all surfaces
b2news_poteq	0	zero electrical potential, zero currents
b2tfhi_mdf	0	no modified fluxes in the temperature equation for ions
b2tfhe_mdf	0	no modified fluxes in the temperature equation for electron
b2tfnb_mdf	0	no modified fluxes in the continuity equation for ions
b2trcl_lluciani	0	no Luciani restriction
b2siav_addvis	0.0	no new expression for viscosity
b2siav_addvis1	0.0	no new expression for parallel viscosity connected with heat flow

D Odd-even decoupling Oscillations in B2.5

For some test cases, oscillations due to odd-even decoupling have been noted in B2.5 results, an effect expected due to the use of collocated grids in B2.5, as opposed to the staggered grid technique in B2 which is not vulnerable to this problem. These oscillations tend to appear for cases where sheath conditions are applied at the targets, when convection becomes more dominant than in the test cases discussed in the main body of this report. In this appendix, a short first analysis of these oscillations is carried out for a specific test case.

D.1 Description of the test case

For this case, all state equations were solved: continuity, momentum and energy equations. The geometry is the same as the other benchmark tests: the simplified inboard half of a double null divertor SOL. Results are shown for a grid with 100x20 cells. The following boundary conditions were applied (i.e. we now use conventional sheath outflow boundary conditions rather an fixed values or their derivatives):

Core plasma

- Density: $n = 3 \cdot 10^{19}$ ions/m^3
- Parallel velocity: $\frac{\partial V_\parallel}{\partial y} = 0$
- Electron temperature: $T_e = 400$ eV
- Ion temperature: $T_i = 400$ eV

Targets

Sheath conditions are applied, with following parameters:

- Density: $\frac{\partial n}{\partial x} = 0$
- Parallel velocity: $M = 1$
- Electron sheath transmission coefficient (B2): $\delta_e = 2$ (+ 3.1 for the sheath potential)
- Ion sheath transmission coefficient (B2): $\delta_i = 2.5$ (and 1 for kinetic energy)

Walls and private flux

- Density: $\frac{\partial n}{\partial y} = 0$
- Parallel velocity: $\frac{\partial V_\parallel}{\partial y} = 0$
- Electron temperature: $\frac{\partial T_e}{\partial y} = 0$
- Ion temperature: $\frac{\partial T_i}{\partial y} = 0$

D.2 Results and Discussion

Fig. 17 shows the density profiles in the poloidal direction (ranging from $x = 0$ m to $x = 1.5$ m, i.e. 8 interior cells on the 100x20 grid) at a radial position $y = -0.0375$ m, for both B2 and B2.5. This corresponds to a location in the private flux region. Clearly, the density of B2.5 shows oscillations from cell to cell. The corresponding velocity profiles are plotted in Fig. 18. Also here, the same type of oscillations are noted. When looking at the particle flux through cell faces, however, we see that this flux is smooth (Fig. 19). Apparently, the oscillations are filtered by the interpolation to the cell faces. This implies that the oscillations are not 'felt' by the pressure correction equation. Thus, despite the compact pressure stencil in this equation, the oscillations in pressure are not corrected.

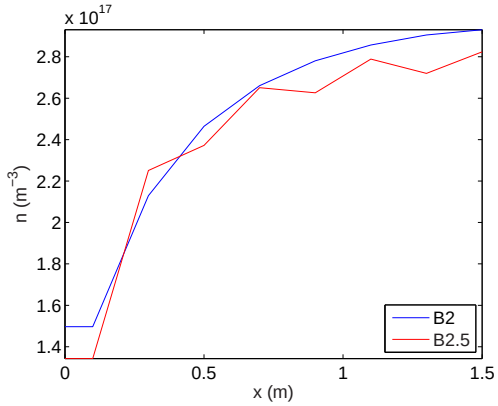


Figure 17: Density profiles in the private flux region (PR), at $y = -0.0375$ m. Blue: B2, Red: B2.5. Differences here, albeit small ($\approx 10\%$) are now significantly larger than the discretization errors.

