The full-F electromagnetic model in toroidal geometry FELTOR

M. Wiesenberger and M. Held

October 4, 2020

Abstract

The purpose of this document is to describe the programs `feltor_hpc.cu`, `feltor.cu`, `feltor_diag.cu` and to an extend `geometry_diag.cu`. The goal is to provide information such that a user can avoid to look into the actual codes on the one side and connect the presented formulas to relevant journal publications on the other.

The program `feltor/inc/geometries/geometry_diag.cu` analyses the magnetic field geometry. `feltor_hpc.cu` and `feltor.cu` are programs for global 3d isothermal electromagnetic full-F gyro-fluid simulations. `feltor/diag/feltor_diag.cu` is a program to analyse the output file(s) of `feltor_hpc.cu`.

Contents

1 The magnetic field

We assume a three-dimensional flat space with arbitrary coordinate system $\boldsymbol{x} := \{x_0, x_1, x_2\}$, metric tensor g_{ij} and volume element $\sqrt{g} := \sqrt{\det g}$. Given a vector field $\boldsymbol{B}(\boldsymbol{x})$ with unit vector $\hat{\boldsymbol{b}}(\boldsymbol{x}) := (\boldsymbol{B}/B)(\boldsymbol{x})$ we can define various differential operations.

Name	Symbol	Definition
Perpendicular Poisson bracket	$[.,.]_\perp$	$[f,g]_\perp := \hat{\boldsymbol{b}} \cdot (\boldsymbol{\nabla} f \times \boldsymbol{\nabla} g) = b_i \varepsilon^{ijk} \partial_j f \partial_k g / \sqrt{g}$
Projection Tensor	h	$h^{ij} := g^{ij} - b^i b^j$
Perpendicular Gradient	$\boldsymbol{\nabla}_\perp$	$\boldsymbol{\nabla}_\perp f := \hat{\boldsymbol{b}} \times (\boldsymbol{\nabla} f \times \hat{\boldsymbol{b}}) \equiv h \cdot \boldsymbol{\nabla} f$
Perpendicular Laplacian	$\Delta_\perp$	$\Delta_\perp f := \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla}_\perp f) = \boldsymbol{\nabla} \cdot (h \cdot \boldsymbol{\nabla} f)$
Curl-b Curvature	$\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}$	$\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(f) := \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} \cdot \boldsymbol{\nabla} f = \frac{1}{B}(\boldsymbol{\nabla} \times \hat{\boldsymbol{b}}) \cdot \boldsymbol{\nabla} f$

Grad-B Curvature	$\mathcal{K}_{\boldsymbol{\nabla} B}$	$\mathcal{K}_{\boldsymbol{\nabla} B}(f) := \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla B}} \cdot \boldsymbol{\nabla} f = \frac{1}{B}(\hat{\boldsymbol{b}} \times \boldsymbol{\nabla} \ln B) \cdot \boldsymbol{\nabla} f$
Curvature	$\mathcal{K}$	$\mathcal{K}(f) := \boldsymbol{\mathcal{K}} \cdot \boldsymbol{\nabla} f = \boldsymbol{\nabla} \cdot \left(\frac{\hat{\boldsymbol{b}} \times \boldsymbol{\nabla} f}{B}\right)$,
Parallel derivative	$\nabla_\parallel$	$\nabla_\parallel f := \hat{\boldsymbol{b}} \cdot \boldsymbol{\nabla} f$
Parallel Laplacian	$\Delta_\parallel$	$\Delta_\parallel f := \boldsymbol{\nabla} \cdot (\hat{\boldsymbol{b}}\hat{\boldsymbol{b}} \cdot \boldsymbol{\nabla} f)$

with b^i the contra- and b_i the co-variant components of $\hat{\boldsymbol{b}}$, and ε^{ijk} the Levi-Civita symbols. Explicit expressions for the above expressions depend on the choice of the magnetic field and the underlying coordinate system. Note that we have

$$\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla} \times \hat{\boldsymbol{b}}} = -\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla B}} = -\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla} \times \hat{\boldsymbol{b}}} \cdot \boldsymbol{\nabla} \ln B, \tag{1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{K}} = 0, \tag{2}$$

$$\boldsymbol{\mathcal{K}} = \boldsymbol{\nabla} \times \frac{\hat{\boldsymbol{b}}}{B} \cdot \boldsymbol{\nabla} f = \mathcal{K}_{\boldsymbol{\nabla} \times \hat{\boldsymbol{b}}}(f) + \mathcal{K}_{\boldsymbol{\nabla} B}(f), \tag{3}$$

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla} \times \hat{\boldsymbol{b}}} - \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla B}} = \frac{1}{B^2}(\boldsymbol{\nabla} \times \boldsymbol{B}), \tag{4}$$

$$\nabla_\parallel \ln B = -\boldsymbol{\nabla} \cdot \hat{\boldsymbol{b}}. \tag{5}$$

The last equality holds if $\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0$. Note that in any arbitrary coordinate system we have

$$(\boldsymbol{\nabla} f)^i = g^{ij}\partial_j f \ , \quad \boldsymbol{\nabla} \cdot \boldsymbol{v} = \frac{1}{\sqrt{g}}\partial_i \left(\sqrt{g} v^i\right) \ , \quad (\boldsymbol{v} \times \boldsymbol{w})^i = \frac{1}{\sqrt{g}}\varepsilon^{ijk} v_j w_k \ . \tag{6}$$

1.1 Coordinate system

We employ cylindrical coordinates (R, Z, φ), with φ anti directed to the geometric toroidal angle (**clockwise** if viewed from above) to obtain a right handed system. The parametric representation in Cartesian (x, y, z) coordinates is therefore simply:

$$x = R\ \sin(\varphi), \qquad y = R\ \cos(\varphi), \qquad z = Z. \tag{7}$$

Note here that the angle $\varphi = 0$ corresponds to the Cartesian y-axis. The unit basis vectors and (covariant) metric tensor are:

$$\hat{\boldsymbol{e}}_R = (\sin(\varphi), \cos(\varphi), 0)^T, \qquad \hat{\boldsymbol{e}}_Z = (0, 0, 1)^T, \qquad \hat{\boldsymbol{e}}_\varphi = (\cos(\varphi), -\sin(\varphi), 0)^T, \tag{8}$$

$$(g_{ij}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & R^2 \end{pmatrix} \tag{9}$$

With the help of the metric elements we get a well behaved volume element $\sqrt{g} = R$. However, we have a coordinate singularity at $R = 0$. The cylindrical coordinate basis vectors are mutually orthogonal to each other.

1.2 Solov'ev equilbrium

In cylindrical coordinates the general axisymmetric magnetic field can be written as (dimensionless)

$$\boldsymbol{B} = \frac{R_0}{R}\left[I(\psi_p)\hat{\boldsymbol{e}}_\varphi + \frac{\partial\psi_p}{\partial Z}\hat{\boldsymbol{e}}_R - \frac{\partial\psi_p}{\partial R}\hat{\boldsymbol{e}}_Z\right], \tag{10}$$

which can obviously not be manipulated to be in Clebsch form. Hence we are dealing with a non-flux aligned coordinate system. For the sake of clarity we define the poloidal magnetic field $\boldsymbol{B}_p = \frac{R_0}{R}\left(\frac{\partial\psi_p}{\partial Z}\hat{\boldsymbol{e}}_R - \frac{\partial\psi_p}{\partial R}\hat{\boldsymbol{e}}_Z\right)$ and the toroidal magnetic field $\boldsymbol{B}_t = \frac{R_0 I}{R}\hat{\boldsymbol{e}}_\varphi$. Note that with a typically convex function ψ_p (second derivative is positive), $I(\psi_p) > 0$ and the previously defined coordinate system the field line winding is a **left handed screw** in the positive $\hat{\boldsymbol{e}}_\varphi$-direction. Also note that then $\boldsymbol{B}\times\boldsymbol{\nabla}\boldsymbol{B}$ points **down**, towards the magnetic X-point, and we have the **favourable** drift direction (since in experiments H-mode is reached easier in this configuration).

We scaled R, Z and R_0 with $\rho_s = \sqrt{T_e m_i}/(eB_0)$, the magnetic field with B_0, the poloidal flux with $\psi_{p0} = B_0\rho_s\hat{R}_0$ and the poloidal equilibrium current streamfunction with $I_0 = B_0\hat{R}_0$ (with $\hat{R}_0 = \rho_s R_0$ the dimensional major radius). We have the equilibrium equations in toroidally symmetric, ideal MHD $\boldsymbol{\nabla}p = \boldsymbol{j}\times\boldsymbol{B}$ and $\boldsymbol{\nabla}\times\boldsymbol{B} = \beta\boldsymbol{j}$ normalized with $p_0 = n_0T_0$, and $j_0 = en_0c_S$, where we introduce $\beta = n_0T_0\mu_0/B_0^2$. Note that this normalization is in line with the one later chosen for the gyrofluid equations but is unnatural for the MHD type equilibrium equations through the introduction of ρ_s and β.

$$\boldsymbol{\nabla}\times\boldsymbol{B} = \frac{R_0}{R}\left[-\Delta^*\psi_p\hat{\boldsymbol{e}}_\varphi + I_Z\hat{\boldsymbol{e}}_R - I_R\hat{\boldsymbol{e}}_Z\right] \equiv \beta\boldsymbol{j} \tag{11}$$

$$\beta j_\parallel = \beta\boldsymbol{j}\cdot\hat{\boldsymbol{b}} = \beta\frac{\mathrm{d}p}{\mathrm{d}\psi_p}\frac{I(\psi_p)}{B} + \frac{\mathrm{d}I}{\mathrm{d}\psi_p}B \quad \text{Pfirsch-Schlüter \& Bootstrap current} \tag{12}$$

$$\beta\boldsymbol{j}_\perp = \beta\hat{\boldsymbol{b}}\times\left(\boldsymbol{j}\times\hat{\boldsymbol{b}}\right) = \beta\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}p}{B} \quad \text{diamagnetic current} \tag{13}$$

$$\beta\boldsymbol{j}\times\boldsymbol{B} = \frac{R_0^2}{R^2}\left[-\Delta^*\psi_p - I\frac{\mathrm{d}I}{\mathrm{d}\psi_p}\right]\boldsymbol{\nabla}\psi_p \equiv \beta\frac{\mathrm{d}p}{\mathrm{d}\psi_p}\boldsymbol{\nabla}\psi_p = \beta\boldsymbol{\nabla}p \tag{14}$$

from where we recover the Grad-Shafranov equation

$$-\Delta^*_\perp\psi_p = \beta\frac{R^2}{R_0^2}\frac{dp}{d\psi_p} + I\frac{dI}{d\psi_p} \equiv \beta\frac{R}{R_0}j_{\hat{\varphi}} \tag{15}$$

with $\Delta^*_\perp\psi_p = R\partial_R(R^{-1}\psi_R) + \psi_{ZZ}$. The Solov'ev assumptions consist of $A/R_0 = -I\frac{dI}{d\psi_p}$ and $(1-A)/R_0 = -\frac{dp}{d\psi_p}$, where A is a constant [2, 3]. By integration over ψ_p we find $p(\psi_p) = (A-1)\psi_p/R_0/\beta + p(0)$, $I(\psi_p) = \sqrt{-2A\psi_p/R_0+1}$, and $j_{\hat{\varphi}} = \left[(A-1)R^2/R_0^2 - A\right]/R/\beta$. Note that if ψ_p, $I(\psi)$ and $p(\psi)$ are a solution to Eq. (15) then so are $\mathcal{P}_\psi\psi_p$, $\mathcal{P}_\psi I(\psi_p)$ and $\mathcal{P}^2_\psi p(\psi_p)$. Also note that for $A = 0$ the constant current I becomes arbitrary $\mathcal{P}_I$.

We introduce $\bar{R} \equiv \frac{R}{R_0}$ and $\bar{Z} \equiv \frac{Z}{R_0}$ and thus represent a general solution to Equation (15)

as [2]

$$\psi_p(R,Z) = \mathcal{P}_\psi R_0 \left[A\left(\frac{1}{2}\bar{R}^2\ln\bar{R} - \frac{1}{8}\bar{R}^4\right) + \frac{1}{8}\bar{R}^4 + \sum_{i=1}^{12} c_i \bar{\psi}_{pi}\right], \tag{16a}$$

$$I(\psi_p) = \mathcal{P}_I\sqrt{-2A\frac{\psi_p}{R_0\mathcal{P}_\psi} + 1}, \tag{16b}$$

with $\mathcal{P}_\psi$ a free constant, $\mathcal{P}_I = \pm\mathcal{P}_\psi$ for $A \neq 0$ and $\mathcal{P}_I$ arbitrary for $A = 0$ (purely toroidal equilibrium current). We have

$$p(\psi_p) = \mathcal{P}_\psi\frac{(A-1)\psi_p}{\beta R_0} + p(0) \qquad j_{\hat{\varphi}} = \frac{\mathcal{P}_\psi}{\beta}\left[\frac{(A-1)R}{R_0^2} - \frac{A}{R}\right] \tag{17}$$

$\bar{\psi}_{p1} = 1$	$\bar{\psi}_{p7} = 8\bar{Z}^6 - 140\bar{R}^2\bar{Z}^4 + 75\bar{R}^4\bar{Z}^2 - 15\bar{R}^6\ln\bar{R} + 180\bar{R}^4\bar{Z}^2\ln\bar{R} - 120\bar{R}^2\bar{Z}^4\ln\bar{R}$
$\bar{\psi}_{p2} = \bar{R}^2$	$\bar{\psi}_{p8} = \bar{Z}$
$\bar{\psi}_{p3} = \bar{Z}^2 - \bar{R}^2\ln\bar{R}$	$\bar{\psi}_{p9} = \bar{Z}\bar{R}^2$
$\bar{\psi}_{p4} = \bar{R}^4 - 4\bar{R}^2\bar{Z}^2$	$\bar{\psi}_{p10} = \bar{Z}^3 - 3\bar{Z}\bar{R}^2\ln\bar{R}$
$\bar{\psi}_{p5} = 2\bar{Z}^4 - 9\bar{R}^2\bar{Z}^2 + 3\bar{R}^4\ln\bar{R} - 12\bar{R}^2\bar{Z}^2\ln\bar{R}$	$\bar{\psi}_{p11} = 3\bar{Z}\bar{R}^4 - 4\bar{Z}^3\bar{R}^2$
$\bar{\psi}_{p6} = \bar{R}^6 - 12\bar{R}^4\bar{Z}^2 + 8\bar{R}^2\bar{Z}^4$	$\bar{\psi}_{p12} = 8\bar{Z}^5 - 45\bar{Z}\bar{R}^4 - 80\bar{Z}^3\bar{R}^2\ln\bar{R} + 60\bar{Z}\bar{R}^4\ln\bar{R}$

As an alternative and in order to better fit experimental equilibria we offer a polynomial expansion of the magnetic flux function

$$\psi_p(R,Z) = \mathcal{P}_\psi R_0 \sum_{i=0}^{N_R-1}\sum_{j=0}^{N_Z-1} c_{ij}\bar{R}^i\bar{Z}^j \tag{18a}$$

$$I(\psi_p) = \mathcal{P}_I \tag{18b}$$

where the number of polynomial coefficients N_R and N_Z can be freely chosen. Since Eqs. (16) and (18) are given analytically we can numerically evaluate ψ_p and I and all their derivatives at arbitrary points to machine precision, which is simple to implement and fast to execute. This translates to an exact representation of the magnetic field and related quantities like the curvature operators in code. In particular, the X-point and O-point can be determined to machine precision via a few Newton iterations.

