GENE-X: A full-f gyrokinetic turbulence code based on the flux-coordinate independent approach

Dominik Michels[a,*], Andreas Stegmeir[a], Philipp Ulbl[a], Denis Jarema[a], Frank Jenko[a,b]

[a] *Max-Planck-Institut für Plasmaphysik, Boltzmannstrasse 2, 85748 Garching, Germany*
[b] *University of Texas at Austin, Austin, TX 78712, USA*

Abstract

Understanding and predicting plasma turbulence in the scrape-off layer of a magnetic confinement fusion device is a key open problem in modern plasma physics. The transitional region between the core and scrape-off layer poses a difficult problem for turbulence simulations. The poloidal magnetic field vanishes at the X-point of a fusion device, which introduces a coordinate singularity in the commonly used field-aligned coordinates. In the present work, we present a full-f gyrokinetic code based on a locally field-aligned coordinate system that is flux-coordinate independent and free of singularities. The coordinate system, as well as the equations and numerical methods are described. In addition, careful numerical and physical verifications in closed magnetic flux surfaces are included.

Keywords: Plasma turbulence, gyrokinetics, flux-coordinate independent (FCI)

1. Introduction

Understanding and predicting plasma turbulence in the edge and scrape-off layer (SOL) of magnetic confinement fusion devices plays an important role for the achievement of a sustainable fusion energy source. The plasma boundary is in contact with its material surroundings which limits the lifetime of the wall and impacts the level of impurity accumulation in the core. Moreover, many parameters of the plasma core are influenced by edge and scrape-off layer phenomena [1, 2].

Plasma turbulence in the core of a fusion device is commonly modeled within gyrokinetic theory [3] and there exist multiple codes featuring different numerical techniques in the community [4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

Extending these simulations to the plasma edge and scrape-off layer poses significant challenges. First of all, fluctuations of the plasma in the scrape-off layer are known to be much larger than in the core. As such, nonlinear effects originating from the coupling between fluctuations become important - which makes a so called full-f treatment of the gyrokinetic equations necessary. Moreover, the poloidal magnetic field vanishes at the X-point of a magnetic confinement fusion device, introducing a coordinate singularity in the commonly used field-aligned coordinates.

In recent years, there have been many investigations into scrape-off layer and edge turbulence with fluid codes [14, 15, 16, 17, 18]. However, it remains an open question whether the assumptions behind the fluid approximation are valid in the plasma edge. As a step forward, it is important to enable the use of gyrokinetic models for scrape-off layer turbulence.

Several projects aim at simulating gyrokinetic turbulence in the edge and scrape-off layer, namely `XGC` [19], `COGENT` [20], `GKEYLL` [21], `GENE` [22], `GYSELA` [23] and `PICLS` [24]. They differ by the numerical scheme used, the model implemented, and their ability to treat geometries complexities. `XGC` and `PICLS` use the so called particle-in-cell (PIC) method. In these codes marker particles are integrated along the characteristics of the gyrokinetic Vlasov equation. While these algorithms allow for a straightforward generalization to X-point geometries, they contain statistical noise which may require a large number of marker particles to resolve the full-f distribution function in edge and scrape-off layer plasmas. The other projects mentioned employ a continuum method and solve for the gyrokinetic Vlasov equation directly. `GKEYLL` features an energy-conservative discontinuous Galerkin discretization and has been used successfully to simulate open field line regions in linear and helical devices [25]. `GENE` uses a combination of a shock-stable finite volume and a finite difference scheme, and `GYSELA` a semi-Lagrangian method.

To allow for continuum turbulence simulations in X-point geometries, locally field-aligned methods based on the flux-coordinate independent approach (FCI) [26, 15] have been developed and implemented in fluid codes [15, 27, 18, 28]. Along with fully non-aligned methods [29], FCI methods have been established as an effective and efficient numerical method for studying edge and SOL turbulence in the fluid community. The use of FCI has enabled, for example, a recent simulation of turbulence in the edge and SOL of ASDEX Upgrade scale with realistic magnetic geometry [30]. Within the gyrokinetic community, locally field-aligned methods have recently been implemented in the gyrokinetic finite-volume code `COGENT` [31] for simulations in X-point geometry.

In this work, we present a new full-f gyrokinetic continuum

*Corresponding author
Email address: `dominik.michels@ipp.mpg.de` (Dominik Michels)

code, GENE-X, implementing a locally field-aligned coordinate system following the FCI approach. The implementation is based on the fluid code GRILLIX [15, 32] and employs a finite difference discretization scheme on unstructured, locally Cartesian grids. These grids allow for a high flexibility regarding the geometry while ensuring good numerical properties and a high computational efficiency. The approach is free of coordinate singularities and capable of handling simulations ranging from the magnetic axis, across the separatrix and into the scrape-off layer. Furthermore, the existence of magnetic flux surfaces is not required and it is possible, in principle, to extend the calculation to stochastic regions and 3D geometries [28]. In this paper, we will focus on simulations performed on closed magnetic flux surfaces, to allow for careful verification of the code.

The present work is structured as follows. We start by deriving the gyrokinetic model used in our simulations in Section 2. In Section 3 we introduce the locally field-aligned coordinate system, derive metric coefficients and express the differential operators in the gyrokinetic Vlasov equation in the new coordinates. We discuss the discretization scheme used in Section 4. Finally, in Section 5 we verify the numerical scheme and the implementation.