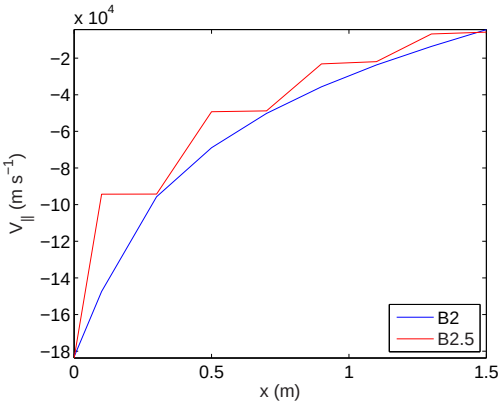


Figure 18: Parallel velocity profiles in the PR, at $y = -0.0375$ m. Note: by coincidence, this kind of oscillation retains the monotonic behavior and therefore would essentially be undetectable in the more commonly employed 2D contour plots of parallel velocity.

Turning to the momentum equation, we investigate the poloidal momentum flux for this solution, Fig. 20. It is quite smooth, probably mainly due to the smooth particle flux through cell faces combined with interpolated velocities. The pressure gradients computed by B2.5 on the cell faces, Fig. 21 (green curve), show strong oscillations due to the oscillating pressure, Fig. 22. Interpolating to cell centered pressure gradients, Fig. 21 (red curve), largely removes these oscillations (typical for the 'broad' stencil used for the pressure gradient in the momentum equation). Thus, also the momentum equation will not remove

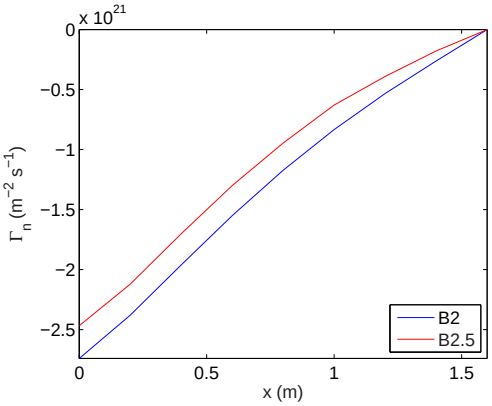


Figure 19: Poloidal particle fluxes in the PR, at $y = -0.0375$ m.

the oscillations in the velocities.

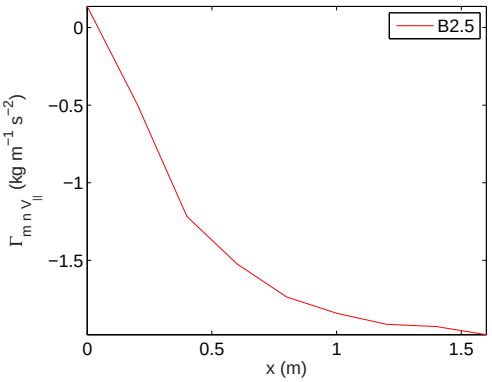


Figure 20: Poloidal momentum flux (B2.5) in the PR, at $y = -0.0375$ m.

Eliminating the oscillations due to odd-even decoupling

The following 'quick' solution to eliminate the oscillations has been investigated with B2.5:

- Use of a compact stencil for the pressure gradient in momentum equation, combined with adapted interpolation for the face values of velocities: e.g. use 'backward differencing' for the pressure gradient, and 'forward differencing' for the face velocities (or vice versa).

Although this strategy is effective in removing the oscillations, the asymmetry introduced with these schemes turned out to be too large for useful comparison of the outputs of the B2 and B2.5 codes. Higher (second) order schemes are thus desirable for the pressure gradient.

For a good 'long-term' solution to the oscillation problem, it is better to keep the current implementation of pressure gradient in momentum equation (see Notes below), and use following improvements to the pressure correction equation:

- Rhie-Chow interpolation for the velocities on cell faces, while making sure that the interpolated face values remain in between the cell centered values.

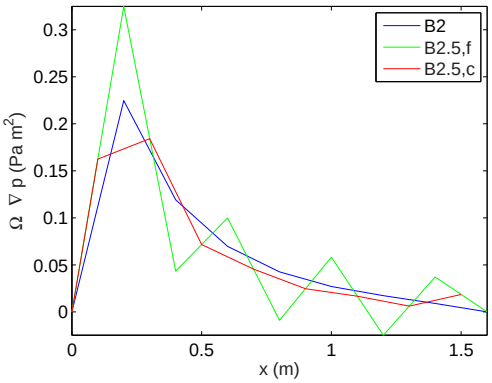


Figure 21: Pressure gradients at $y = -0.0375$ m. For B2.5, both face values (f) and cell center values (c) are shown.