The choice of the coefficients c_i and A, respectively c_{ij} determines the actual form of the magnetic field. We can for example represent single and asymmetric double X-point configurations, force-free states, field reversed configurations and low and high beta tokamak equilibria. R_0 appears as an artificial scaling factor (note here that a change in ρ_s changes R_0 but not the form or size of the dimensional equilibrium magnetic field). The scaling factors $\mathcal{P}_\psi$ and $\mathcal{P}_I$ are

mainly introduced to maximize the flexibility e.g. to adapt the solution to experimental equilibria or to reverse the sign of the magnetic field. If an X-point is present, we choose c_1 such that $\psi_p(R_X, Z_X) = 0$ that is the separatrix is given by $\psi_p(R, Z) = 0$.

Note that

$$B^R = B_R = R_0\psi_Z/R \tag{19}$$
$$B^Z = B_Z = -R_0\psi_R/R \tag{20}$$
$$B^\varphi = B_\varphi/R^2 = R_0 I/R^2 \tag{21}$$

(contra- and covariant components of $\boldsymbol{B}$). By construction we have $\partial_\varphi B = 0$ with

$$B = \frac{R_0}{R}\sqrt{I^2 + |\boldsymbol{\nabla}\psi_p|^2}. \tag{22}$$

Furthermore, we have

$$\nabla_\parallel f(R,Z) = \frac{R_0}{RB}[f,\psi_p]_{RZ} \Rightarrow \nabla_\parallel \ln B = \frac{R_0}{RB^2}\left[B,\psi_p\right]_{RZ} = -\boldsymbol{\nabla}\cdot\hat{\boldsymbol{b}}. \tag{23}$$

We allow various simplifications to the curvature operator for the Solov'ev equilibrium.

1.2.1 Toroidal (and negative toroidal) field line approximation

The toroidal/negative toroidal field line approximation applies $\hat{\boldsymbol{b}} \approx \pm\hat{\boldsymbol{e}}_\varphi$ to all perpendicular operators (e.g.: Poisson bracket, perpendicular elliptic operator and curvature operators) but retains the full expression for the magnetic field unit vector $\hat{\boldsymbol{b}}$ for parallel operators ($\nabla_\parallel$ and $\Delta_\parallel$). (Note that we allow the negative sign $-\hat{\boldsymbol{e}}_\varphi$ to enable a sign reversal of the magnetic field, see Section 2.6.1). In cylindrical coordinates that is

$$[f,g]_\perp \equiv [f,g]_{RZ} = \pm\frac{1}{R}\left(\partial_R f\partial_Z g - \partial_Z f\partial_R g\right) \tag{24}$$
$$\boldsymbol{\nabla}_\perp f = \partial_R f\hat{\boldsymbol{e}}_R + \partial_Z f\hat{\boldsymbol{e}}_Z \tag{25}$$
$$\Delta_\perp f = \frac{1}{R}\partial_R\left(R\partial_R f\right) + \partial_Z(\partial_Z f) \tag{26}$$

The curl of $\hat{\boldsymbol{b}}$ reduces to $\boldsymbol{\nabla}\times\hat{\boldsymbol{b}} \approx -\frac{\pm 1}{R}\hat{\boldsymbol{e}}_Z$. This simplifies the curvature operators to:

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} \approx -\frac{\pm 1}{BR}\hat{\boldsymbol{e}}_Z, \qquad \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}B} \approx -\frac{\pm 1}{B^2}\frac{\partial B}{\partial Z}\hat{\boldsymbol{e}}_R + \frac{\pm 1}{B^2}\frac{\partial B}{\partial R}\hat{\boldsymbol{e}}_Z \qquad \boldsymbol{\mathcal{K}} \approx \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}B} + \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}, \tag{27}$$

and

$$\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} \approx \frac{\pm 1}{RB^2}\frac{\partial B}{\partial Z}, \tag{28}$$

which results in a vanishing divergence of the curvature operators $\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}} = 0$.

Note that in an actual toroidal field we have

$$\boldsymbol{B}(R) := \pm\frac{R_0}{R}\hat{\boldsymbol{e}}_\varphi \tag{29}$$

We then have $\hat{\boldsymbol{b}} = \pm\hat{\boldsymbol{e}}_\varphi$ and the curvature operators further simplify to

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\boldsymbol{B}} = -\frac{\pm 1}{R_0}\hat{\boldsymbol{e}}_Z = \boldsymbol{\mathcal{K}}/2 \tag{30}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = \nabla_\parallel \ln B = 0 \tag{31}$$

Note: the negative sign is automatically chosen in code if $I(R_0, 0) < 0$.

1.2.2 Low beta approximation

In this approximation we apply the toroidal field line approximation as in Section 1.2.1 but approximate the curvature operator $\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} \approx \hat{\boldsymbol{b}}\times\boldsymbol{\kappa}$ with $\boldsymbol{\kappa} := \hat{\boldsymbol{b}}\cdot\boldsymbol{\nabla}\hat{\boldsymbol{b}} = -\hat{\boldsymbol{b}}\times(\boldsymbol{\nabla}\times\hat{\boldsymbol{b}})$. For an isotropic pressure plasma $\boldsymbol{P} = \boldsymbol{I}P_\perp + \boldsymbol{bb}P_\Delta \approx \boldsymbol{I}P_\perp$ and with the definition of the plasma beta parameter $\beta = \frac{P}{B^2/(2\mu_0)}$ we can rewrite the curvature to

$$\boldsymbol{\kappa} \approx \frac{\beta}{2}\boldsymbol{\nabla}\ln(P) + \boldsymbol{\nabla}_\perp \ln B. \tag{32}$$

In low beta plasmas $\beta \ll 1$ the curvature reduces to:

$$\boldsymbol{\kappa} \approx \boldsymbol{\nabla}_\perp \ln B. \tag{33}$$

This simplifies the curvature operators to:

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(f) \approx \boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}B} \approx -\frac{1}{B^2}\frac{\partial B}{\partial Z}\hat{\boldsymbol{e}}_R + \frac{1}{B^2}\frac{\partial B}{\partial R}\hat{\boldsymbol{e}}_Z \quad \mathcal{K}(f) \approx 2\mathcal{K}_{\boldsymbol{\nabla}B}(f), \quad \boldsymbol{\nabla}\times\hat{\boldsymbol{b}}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}B} = 0. \tag{34}$$

The divergence over the curvature vanishes $\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}} = 0$ only if $\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}B} = 0$. In general, the divergence $\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}} \approx 0$ is only approximately vanishing.

1.2.3 True perpendicular terms

Without any approximations we have

$$b^R = \frac{\partial\psi}{\partial Z}\left(I^2 + |\boldsymbol{\nabla}\psi|^2\right)^{-1/2} \quad b^Z = -\frac{\partial\psi}{\partial R}\left(I^2 + |\boldsymbol{\nabla}\psi|^2\right)^{-1/2} \quad b^\varphi = \frac{I}{R}\left(I^2 + |\boldsymbol{\nabla}\psi|^2\right)^{-1/2} \tag{35}$$

$$\boldsymbol{\nabla}\cdot\hat{\boldsymbol{b}} = -\nabla_\parallel \ln B = -\frac{R_0}{RB^2}[B, \psi_p]_{RZ} \tag{36}$$

$$\left(\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}\right)\cdot\hat{\boldsymbol{b}} = (I'(\boldsymbol{\nabla}\psi_p)^2 - I\Delta^*_\perp\psi_p)\frac{R_0^2}{R^2B^2} \propto 1/R_0 \tag{37}$$

where for the last estimate we inserted the Grad-Shafranov equation and the Solov'ev assumptions. We can then insert $\hat{\boldsymbol{b}}$ into the exact definitions for $[.,.]_\perp$, $\boldsymbol{\nabla}_\perp$ and $\Delta_\perp$ from Section 1.

For the curvature terms we can explicitly write

$$K^R_{\boldsymbol{\nabla}B} = -\frac{R_0 I}{B^3 R}\frac{\partial B}{\partial Z} \equiv -\frac{1}{B^2}\frac{\partial B}{\partial Z}b^\varphi \tag{38}$$

$$K^Z_{\boldsymbol{\nabla}B} = \frac{R_0 I}{B^3 R}\frac{\partial B}{\partial R} \equiv \frac{1}{B^2}\frac{\partial B}{\partial R}b^\varphi \tag{39}$$

$$K^\varphi_{\boldsymbol{\nabla}B} = \frac{R_0}{B^3R^2}\left(\frac{\partial\psi}{\partial Z}\frac{\partial B}{\partial Z} + \frac{\partial\psi}{\partial R}\frac{\partial B}{\partial R}\right) \tag{40}$$

and

$$K^R_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = \frac{R_0}{RB^3}\left(B\frac{\partial I}{\partial Z} - I\frac{\partial B}{\partial Z}\right) \tag{41}$$

$$K^Z_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = \frac{R_0}{RB^3}\left(I\frac{\partial B}{\partial R} - B\frac{\partial I}{\partial R}\right) \tag{42}$$

$$K^\varphi_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = \frac{R_0}{R^2B^2}\left(+\frac{1}{B}\frac{\partial\psi}{\partial Z}\frac{\partial B}{\partial Z} + \frac{1}{B}\frac{\partial\psi}{\partial R}\frac{\partial B}{\partial R} - R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\psi}{\partial R}\right) - \frac{\partial^2\psi}{\partial Z^2}\right) \tag{43}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} = -\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\boldsymbol{B}} = -\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\cdot\boldsymbol{\nabla}\ln B = \frac{R_0}{RB^3}[I,B]_{RZ} \tag{44}$$

1.3 Flux surface averaging and safety factor

1.3.1 Preliminary

Recall that the **Dirac delta-function** has the property (in any dimension):

$$\int_V f(\boldsymbol{x})\delta(h(\boldsymbol{x})-h')\mathrm{dV} = \int_{h=h'}\frac{f(\boldsymbol{x})}{|\boldsymbol{\nabla}h|}\,\mathrm{dA} \tag{45}$$

which means that the delta-function can be used to express area integrals of the submanifold given as a contour of the function $h(\boldsymbol{x})$. A numerically tractable approximation to the delta-function reads

$$\delta(h(\boldsymbol{x})-h') = \frac{1}{2\pi\epsilon^2}\exp\left(-\frac{(h(\boldsymbol{x})-h')^2}{2\epsilon^2}\right) \tag{46}$$

where ϵ is a small, free parameter. In the DG framework the left-hand side of Eq. (45) can thus readily be computed via Gauss-Legendre quadrature, which we propse as a first method to compute area integrals even if our coordinate system is not aligned to the area. Note: in order for this to work the Delta function needs to be numerically resolved and cannot be made arbitrarily small. This introduces a smoothing effect over neighboring contour lines which is given by the grid distance.

Furthermore, recall the **co-area formula**

$$\int_{\Omega_0} f(\boldsymbol{x})\mathrm{dV} = \int_0^{h_0}\left(\int_{h=h'}\frac{f(\boldsymbol{x})}{|\boldsymbol{\nabla}h|}\,\mathrm{dA}\right)\mathrm{d}h' \tag{47}$$

where Ω_0 is the volume enclosed by the contour $h = h_0$. The co-area formula can be viewed as a change of variables in the volume integral.

We define the **toroidal average** of a function $f(R,Z,\varphi)$ as

$$\langle f\rangle_\varphi(R,Z) := \frac{1}{2\pi}\oint f(R,Z,\varphi)\mathrm{d}\varphi \tag{48}$$

In arbitrary coordinates the area integral is defined by the pull back of the flux 2-form and the metric

$$\mathrm{dA}^2 = i_{\hat{\psi}_p}vol^3 \quad \hat{\psi}_p = \frac{\boldsymbol{\nabla}\psi_p}{|\boldsymbol{\nabla}\psi_p|} \tag{49}$$

to a parameterization of the flux-surface. In a flux-aligned coordinate system $\{\zeta, \eta, \varphi\}$ the pull-back is trivial ($\zeta = const$) and we have

$$\mathrm{dA} = \sqrt{g^{\zeta\zeta}}\sqrt{g}\mathrm{d}\eta\mathrm{d}\varphi = f_0|\boldsymbol{\nabla}\psi_p|\sqrt{g}\mathrm{d}\eta\mathrm{d}\varphi, \tag{50}$$

$$\mathbf{dA} := \hat{\psi}_p\,\mathrm{dA} = f_0(\boldsymbol{\nabla}\psi_p)\sqrt{g}\mathrm{d}\eta\mathrm{d}\varphi, \tag{51}$$

where we used that $g^{\zeta\zeta} = (\boldsymbol{\nabla}\zeta)^2 = f_0^2(\boldsymbol{\nabla}\psi_p)^2$. Notice that numerically we can integrate in flux-aligned coordinates by generating a corresponding grid and pulling back (interpolating) the relevant fields to this grid. This is the second method to numerically compute area integrals.

1.3.2 Flux surface average

The flux surface average (as a **volume average** after [4]) is defined as an average over a small volume - a shell centered around the flux-surface - defined by two neighboring flux-surfaces. With the help of the volume flux label (notice that both the volume v as well as the poloidal flux ψ_p have physical meaning while the coordinate $\zeta(\psi_p)$ is an arbitrary choice) we define

$$v(\psi_p) := \int_{\psi_{p,O}}^{\psi} \mathrm{dV} = \int^{\zeta(\psi_p)} \sqrt{g}\mathrm{d}\zeta\mathrm{d}\eta\mathrm{d}\varphi, \tag{52}$$

$$\frac{\mathrm{d}v}{\mathrm{d}\psi_p} = \int \mathrm{dA}|\boldsymbol{\nabla}\psi_p|^{-1} = 2\pi f_0 \oint_{\zeta(\psi_p)} \sqrt{g}\mathrm{d}\eta \tag{53}$$

$$\begin{aligned}\langle f\rangle_\psi :=& \frac{\partial}{\partial v}\int \mathrm{dV}\, f = \frac{1}{\int \mathrm{dA}|\boldsymbol{\nabla}\psi_p|^{-1}}\int_{\psi_p}\frac{f(\boldsymbol{x})}{|\boldsymbol{\nabla}\psi_p|}\,\mathrm{dA}\\ =& \frac{\int_\Omega \langle f\rangle_\varphi(R,Z)\delta(\psi_p(R,Z)-\psi_p)H(Z-Z_X)\,R\mathrm{d}R\mathrm{d}Z}{\int_\Omega \delta(\psi_p(R,Z)-\psi_p)H(Z-Z_X)\,R\mathrm{d}R\mathrm{d}Z}\\ =& \left(\frac{\mathrm{d}v}{\mathrm{d}\psi_p}\right)^{-1} 2\pi f_0 \oint_0^{2\pi}\langle f\rangle_\varphi(\zeta,\eta)\sqrt{g}\mathrm{d}\eta = \frac{1}{\oint\sqrt{g}\mathrm{d}\eta}\oint_0^{2\pi}\langle f\rangle_\varphi(\zeta,\eta)\sqrt{g}\mathrm{d}\eta\end{aligned} \tag{54}$$

where we used the co-area formula Eq. (47) for the second identity and we use the Heaviside function $H(Z - Z_X)$ to cut away contributions from below the X-point in our domain Ω. We immediately see that this definition is particularly easy to compute in a flux-aligned coordinate system. Notice however that the volume element does appear (unlike e.g. Tokam3X papers). We use our grid construction algorithm with constant monitor metric described in Reference [10] to construct a flux-aligned grid and interpolate the values of any function onto its grid points. Even though this grid is unusable for simulations due to the diverging metric at the X-point the evaluation of integrals works well as the singularity is integrable.