2. Gyrokinetic Model

We start from the particle Lagrangian in centimeter-gram-second (CGS) units given by [33, 34]

$$L_p(\boldsymbol{Z}, \dot{\boldsymbol{Z}}) = \left(\frac{q_\sigma}{c}\boldsymbol{A} + p_\parallel \boldsymbol{b}\right)\cdot \dot{\boldsymbol{R}} + \frac{m_\sigma c}{q_\sigma}\mu\dot{\theta} - H\,. \tag{1}$$

m_σ represents the mass and q_σ the charge of species σ. The particle coordinates are denoted by $Z = (\boldsymbol{R}, p_\parallel, \mu, \theta)$, where $\boldsymbol{R}$ represents the gyrocenter position [3], $p_\parallel$ the canonical parallel momentum, μ the gyration magnetic moment, and θ the gyrophase angle.

The background magnetic field $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ is represented by the vector potential $\boldsymbol{A}$. $\boldsymbol{b}$ denotes the magnetic field unit vector. From the particle Lagrangian the total gyrokinetic Lagrangian can be derived [35]. It yields

$$L = \sum_\sigma \int f(Z_0, t_0) L_p(Z(Z_0, t_0; t), \dot{Z}(Z_0, t_0; t), t)\,\mathrm{d}W_0\,\mathrm{d}V_0 + \int \frac{E^2 - B_\perp^2}{8\pi}\,\mathrm{d}V, \tag{2}$$

where $\mathrm{d}W = (2\pi/m_\sigma)B_\parallel^*\,\mathrm{d}v_\parallel\,\mathrm{d}\mu$ denotes the volume element in velocity space, $\mathrm{d}V$ the volume element in real space and $\boldsymbol{B}_\perp = \boldsymbol{\nabla}_\perp A_\parallel$ the fluctuation of the magnetic field. The guiding center magnetic field $\boldsymbol{B}^*$ and its parallel component $B_\parallel^*$ are defined as

$$\boldsymbol{B}^* = \boldsymbol{B} + \frac{m_\sigma c}{q_\sigma} v_\parallel \nabla \times \boldsymbol{b}, \tag{3}$$

$$B_\parallel^* = B + \frac{m_\sigma c}{q_\sigma} v_\parallel \boldsymbol{b}\cdot(\nabla \times \boldsymbol{b})\,. \tag{4}$$

We simplify the gyrokinetic model by neglecting electromagnetic fluctuations, setting the magnetic potential $A_\parallel = 0$. This implies that the parallel velocity $v_\parallel = \frac{\partial H}{\partial p_\parallel}$ is given by $v_\parallel = p_\parallel/m_\sigma$ [33] and the magnetic field perturbation $\boldsymbol{B}_\perp$ vanishes. Furthermore, we employ a long wavelength limit such that gyroaverages reduce to the identity. With these simplifications the particle Hamiltonian reads

$$H = \frac{p_\parallel^2}{2m_\sigma} + \mu B + q_\sigma\phi - \frac{m_\sigma c^2}{2B^2}|\nabla_\perp\phi|^2\,. \tag{5}$$

In this work we employ the quasi-neutrality approximation and neglect the E^2 term in Eq. (2) [35]. We linearize the quasi-neutrality equation by requiring that the field part of the Hamiltonian $H_2 = -\frac{m_\sigma c^2}{2B^2}|\nabla_\perp\phi|^2$ contributes to the field equation only [33, 35]. This implies that we replace the term $H_2 f_\sigma$ with $H_2 f_{0\sigma}$, where $f_{0\sigma}$ denotes a static initial distribution function.

The electrostatic gyrokinetic Lagrangian in the long wavelength limit, describing the implemented system, is given by

$$L = \sum_\sigma \int \left[\left(\frac{q_\sigma}{c}\boldsymbol{A} + p_\parallel\boldsymbol{b}\right)\cdot\dot{\boldsymbol{R}} + \frac{m_\sigma c}{q_\sigma}\mu\dot{\theta} - \frac{p_\parallel^2}{2m_\sigma} - \mu B - q_\sigma\phi\right] f_\sigma\,\mathrm{d}W\,\mathrm{d}V + \sum_\sigma \int \frac{m_\sigma c^2}{2B^2}|\nabla_\perp\phi|^2 f_{0\sigma}\,\mathrm{d}W\,. \tag{6}$$

2.1. The gyrokinetic Vlasov-Poisson system

The gyrokinetic Vlasov equation is derived from the Liouville theorem stating that the distribution function is constant along particle trajectories

$$\frac{\mathrm{d}f_\sigma}{\mathrm{d}t} = \frac{\partial f_\sigma}{\partial t} + \dot{\boldsymbol{R}}\cdot\nabla f_\sigma + \dot{v}_\parallel\frac{\partial f_\sigma}{\partial v_\parallel} = 0\,. \tag{7}$$

The trajectories of the gyrocenters are derived by solving the Euler-Lagrange equations [34]

$$\dot{\boldsymbol{R}} = v_\parallel\frac{\boldsymbol{B}^*}{B_\parallel^*} + \frac{c}{q_\sigma B_\parallel^*}\boldsymbol{b}\times(\mu\nabla B + q_\sigma\nabla\phi)\,, \tag{8}$$

$$\dot{v}_\parallel = -\frac{\boldsymbol{B}^*}{m_\sigma B_\parallel^*}\cdot(\mu\nabla B + q_\sigma\nabla\phi)\,. \tag{9}$$