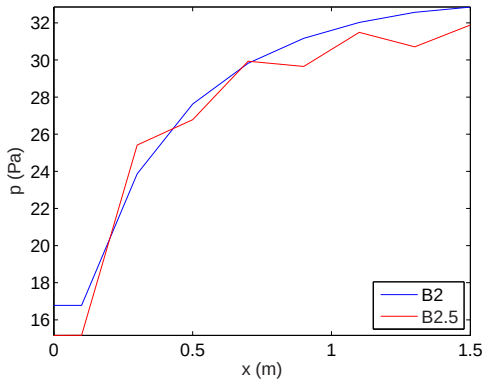


Figure 22: Pressure in the PR, at $y = -0.0375$ m.

- Store velocity corrections on cell faces (not interpolated to cell centers, as this could average out the corrections which fight the oscillations), and use them to update face fluxes.

Notes

- The interpolation procedure used in B2.5 to calculate pressure gradients in cell centers from cell face values, could lead to 'overshooting': the sum of the coefficients specifying the linear interpolation is not 1. This tends to happen in the cells next to a guard cell if the pressure gradient on the boundary is *not* zero, due to the very small volume of the guard cell. Could this explain the sudden jumps in parallel velocity sometimes found in front of the targets in B2.5?

  The interpolation scheme:

$$(\nabla p)_P = V_P \cdot \left(\frac{(\nabla p)_w}{V_W + V_P} + \frac{(\nabla p)_e}{V_P + V_E} \right) \tag{61}$$

  with V the volume of a cell, W, P and E referring to cell centers, and w and e to cell faces.

  Possibly, it is more desirable to calculate cell center gradients based on surface integrals (and replace the cell center gradients in the guard cells by face gradients on the boundary of the domain).

E B2 Bug Fixes

E.1 Interpolation of radial viscous momentum flux

Due to an incorrect interpolation, the radial viscous momentum flux is unintentionally doubled in the3 ITER-IO version of B2 (see Sec. 2.2.1). The factor of two can easily be eliminated by making the correction in subroutine B2CMOM as suggested below.

- **Old code**

```
* - - - - - - - - - - - - - - - - - - - - - - - -
* COMPUTE THE RESIDUAL
* - - - - - - - - - - - - - - - - - - - - - - - -


...
* - - EVALUATE FLOY AND CONY - -

...
CONY(IX,IY) = SY(IX,IY) *
. 2*(DY(IX,IY)*VISY(IX,IY,IS)+DY(IX,IY+1)*VISY(IX,IY+1,IS))/
. (DY(IX,IY)+DY(IX,IY+1))**2
```

- **Corrected code**

```
* - - - - - - - - - - - - - - - - - - - - - - - -
* COMPUTE THE RESIDUAL
* - - - - - - - - - - - - - - - - - - - - - - - -


...
* - - EVALUATE FLOY AND CONY - -

...
CONY(IX,IY) = SY(IX,IY) *
. (DY(IX,IY)*VISY(IX,IY,IS)+DY(IX,IY+1)*VISY(IX,IY+1,IS))/
. (DY(IX,IY)+DY(IX,IY+1))**2
```

E.2 Correction to viscous energy terms

In B2 subroutine B2CENE, the viscosity coefficient appearing in the viscous contribution to the ion energy balance is incorrectly coded with a factor 2 instead of 4/3. Furthermore, this viscosity coefficient is not flux limited, even though a flux limit may have been applied in the momentum equation. Below, the original code is given, and also a correction is suggested that makes the viscous energy terms in B2CENE consistent with the momentum equation.

Note that similar changes may turn out to be required for the boundary conditions.