The flux-surface average fulfills the basic identities

$$\langle \mu f + \lambda g\rangle = \mu\langle f\rangle + \lambda\langle g\rangle \tag{55}$$

$$\langle f(\psi_p)\rangle = f(\psi_p) \tag{56}$$

The volume average is well-suited for density-like quantities as we can see with the following identity. Assume we have a quantity X with $\partial_t X + \boldsymbol{\nabla}\cdot\boldsymbol{j}_X = \Lambda_X$. Then we can use the volume

average to write

$$\frac{\partial}{\partial t}\langle X\rangle + \frac{\partial}{\partial v}\langle \boldsymbol{j}_X \cdot \boldsymbol{\nabla} v\rangle = \langle \Lambda_X\rangle \tag{57}$$

where again $v = v(\psi_p)$ is the volume flux label. The **total flux** of a given flux density $\boldsymbol{j}_X$ through the flux surface $\psi_p = \psi_{p0}$ is given by

$$\langle \boldsymbol{j}_X \cdot \boldsymbol{\nabla} v\rangle := J_X = \oint_{\psi_p=\psi_{p0}} \boldsymbol{j}_X \cdot \mathbf{dA} = \frac{\mathrm{d}v}{\mathrm{d}\psi_p}\langle \boldsymbol{j}_X \cdot \boldsymbol{\nabla}\psi_p\rangle \tag{58}$$

$$= 2\pi f_0 \oint_0^{2\pi} \langle \boldsymbol{j}_X \cdot \boldsymbol{\nabla}\psi_p\rangle_\varphi(\zeta,\eta)\sqrt{g}\mathrm{d}\eta \tag{59}$$

Once we have the flux-surface averaged equation we can easily get the volume integrated version (again with the help of the co-area formula)

$$\frac{\partial}{\partial t}\int_0^{v(\psi_p)} \langle X\rangle\,\mathrm{d}v + \langle \boldsymbol{j}_X \cdot \boldsymbol{\nabla} v\rangle(v(\psi_p)) = \int_0^{v(\psi_p)} \langle \Lambda_X\rangle\,\mathrm{d}v \tag{60}$$

1.3.3 The safety factor

Assume that we pick a random field line and follow it (integrate it) for exactly one poloidal turn. The **safety factor** is defined as the ratio between the resulting toroidal angle ($\Delta\varphi$) to the poloidal angle (2π)

$$q := \frac{\Delta\varphi}{2\pi} \tag{61}$$

Since our magnetic field is symmetric in φ and we used one full poloidal turn this definition is independent of which fieldline we pick on a given flux surface.

We define the geometric poloidal angle Θ as the fieldline following parameter i.e. $\boldsymbol{B}\cdot\boldsymbol{\nabla}\Theta = R_0(\psi_R(R-R_0)+\psi_Z Z)/r^2 R$. We can then directly integrate the safety factor as

$$\frac{\mathrm{d}R}{\mathrm{d}\Theta} = \frac{B^R}{B^\Theta} \quad \frac{\mathrm{d}Z}{\mathrm{d}\Theta} = \frac{B^Z}{B^\Theta} \quad \frac{\mathrm{d}\varphi}{\mathrm{d}\Theta} = \frac{B^\varphi}{B^\Theta} \tag{62}$$

$$q \equiv \frac{1}{2\pi}\oint \frac{B^\varphi}{B^\Theta}\mathrm{d}\Theta \tag{63}$$

We integrate this equation with the help of one of our ODE integrators, i.e. we use a high-order Runge-Kutta method and refine the stepsize until machine-precision is reached. Notice that the safety factor diverges on the last closed flux surface whereas Eq. (59) remains finite due to the $\boldsymbol{\nabla}\psi_p$ factor.

1.3.4 Toroidal averages

Here, we comment on the φ average that is part of the flux-surface average Eq. (52). One simple approach is quadrature of the form

$$\bar{f} = \frac{1}{N}\sum_{i=0}^{N-1} f_i(R,Z) \tag{64}$$

where $N = 32$ in most of our simulations and f_i is the i-th toroidal plane. Since the boundary conditions in φ are periodic this amounts to the trapezoidal rule. A low number of toroidal planes is sufficient in simulations when we use the toroidal field approximation in combination with the flux-coordinate independent (FCI) approach for the parallel derivatives. However, since the actual φ direction is under-resolved the integration gives a wrong answer to the actual φ average (seen in 2d plots as little humps). This is because the resulting structures are predominantly field aligned and not toroidally symmetric.

In order to improve the toroidal average we now have the following idea: if we, before we do the φ integration, interpolate the function to integrate onto a large number of toroidal planes then the result should be more accurate than before. In other words we interpolate the function given on the coarse φ simulation grid onto a hypothetic fine φ grid along the magnetic field lines and only then compute the φ average.

Let us divide the φ direction between two original planes into $N_\varphi+1$ (a large number) equidistant planes of distance $\delta\varphi$ and integrate the magnetic field $\boldsymbol{B}$ in between.

$$\frac{\mathrm{d}R}{\mathrm{d}\varphi} = \frac{B^R}{B^\varphi}, \tag{65a}$$

$$\frac{\mathrm{d}Z}{\mathrm{d}\varphi} = \frac{B^Z}{B^\varphi}, \tag{65b}$$

$$\tag{65c}$$

We integrate Eqs. (65) from $\varphi = 0$ to $\varphi = \pm\Delta\varphi$ with initial condition

$$(R(0), Z(0)) = (R, Z). \tag{66}$$

Let us characterize the solution $(R(\pm\delta\varphi), Z(\pm\delta\varphi))$ to Eqs. (65) as the flow generated by $\boldsymbol{B}/B^\varphi$

$$\mathcal{T}^\pm_{\delta\varphi}\boldsymbol{z} \equiv \mathcal{T}^\pm_{\delta\varphi}[R, Z, \varphi] := (R(\pm\delta\varphi), Z(\pm\delta\varphi), \varphi \pm \delta\varphi), \tag{67}$$

Obviously we have $\mathcal{T}^-_{\delta\varphi} \circ \mathcal{T}^+_{\delta\varphi} = 1$, but $\mathcal{T}^\pm_{\delta\varphi}$ is not necessarily unitary since $\boldsymbol{B}/B^\varphi$ is in general not divergence free. We are now able to extend the function f given on the coarse φ grid unto the fine φ grid via

$$f(R, Z, \varphi_0 + j\delta\varphi) = \mathcal{T}^{-j}_{\delta\varphi} f(R, Z, \varphi_0) \tag{68}$$

$$f(R, Z, \varphi_0 - j\delta\varphi) = \mathcal{T}^{+j}_{\delta\varphi} f(R, Z, \varphi_0) \tag{69}$$

This gives simple 0-th order extrapolation of our function. Let us call $f_i := f(R, Z, \varphi_i)$ the i-th toroidal plane and N_φ even. Then we have the following integration, where we consider the

original toroidal planes as cell-centered

$$\begin{aligned}\langle f\rangle_\varphi &= \frac{1}{(N_\varphi+1)N}\left[\left(\mathcal{T}_{\delta\varphi}^{+\,N_\varphi/2} f_0 + ... + \mathcal{T}_{\delta\varphi}^{+} f_0 + f_0 + \mathcal{T}_{\delta\varphi}^{-} f_0 ... + \mathcal{T}_{\delta\varphi}^{-\,N_\varphi/2} f_0\right)\right.\\ &\left.+\left(\mathcal{T}_{\delta\varphi}^{+\,N_\varphi/2} f_1 + ... + \mathcal{T}_{\delta\varphi}^{+} f_1 + f_1 + \mathcal{T}_{\delta\varphi}^{-} f_1 ... + \mathcal{T}_{\delta\varphi}^{-\,N_\varphi/2} f_1\right) + ...\right]\\ &= \frac{1}{N(N_\varphi+1)}\sum_{i=0}^{N-1}\left[f_i + \sum_{j=1}^{N_\varphi/2}\left(\mathcal{T}_{\delta\varphi}^{-\,j} f_i + \mathcal{T}_{\delta\varphi}^{+\,j} f_i\right)\right]\\ &= \frac{1}{N_\varphi+1}\left[\sum_{j=0}^{N_\varphi/2}\mathcal{T}_{\delta\varphi}^{-\,j}\left(\frac{1}{N}\sum_{i=0}^{N} f_i\right) + \sum_{j=1}^{N_\varphi/2}\mathcal{T}_{\delta\varphi}^{+\,j}\left(\frac{1}{N}\sum_{i=0}^{N} f_i\right)\right]\\ &= \frac{1}{N_\varphi+1}\left[\sum_{j=0}^{N_\varphi/2}\mathcal{T}_{\delta\varphi}^{-\,j}\bar{f}(R,Z) + \sum_{j=1}^{N_\varphi/2}\mathcal{T}_{\delta\varphi}^{+\,j}\bar{f}(R,Z)\right]\end{aligned} \tag{70}$$

Here, we used that the push-forward operator $\mathcal{T}_{\delta\varphi}^{-}$ is linear that is $\mathcal{T}_{\delta\varphi}^{-} f_0 + \mathcal{T}_{\delta\varphi}^{-} f_1 = \mathcal{T}_{\delta\varphi}^{-}(f_0+f_1)$ and recover the simple toroidal summation $\bar{f}$ Eq. (64). Now, we can see that in the limit $N_\varphi \to \infty$ the discrete sum represents the integral of the form

$$\langle f\rangle_\varphi(R,Z) = \frac{1}{\Delta\varphi}\int_{-\Delta\varphi/2}^{\Delta\varphi/2} \mathrm{d}\varphi \bar{f}(R(\varphi),Z(\varphi)) \tag{71}$$

A consistency test of this approach is to simply use $\boldsymbol{B} = e_\varphi$. Then $\mathcal{T}_{\delta\varphi}^{\pm} = 1$ and we recover the original integration $\langle f\rangle_\varphi = \bar{f}$. Now, instead of doing a 0-th order interpolation let us try a linear interpolation along field-lines in between planes that is (assuming N_φ toroidal planes)

$$f(R,Z,\varphi_i + j\delta\varphi) = \left(1 - \frac{j}{N_\varphi}\right)\mathcal{T}_{\delta\varphi}^{-\,j} f_i + \frac{j}{N_\varphi}\mathcal{T}_{\delta\varphi}^{+\,N_\varphi - j} f_{i+1} \tag{72}$$

$$\begin{aligned}\langle f\rangle_\varphi &= \frac{1}{N_\varphi N}((f_0 + (1-\alpha_1)\mathcal{T}_{\delta\varphi}^{-} f_0 + \alpha_1(\mathcal{T}_{\delta\varphi}^{+})^{N_\varphi-1} f_1 + (1-\alpha_2)(\mathcal{T}_{\delta\varphi}^{-})^2 f_0 + \alpha_2(\mathcal{T}_{\delta\varphi}^{+})^{N_\varphi-2} f_1 ...) + (f_1 + ...) + ...)\\ &= \frac{1}{N_\varphi}\sum_{j=0}^{N_\varphi-1}(1-\alpha_j)(\mathcal{T}_{\delta\varphi}^{-})^j \bar{f} + \alpha_j(\mathcal{T}_{\delta\varphi}^{+})^{N_\varphi-j}\bar{f} = \frac{1}{\Delta\varphi}\int_{-\Delta\varphi}^{\Delta\varphi}\mathrm{d}\varphi w(\varphi)\bar{f}(R(\varphi),Z(\varphi))\end{aligned} \tag{73}$$

with $\alpha_j = \frac{j}{N_\varphi}$ and $w(\varphi)$ a linear weight function (pyramid shape) with $\int_{-\Delta\varphi}^{\Delta\varphi} w(\varphi) = \Delta\varphi$. Taking $\mathcal{T}_{\delta\varphi}^{\pm} = 1$ again leads to the old result.

Now, an interesting question is, what happens if we are trying to apply the above results to a function that is not field-aligned like $B(R,Z)$ of $\boldsymbol{K}\cdot\boldsymbol{\nabla}\psi_p$ for instance? For those functions $\bar{f}_{\mathrm{old}}$ actually yields the exact result, while the convolution is an approximation. Here, we have to test. Typically, the functions that we use are slowly varying in R and Z and so the convolution should not change the result too much. A good test candidate is still $\langle\mathcal{K}(\psi_p)\rangle_{\psi_p} = 0$.

In all practical tests so far the flux-suface average is not or only very slightly changed by this procedure. This means that it is not necessary to follow the smoothing procedure if one is only interested in the flux-surface average. This makes sense because the toroidal and poloidal averages commute.

1.4 Alternative flux labels

We find the toroidal flux ψ_t by integrating the q-profile $\psi_t = \int^{\psi_p} \mathrm{d}\psi_p q(\psi_p)$. Since q diverges, ψ_t, in contrast to ψ_p, is not defined outside the last closed flux-surface (but has a finite value on the last closed flux surface). We now define the normalized poloidal and toroidal flux labels ρ_p and ρ_t

$$\rho_p := \sqrt{1 - \frac{\psi_p}{\psi_{p,O}}} \;\leftrightarrow\; \psi_p = (1-\rho_p^2)\psi_{p,O} \tag{74}$$

$$\rho_t := \sqrt{\frac{\psi_t}{\psi_{t,\mathrm{sep}}}}, \tag{75}$$

$$\text{with } \psi_{p,O} = \psi_p(R_O, Z_O) \tag{76}$$

where R_O, Z_O are the coordinates of the O-point. The labels ρ_t and ρ_p are useful because equidistant ρ_p and ρ_t values tend to translate to equidistant flux-surfaces in configuration space.

1.5 Modified ψ_p

Our computational domain is a box and in particular not aligned with the magnetic flux surfaces. This means that particularly in the corners of the domain the field lines inside the domain are very short (in the sense that the distance between the entry point and leave point is short). It turns out that this behaviour is numerically disadvantageous (may blow up the simulation in the worst case) in the computation of parallel derivatives. In order to remedy this situation we propose to modify the flux surfaces ψ_p to a constant value if ψ_p exceeds a certain critical value. In this way the poloidal field component vanishes in the corners of the domain at the cost of introducing a strong shear layer limiting the scrape-off layer width.

We define an approximation to the step function with a transition layer of radius a around the origin

$$\Theta_a(x) := \begin{cases} 0 & \text{for } x \le -a \\ \frac{1}{32a^7}\left(16a^3 - 29a^2x + 20ax^2 - 5x^3\right)(a+x)^4 & \text{for } -a < x \le a \approx H(x) \\ 1 & \text{for } x > a \end{cases} \tag{77}$$

where $H(x)$ is the Heaviside step function. An integral of this function is

$$\theta_a(x) := \begin{cases} 0 & \text{for } x \le -a \\ \frac{1}{256a^7}\left(35a^3 - 47a^2x + 25ax^2 - 5x^3\right)(a+x)^5 & \text{for } -a < x \le a \approx xH(x) \\ x & \text{for } x > a \end{cases} \tag{78}$$

Note that $\Theta_a(0) = 0.5$ and $\theta_a(0) = 35a/256$.

We now use

$$-\theta_{\alpha/2}\left(\psi_{p,b} + \frac{\alpha}{2} - \psi\right) + \psi_{p,b} + \frac{\alpha}{2} \approx (\psi - \psi_{p,b})H(\psi_{p,b} - \psi) + \psi_{p,b} \tag{79}$$

instead of ψ_p for the computation of the magnetic field, which introduces a shear layer between $\psi_{p,b}$ and $\psi_{p,b}+\alpha$ where the fieldlines are straightened to match $\hat{\boldsymbol{e}}_\varphi$. In order to simplify the setup of this region we give $\psi_{p,b}$ and α in terms of ρ_p and α_p via $\psi_{p,b} = (1-\rho_{p,b}^2)\psi_{p,O}$ and $\alpha = -(2\rho_{p,b}\alpha_p + \alpha_p^2)\psi_{p,O}$. In case we change the sign of ψ_p via $\mathcal{P}_\psi$ (to make it concave) note that α becomes negative and $\psi_{p,O}$ is positive). We then need to point mirror Eq. (79) at $\psi_{p,b}+\frac{\alpha}{2}$.