The quasi-neutrality equation is derived by requiring that the variation of the action functional, generated by the Lagrangian, with respect to electrostatic potential vanishes. Explicit calculation yields [34]

$$-\nabla_\perp\cdot\left(\sum_\sigma \frac{m_\sigma c^2 n_{0\sigma}}{B^2}\nabla_\perp\phi\right) = \sum_\sigma q_\sigma\int f_\sigma\,\mathrm{d}W, \tag{10}$$

where $n_{0\sigma}$ is the initial density given by the Maxwellian distribution function $f_{0\sigma}$. Eqs. (7) - (10) describe the system solved in the code. Due to the derivation from a Lagrangian the system fulfills an energy conservation law. The conserved energy is [34]

$$E = \sum_\sigma \int\left(\frac{1}{2}m_\sigma v_\parallel^2 + \mu B + \frac{1}{2}q_\sigma\phi\right) f_\sigma\,\mathrm{d}V\,\mathrm{d}W\,. \tag{11}$$

After having described the gyrokinetic model and the energy theorem we continue with deriving the coordinate system for our simulations in the next section.

3. Locally field-aligned coordinates

Due to the fast dynamics along the magnetic field lines, parallel gradients are much smaller than perpendicular gradients, which makes structures in the plasma field-aligned. Multiple gyrokinetic codes present in the community, e.g. GENE [6, 13], GYRO [8], GYSELA [9], ORB5 [11], GKW [12], exploit this anisotropy by implementing a field-aligned coordinate system to reduce the resolution needed for a simulation. These coordinates are usually composed of a flux-surface label and two coordinates characterizing the position on the flux surface, one of which is the position along a given magnetic field line. The field-aligned coordinate system requires the existence of flux-surfaces and becomes ill defined at the separatrix.

For plasma edge and scrape-off layer simulations it is important to incorporate the separatrix and the X-point in the simulations. Therefore, changes to the field-aligned coordinate system are necessary. We solve this problem implementing a *locally* field-aligned coordinate system (x, y, z) following the flux-coordinate independent (FCI) approach [36, 26, 15, 32]. This coordinate system exploits the anisotropy by aligning the y coordinate to the magnetic field line while x and z coordinates are independent of magnetic flux surfaces. Due to the flux-coordinate independent nature of the x and z coordinate, numerical diffusion, arising from the fast electron streaming along the magnetic field line, can interfere with the turbulent orthogonal dynamics. For a thorough discussion of this phenomenon we refer to studies in previous work with fluid and gyrofluid models [26, 37, 27, 15, 32]. In the following section, we give a precise definition of the coordinate system and derive expressions for the differential operators used in the gyrokinetic Vlasov-Poisson system.

3.1. Coordinate transformation

Consider a cylindrical coordinate system (R, φ, Z) and a collection of poloidal planes located at φ_k. We define the locally field-aligned coordinate system in a region $[\varphi_k - \Delta\varphi, \varphi_k + \Delta\varphi]$ around the poloidal plane. We choose the poloidal range $\Delta\varphi$ of the coordinate system large enough such that the coordinate systems for different poloidal planes overlap. The collection of all coordinate systems covers the fusion device. In the following, we assume without loss of generality that $\varphi_k = 0$ because we can always rotate the embedding cylindrical coordinate system. In order to derive the coordinate transformation we choose a Cartesian reference frame (x_c, y_c, z_c) on the poloidal plane where x_c points in the direction of the cylindrical R coordinate, z_c in the direction of the cylindrical Z coordinate, and y_c out of the poloidal plane.

Let $\boldsymbol{\gamma} : u \mapsto \boldsymbol{\gamma}(u)$ be the curve of a magnetic field line starting at $\boldsymbol{\gamma}(0) = \boldsymbol{R}_0 = (x_c, 0, z_c)$ defined by the differential equation

$$\frac{\mathrm{d}\boldsymbol{\gamma}}{\mathrm{d}u} = \boldsymbol{b}(\boldsymbol{\gamma}) \,. \tag{12}$$

We define the locally field-aligned coordinate system, by providing the coordinate transformation τ from locally field-aligned to Cartesian coordinates

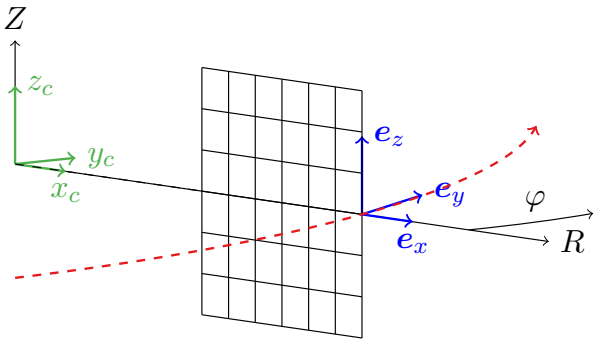

Figure 1: Figure displaying the cylindrical (R, φ, Z) coordinate system, the Cartesian (x_c, y_c, z_c) coordinate system, and the unit vectors $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ of the the locally field-aligned (x, y, z) coordinate system. The unit vectors of the locally field aligned coordinate system are not orthogonal and change for every point in the poloidal plane.

$$\tau : \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} x + \int_0^y b_{x_c}(\boldsymbol{\gamma}(u))\,\mathrm{d}u \\ \int_0^y b_{y_c}(\boldsymbol{\gamma}(u))\,\mathrm{d}u \\ z + \int_0^y b_{z_c}(\boldsymbol{\gamma}(u))\,\mathrm{d}u \end{pmatrix} . \tag{13}$$