- **Old code**

```
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
* COMPUTE THE KINETIC AND VISCOUS ION ENERGY FLOW
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


...
* - - COMPUTE FLOXI AND CONXI - -
...
CONXI(IX,IY) = SX(IX,IY) *
. AVE(VISX(IX,IY,IS)/DX(IX,IY),
. VISX(IX+1,IY,IS)/DX(IX+1,IY))
```

- **Corrected code**

```
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
* COMPUTE THE KINETIC AND VISCOUS ION ENERGY FLOW
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


...
* - - COMPUTE FLOXI AND CONXI - -
...
CONXI(IX,IY) = SX(IX,IY) * 2.0D0/3.0D0 *
. AVE(VISX(IX,IY,IS)/DX(IX,IY),
. VISX(IX+1,IY,IS)/DX(IX+1,IY))


IF (FLIMVI .EQ. 1) THEN
CSH = -CONXI(IX,IY)*(UP(IX,IY,IS) - UP(IX+1,IY,IS))
QFL = FLIMV*AVE(NI(IX,IY,IS)*TI(IX,IY)*
. ABS(PIT(IX,IY))*VOL(IX,IY)/DX(IX,IY),
. NI(IX+1,IY,IS)*TI(IX+1,IY)*
. ABS(PIT(IX+1,IY))*VOL(IX+1,IY)/DX(IX+1,IY))
CONXI(IX,IY)=CONXI(IX,IY)/(1.0D0 + ABS(CSH/QFL))
ENDIF
```

Note that also some extra parameters will need to be passed in the call to B2CENE (FLIMV and FLIMVI). FLIMVI is a newly defined input flag. The viscous flux limit will be applied in the energy equation if FLIMVI = 1.

References

[1] V. Kotov, D. Reiter, A.S. Kukushkin, *Numerical study of the ITER divertor plasma with the B2-EIRENE code package*, Report, Forschungszentrum Jülich Jül-4257, November 2007.

[2] B.J. Braams, *Computational studies in tokamak equilibrium and transport* PhD thesis, University of Utrecht, the Netherlands, 1986, and: B.J. Braams, *A Multi-Fluid Code for Simulation of the Edge Plasma in Tokamaks*, NET Report Nr. 68, 1987.

[3] B.J. Braams, *Radiative divertor modelling for ITER and TPX*, Contrib. Plasma Phys. 36 (1996) No. 2/3, 276-281.

[4] X. Bonnin, A.S. Kukushkin, D.P. Coster, *Code development for ITER edge modelling – SOLPS5.1*, J. Nucl. Mater. 390-391 (2009) 274-277.

[5] V. Kotov, D. Reiter, D.P. Coster, A.S. Kukushkin, *Verification of the 2D Tokamak Edge Modelling Codes for Conditions of Detached Divertor Plasma*, Contrib. Plasma Phys. 50 (2010) No. 3-5, 292-298.

[6] X. Bonnin, D. Coster, *Transition of ITER edge simulations from SOLPS4 to SOLPS5 with drifts and currents*, Final report on Contracts ITER/CT/07/346 and ITER/CT/07/4100001312, 2009, ITER document reference: 2WBE8W.

[7] V. Kotov (FZ Jülich), I. Veselova (St. Petersburg State Polytechnical University), *Benchmarks of SOLPS4.3 (ITER/FZJ) versus SOLPS5.2 (SPb)*, Sept. 2009 - March 2010, private communication.

[8] M. Baelmans, *Code Improvements and Applications of a Two-dimensional Edge Plasma Model for Toroidal Devices*, PhD thesis, Departement Mechanica, Faculteit der Toegepaste Wetenschappen, Katholieke Universiteit Leuven, October 1993.

[9] M. Baelmans, D. Reiter, R.R. Weynants, *New developments in plasma edge modelling with particular emphasis on drift flows and electric fields* Contrib. Plasma Phys. 36 (1996) 117-126.

[10] R. Balescu, *Transport Processes in Plasmas*, Vol 1, North-Holland, Amsterdam, 1988.

[11] S.I. Braginskii, *Transport processes in a plasma*, Reviews of Plasma Physics 1 (1965) 250-311.

[12] J.H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 2002.

[13] V. Rozhansky, S. Voskoboynikov, E. Kaveeva, D. Coster, R. Schneider, *Simulation of tokamak edge plasma including self-consistent electric fields*, Nucl. Fusion 41 (2001) 387.

Jül-4337
Januar 2011
ISSN 0944-2952