2 The model

2.1 Conservative form

We scale all spatial lengths by $\rho_s = \sqrt{T_e m_i}/(eB_0)$ and time by the ion gyro-frequency $\Omega_0 = eB_0/m_i$. The magnetic field is scaled with B_0, densities with n_0 and the parallel velocity is scaled with $c_s = \sqrt{T_e/m_i}$. The potential is scaled with $\hat{\phi} = e/T_e$ and the vector potential with $\hat{A}_\parallel = \rho_s B_0$. We introduce the dimensionless parameters

$$\tau_a = \frac{T_a}{z_a T_e}\ , \quad \mu_a = \frac{m_a}{z_a m_i} \text{ and } \beta := \frac{\mu_0 n_0 T_e}{B_0^2} \tag{80}$$

where $a \in \{e,i\}$ is the species label and z is the charge number. Omitting the species label we arrive at (dividing the density equation by $\Omega_0 n_0$ and the velocity equation by $\Omega_0 c_s$)

$$\frac{\partial}{\partial t}N + \boldsymbol{\nabla}\cdot\left(N\left(\boldsymbol{u}_E + \boldsymbol{u}_K + \boldsymbol{u}_C + U_\parallel\left(\hat{\boldsymbol{b}}+\boldsymbol{b}_\perp\right)\right)\right) = \Lambda_N + S_N \tag{81}$$

$$\begin{aligned}
\mu\frac{\partial}{\partial t}\left(NU_\parallel\right) &+ \mu\boldsymbol{\nabla}\cdot\left(NU_\parallel\left(\boldsymbol{u}_E + \boldsymbol{u}_K + \boldsymbol{u}_C + U_\parallel\left(\hat{\boldsymbol{b}}+\boldsymbol{b}_\perp\right)\right)\right)\\
&+ 2\mu\boldsymbol{\nabla}\cdot\left(NU_\parallel \boldsymbol{u}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\right) - \mu N U_\parallel \boldsymbol{\nabla}\cdot\boldsymbol{u}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} + \mu N U_\parallel \mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi)\\
=& -\tau\left(\hat{\boldsymbol{b}}+\boldsymbol{b}_\perp\right)\cdot\boldsymbol{\nabla}N - N\left(\left(\hat{\boldsymbol{b}}+\boldsymbol{b}_\perp\right)\cdot\boldsymbol{\nabla}\psi + \frac{\partial A_\parallel}{\partial t}\right) - \eta n_e^2(U_{\parallel,i} - u_{\parallel,e}) + \mu\nu_\parallel\Delta_\parallel U\\
&+ \mu N\left(\Lambda_U + S_U\right) + \mu U_\parallel\left(\Lambda_N + S_N\right)
\end{aligned} \tag{82}$$

with

$$\begin{aligned}
\boldsymbol{u}_E &:= \frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\psi}{B}, \quad \boldsymbol{u}_K := \tau\left(\mathcal{K}_{\boldsymbol{\nabla}\boldsymbol{B}} + \mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\right) = \tau\boldsymbol{\mathcal{K}},\\
\boldsymbol{u}_C &:= \mu U_\parallel^2 \mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}, \quad \boldsymbol{u}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} := \tau\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}, \quad \boldsymbol{b}_\perp = \frac{\boldsymbol{\nabla}\times A_\parallel\hat{\boldsymbol{b}}}{B}.
\end{aligned} \tag{83}$$

The electric potential ϕ and parallel magnetic vector potential $A_\parallel$ are computed by the polarisation and induction equations (with $q_e = -e$ and $q_i = +e$)

$$-\boldsymbol{\nabla}\cdot\left(\frac{\mu_i N_i}{B^2}\boldsymbol{\nabla}_\perp\phi\right) = \Gamma_{1,i}N_i - n_e, \quad \Gamma_{1,i}^{-1} := 1 - \frac{1}{2}\mu_i\tau_i\Delta_\perp, \tag{84}$$

$$-\frac{1}{\beta}\Delta_\perp A_\parallel = \left(N_i U_{\parallel,i} - n_e u_{\parallel,e}\right) \tag{85}$$

Given ϕ we define the generalised electric potential

$$\psi_e := \phi, \quad \psi_i := \Gamma_{1,i}\phi - \frac{\mu_i}{2}\left(\frac{\boldsymbol{\nabla}_\perp \phi}{B}\right)^2 \tag{86}$$

In total we have an isothermal 3d gyro-fluid model with up to 2nd order FLR effects on in the electric potential ϕ and 0th order FLR effects in the parallel magnetic potential $A_\parallel$. We have the continuity equation for the electron density n_e and the ion gyro-centre density N_i and the momentum conservation equation for the parallel electron velocity $u_{\parallel,e}$ and the parallel ion gyro-centre velocity $U_{\parallel,i}$ [5, 9].

2.2 Diffusive terms

Since the gyro-fluid derivation does not include collisional terms we copied the parallel resistive and viscous terms from the Braginskii fluid equations [1]. The electron-ion and ion-ion collision frequencies are given by $\nu_{ei} = \sqrt{2}z^2e^4\ln\Lambda/(12\pi^{3/2}\sqrt{m_e}\epsilon_0^2)n_e/T_e^{3/2}$, $\nu_{ee} = \nu_{ei}/\sqrt{2}$ and $\nu_{ii} = z^4e^4\ln\Lambda/(12\pi^{3/2}\sqrt{m_i}\epsilon_0^2)n_i/T_i^{3/2}$. We define with the parallel Spitzer resistivity $\eta_\parallel := \frac{0.51 m_e\nu_{ei}}{n_e e^2}$ and the parallel electron and ion viscosities $\mu_{\parallel,e} := 0.73\frac{n_eT_e}{\nu_{ei}}$ and $\mu_{\parallel,i} = 0.96\frac{n_iT_i}{\nu_{ii}}$ [1]

$$\eta := \frac{en_0\eta_\parallel}{B_0} = \frac{\nu_{ei,0}}{\Omega_{ce}} = 8.45\cdot 10^{-5}\ln\lambda\left(\frac{n_0}{10^{19}\mathsf{m}^3}\right)\left(\frac{T_e}{\mathsf{eV}}\right)^{-3/2}\left(\frac{B_0}{\mathsf{T}}\right)^{-1}, \tag{87a}$$

$$\nu_{\parallel,e} := \frac{\mu_{\parallel,e,0}}{m_en_0\rho_s^2\Omega_{ci}} = 0.73\frac{\Omega_{ce}}{\nu_{ei,0}} = \frac{0.73}{\eta} \tag{87b}$$

$$\nu_{\parallel,i} := \frac{\mu_{\parallel,i,0}}{m_in_0\rho_s^2\Omega_{ci}} = 0.96\frac{\Omega_{ci}}{\nu_{ii,0}} = \sqrt{\frac{m_e}{m_i}}\frac{1.36}{\eta} \tag{87c}$$

with $\ln\lambda \approx 10$ and $T_e = T_i$ for the purpose of computing the diffusive coefficients. The approximate Spitzer current $J_{\parallel,s} := n_e\left(U_{\parallel,i} - u_{\parallel,e}\right)$ determines the parallel resistive terms to $R_\parallel := n_e\eta J_{\parallel,s}$.

For the perpendicular terms we use ad-hoc numerical diffusion terms for numerical stabilisation.

$$\Lambda_{n_e} = \nu_\perp\Delta_\perp n_e \text{ or } -\nu_\perp\Delta_\perp^2 n_e \qquad \Lambda_{N_i} = \nu_\perp\Delta_\perp N_i \text{ or } -\nu_\perp\Delta_\perp^2 N_i \tag{88}$$

$$\Lambda_{u_e} = \nu_\perp\Delta_\perp u_{\parallel,e} \text{ or } -\nu_\perp\Delta_\perp^2 u_{\parallel,e} \qquad \Lambda_{U_i} = \nu_\perp\Delta_\perp U_{\parallel,i} \text{ or } -\nu_\perp\Delta_\perp^2 U_{\parallel,i} \tag{89}$$

Here the mass diffusion coefficient coincides with the viscous coefficient, hence we fixed the Schmidt number $Sc_\parallel := \frac{\nu_U}{\nu_N}$ to unity.

We do not derive resistive drifts in the gyro-fluid approach. The drift-fluid corresponding resistive drift gives an order-of-magnitude estimate for $\nu_\perp$. We have $D_i = \rho_i^2\nu_{ii}$ and $T_{i0} = T_{e0}$. By dividing by $\rho_s^2\Omega_{ci}$ we arrive at $\nu_\perp = \nu_{ii0}/\Omega_{ci}$.

$$\nu_\perp = 5\cdot 10^{-3}\ln\lambda\left(\frac{n_0}{10^{19}\mathsf{m}^3}\right)\left(\frac{T_e}{\mathsf{eV}}\right)^{-3/2}\left(\frac{B_0}{\mathsf{T}}\right)^{-1}\left(\frac{m_i}{m_H}\right)^{1/2}, \tag{90}$$

2.3 Boundary and initial conditions

We define the simulation box as $[R_{\min}, R_{\max}] \times [Z_{\min}, Z_{\max}] \times [0, 2\pi]$, where we define

$$\begin{aligned} R_{\min} &= R_0 - \varepsilon_{R-} a & R_{\max} &= R_0 + \varepsilon_{R+} a \\ Z_{\min} &= -\varepsilon_{Z-} a e & Z_{\max} &= \varepsilon_{Z+} a e \end{aligned} \tag{91}$$

where a is the minor radius, e is the elongation of the flux surfaces and the ε are free parameters to be specified by the user.

We choose boundary conditions separately on input for the variables n_e, $u_{\parallel,e}$ and ϕ. The boundary condition for N_i, $U_{\parallel,i}$ and ψ are equal to n_e, $u_{\parallel,e}$ and ϕ respectively. We choose $A_\parallel$ to have equal boundary conditions as $u_{\parallel,e}$ and $U_{\parallel,i}$. This will later enable us to treat the sum of $U_\parallel$ and $A_\parallel$ in the same way as $U_\parallel$. Typically,

$$n_e = n_0, \quad u_{\parallel,e} = \phi = 0 \text{ or } \hat{n} \cdot \boldsymbol{\nabla} n_e = \hat{n} \cdot \boldsymbol{\nabla} u_{\parallel,e} = 0 \tag{92}$$

where $\hat{n}$ is the normal vector to the boundary.

We initialize the parallel velocity to zero

$$u_{\parallel,e}(R, Z, \varphi, 0) = U_{\parallel,i}(R, Z, \varphi, 0) = 0 \tag{93}$$

which in turn initializes $A_\parallel = 0$ and initialize the electron density with

$$n_e(R, Z, \varphi, 0) = n_{prof}(R, Z) + \tilde{n}(R, Z, \varphi) \tag{94}$$

consisting of a toroidally symmetric background profile $n_{\text{prof}}(R, Z)$ and a perturbation $\tilde{n}(R, Z, \varphi)$, which breaks the toroidal symmetry. Note that we should take care to intitialize a smooth profile with ideally well-defined $\Delta_\perp^2 n_e$.

We define a flux-aligned density profile as

$$n_{\text{prof}}(R, Z) = n_0 + \triangle n_{peak} \frac{\psi_p(R, Z)}{\psi_{p,O}} \Theta_{\alpha_p/2} \left(1 - \rho_p(R, Z) - \frac{\alpha_p}{2}\right) H(Z - Z_X) \tag{95}$$

The second Heaviside is multiplied only if the equilibrium ψ_p has an X-point and avoids a profile in the private flux region. The factor α_p provides a smooth transition zone that avoids numerical oscillations.

We have two possibilities to initialize the ion density

$$N_i = \Gamma_{1,i}^{-1} n_e \text{ or } N_i = \Gamma_{1,i} n_e \approx \left(1 + \frac{1}{2}\tau_i \mu_i \Delta_\perp\right) n_e \tag{96}$$

In the first case the potential $\phi = 0$ while in the second case the $E \times B$ and ion diamagnetic vorticity coincide $\Delta_\perp N_i \propto \Delta_\perp \phi$ in the long-wavelength limit. Note that α must not be too small to avoid $N_i < 0$. We can choose between several initial conditions for $\tilde{n}$:

2.3.1 Blob and Straight blob

We initialize a blob in the R-Z plane

$$\tilde{n}_{blob}(R, Z, 0) = \triangle n \exp\left(-\frac{(R - R_0 - p_x a)^2 + (Z - p_y a)^2}{\sigma^2}\right) \tag{97}$$

Then, we use fieldline integration modulated by

$$m_{blob}(s) = \exp\left(-\frac{s^2}{\pi^2\sigma_z^2}\right) \tag{98}$$

to transform this blob to all other poloidal planes. We either follow fieldlines around the torus several times ("blob") or only once ("straight blob").

2.3.2 Turbulent bath

We can initialize the R-Z plane with a turbulent bath with a certain amplitude A. This especially has the goal to destabilize the edge region right inside the last closed flux surface. Notice that the core region is rather stable and quickly damps away fluctuations. Again, we transform this to all poloidal planes along the magnetic field lines and multiply the bath with

$$\tilde{n}_e(R,Z,\varphi) = \tilde{n}_{\text{bath}}(R,Z,\varphi)\Theta_{\alpha_p/2}(-\rho_p(R,Z) - \alpha_p/2)H(Z-Z_X) \tag{99}$$

2.3.3 Zonal flows

We can initialize the R-Z plane with zonal flows of amplitude A and wavelength k_ψ aligned with the magnetic flux surfaces.

$$\begin{aligned}\tilde{n}_{\text{zonal}}(R,Z) &= A\sin(2\pi k_\psi \psi_p(R,Z))\\ \tilde{n}_e(R,Z,\varphi) &= \tilde{n}_{\text{zonal}}(R,Z)\Theta_{\alpha_p}\left(-\rho_p(R,Z) - \frac{\alpha_p}{2}\right)H(Z-Z_X)\end{aligned} \tag{100}$$

2.3.4 Turbulence on Gaussian profile

Instead of the flux-aligned profile we can also choose a toroidally symmetric Gaussian profile

$$n_{prof}(R,Z) = n_0 + \triangle n_{peak}\exp\left(-\frac{(R-R_0-p_x a)^2 + (Z-p_y a)^2}{\sigma^2}\right) \tag{101}$$

on top of which we can add the turbulent bath $\tilde{n}_{\text{bath}}$ and finally dampen it by

$$n_e(R,Z,\varphi,0) = (n_{prof}(R,Z) + \tilde{n}_{\text{bath}})\Theta_{\alpha_p/2}\left(1 - \sqrt{(R-R_0)^2+Z^2}/a\right) \tag{102}$$

2.4 Sinks and sources

We can choose the source terms S_N to either force a profile n_{prof} or provide a constant influx of particles in the core of our domain, where our model does not apply. We thus define a particle sink/source for electrons as

$$S_{n_e}(R,Z,\varphi,t) = \omega_s \begin{cases} (n_{prof}(R,Z) - n_e(R,Z,\varphi,t))\Theta_{\alpha_p/2}\left(\rho_{p,s} - \frac{\alpha_p}{2} - \rho_p(R,Z)\right)H(Z-Z_X) & \text{forced} \\ S_{prof}(R,Z) & \text{influx} \end{cases} \tag{103}$$

where ω_s is the source strength parameter. The shift of Θ is chosen such that the source vanishes exactly outside $\psi_{p,s}$. The forced source will result in exponential adaption of the core density profile of the form $n_e \propto n_{prof} + (n_{prof} - n_{e,0})e^{-\omega_s t}$.

We can choose the constant influx

$$S_{prof}(R,Z) = \Theta_{\alpha_p/2}\left(\rho_{p,s} - \frac{\alpha_p}{2} - \rho_p(R,Z)\right) H(Z - Z_X) \tag{104}$$

or a ringed Gaussian TCV source of the form

$$S_{prof}(R,Z) = \exp\left(-\frac{(\psi_p - \psi_{p,0})^2}{\sigma^2}\right) H(Z - Z_X) \tag{105}$$

with $\psi_{p,0} = \psi_p(1075, -10)$ and $\sigma = 0.0093\psi_{p,0}/0.4$, or a Torpex inspired source profile

$$S_{prof}(R,Z) = \begin{cases} \exp\left(-\frac{(R-R_0)^2}{a^2} - \frac{(Z-Z_0)^2}{b^2}\right) & \text{if} R > R_0 \\ \frac{1}{2}\exp\left(-\frac{(R-R_0)^2}{a^2} - 2c(R-R_0)(Z-Z_0) - \frac{(Z-Z_0)^2}{b^2}\right) & \\ +\frac{1}{2}\exp\left(-\frac{(R-R_0)^2}{a^2} + 2c(R-R_0)(Z-Z_0) - \frac{(Z-Z_0)^2}{b^2}\right) & \text{else} \end{cases} \tag{106}$$

with $a = 0.0335$m, $b = 0.05$m, $c = 565m^{-2}$, $R_0 = 0.98$m and $Z_0 = -0.02$m.