The new coordinate system can be interpreted in a simple way. The y coordinate corresponds to the length of the magnetic field line starting at the point $(x_c, y_c = 0, z_c)$. The x and z coordinates correspond to the x_c and z_c coordinates of the point the field line is originating from. On the poloidal plane, i.e. $y_c = \varphi = y = 0$, the locally field-aligned coordinates coincide with the Cartesian coordinates. We continue by deriving the metric tensor in the new coordinate system. We start with the Jacobian matrix of the coordinate transformation evaluated on the poloidal plane at $y = 0$:

$$\begin{aligned} J(x, 0, z) &= \left(\left(\frac{\partial\tau}{\partial x}\right)(x, 0, z), \left(\frac{\partial\tau}{\partial y}\right)(x, 0, z), \left(\frac{\partial\tau}{\partial z}\right)(x, 0, z)\right), \\ &= \begin{pmatrix} 1 & b_{x_c} & 0 \\ 0 & b_{y_c} & 0 \\ 0 & b_{z_c} & 1 \end{pmatrix} . \end{aligned} \tag{14}$$

For clarity we omit the dependency of b_{x_c}, b_{z_c} and b_{y_c} on x and z. The columns of the Jacobi matrix can be interpreted as a basis of the tangent space of the coordinate system and point in the direction of the coordinate lines

$$\boldsymbol{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \boldsymbol{e}_y = \begin{pmatrix} b_{x_c} \\ b_{y_c} \\ b_{z_c} \end{pmatrix}, \qquad \boldsymbol{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} . \tag{15}$$

The basis vectors are not orthogonal and the vector $\boldsymbol{e}_y$ is equal to the unit vector of the magnetic field. The coordinate system is depicted in Figure 1.

The components of the metric tensor are given by

$$g_{ij} = J^T J = \begin{pmatrix} 1 & b_{x_c} & 0 \\ b_{x_c} & 1 & b_{z_c} \\ 0 & b_{z_c} & 1 \end{pmatrix}, \tag{16}$$

and the determinant of the metric tensor reads

$$\sqrt{g} := \sqrt{\det\left(g_{ij}\right)} = b_{y_c} \,. \tag{17}$$

The basis of the dual space is given by

$$\begin{aligned} \boldsymbol{e}^x &= \left(1, -\tfrac{b_{x_c}}{b_{y_c}}, 0\right), \\ \boldsymbol{e}^y &= \left(0, \ \tfrac{1}{b_{y_c}}, 0\right), \\ \boldsymbol{e}^z &= \left(0, -\tfrac{b_{z_c}}{b_{y_c}}, 1\right), \end{aligned} \tag{18}$$

and the basis expansion of the magnetic field unit vector $\boldsymbol{b}$ reads

$$\boldsymbol{b} = \boldsymbol{e}_y = g_{2j}\boldsymbol{e}^j = b_{x_c}\boldsymbol{e}^x + \boldsymbol{e}^y + b_{z_c}\boldsymbol{e}^z \,. \tag{19}$$

3.2. Differential operations in the governing equation

There are three types of differential operators in the gyrokinetic Vlasov equation which we express in the locally field-aligned coordinate system

$$\boldsymbol{b}\cdot\nabla f, \qquad (\nabla\times\boldsymbol{b})\cdot\nabla f \qquad \text{and} \qquad (\boldsymbol{b}\times\nabla h)\cdot\nabla f,$$

where h, f are arbitrary scalar functions. The first operation corresponds to the parallel derivative and is given by

$$\boldsymbol{b}\cdot\nabla f = \boldsymbol{e}_y\cdot\boldsymbol{e}^i\frac{\partial f}{\partial x^i} = \frac{\partial f}{\partial y} \,. \tag{20}$$

For the second and third operator we have to express the curl and cross products in the new coordinate system. The curl of the magnetic field unit vector $\boldsymbol{b}$ is given by

$$\begin{aligned} \nabla\times\boldsymbol{b} &= \frac{1}{\sqrt{g}}\sum_{k=1}^{3}\epsilon_{ijk}\left(\frac{\partial b_j}{\partial x^i} - \frac{\partial b_i}{\partial x^j}\right)\boldsymbol{e}_k, \\ &= \frac{1}{b_{y_c}}\left[\frac{\partial b_{z_c}}{\partial y}\boldsymbol{e}_x - \frac{\partial b_{x_c}}{\partial y}\boldsymbol{e}_z + \left(\frac{\partial b_{x_c}}{\partial z} - \frac{\partial b_{z_c}}{\partial x}\right)\boldsymbol{e}_y\right], \end{aligned} \tag{21}$$

and the gradient of the function f in a curvilinear coordinate system can be expressed as

$$\nabla f = \frac{\partial f}{\partial u^i}\boldsymbol{e}^i \,. \tag{22}$$

We obtain the second operation by combining Eqs. (21) and (22) and taking into account that $\boldsymbol{e}_i\cdot\boldsymbol{e}^j = \delta_i^j$

$$(\nabla\times\boldsymbol{b})\cdot\nabla f = \frac{1}{b_{y_c}}\left[\frac{\partial b_{z_c}}{\partial y}\frac{\partial f}{\partial x} - \frac{\partial b_{x_c}}{\partial y}\frac{\partial f}{\partial z} + \left(\frac{\partial b_{x_c}}{\partial z} - \frac{\partial b_{z_c}}{\partial x}\right)\frac{\partial f}{\partial y}\right]. \tag{23}$$

The third operation is a triple vector product of the magnetic unit vector and two gradients of scalar functions. The gradients of the functions h and f are given by

$$\nabla h = \frac{\partial h}{\partial u^i}\boldsymbol{e}^i, \qquad \nabla f = \frac{\partial f}{\partial u^j}\boldsymbol{e}^j \,. \tag{24}$$

The vector product in the curvilinear coordinate system takes the form

$$\boldsymbol{A}\times\boldsymbol{B} = \frac{1}{\sqrt{g}}\sum_{k=1}^{3}\epsilon_{ijk}\left(A_iB_j - A_jB_i\right)\boldsymbol{e}_k \,. \tag{25}$$

By combining Eqs. (19), (24) and (25) we obtain the vector product of the magnetic field unit vector with the gradient of h:

$$\begin{aligned} \boldsymbol{b}\times\nabla h = \frac{1}{b_{y_c}}\Bigg[&\left(\frac{\partial h}{\partial z} - b_{z_c}\frac{\partial h}{\partial y}\right)\boldsymbol{e}_x + \left(b_{x_c}\frac{\partial h}{\partial y} - \frac{\partial h}{\partial x}\right)\boldsymbol{e}_z \\ &+ \left(b_{z_c}\frac{\partial h}{\partial x} - b_{x_c}\frac{\partial h}{\partial z}\right)\boldsymbol{e}_y\Bigg]. \end{aligned} \tag{26}$$

Taking the dot product of the last expression with the gradient of the function f yields the third operation

$$\begin{aligned} (\boldsymbol{b}\times\nabla h)\cdot\nabla f = \frac{1}{b_{y_c}}\Bigg[&\left(\frac{\partial h}{\partial z} - b_{z_c}\frac{\partial h}{\partial y}\right)\frac{\partial f}{\partial x} + \left(b_{x_c}\frac{\partial h}{\partial y} - \frac{\partial h}{\partial x}\right)\frac{\partial f}{\partial z} \\ &+ \left(b_{z_c}\frac{\partial h}{\partial x} - b_{x_c}\frac{\partial h}{\partial z}\right)\frac{\partial f}{\partial y}\Bigg]. \end{aligned} \tag{27}$$

We finish this section by rewriting Eq. (27) in terms of Poisson brackets

$$\{h, f\}_{x,y} = \frac{\partial h}{\partial x}\frac{\partial f}{\partial y} - \frac{\partial h}{\partial y}\frac{\partial f}{\partial x}. \tag{28}$$

This rewriting will be useful in the discretization where we employ the Arakawa scheme for discretizing Poisson brackets [38]

$$\begin{aligned} (\boldsymbol{b}\times\nabla h)\cdot\nabla f &= \frac{b_{x_c}}{b_{y_c}}\left(\frac{\partial h}{\partial y}\frac{\partial f}{\partial z} - \frac{\partial h}{\partial z}\frac{\partial f}{\partial y}\right) \\ &\quad + \frac{b_{z_c}}{b_{y_c}}\left(\frac{\partial h}{\partial x}\frac{\partial f}{\partial y} - \frac{\partial h}{\partial y}\frac{\partial f}{\partial x}\right) + \frac{1}{b_{y_c}}\left(\frac{\partial h}{\partial z}\frac{\partial f}{\partial y} - \frac{\partial h}{\partial x}\frac{\partial f}{\partial z}\right), \\ &= \frac{b_{x_c}}{b_{y_c}}\{h, f\}_{y,z} + \frac{b_{z_c}}{b_{y_c}}\{h, f\}_{x,y} + \frac{1}{b_{y_c}}\{h, f\}_{z,x} \,. \end{aligned} \tag{29}$$

In summary, Eqs. (20), (23) and (29) provide an explicit form of the operations in the gyrokinetic Vlasov equation in the locally field aligned coordinate system.

In the following, we assume that $\nabla_\perp \approx \frac{\partial}{\partial x}\boldsymbol{e}^x + \frac{\partial}{\partial z}\boldsymbol{e}^z$ in the quasi-neutrality equation. This approximation can be made in the field part of the Lagrangian such that the gyrokinetic Vlasov-Poisson system remains consistent. We use this assumption for the following computational purposes. As the basis vectors $\boldsymbol{e}^x$ and $\boldsymbol{e}^z$ are not orthogonal to the magnetic field, the quasi-neutrality equation contains derivatives in all three coordinate variables x, y and z. This makes it necessary to solve the quasi-neutrality equation for all poloidal planes at once. By removing the y dependence from $\nabla_\perp$ the quasi-neutrality equation can be solved on every poloidal independently.

In the next section we describe the numerical techniques used to solve the gyrokinetic Vlasov-Poisson system in the locally field-aligned coordinate system.