In order to not generate potential with the source term the ion source needs to fulfill $S_{n_e} = \Gamma_{1,i}S_{N_i} + \boldsymbol{\nabla}\cdot\left(\frac{\mu_i S_{N_i}}{B^2}\boldsymbol{\nabla}_\perp\phi\right)$ which in the long wavelength limit can be inverted to (the long wavelength limit should be well-fulfilled for a realistic source term since the amplitude is typically quite small)

$$S_{N_i} = \left(1 - \frac{1}{2}\mu_i\tau_i\Delta_\perp\right) S_{n_e} - \boldsymbol{\nabla}\cdot\left(\frac{\mu_i S_{n_e}}{B^2}\boldsymbol{\nabla}_\perp\phi\right) \tag{107}$$

Note that the additional terms besides S_{n_e} are total divergences which means they do not change the volume integrated "total" particle number created by the source. Note that S_{n_e} needs to be smooth so that $\boldsymbol{\nabla}_\perp^2 S_{n_e}$ is well defined. Also note that with our definition of Λ_{n_e} and Λ_{N_i} and the polarisation equation we have $\Lambda_{n_e} = \Gamma_{1,i}\Lambda_{N_i} + \boldsymbol{\nabla}\cdot\left(\frac{\mu_i\Lambda_{N_i}}{B^2}\boldsymbol{\nabla}_\perp\phi\right)$ in the long wavelength limit (swap the operators). This means that diffusion does not generate potential either.

Now, our idea is to dampen the density and velocity in the region defined by the magnetic field straightening. The idea for the terms S_U is mainly to provide more numerical stability in the corner regions of the domain, where the parallel derivative may lead to unwelcome numerical instabilities. For both electrons and ions we choose

$$S^d_{n_e}(R,Z,\varphi,t) := -\omega_d(n_e - 1)\Theta_{\alpha_p/2}\left(\rho_p(R,Z) - \rho_{p,b} - \frac{\alpha_p}{2}\right) \tag{108a}$$

$$S^d_{N_i}(R,Z,\varphi,t) = \left(1 - \frac{1}{2}\mu_i\tau_i\Delta_\perp\right) S^d_{n_e} - \boldsymbol{\nabla}\cdot\left(\frac{\mu_i S^d_{n_e}}{B^2}\boldsymbol{\nabla}_\perp\phi\right) \tag{108b}$$

$$S^d_U(R,Z,\varphi,t) := -\omega_d U_\parallel\Theta_{\alpha_p/2}\left(\rho_p(R,Z) - \rho_{p,b} - \frac{\alpha_p}{2}\right) - \frac{U_\parallel}{N}S_N \tag{108c}$$

The last term is there to avoid introducing an unintentional parallel momentum source through the density sources.

2.5 Implemented form

The form that we implement avoids derivatives on the product of two functions for which we have no boundary conditions

$$\begin{aligned}\frac{\partial}{\partial t}N =& -\frac{1}{B}[\psi,N]_\perp - \bar{\nabla}_\parallel\left(NU_\parallel\right) - NU_\parallel\left(\boldsymbol{\nabla}\cdot\hat{\boldsymbol{b}} + \boldsymbol{\nabla}\cdot\boldsymbol{b}_\perp\right) - \tau\mathcal{K}(N)\\ &- N\mathcal{K}(\psi) - \mu\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(NU_\parallel^2) - \mu NU_\parallel^2\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} - \nu_\perp\Delta_\perp^2 N + S_N, \qquad (109a)\\ \frac{\partial}{\partial t}W_\parallel =& -\frac{1}{B}\left[\psi,U_\parallel\right]_\perp - \frac{1}{\mu}\bar{\nabla}_\parallel\psi - \frac{1}{2}\bar{\nabla}_\parallel U_\parallel^2 - \frac{\tau}{\mu}\bar{\nabla}_\parallel\ln N - U_\parallel\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi) - \tau\mathcal{K}(U_\parallel) - \tau U_\parallel\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\\ &- \left(2\tau+\mu U_\parallel^2\right)\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(U_\parallel) - 2\tau U_\parallel\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\ln N) - \frac{\eta}{\mu}\frac{n_e}{N}n_e(U_{\parallel,i}-u_{\parallel,e}) + \frac{\nu_\parallel}{N}\Delta_\parallel U_\parallel\\ &- \nu_\perp\Delta_\perp^2 U_\parallel + S_U, \qquad (109b)\\ W_\parallel :=& \left(U_\parallel + \frac{A_\parallel}{\mu}\right) \qquad (109c)\end{aligned}$$

together with $\bar{\nabla}_\parallel f = \nabla_\parallel f + A_\parallel\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(f) + \frac{1}{B}[f,A_\parallel]_\perp$ and $\boldsymbol{\nabla}\cdot\boldsymbol{b}_\perp = A_\parallel\boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{K}}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} - \mathcal{K}_{\boldsymbol{\nabla}B}(A_\parallel)$ and

$$-\boldsymbol{\nabla}\cdot\left(\frac{N_i}{B^2}\boldsymbol{\nabla}_\perp\phi\right) = \Gamma_{1,i}N_i - n_e, \qquad \Gamma_{1,i}^{-1} = 1 - \frac{1}{2}\tau_i\mu_i\Delta_\perp \qquad (110a)$$

$$\psi_e = \phi, \quad \psi_i = \Gamma_{1,i}\phi - \frac{\mu_i}{2}\frac{(\boldsymbol{\nabla}_\perp\phi)^2}{B^2} \qquad (110b)$$

$$\left(\frac{\beta}{\mu_i}N_i - \frac{\beta}{\mu_e}n_e - \Delta_\perp\right)A_\parallel = \beta\left(N_iW_i - n_ew_e\right) \qquad (110c)$$

Note that the negative signs make the operators in Eqs. (110) positive definite.

In the output file we have

Name	Equation	Name	Equation
electrons	n_e	ions	N_i
Ue	$u_{\parallel,e}$	Ui	$U_{\parallel,i}$
potential	ϕ	psi	ψ
induction	$A_\parallel$		

2.6 Scale invariance

2.6.1 Sign reversals of the magnetic field

If we change the direction of the magnetic field vector $\hat{\boldsymbol{b}}$, we immediately see that all perpendicular drifts and $U_\parallel\hat{\boldsymbol{b}}$ change directions. On the other side, the diffusive and resistive terms remain unchanged. Without resistivity and diffusion a change in direction of the magnetic field thus corresponds to a time reversal $t \to t' = -t$. In the code $\hat{\boldsymbol{b}}$ changes sign by using both $-\mathcal{P}_\psi$ and $-\mathcal{P}_I$.

Also note that changing the sign of the magnetic field only in the parallel derivatives $\nabla_\parallel \to -\nabla_\parallel$ does not have any effect. This can be seen by simply renormalizing $U'_\parallel = -U_\parallel$. This reverts the equations back to the original equations.

2.6.2 Scaling of density

If $N, U_\parallel, \phi, A_\parallel$ are a solution to the model equations then so are $N' = \alpha N$, $U'_\parallel = U_\parallel$, $\phi' = \phi$ and $A'_\parallel = A_\parallel$ with the changed parameters $S'_N = \alpha S_N$, $\eta' = \eta/\alpha$ and $\beta' = \beta/\alpha$. If N has a Dirichlet boundary condition, then N' satisfies a correspondingly scaled boundary condition.

2.6.3 Helicity

The standard helicity of the magnetic field would be a right handed screw, where the magnetic field and the toroidal plasma current point in the clockwise direction if the tokamak is seen from above. The helicity should however not have any influence on the plasma dynamics as the tokamak viewed in the mirror has the reversed helicity.

2.7 Conservation laws

2.7.1 Mass conservation

The density equations directly yield the particle conservation

$$\frac{\partial}{\partial t} N + \boldsymbol{\nabla} \cdot \boldsymbol{j_N} = \Lambda_N + S_N \tag{111}$$

The terms of the particle conservation thus read

$$N = N, \tag{112}$$

$$\boldsymbol{j}_N = N\left(\boldsymbol{u}_\psi + \boldsymbol{u}_C + \boldsymbol{u}_K + U_\parallel\left(\hat{\boldsymbol{b}} + \boldsymbol{b}_\perp\right)\right) \tag{113}$$

$$\Lambda_N = \nu_\perp \Delta_\perp N \tag{114}$$

$$S_N = S_N \tag{115}$$

Notice that

$$\tau N \boldsymbol{K} = \tau N \boldsymbol{\nabla} \times \frac{\hat{\boldsymbol{b}}}{B} = \tau \boldsymbol{\nabla} \times N\frac{\hat{\boldsymbol{b}}}{B} + \tau \frac{\hat{\boldsymbol{b}} \times \boldsymbol{\nabla} N}{B} \tag{116}$$

such that we can define the diamagnetic flux in the particle flux since the rotation vanishes under the divergence.

We here also derive the particle flux (113) through a flux surface

$$\begin{aligned}\boldsymbol{j}_N \cdot \boldsymbol{\nabla} v =& \frac{\mathrm{d}v}{\mathrm{d}\psi_p} N\left[\frac{1}{B}[\psi, \psi_p]_\perp + \left(\tau + \mu U_\parallel^2\right)\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p) + \tau \mathcal{K}_{\boldsymbol{\nabla} B}(\psi_p)\right] \\ &+ N U_\parallel \frac{\mathrm{d}v}{\mathrm{d}\psi_p}\left[\left(A_\parallel \mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p) + \frac{1}{B}[\psi_p, A_\parallel]_\perp\right)\right]\end{aligned} \tag{117}$$

The relevant terms in the output file are

Name	Equation	Name	Equation
electrons	n_e	jsneC_tt	$n_e(\boldsymbol{u}_K + \boldsymbol{u}_C)\cdot\boldsymbol{\nabla}\psi_p$
jsneA_tt	$n_e u_{\parallel,e}\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p$	jsneE_tt	$n_e\boldsymbol{u}_E\cdot\boldsymbol{\nabla}\psi_p$
lneperp_tt	$\Lambda_{\perp,n_e} = \nu_\perp\Delta_\perp n_e$ or $-\nu_\perp\Delta_\perp^2 n_e$		
sne_tt	S_{n_e}	jsdiae_tt	$\tau_e\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}n_e\cdot\boldsymbol{\nabla}\psi_p/B$
dnepar_tt	$\boldsymbol{\nabla}\cdot(\hat{\boldsymbol{b}}n_e u_{e,\parallel})$		
ions	N_i	jsniC_tt	$N_i(\boldsymbol{u}_K + \boldsymbol{u}_C)\cdot\boldsymbol{\nabla}\psi_p$
jsniA_tt	$N_i U_{\parallel,i}\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p$	jsniE_tt	$N_i\boldsymbol{u}_E^i\cdot\boldsymbol{\nabla}\psi_p$
lniperp_tt	$\Lambda_{\perp,N_i} = \nu_\perp\Delta_\perp N_i$ or $-\nu_\perp\Delta_\perp^2 N_i$		
sni_tt	S_{N_i}	jsdiai_tt	$\tau_i\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}N_i\cdot\boldsymbol{\nabla}\psi_p/B$
dnipar_tt	$\boldsymbol{\nabla}\cdot(\hat{\boldsymbol{b}}N_i U_{i,\parallel})$		

2.7.2 Energy theorem

The terms of the energy theorem are

$$\partial_t\mathcal{E} + \boldsymbol{\nabla}\cdot\boldsymbol{j}_\mathcal{E} = \Lambda_\mathcal{E} + S_\mathcal{E} + R_\mathcal{E} \tag{118}$$

with ($z_e = -1$ and $z_i = +1$) and $\boldsymbol{u}_E := \hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi/B$

$$\begin{aligned}
\mathcal{E} =& z_e\tau_e n_e\ln(n_e) + z_i\tau_i N_i\ln(N_i) + \frac{1}{2\beta}\left(\boldsymbol{\nabla}_\perp A_\parallel\right)^2 + \frac{1}{2}z_i\mu_i N_i u_E^2 \\
&+ \frac{1}{2}z_e\mu_e n_e u_{\parallel,e}^2 + \frac{1}{2}z_i\mu_i N_i U_{\parallel,i}^2, \qquad (119)\\
\boldsymbol{j}_\mathcal{E} =& \sum_s z\left[\left(\tau\ln N + \frac{1}{2}\mu U_\parallel^2 + \psi\right)N\left(\boldsymbol{u}_E + \boldsymbol{u}_C + \boldsymbol{u}_K + U_\parallel\left(\hat{\boldsymbol{b}} + \boldsymbol{b}_\perp\right)\right)\right] \\
&+ \sum_z z\left[\mu\tau N U_\parallel^2\boldsymbol{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}} + \tau N U_\parallel\left(\hat{\boldsymbol{b}} + \boldsymbol{b}_\perp\right)\right], \qquad (120)\\
\Lambda_\mathcal{E} =& \sum_s z\left[\left(\tau(1+\ln N) + \psi + \frac{1}{2}\mu U_\parallel^2\right)(\nu_\perp\Delta_\perp N) + \mu N U_\parallel\left(\nu_\perp\Delta_\perp U_\parallel + \nu_\parallel\Delta_\parallel U_\parallel\right)\right] \\
S_\mathcal{E} =& \sum_s z\left[\left(\tau(1+\ln N) + \psi + \frac{1}{2}\mu U_\parallel^2\right)S_N\right] \\
R_\mathcal{E} =& -\eta_\parallel\left[n_e(U_{\parallel,i} - u_{\parallel,e})\right]^2. \qquad (121)
\end{aligned}$$

where in the energy flux $\boldsymbol{j}_\mathcal{E}$ we neglect terms containing time derivatives of the eletric and magnetic potentials and we sum over all species. The energy density $\mathcal{E}$ consists of the Helmholtz free energy density for electrons and ions, the $\boldsymbol{E}\times\boldsymbol{B}$ energy density, the parallel energy densities for electrons and ions and the perturbed magnetic field energy density. In Λ we insert the dissipative terms of Section 2.2.

Replace $\Delta_\perp$ with $-\Delta_\perp^2$ when hyperviscous diffusion is chosen for the diffusion terms in the above equations.