4. Numerical scheme and implementation

4.1. Normalization

We use the normalization conventions employed in GENE [13]. Reference quantities are denoted with a "$_{\text{ref}}$" subscript and renormalized quantities with a hat, i.e. for the mass m_σ =

Elementary charge	e
Reference mass	m_{ref}
Reference temperature	T_{ref}
Reference length	L_{ref}
Reference magnetic field	B_{ref}
Reference density	n_{ref}
Reference velocity	$c_{\mathrm{ref}} = \sqrt{T_{\mathrm{ref}}/m_{\mathrm{ref}}}$
Reference gyrofrequency	$\Omega_{\mathrm{ref}} = eB_{\mathrm{ref}}/(m_{\mathrm{ref}}c)$
Reference gyroradius	$\rho_{\mathrm{ref}} = c_{\mathrm{ref}}/\Omega_{\mathrm{ref}}$

Table 1: Elementary reference values for mass, temperature, length, magnetic field strength and density and derived reference values for velocity, gyrofrequency and gyroradius.

$\hat{m}_\sigma m_{\mathrm{ref}}$. The reference values defining the scales of the simulation are given in Table 1.

With those definitions we normalize the gyrocenter coordinates as

$$\begin{aligned} R &= \hat{R}L_{\mathrm{ref}}, & Z &= \hat{Z}L_{\mathrm{ref}}, & \varphi &= \hat{\varphi}, \\ v_\parallel &= \hat{v}_\parallel v_{T_\sigma}(\boldsymbol{R}_0), & \mu &= \hat{\mu}\frac{T_{0\sigma}(\boldsymbol{R}_0)}{B_{\mathrm{ref}}}, & t &= \hat{t}\frac{L_{\mathrm{ref}}}{c_{\mathrm{ref}}}, \end{aligned} \tag{30}$$

where $T_{0\sigma}(\boldsymbol{R}_0) = T_{\mathrm{ref}}\hat{T}_{0\sigma}(\boldsymbol{R}_0)$ denotes the initial temperature at a given radial position $\boldsymbol{R}_0$ and $v_{T_\sigma}(\boldsymbol{R}_0) = \sqrt{2T_{0\sigma}(\boldsymbol{R}_0)/m_\sigma} = c_{\mathrm{ref}}\hat{v}_{T_\sigma}(\boldsymbol{R}_0)$ the corresponding thermal velocity. Finally, the distribution function and electrostatic potential are normalized according to

$$f_\sigma = \frac{n_{0\sigma}(\boldsymbol{R}_0)}{v^3_{T_\sigma}(\boldsymbol{R}_0)}\hat{f}_\sigma, \tag{31}$$

$$\phi = \frac{T_{\mathrm{ref}}}{e}\hat{\phi}\,. \tag{32}$$

4.2. Discretization

4.2.1. Discretization of the Vlasov equation

We discretize the Vlasov equation with a finite difference discretization scheme. In the coordinate $v_\parallel$ we use a Cartesian grid, and in the coordinates x and z in the poloidal plane an unstructured grid, that is required to be Cartesian in the neighborhood of any point. We label each grid point in the poloidal plane by a one dimensional index i and store the x and z coordinates in one dimensional arrays `x[i]` and `z[i]`. This makes the implementation flexible because one can remove points from and add points to the grid and thus model the shape of the fusion device. An example of the grid demonstrating the capability to model realistic geometries is shown in Fig. 2. Requiring that the grid is Cartesian in the neighborhood of every point allows us to use efficient numerical schemes for Cartesian grids.

We discretize the advection terms in x, z and $v_\parallel$ with exception of the nonlinear $\boldsymbol{E}\times\boldsymbol{B}$ terms as

$$a\left(\frac{\partial f_\sigma}{\partial x}\right)(x_i) = \frac{a}{12\Delta x}\Big[f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})\Big] + O\left((\Delta x)^4\right). \tag{33}$$

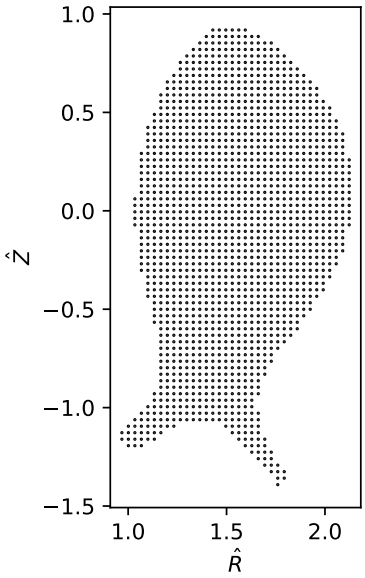

Figure 2: Figure displaying the grid in the poloidal plane in an X-point geometry. The grid is unstructured and models the boundary and the divertor region of the fusion device. In the neighborhood of every point the grid is Cartesian. For a clear presentation only every 10th grid point is displayed.

For the $\boldsymbol{E}\times\boldsymbol{B}$ terms, represented by the Poisson bracket, we use the scheme introduced by Arakawa [38].

In the y direction, along the field line, our grid is non-uniform. The grid points are defined by the intersection points of a field line with neighboring poloidal planes. The intersection points, as well as the length of the magnetic field line Δy, are calculated once upon initialization of the code up to machine precision with the 8th order embedded Runge-Kutta method `dop853` [39, p. 2, II. 10]. In general, the intersection points do not coincide with a grid point. We use bicubic interpolation to approximate the value of the distribution function on the intersection points. The intersection of the magnetic field line with neighboring poloidal planes is visualized in Fig. 3. Advection terms in the parallel y direction are expressed with a second order symmetric finite difference scheme as

$$a\left(\frac{\partial f_\sigma}{\partial y}\right)(x_i, 0, z_j) = \frac{a}{\Delta y}\Big[f(x_i, y_1, z_j) - f(x_i, y_{-1}, z_j)\Big] + O\left((\Delta y)^2\right) + O\left(\frac{(\Delta x)^3}{\Delta y}\right) + O\left(\frac{(\Delta z)^3}{\Delta y}\right), \tag{34}$$

where $\Delta y = y_{-1} + y_1$. The calculation of the parallel derivatives is the most performance demanding part of the code. The bicubic interpolation comprises the 16 grid points surrounding the intersection point. This gives rise to a 32 point stencil to calculate the parallel derivative with the second order centered finite difference scheme mentioned above.