We have the energy flux through a flux surface

$$\begin{aligned}\boldsymbol{j}_\mathcal{E}\cdot\boldsymbol{\nabla}v =& \frac{\mathrm{d}v}{\mathrm{d}\psi_p}\sum_s z\left(\tau\ln N + \frac{1}{2}\mu U_\parallel^2 + \psi\right)\boldsymbol{j}_N\cdot\boldsymbol{\nabla}\psi_p + z\mu\tau N U_\parallel^2\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p)\\ &+ z\tau N U_\parallel\left(A_\parallel\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p) + \frac{1}{B}[\psi_p, A_\parallel]_\perp\right)\end{aligned} \tag{122}$$

The relevant terms in the output file are

Name	Equation
nelnne	$z_e\tau_e n_e\ln n_e$
nilnni	$z_i\tau_i N_i\ln N_i$
aperp2	$(\boldsymbol{\nabla}_\perp A_\parallel)^2/2/\beta$
ue2	$z_i\mu_i N_i u_E^2/2$
neue2	$z_e\mu_e n_e u_{\parallel,e}^2/2$
niui2	$z_i\mu_i N_i U_{\parallel,i}^2/2$
see_tt	$z_e(\tau_e(1+\ln n_e)+\phi+\frac{1}{2}\mu_e u_{\parallel,e}^2)S_{n_e}$
sei_tt	$z_i(\tau_i(1+\ln N_i)+\psi+\frac{1}{2}\mu_i U_{\parallel,i}^2)S_{N_i}$
resistivity_tt	-$\eta_\parallel n_e^2(U_{\parallel,i}-u_{\parallel,e})^2$
jsee_tt	$z_e(\tau_e\ln n_e+\mu_e u_{\parallel,e}^2/2+\phi)n_e(\boldsymbol{u}_E+\boldsymbol{u}_C+\boldsymbol{u}_K)\cdot\boldsymbol{\nabla}\psi_p + z_e\tau_e n_e u_{\parallel,e}^2\boldsymbol{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\cdot\boldsymbol{\nabla}\psi_p$
jsei_tt	$z_i(\tau_i\ln N_i+\mu_i U_{\parallel,i}^2/2+\psi_i)N_i(\boldsymbol{u}_E^i+\boldsymbol{u}_C+\boldsymbol{u}_K)\cdot\boldsymbol{\nabla}\psi_p + z_i\tau_i N_i U_{\parallel,i}^2\boldsymbol{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\cdot\boldsymbol{\nabla}\psi_p$
jseea_tt	$z_e(\tau_e\ln n_e+\mu_e u_{\parallel,e}^2+\phi)n_e\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p + z_e\tau_e n_e u_{\parallel,e}\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p$
jseia_tt	$z_i(\tau_i\ln N_i+\mu_i U_{\parallel,i}^2+\psi_i)N_i\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p + z_i\tau_i N_i U_{\parallel,i}\boldsymbol{b}_\perp\cdot\boldsymbol{\nabla}\psi_p$
leeperp_tt	$z_e(\tau_e(1+\ln n_e)+\phi+\mu_e u_{\parallel,e}^2/2)\nu_\perp\Delta_\perp n_e + z_e\mu_e n_e u_{\parallel,e}\nu_\perp\Delta_\perp u_{\parallel,e}$
leiperp_tt	$z_i(\tau_i(1+\ln N_i)+\psi_i+\mu_i U_{\parallel,i}^2/2)\nu_\perp\Delta_\perp N_i + z_i\mu_i N_i U_{\parallel,i}\nu_\perp\Delta_\perp U_{\parallel,i}$
leeparallel_tt	$z_e\mu_e n_e u_{\parallel,e}\nu_\parallel\Delta_\parallel u_{\parallel,e}$
leiparallel_tt	$z_i\mu_i N_i U_{\parallel,i}\nu_\parallel\Delta_\parallel U_{\parallel,i}$

2.7.3 Toroidal ExB angular momentum equation

We integrate the polarisation equation over volume, multiply by $\mathrm{d}\psi_p/\mathrm{d}v$ and derive by time. In the drift-ordering up to order $\mathcal{O}(\delta^3)$ we get

$$\partial_t \langle \Omega \rangle + \frac{\partial}{\partial v}\frac{\mathrm{d}v}{\mathrm{d}\psi_p}\langle \boldsymbol{j}_\Omega \cdot \boldsymbol{\nabla}\psi_p\rangle = -\langle F_{L,\varphi}\rangle + \langle \mathcal{S}_\Omega\rangle \tag{123}$$

$$\Omega := \mu_i N_i \left(\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2} + \tau_i \boldsymbol{\nabla}\ln N_i\cdot\boldsymbol{\nabla}\psi_p\right) \equiv \mu_i N_i (u_{E,\varphi}+u_{D,\varphi}) \tag{124}$$

$$\boldsymbol{j}_\Omega := \Omega \boldsymbol{u}_E - \left(\frac{1}{\beta}\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}A_\parallel + \frac{1}{2}\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}(N_iU_{\parallel,i})\right)\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}A_\parallel}{B} \tag{125}$$

$$F_{L,\varphi} := -(z_e\tau_e n_e + z_i\tau_i N_i)\mathcal{K}(\psi_p) - (z_e\mu_e n_e u_{\parallel,e}^2 + z_i\mu_i N_i U_{\parallel,i}^2)\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p) \tag{126}$$

$$\mathcal{S}_\Omega := \mu_i S_{n_e}\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2} + \mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}S_{n_e} \tag{127}$$

Equation (123) can be rewritten by inserting the continuity equation to yield an equation only for the $\boldsymbol{E}\times\boldsymbol{B}$ angular momentum. Again up to order $\mathcal{O}(\delta^3)$ in the drift ordering we obtain (the diffusive term is for testing purposes)

$$\partial_t \langle \Omega_E \rangle + \frac{\partial}{\partial v}\frac{\mathrm{d}v}{\mathrm{d}\psi_p}\langle \boldsymbol{j}_{\Omega_E}\cdot\boldsymbol{\nabla}\psi_p\rangle = -\langle F_{L,\varphi}\rangle + \langle \mathcal{S}_{\Omega_E}\rangle + \langle \Lambda_{\Omega_E}\rangle \tag{128}$$

$$\Omega_E := \mu_i N_i \frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2} \equiv \mu_i N_i u_{E,\varphi} \tag{129}$$

$$\boldsymbol{j}_{\Omega_E} := \Omega_E(\boldsymbol{u}_E+\boldsymbol{u}_D) - \boldsymbol{\nabla}A_\parallel\cdot\boldsymbol{\nabla}\psi_p\left(\frac{1}{\beta}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}A_\parallel}{B} + \frac{1}{2}\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\mu_i\tau_i N_i U_{\parallel,i}\right) \tag{130}$$

$$\mathcal{S}_{\Omega_E} := \mu_i S_{n_e}\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2} \quad \Lambda_{\Omega_E} := \mu_i\Lambda_{n_e}\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2} \tag{131}$$

where here we also monitor the source and diffusion terms. In the output file we have

Name	Equation	Name	Equation
oexbe	$\mu_i n_e\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}$	oexbi	$\mu_i N_i\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}$
odiae	$\mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}n_e$	odiai	$\mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}N_i$
jsoexbi_tt	$\mu_i N_i\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\psi_p}{B}$	jsoexbe_tt	$\mu_i n_e\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\psi_p}{B}$
jsodiaiUE_tt	$\mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}N_i\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\psi_p}{B}$	jsodiaeUE_tt	$\mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}n_e\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\psi_p}{B}$
jsoexbiUD_tt	$\mu_i\tau_i\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}N_i\cdot\boldsymbol{\nabla}\psi_p}{B}$	jsoexbeUD_tt	$\mu_i\tau_i\frac{\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi}{B^2}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}n_e\cdot\boldsymbol{\nabla}\psi_p}{B}$
jsoapar_tt	$-\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}A_\parallel\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}A_\parallel\cdot\boldsymbol{\nabla}\psi_p}{B\beta}$	jsodiaApar_tt	$-\frac{1}{2}\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}(N_iU_{\parallel,i})\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}A_\parallel}{B}\cdot\boldsymbol{\nabla}\psi_p$
jsoexbApar_tt	$-\frac{1}{2}\tau_i\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}(N_iU_{\parallel,i})\cdot\boldsymbol{\nabla}\psi_p\boldsymbol{\nabla}A_\parallel\cdot\boldsymbol{\nabla}\psi_p$	socurve_tt	$z_e\tau_e n_e\mathcal{K}(\psi_p)$
socurvi_tt	$z_i\tau_i N_i\mathcal{K}(\psi_p)$	socurvkappae_tt	$z_e\mu_e n_e u_{\parallel,e}^2\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p)$
socurvkappai_tt	$z_i\mu_i N_i U_{\parallel,i}^2\mathcal{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}(\psi_p)$		
sosne_tt	$\mu_i S_{n_e}\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}\phi/B^2$	sospi_tt	$\mu_i\tau_i\boldsymbol{\nabla}\psi_p\cdot\boldsymbol{\nabla}S_{n_e}$

loexbe_tt	$\mu_i \Lambda_{n_e} \boldsymbol{\nabla}\psi_p \cdot \boldsymbol{\nabla}\phi / B^2$

2.7.4 Parallel momentum balance

The flux surface average over the parallel momentum equation under species summation yields up to order $\mathcal{O}(\delta^3)$ in the drift-ordering

$$\frac{\partial}{\partial t}\left\langle \mu_i N_i U_{\parallel,i}\right\rangle + \frac{\partial}{\partial v}\frac{\mathrm{d}v}{\mathrm{d}\psi_p}\left\langle \mu_i N_i U_{\parallel,i}\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi}{B}\cdot\boldsymbol{\nabla}\psi_p + \sum_s (z_s\tau_s N_s + z_s\mu_s N_s U_{\parallel,s}^2) b_\perp^v\right\rangle = \sum_s \left\langle -z_s\tau_s N_s \nabla_\parallel \ln B\right\rangle + \mu_i\left\langle S_{N_i}U_{\parallel,i} + N_i S_{U_\parallel}\right\rangle \tag{132}$$

while the toroidal parallel angular momentum contribution reads up to order $\mathcal{O}(\delta^3)$

$$\frac{\partial}{\partial t}\left\langle \mu_i N_i U_{\parallel,i} b_\varphi\right\rangle + \frac{\partial}{\partial v}\frac{\mathrm{d}v}{\mathrm{d}\psi_p}\left\langle \mu_i N_i U_{\parallel,i} b_\varphi\frac{\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi}{B}\cdot\boldsymbol{\nabla}\psi_p + \sum_s (z_s\tau_s N_s + z_s\mu_s N_s U_{\parallel,s}^2) b_\varphi b_\perp^v\right\rangle = \left\langle F_{L,\varphi}\right\rangle + \mu_i\left\langle (S_{N_i}U_{\parallel,i} + N_i S_{U_\parallel}) b_\varphi\right\rangle \tag{133}$$

The relevant terms in the output file are (the Lorentz force term is described in the previous subsection 2.7.3)

Name	Equation	Name	Equation
neue	$n_e u_{\parallel,e}$	niui	$\mu_i N_i U_{\parallel,i}$
neuebphi	$n_e u_{\parallel,e} b_\varphi$	niuibphi	$\mu_i N_i U_{\parallel,i} b_\varphi$
jsparexbi_tt	$\mu_i N_i U_{\parallel,i}(\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi)\cdot\boldsymbol{\nabla}\psi_p/B$	jsparbphiexbi_tt	$\mu_i N_i U_{\parallel,i} b_\varphi(\hat{\boldsymbol{b}}\times\boldsymbol{\nabla}\phi)\cdot\boldsymbol{\nabla}\psi_p/B$
jspardiai_tt	$\mu_i\tau_i N_i U_{\parallel,i}\boldsymbol{K}\cdot\boldsymbol{\nabla}\psi_p$	jsparbphdiai_tt	$\mu_i\tau_i N_i U_{\parallel,i} b_\varphi\boldsymbol{K}\cdot\boldsymbol{\nabla}\psi_p$
jsparkappai_tt	$\mu_i N_i U_{\parallel,i}(\mu_i U_{\parallel,i}^2 + 2\tau_i)\boldsymbol{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\cdot\boldsymbol{\nabla}\psi_p$	jsparbphikappai_tt	$\mu_i N_i U_{\parallel,i} b_\varphi(\mu_i U_{\parallel,i}^2 + 2\tau_i)\boldsymbol{K}_{\boldsymbol{\nabla}\times\hat{\boldsymbol{b}}}\cdot\boldsymbol{\nabla}\psi_p$
jsparApar_tt	$\sum_s (z_s\tau_s N_s + z_s\mu_s N_s U_{\parallel,s}^2) b_\perp^v$	jsparbphiApar_tt	$\sum_s (z_s\tau_s N_s + z_s\mu_s N_s U_{\parallel,s}^2) b_\varphi b_\perp^v$
sparmirrore_tt	$-z_e\tau_e n_e\nabla_\parallel \ln B$	sparmirrori_tt	$-z_i\tau_i N_i\nabla_\parallel \ln B$
sparsni_tt	$\mu_i S_{N_i} U_{\parallel,i} + \mu_i S_{U_i} N_i$	sparsnibphi_tt	$\mu_i S_{N_i} U_{\parallel,i} b_\varphi + \mu_i S_{U,i} N_i b_\varphi$
lparpar_tt	$\nu_\parallel \Delta_\parallel U_{\parallel,i}$	lparperp_tt	$-\nu_\perp U_{\parallel,i}\Delta_\perp^2 N_i - \nu_\perp N_i \Delta_\perp^2 U_{\parallel,i}$

2.7.5 Parallel electron force balance

We gather the dominant terms in the electron momentum equation (neglecting all terms as $\mu_e = 0$). This leaves the parallel force balance

$$-(\hat{\boldsymbol{b}} + \boldsymbol{b}_\perp)\cdot\boldsymbol{\nabla} n_e + n_e\left(\left(\hat{\boldsymbol{b}} + \boldsymbol{b}_\perp\right)\cdot\boldsymbol{\nabla}\phi + \frac{\partial A_\parallel}{\partial t}\right) + \eta n_e^2\left(U_{\parallel,i} - u_{\parallel,e}\right) = 0 \tag{134}$$

Name	Equation	Name	Equation
sparphie_tt	$n_e\nabla_\parallel\phi$	friction_tt	$\eta n_e^2(U_{\parallel,i}-u_{\parallel,e})$
sparmirrore_tt	$-\nabla_\parallel n_e$	sparmirrorAe_tt	$-\boldsymbol{\nabla}A_\parallel\times\hat{\boldsymbol{b}}\cdot\boldsymbol{\nabla}n_e/B$
sparphiAe_tt	$n_e\boldsymbol{\nabla}A_\parallel\times\hat{\boldsymbol{b}}\cdot\boldsymbol{\nabla}\phi/B$	spardotAe_tt	$n_e\partial A_\parallel/\partial t$

2.7.6 Zonal flow energy

$$E_{\text{zonal}} = \frac{1}{2}\left\langle\rho_M\right\rangle\left[\left[\iota^{-2}\mathcal{I}^{\vartheta\vartheta}+2\iota^{-1}\mathcal{I}^{\vartheta\varphi}+\mathcal{I}^{\varphi\varphi}\right]\right]\left[\left[u_{E,\varphi}\right]\right]^2 \equiv \frac{1}{2}\left\langle\rho_M\right\rangle\left[\left[u_{E,\varphi}\right]\right]^2\left[\left[\mathcal{I}_0\right]\right] \tag{135}$$

For symmetry flux coordinates we have $g_{\vartheta\vartheta}=R^2(\boldsymbol{\nabla}\psi_p)^2/I^2\iota^2$, $g_{\varphi\vartheta}=0$ and $g_{\varphi\varphi}=R^2$ and thus $\mathcal{I}_0=R^{-2}(1+I^2/|\boldsymbol{\nabla}\psi_p|^2)=B^2/|\boldsymbol{\nabla}\psi_p|^2$.

$$\begin{aligned}\frac{\partial}{\partial t}E_{\text{zonal}}+\frac{\partial}{\partial v}\left(E_{\text{zonal}}\left[\left[u^v\right]\right]\right) =& -\left[\left[\mathcal{I}_0\right]\right]\left[\left[u_{E,\varphi}\right]\right]\left(\frac{\partial}{\partial v}\Theta^v_\varphi+\left\langle(\boldsymbol{j}_f\times\boldsymbol{B})_\varphi\right\rangle\right)\\ &-\frac{1}{2}\left[\left[u_{E,\varphi}\right]\right]^2\frac{\partial}{\partial v}\left(\left\langle\rho_M\right\rangle\left[\left[\widehat{\mathcal{I}_0}\widehat{u^v}\right]\right]\right)+\mathcal{S}_{\text{zonal}}\end{aligned} \tag{136}$$

where we neglected the term $\langle n\boldsymbol{u}\cdot\boldsymbol{\nabla}\mathcal{I}_0\rangle$ in the continuity equation as small in our ordering and we have

$$\mathcal{S}_{\text{zonal}} := \left[\left[\mathcal{I}_0\right]\right]\left[\left[u_{E,\varphi}\right]\right]\mathcal{S}_{u_{E,\varphi}}+\frac{1}{2}\left[\left[u_{E,\varphi}\right]\right]^2\left\langle mS_n\mathcal{I}_0\right\rangle \tag{137}$$

and in the output file

Name	Equation	Name	Equation
nei0	$n_e\mathcal{I}_0$	snei0_tt	$S_{n_e}\mathcal{I}_0$