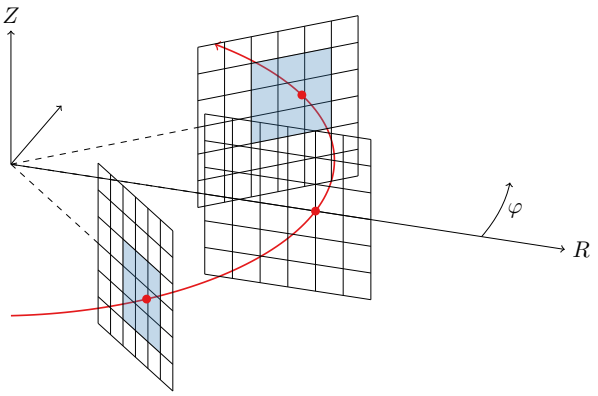

Figure 3: Figure displaying a magnetic field line originating at the poloidal plane at $\varphi = 0$ and hitting different poloidal planes at $\varphi \in \{-\pi/4, \pi/4\}$. The points where the field line hits are denoted by red dots. The value of the distribution function is obtained via bicubic interpolation around the intersection point. The area included in the interpolation on every poloidal plane are shaded in blue.

4.2.2. Discretization of the quasi-neutrality equation

The quasi-neutrality equation can be written in a general form as

$$-\nabla_\perp \cdot (c\,\nabla_\perp \phi) = b \sum_\sigma q_\sigma \int f_\sigma \,\mathrm{d}v_\parallel \,\mathrm{d}\mu, \tag{35}$$

with coefficients c and b. We discretize the left hand side of the equation with second order symmetric finite differences

$$\begin{aligned}
&-\nabla_\perp \cdot (c\nabla_\perp \phi) = \\
&-\left[\left(\frac{c(x_{i+1},z_j)+c(x_i,z_j)}{2\Delta x}\right)\left(\frac{\phi(x_{i+1},z_j)-\phi(x_i,z_j)}{\Delta x}\right)\right. \\
&\quad\left.-\left(\frac{c(x_i,z_j)+c(x_{i-1},z_j)}{2\Delta x}\right)\left(\frac{\phi(x_i,z_j)-\phi(x_{i-1},z_j)}{\Delta x}\right)\right] \\
&-\left[\left(\frac{c(x_i,z_{j+1})+c(x_i,z_j)}{2\Delta z}\right)\left(\frac{\phi(x_i,z_{j+1})-\phi(x_i,z_j)}{\Delta z}\right)\right. \\
&\quad\left.-\left(\frac{c(x_i,z_j)+c(x_i,z_{j-1})}{2\Delta z}\right)\left(\frac{\phi(x_i,z_j)-\phi(x_i,z_{j-1})}{\Delta z}\right)\right] \\
&+O\left((\Delta x)^2\right)+O\left((\Delta z)^2\right).
\end{aligned} \tag{36}$$

The integral over the distribution function in the right hand side of the quasi-neutrality equation is discretized using the trapezoidal rule

$$\int f \,\mathrm{d}v_\parallel \,\mathrm{d}\mu = \Delta v_\parallel \Delta\mu \sum_{l=1}^{N_l}\sum_{m=1}^{N_m} w_{lm} f(v_{\parallel l}, \mu_m) + O\left((\Delta v_\parallel)^2\right) + O\left((\Delta\mu)^2\right), \tag{37}$$

where

$$w_{lm} = \begin{cases} 1/2 & l \in \{1, N_l\} \quad \text{or} \quad m \in \{1, N_m\}, \\ 1 & \text{otherwise}. \end{cases} \tag{38}$$

Due to the quadrature, the parallel discretization, the elliptic operator and the Arakawa scheme, the spatial discretization is of order two. We solve the remaining initial value problem with an explicit Runge-Kutta method of order four [39, pp. 132–140]. This has proven to be a solid integrator in combination with the Arakawa scheme and central finite differences in the gyrokinetic code `GENE` [13]. Furthermore, we have implemented the option to add numerical diffusion in the dimensions x, y, z and $v_\parallel$. For the simulations presented in this work no additional numerical diffusion was added.

4.3. Boundary conditions

For the simulations presented in the next chapter we choose Dirichlet boundary conditions. We set the distribution function to zero on the velocity space boundary. In real-space we choose periodic boundary conditions in φ and set the distribution function equal to the initial profile and the electrostatic potential equal to zero on the R, Z boundary. The same boundary condition is used for the advective inflow and outflow through the boundary. Furthermore, we have the option to apply diffusion in a small buffer zone around the R, Z boundary to shield the effect of the boundary conditions in the poloidal plane. For the simulation results presented in the next chapter no buffer zone was added.