2.8 Manufactured Solution

In order to test the implementation we manufacture a solution to Eqs. (109) and (110) of the form

$$\begin{aligned}n_e(R,Z,\varphi,t) &:= 1+0.5\sin(\pi(R-R_0))\sin(\pi Z)\sin(\varphi)\sin(\pi t)\\ N_i(R,Z,\varphi,t) &:= n_e(R,Z,\varphi,t)=\gamma_{N_i}\\ u_{\parallel,e}(R,Z,\varphi,t) &:= \sin(2\pi(R-R_0))\sin(2\pi Z)\sin(2\varphi)\sin(2\pi t)/(3\sqrt{-\mu_e})\\ U_{\parallel,i}(R,Z,\varphi,t) &:= \sqrt{-\mu_e}u_{\parallel,e}(R,Z,\varphi,t)\\ \phi(R,Z,\varphi,t) &:= \sin(3\pi(R-R_0))\sin(3\pi Z)\sin(3\varphi)\sin(3\pi t)/5;\\ \psi(R,Z,\varphi,t) &:= \phi(R,Z,\varphi,t)=\gamma_\phi\\ A_\parallel(R,Z,\varphi,t) &:= \beta\sin(4\pi(R-R_0))\sin(4\pi Z)\sin(4\varphi)\sin(4\pi t)/4;\end{aligned}$$

We choose circular flux surfaces of the form

$$\psi_p(R,Z) := 0.5((R-R_0)^2 + Z^2), \quad I_p(R,Z) := I_0$$

with $R_0 = 10$ and $I_0 = 20$ and a simulation box $[R_0 - a, R_0 + a] \times [-a, a] \times [0, 2\pi]$. We then symbolically compute (with the help of Mathematica) source terms that we insert to the right hand side of the corresponding equation in code (`manufactured.h`) and simulate from $t = 0...10^{-3}$. By comparing the numerical solution to the manufactured one we can observe the convergence of our numerical methods. Note that in order to better distinguish the convergence of the DG discretized terms from our parallel derivative we can selectively choose to only activate perpendicular (including $A_\parallel$ terms) or parallel terms (those that involve derivatives along $\hat{\boldsymbol{b}}$).

Unfortunately, we were unable to find a closed solution for the energy integrals with the above fields.

3 Numerical methods

discontinuous Galerkin on structured grid

Term	Method	Description
coordinate system	Cylindrical	equidistant discretization of $[R_{\min}, R_{\max}] \times [Z_{\min}, Z_{\max}] \times [0, 2\pi]$ (Eq. (91), equal number of Gaussian nodes in R and Z, equidistant planes in φ with one Gaussian node
Advection terms	direct DG	DG approximation with centered flux of derivatives
Elliptic terms	local DG	The local DG approximation with centered flux
Helmholtz and Elliptic matrix inversions	multigrid/ conjugate gradient	Use previous two solutions to extrapolate initial guess and $1/\chi$ as preconditioner
Parallel derivatives	regular FCI	cf. [6, 8]. All terms use the direct centered difference, which turns out is best at keeping the numerical flux-surface leakage in $\left\langle \boldsymbol{\nabla} \cdot (\hat{\boldsymbol{b}} N U_\parallel) \right\rangle$ to a minimum, even though it is not exactly zero like in the adjoint discretizations, which in turn are unusable because they do not reliably converge. There seems to be no benefit in using the grid refinement technique except when field-aligning the initial condition.
time	Multistep "Karniadakis"	
explicit	Multistep "Karniadakis"	3rd order explicit

implicit	Multistep	"Karniadakis"	2nd order implicit, contains perp. Diffusion and Resistive terms. In every iteration of the implicit inversion we need to solve for $A_{\parallel}$

4 Usage

Compilation:
`make feltor device={gpu,omp}` Compile `feltor.cu` (only shared memory)
`make feltor_hpc device={gpu,omp}` Compile `feltor_hpc.cu` for shared memory system. Needs *serial netcdf*
`make feltor_mpi device={gpu,omp,skl,knl}` Compile `feltor_hpc.cu` for distributed memory systems. Also needs *serial netcdf*
Usage:

```
./feltor_hpc input.json geometry.json output.nc [initial.nc]
echo npx npy npz | mpirun -n np ./feltor_mpi input.json geometry.json output.nc [initial.nc]
./feltor input.json geometry.json
```

The programs `feltor_hpc.cu` and `feltor.cu` expect two input files `input.json` and `geometry.json`, described in Sections 4.1 and 4.2. The first is for the physical and numerical parameters of the model equations while the latter describes the Solov'ev equilibrium. The program `feltor.cu` plots the results directly to the screen using `glfw3`. The program `feltor_hpc.cu` writes results into the output file `output.nc`. The output file is described in Section 4.3. The optional file `initial.nc` can be used to initialize a simulation from an existing file. This behavior is described in Section 4.4. Both programs write unstructured human readable performance information of the running simulation to `std::cout`.

Note that when compiled for mpi, the program `feltor_hpc.cu` expects the partition of the total number of processes np into the three directions x, y and z as an input from the command line. Make sure that `npx*npy*npz==np` and that they evenly divide the number of grid points in the respective direction! The number of stages in the multigrid algorithm and the compression parameters further restrict this choice. Also note that the number of processes in a direction must not equal the number of grid points in that direction!

4.1 Input file structure

Input file format: json

Name	Type	Example	Default	Description
n	integer	3	-	Number of Gaussian nodes in R and Z (we practically always take 3)
Nx	integer	52	-	Number of grid points in R (increase if your simulations crash)
Ny	integer	52	-	Number of grid points in Z (increase if your simulations crash)

Nz	integer	16	-	Number of grid points in φ (determines dt since parallel velocity dominates timestep)
dt	integer	1e-2	-	time stepsize in units of c_s/ρ_s
compression	integer[2]	[2,2]	[1,1]	Compress output file by reducing points in x and y (pojecting the polynomials onto a coarser grid): output contains n*Nx/c[0] points in x, (has to divde Nx evenly), and n*Ny/c[1] points in y, (has to divde Ny evenly). 2 or 3 are reasonable values.
inner_loop	integer	2	1	Number of time steps between updates to the time integrated quantities. (Although the diagnostics is quite fast sometimes you need to amortize the time spent on it). Note that integrating selected quantities in time during the simulation is how we maintain the time-resolution in the file output (cf. 4.3). Choose as low as you can get away with (between 1 and 10).
itstp	integer	2	-	`inner_loop*itstp` is the number of timesteps between file outputs (2d and 3d quantities); Note that 1d and 0d quantities can only be computed post-simulation since we can't compute flux-integrals in parallel in MPI.
maxout	integer	10	-	Total Number of fields outputs excluding first (The total number of time steps is `maxout·itstp·inner_loop`) If you want to let the simulation run for a certain time instead just choose this parameter very large and let the simulation hit the time-limit.
eps_time	float	1e-7	-	Tolerance for solver for implicit part in time-stepper (if too low, you'll see oscillations in $u_{\parallel,e}$ and/or ϕ)
rtol	float	1e-6	-	Tolerance of adaptive time-stepper. (Ignored in Multistep)
stages	integer	3	3	number of stages in multigrid, $2^{\text{stages-1}}$ has to evenly divide both N_x and N_y

eps_pol	float[stages]	[1e-6,1,1]	-	The first number is the tolerance for residual of the inversion of polarisation and induction Eq.. The second number is a multiplicative factor for the accuracy on the second grid in a multigrid scheme, the third for the third grid and so on. (i.e. $\varepsilon_0\varepsilon_i$ is the accuracy on the i-th grid) Tuning those factors is a major performance tuning oppourtunity!! For saturated turbulence the suggested values are [1e-6, 2000, 100].
jumpfactor	float	1	1	Jumpfactor $\in [0.01, 1]$ in the local DG method for the elliptic terms. (Don't touch unless you know what you're doing.
eps_gamma	float	1e-6	-	Tolerance for Γ_1
FCI	dict			Parameters for Flux coordinate independent approach
refine	integer[2]	[1,1]	[1,1]	refinement factor in FCI approach in R- and Z-direction. We use [1,1], higher values take more time, but possibly stabilize the simulation.
rk4eps	float	1e-6	1e-6	Accuracy of fieldline integrator in FCI. The default is reasonable.
periodify	bool	true	true	Indicate if flux function is periodified beyond grid boundaries such that the contours are perpendicular to the boundaries. This is not entirely consistent but works better for small toroidal resolution
mu	float	-0.000272121	-	$\mu_e = -m_e/m_i$. One of $\{-0.000544617, -0.000272121, -0.000181372\}$
tau	float	1	-	$\tau = T_i/T_e$
beta	float	5e-6	0	Plasma beta $5 \cdot 10^{-6}$ (TJK), $4 \cdot 10^{-3}$ (Compass), If 0, then the model is electrostatic
nu_perp	float	1e-3	-	perpendicular viscosity $\nu_\perp$, increase this or the resolution if you see vertical or horizontal oscillations (likely from the advection terms) in your simulation box, decrease if it dampens all instabilities

perp_diff	string	"viscous"	"viscous"	"viscous": $\Lambda_\perp \propto \nu_\perp \Delta_\perp$, "hyperviscous": $\Lambda_\perp \propto -\nu_\perp \Delta_\perp^2$
resistivity	float	1e-4	-	parallel resistivity parameter Eq. (87)
curvmode	string	"low beta"	"toroidal"	curvature mode ("low beta", "true": no approximation - requires significantly more resolution in Nz, "toroidal": toroidal field approx - elliptic equation does not need communication in z)
symmetric	bool	false	false	If true, initialize all quantities symmetric in φ (effectively reducing the problem to 2d). The input N_z is used to construct the parallel derivatives and then overwritten to $N_z \equiv 1$.
bc	dict			Boundary conditions (note that $A_\parallel$ has the same bc as $U_\parallel$) ...
density	char[2]	[DIR,DIR]	-	boundary conditions in x and y for n_e and N_i, DIR (density 1 on boundary) means both convective and diffusive outflow while NEU (gradient 0) means no outflow by diffusion
velocity	char[2]	[NEU,NEU]	-	boundary conditions in x and y for $u_{\parallel,e}$ and $U_{\parallel,i}$ and $A_\parallel$, DIR is in general not very stable, NEU works better
potential	char[2]	[DIR,DIR]	-	boundary conditions in x and y for ϕ and ψ, DIR means that the $v_{E,\perp} = 0$ on the boundary (i.e. no outflow by $\boldsymbol{E}\times\boldsymbol{B}$ drift), NEU can can have a detrimental effect on timestep
box	dict			Bounding box
scaleR	float[2]	[1.1,1.1]	[1.05,1.05]	$[\varepsilon_{R-}, \varepsilon_{R+}]$ scale left and right boundary in units of a Eq. (91)
scaleZ	float[2]	[1.2,1.1]	[1.05,1.05]	$\varepsilon_{Z-}, \varepsilon_{Z+}$ scale lower and upper boundary in units of ae Eq. (91)

initne	string	”turbulence”	”blob”	initial condition for the perturbation $\tilde{n}$ in (94). ”zonal” (Eq. (100)), ”blob” = blob simulations (several rounds field-aligned), ”straight blob” = straight blob simulation(1 round fieldaligned), ”turbulence” = turbulence simulations (1 round fieldaligned, Eq. (99)) ”turbulence on gaussian” = Gaussian bg. profile with turbulence perturbation Eq. (102) See the file `init.h` to add your own custom condition.
initphi	string	”zero”	”balance”	(ignored if $\tau_i = 0$, then $\phi = 0$) initial condition for ϕ and thus N_i (Eq. (96): ”zero” : $\phi = 0$, vanishing electric potential, ”balance”: ExB vorticity equals ion diamagnetic vorticity
amplitude	float	0.01	-	amplitude A of initial perturbation (blob, turbulent bath or zonal flow)
sigma	float	2	-	Gaussian variance in units of ρ_s
posX	float	0.3	-	Gaussian R-position in units of a
posY	float	0.0	-	Gaussian Z-position in units of a
sigma_z	float	0.25	-	toroidal variance in units of π of the fieldline-following initialization
k_psi	float	0	-	zonal mode wave number (only for ”zonal” initial condition)
profile	Dict			Density profile
amp	float	4	0	Profile amplitude $\triangle n_{peak}$ in Eq. (95) and Eq. (102)
alpha	float	0.2	0.2	Transition width α_p in the Heaviside at the separatrix (must not be zero - even if amp is zero - it is also used for the perturbation)
source	dict			Density source, cf. the output `sne_tt_ifs` in `feltordiag` (or `SourceProfile_ifs` in `geometry_diag`) to see how much mass the source with the parameters below generates and compare to `jsne_tt_fsa` to see how much mass is lost.
rate	float	0	0	profile source rate ω_s in Eq. (103).

type	string	”profile”	”profile”	The type of source to use: ”profile” the source is multiplied by $(n_{prof} - n)$ to relax to the initial profile Eq. (103); ”influx” the source has a constant source rate Eq. (104), ”torpex”: Torpex inspired source profile Eq. (106), ”gaussian”: Gaussian shaped source profile - uses `posX`, `posY` and `sigma`, See the file `init.h` to add your own custom source.
boundary	float	0.2	0.5	Source region boundary $\rho_{p,b}$: yields in Eq. (103) and Eq. (104)
alpha	float	0.2	0.2	Transition width α_p in the Heaviside in the density Eq. (95) (with $\rho_{p,b} = 0$ and source profiles Eq. (103) (should be small but cannot be too small if $\tau_i > 0$ else $\Delta_\perp n_e$ explodes, must not be zero)
damping	dict			magnetic and density damping region
modifier	string	”sol_pfr”	”sol_pfr”	One of “none”, “heaviside” or “sol_pfr”
rate	float	0	0	Friction coefficient ω_d in density and velocity damping Eq. (108)
boundary	float[2]	[1.2,0.8]	[1.2,0.8]	Modification region boundary $\rho_{p,b}$: yields $\psi_0 = (1 - \rho_{p,b}^2)\psi_{p,O}$ in Eq. (79).
alpha	float[2]	[0.25,0.25]	0	Transition width α_p: yields $\alpha = -2\rho_{p,b}\alpha_p + \alpha_p^2)\psi_{p,O}$ for the Heaviside in the modified ψ_p function (79). If zero, then we do not modify the magnetic field and damping is ignored.

4.2 Geometry file structure

File format: json

The file structure of the geometry file depends on which expansion for ψ_p is chosen Eq. (16) or Eq. (18). A solovev magnetic field equilibrium

Name	Type	Example	Default	Description
A	float	0	0	Solovev parameter in Eq. (16)
c	float[12]	-	-	Solovev coefficients in Eq. (16)

A polynomial magnetic field equilibrium

Name	Type	Example	Default	Description
M	float	1	1	Number of polynomial coefficients in R in Eq. (18)
N	float	1	1	Number of polynomial coefficients in Z in Eq. (18)
c	float[MN]	-	-	Polynomial coefficients in Eq. (18)

In addition both files must contain

Name	Type	Example	Default	Description
PP	float	1	1	Prefactor $\mathcal{P}_\psi$ for ψ_p
PI	float	1	1	Prefactor $\mathcal{P}_I$ for I
R_0	float	-	-	Major radius R_0 in units of ρ_s (This is the only geometry quantity to change if ρ_s changes)
elongation	float	1	-	Elongation e, used in determining the box size Eq. (91) and the initial guess for the location of the X-point $Z_X = -1.1ea$
triangularity	float	0	-	Triangularity δ, used in the initial guess for the location of the X-point $R_X = R_0 - 1.1\delta a$
inverseaspectratio	float	0.16667	-	minor to major radius a/R_0 (used to compute a from R_0)
equilibrium	string	polynomial	solovev	Tells the magnetic field generation which type of expansion to use for the flux function
description	string	standardX	standardX	Tells the magnetic field modifier where to look for the SOL and PFR regions

4.3 Output

Output file format: netcdf-4/hdf5; A *coordinate variable (Coord. Var.)* is a Dataset with the same name as a dimension. We follow **CF Conventions CF-1.7** http://cfconventions.org/Data/cf-conventions/cf-conventions-1.7/cf-conventions.html and write according attributes into the file. The command `ncdump -h output.nc` gives a full list of what a file contains. Here, we list the content without attributes since the internal netcdf information does not display equations.

Name	Type	Dimension	Description
inputfile	text attribute	-	verbose input file as a string (valid JSON, C-style comments are allowed but discarded)

geomfile	text attribute	-	verbose geometry input file as a string (valid JSON, C-style comments are allowed but discarded)
x	Coord. Var.	1 (x)	R-coordinate (computational space, compressed size: nN_x/c_x)
y	Coord. Var.	1 (y)	Z-coordinate (computational space, compressed size: nN_y/c_y)
z	Coord. Var.	1 (z)	φ-coordinate (computational space, size: N_z)
time	Coord. Var.	1 (time)	time at which fields are written (variable size: maxout$+1$, dimension size: unlimited)
xc	Dataset	3 (z,y,x)	Cartesian x-coordinate $x = R\sin(\varphi)$
yc	Dataset	3 (z,y,x)	Cartesian y-coordinate $y = R\cos(\varphi)$
zc	Dataset	3 (z,y,x)	Cartesian z-coordinate $z = Z$
Psip	Dataset	3 (z,y,x)	Flux function $\psi_p(R, Z)$
Nprof	Dataset	3 (z,y,x)	Density profile n_{prof} used in the forcing source
Source	Dataset	3 (z,y,x)	Source profile S_{prof}
BR	Dataset	3 (z,y,x)	Contravariant magnetic field component B^R
BZ	Dataset	3 (z,y,x)	Contravariant magnetic field component B^Z
BP	Dataset	3 (z,y,x)	Contravariant magnetic field component B^φ
electrons	Dataset	4 (time, z, y, x)	electron density n_e
ions	Dataset	4 (time, z, y, x)	ion density N_i
Ue	Dataset	4 (time, z, y, x)	electron velocity $u_{\parallel,e}$
Ui	Dataset	4 (time, z, y, x)	ion velocity $U_{\parallel,i}$
potential	Dataset	4 (time, z, y, x)	electric potential ϕ
induction	Dataset	4 (time, z, y, x)	parallel vector potential $A_\parallel$
X_2d	Dataset	3 (time,y,x)	Selected plane $X(\varphi = 0)$
X_ta2d	Dataset	3 (time,y,x)	Toroidal average $\langle X\rangle_\varphi$ Eq. (48)
Y_tt_2d	Dataset	3 (time,y,x)	Time integrated (between two outputs, Simpson's rule) selected plane $\int_{t_0}^{t_1} \mathrm{d}t Y(\varphi = 0)$ where $t_1 - t_0$ =`dt*inner_loop*itstp` and `itstp` is the number of discretization points
Y_tt_ta2d	Dataset	3 (time,y,x)	Time integrated (between two outputs, Simpson's rule) toroidal average (Eq. (48)) $\int_{t_0}^{t_1} \mathrm{d}t \langle Y\rangle_\varphi$ where $t_1 - t_0$ =`dt*inner_loop*itstp` and `itstp` is the number of discretization points

where X and Y_tt represent the quantities described in the tables in previous sections and the miscellaneous quantities

Name	Equation	Name	Equation
vorticity	$-\Delta_\perp \phi$	apar_vorticity	$-\Delta_\perp A_\parallel$
dssue	$\nabla_\parallel^2 u_{\parallel,e}$	lperpinv	$L_\perp^{-1} := \|\boldsymbol{\nabla}_\perp n_e\|/n_e$
perpaligned	$(\boldsymbol{\nabla}_\perp n_e)^2/n_e$	lparallelinv	$L_\parallel^{-1} := \|\nabla_\parallel n_e\|/n_e$
aligned	$(\nabla_\parallel n_e)^2/n_e$	ne2	n_e^2
phi2	ϕ^2	nephi	$n_e\phi$

The computation time spent on diagnostics is negligible if `inner_loop` parameter is greater than 1. Also remember that the X and Y fields are all two-dimensional, which takes up much less disk-space than three-dimensional fields.

4.4 Restart file

The program `feltor_hpc.cu` has the possibility to initialize time and the fields with the results of a previous simulation. In particular, this feature is motivated by chain jobs on a cluster (see e.g. the –dependency option in SLURM). This behaviour is enabled by giving an additional file `initial.nc` to the command line. In this case the `initne` and `initphi` parameters of the input file are ignored. Instead, the fields `electrons, ions, Ue, Ui, induction` at the latest timestep are read from the given file to initialize the simulation. Note that to enable a loss-less continuation of the simulation we output special restart fields into the output file that in contrast to the other fields are not compressed. Apart from that the behaviour of the program is unchanged i.e. the magnetic field, profiles, resolutions, etc. are all taken from the regular input files. This means that the user must take care that these are consistent with the paramters in the existing `initial.nc` file. Also note that we try to discourage appending new results to an exisiting file directly, because if for some reason the cluster crashes and the file is corrupted the whole simulation is lost. It is safer to just merge files afterwards with for example

```
ncrcat output1.nc output2.nc output.nc
```

5 Diagnostics

`feltor/src/feltor/feltordiag.cu` reads one or more previously generated simulation file(s) `input0.nc ... inputN.nc` described in Section 4.3 and writes into a single second output file `output.nc` described as follows.

Compilation

```
make feltordiag device={gpu,omp}
```

Usage

```
./feltordiag input0.nc ... inputN.nc output.nc
```

Output file format: netcdf-4/hdf5, Conventions: CF-1.7; A *coordinate variable (Coord. Var.)* is a Dataset with the same name as a dimension.

Name	Type	Dimension	Description
inputfile	text attribute	-	verbose input file as a string (valid JSON, C-style comments are allowed but discarded)
geomfile	text attribute	-	verbose geometry input file as a string (valid JSON, C-style comments are allowed but discarded)
x	Coord. Var.	1 (x)	R-coordinate (computational space, compressed size: nN_x/c_x)
y	Coord. Var.	1 (y)	Z-coordinate (computational space, compressed size: nN_y/c_y)
psi	Coord. Var.	1 (psi)	ψ_p-coordinate (default size: $3 \cdot 64$)
time	Coord. Var.	1 (time)	time at which fields are written (variable size: maxout$+1$, dimension size: unlimited)
dvdpsip	Dataset	1 (psi)	$\mathrm{d}v/\mathrm{d}\psi_p$
psi_vol	Dataset	1 (psi)	The volume enclosed by the flux surfaces $v(\psi_p) = \int_{\psi_p} \mathrm{dV}$
psi_area	Dataset	1 (psi)	The area of the flux surfaces $A(\psi_p) = 2\pi \int_\Omega \lvert\boldsymbol{\nabla}\psi_p\rvert \delta(\psi_p - \psi_{p0}) H(Z - Z_X) R \mathrm{d}R\mathrm{d}Z$
q-profile	Dataset	1 (psi)	The safety factor $q(\psi_p)$ (62) using direct integration (accurate but unavailable outside separatrix)
psi_psi	Dataset	1 (psi)	explicit ψ_p values; Same as psi
psit1d	Dataset	1 (psi)	Toroidal flux (integrated q-profile) $\psi_t = \int^{\psi_p} \mathrm{d}\psi_p q(\psi_p)$
rho	Dataset	1 (psi)	Transformed flux label $\rho := 1 - \psi_p/\psi_{p,O}$
rho_p	Dataset	1 (psi)	poloidal flux label $\rho_p := \sqrt{1 - \psi_p/\psi_{p,O}}$
rho_t	Dataset	1 (psi)	Toroidal flux label $\rho_t := \sqrt{\psi_t/\psi_{t,\mathrm{sep}}}$ (is similar to ρ in the edge but ρ_t is nicer in the core domain, because equidistant ρ_t make more equidistant flux-surfaces)
Z_fluc2d	Dataset	3 (time,y,x)	Fluctuation level on selected plane ($\varphi = 0$) $\delta Z := Z(R, Z, 0) - \langle Z \rangle (R, Z)$
Z_fsa2d	Dataset	3 (time, y,x)	Flux surface average $\langle Z \rangle$ interpolated onto 2d plane Eq. (52)
Z_cta2d	Dataset	3 (time, y,x)	Convoluted toroidal average Eq. (73)
Z_fsa	Dataset	2 (time, psi)	Flux surface average $\langle Z \rangle$ Eq. (52)

Z_std_fsa	Dataset	2 (time, psi)	Standard deviation of flux surface average on outboard midplane $\sqrt{\langle(\delta Z)^2\rangle}$
Z_ifs	Dataset	2 (time, psi)	Volume integrated flux surface average $\int \mathrm{d}v \langle Z\rangle$ unless Z is a current, then it is the volume derived flux-surface average $\partial_v \langle Z\rangle$
Z_ifs_lcfs	Dataset	1 (time)	Volume integrated flux surface average evaluated on last closed flux surface $\int_0^{v(0)} \mathrm{d}v \langle Z\rangle$ unless Z is a current, then it is the fsa evaluated $\langle j_v\rangle(0)$
Z_ifs_norm	Dataset	1 (time)	Volume integrated square flux surface average $\sqrt{\int \mathrm{d}v \langle Z\rangle^2}$, unless Z is a current, then it is the square derivative of the flux surface average $\sqrt{\int \mathrm{d}v(\partial_v \langle j^v\rangle)^2}$

where Z $\in$ {X, Y_tt} Note that feltoridag converts all jsX quantities into jvX by multiplying $\mathrm{d}v/\mathrm{d}\psi_p$ in the sense that $\boldsymbol{j}\cdot\boldsymbol{\nabla}v = \boldsymbol{j}\cdot\boldsymbol{\nabla}\psi_p \mathrm{d}v/\mathrm{d}\psi_p$. The parameters used for the X-point flux-aligned grid construction are $f_x = 1/8$, $f_y = 0$, $n_\psi = 3$, $N_\zeta = 64$ and $N_\eta = 640$ and the constant monitor metric.

We also have a useful geometry diagnostic program: `feltor/inc/geometries/geometry_diag.cu` reads either a previously generated simulation file `input.nc` or the input json files `input.json` and `geometry.json` and writes an output file `diag_geometry.nc` as

Compilation

```
make geometry_diag device={gpu,omp}
```

Usage

```
./geometry_diag input.json geometry.json diag_geometry.nc
```

The program outputs a host of static 1d, 2d and 3d geometric quantities. The output file is for example useful in connection with the "Group Datasets" filter in paraview, which merges Datasets from different files into one using shallow copy only.

6 Error conditions

All previously mentioned codes can crash for various reasons. Here, we list and describe situations, which generally may lead to program termination

Error condition	Handling
An input file does not exist or is otherwise invalid	Program terminates with an error message to `std::cerr`. `feltordiag.cu` writes an error to `std::cerr` and continues with the next input file.
An input netcdf file misses a required field	Program terminates with a NetCDF error message to `std::cerr`

No write permission for the output file location	Program terminates with an error message to `std::cerr`
An input Json file misses a key or contains a typo in a key	The programs `feltor.cu` and `feltor_hpc.cu` will exit with an error message. (The reason why we do not silently use the default value is that the danger of wasting valuable computing time on the cluster due to a typo is bigger than the added convenience. We want to be sure that the program does what the user wants). The other programs just issue warnings if a key is not found and use a default value which is 0 if not otherwise specified.
An input Json file has an invalid value, e.g. a typo in a string value	Invalid values lead to termination with an error message to `std::cerr`, once and if program tries to use the value
Number of processes in x, y and z direction does not match total number of Processes	Program terminates with an error message to `std::cerr`.
2^{s-1} or c_x or c_y does not evenly divide N_x and N_y, where s is the number of stages in the multigrid algorithm.	Program terminates on thrown error. Make sure the numbers add up.
Number of processes in x, y and z direction does not evenly divide or is greater or equal $N_x/2^{s-1}$, $N_y/2^{s-1}$ and N_z, where s is the number of stages in the multigrid algorithm.	Program terminates on failed assert
An MPI error occurs	Program crashes horribly printing cryptic error messages (stack trace) to `std::cerr`
A numerical instability occurs	The program terminates usually caused by a NaN exception raised. However, the cause for the instability has to be determined inspecting the last output in the output file.
large fieldaligned oscillations in $u_{\parallel,e}$ paired with instability in the edge of the box	Apply damping region
Perpendicular grid oscillations in $u_{\parallel,e}$ and $\Delta_\perp\phi$ in the damping region, symmetric in φ	Increase damping $alpha$, increase damping boundary, make the box larger/smaller, increasing DS refinement might help.
Spike in $u_{\parallel,e}$ shortly after simulation start	Increase $\nu_\perp$, increase N_x, N_y, decrease perturbation amplitude
Grid oscillations far away from the edge	Probably caused by the perpendicular transport that goes unstable. Increase $\nu_\perp$ and/or N_x, N_y. Increasing DS refinement might also help.

Oscillations where fieldlines intersect the wall	Caused by boundary conditions in FCI method and necessarily underresolved toroidal direction. Increase N_z, decrease N_x, N_y or decrease q value by decreasing $\mathcal{P}_\psi$ in geometry input file

References

[1] S. Braginskii. Transport processes in a plasma. *Rev. Plasma Phys.*, 1:205, 21965. URL http://people.hao.ucar.edu/judge/homepage/PHSX515/fall2012/Braginskii1965.pdf.

[2] A. J. Cerfon and J. P. Freidberg. "one size fits all" analytic solutions to the grad-shafranov equation. *Phys. Plasmas*, 17(3):032502, 2010. doi:10.1063/1.3328818.

[3] A. J. Cerfon and M. O'Neil. Exact axisymmetric taylor states for shaped plasmas. *Phys. Plasmas*, 21(6):064501, 2014. doi:10.1063/1.4881466.

[4] W. D'haeseleer, W. Hitchon, J. Callen, and J. Shohet. *Flux Coordinates and Magnetic Field Structure*. Springer Series in Computational Physics. Springer-Verlag, 1991.

[5] M. Held. *Full-F gyro-fluid modelling of the tokamak edge and scrape-off layer*. PhD thesis, University of Innsbruck, 2016. URL http://resolver.obvsg.at/urn:nbn:at:at-ubi:1-6853.

[6] M. Held, M. Wiesenberger, and A. Stegmeir. Three discontinuous galerkin schemes for the anisotropic heat conduction equation on non-aligned grids. *Computer Physics Communications*, 199:29–39, 2016. doi:10.1016/j.cpc.2015.10.009.

[7] J. Madsen, V. Naulin, A. H. Nielsen, and J. J. Rasmussen. Collisional transport across the magnetic field in drift-fluid models. *Physics of Plasmas*, 23:032306, 2016. doi:10.1063/1.4943199.

[8] A. Stegmeir, O. Maj, D. Coster, K. Lackner, M. Held, and M. Wiesenberger. Advances in the flux-coordinate independent approach. *Comput. Phys. Commun.*, 213:111 – 121, 2017. doi:10.1016/j.cpc.2016.12.014.

[9] M. Wiesenberger. *Gyrofluid computations of filament dynamics in tokamak scrape-off layers*. PhD thesis, University of Innsbruck, 2014. URL http://resolver.obvsg.at/urn:nbn:at:at-ubi:1-1799.

[10] M. Wiesenberger, M. Held, L. Einkemmer, and A. Kendl. Streamline integration as a method for structured grid generation in x-point geometry. *J. Comput. Phys.*, 373:370–384, 2018. doi:10.1016/j.jcp.2018.07.007.