5. Verification

5.1. Method of manufactured solutions

The gyrokinetic Vlasov-Poisson system is a complicated integro-differential system of equations and exact solutions are only known for special cases. Therefore, it is not possible to comprehensively check the convergence of the numerical solution by comparing it against exact solutions. Nevertheless it is possible to add extra terms to the Vlasov-Poisson system such that it obtains any solution. This is known as the method of manufactured solutions (MMS) [40]. In order to explain it in more detail we rewrite the gyrokinetic Vlasov-Poisson system in a different form. Consider the integro-differential operators $L(\phi)$, $N(f)$ such that

$$\begin{aligned} L(\phi)[f] &= 0, \\ N(f)[\phi] &= 0, \end{aligned}$$

represents the gyrokinetic Vlasov-Poisson system. L represents the Vlasov operator which depends on the electrostatic potential and N represents the operator that forms the quasi-neutrality equation. In the method of manufactured solutions we solve a modified system of equations

$$\begin{aligned} L(\phi)[f] &= S_f, \\ N(f)[\phi] &= S_\phi, \end{aligned} \tag{39}$$

with sources S_ϕ and S_f included in the right hand side of the equations such that given functions f_{MMS} and ϕ_{MMS} are a solution to the system. The functions f_{MMS} and ϕ_{MMS} can be chosen arbitrarily. The sources S_ϕ and S_f are calculated by plugging the solutions into the Vlasov and quasi-neutrality equation. As the solutions are known this is a straightforward process and can be done with a computer algebra system. We use `Mathematica`

[41]. The functions f_{MMS} and ϕ_{MMS} are called the manufactured solutions.

Numerical solutions of the modified Vlasov-Poisson system Eq. (39) can be compared to the exact solution functions f_{MMS} and ϕ_{MMS}. This makes it possible to calculate relative errors of the numerical solution and to study the order of convergence of the numerical scheme.

As manufactured solutions we choose the functions

$$\begin{aligned} f_{\text{MMS}}(\hat{r},\varphi,\theta,\hat{v}_\parallel,\hat{\mu},\hat{t}) &= \cos^2(\hat{\omega}\hat{t})\cos^2(n_{\text{tor}}\varphi)\sin^2(n_{\text{pol}}\theta) \\ &\quad\times \sin^2(\pi n_{\text{rad}}(\hat{r}-\hat{r}_{\text{min}})/(\hat{r}_{\text{max}}-\hat{r}_{\text{min}}))e^{-\left(\hat{v}_\parallel^2+\hat{\mu}\hat{B}\right)}, \\ \phi_{\text{MMS}}(\hat{r},\varphi,\theta,\hat{t}) &= 2\cos^2(\hat{\omega}\hat{t})\cos(n_{\text{tor}}\varphi)\sin(2n_{\text{pol}}\theta) \\ &\quad\times \sin(2\pi n_{\text{rad}}(\hat{r}-\hat{r}_{\text{min}})/(\hat{r}_{\text{max}}-\hat{r}_{\text{min}})), \end{aligned} \tag{40}$$

representing a mode. We run the verification in three different magnetic geometries testing different aspects of the implementation. For the first test, we choose slab geometry where the equilibrium magnetic field is given by

$$\boldsymbol{B}_0 = B_0\boldsymbol{e}_\varphi\,. \tag{41}$$

In slab geometry multiple terms in the Vlasov equations like the curvature or ∇B_0 drift vanish. Furthermore, the magnetic field is uniform and points in the φ direction. Therefore, the magnetic field is automatically aligned to the coordinate system and the locally field-aligned coordinate system coincides with the intrinsic coordinate system of the manifold. This renders the field line tracing trivial and thus enables us to test the implementation without complications arising from the locally field aligned coordinate system. The second, circular equilibrium is defined via

$$\boldsymbol{B}_0 = B_0\left(\boldsymbol{e}_\varphi + \frac{\hat{r}}{q_0}\boldsymbol{e}_\theta\right). \tag{42}$$

In the circular geometry the magnetic field lines are twisted and the field line tracing as well as the interpolation is tested. Third, we test a toroidal magnetic equilibrium that incorporates all terms in the Vlasov equation

$$\boldsymbol{B}_0 = \frac{B_0}{\hat{R}_0+\hat{r}\cos(\theta)}\left(\boldsymbol{e}_\varphi + \frac{\hat{r}}{q(\hat{r})}\boldsymbol{e}_\theta\right), \tag{43}$$

with q profile

$$q(\hat{r}) = q_0 + \alpha\hat{r}^2, \tag{44}$$

and major radius R_0. We run the MMS analysis with the parameters given in Table 2 with four different resolutions presented in Table 3. The radial box size is limited by $\hat{r}_{\text{min}} = 0$ and $\hat{r}_{\text{max}} = 1$ in slab geometry and $\hat{r}_{\text{min}} = 0.5$ and $\hat{r}_{\text{max}} = 0.8$ in both circular and toroidal geometry. In velocity space $(\hat{v}_\parallel,\hat{\mu})$ we choose a domain size of $[-3,3]\times[0,9]$ and in the toroidal angle φ of $[0,2\pi)$.

We calculate one period of the manufactured solution up to $\hat{t} = 1$. The initial ion density and the corresponding final error are shown in Fig. 4 and show no sign of numerical artifacts. Further, we analyze the results by calculating the relative L_2 and L_∞ error of the numerical solution given by $\|f - f_{\text{MMS}}\|_p/\|f_{\text{MMS}}\|_p$ where $p\in\{2,\infty\}$. The results are shown in Fig. 5.

We observe that the numerical scheme converges to second order for all three geometries.

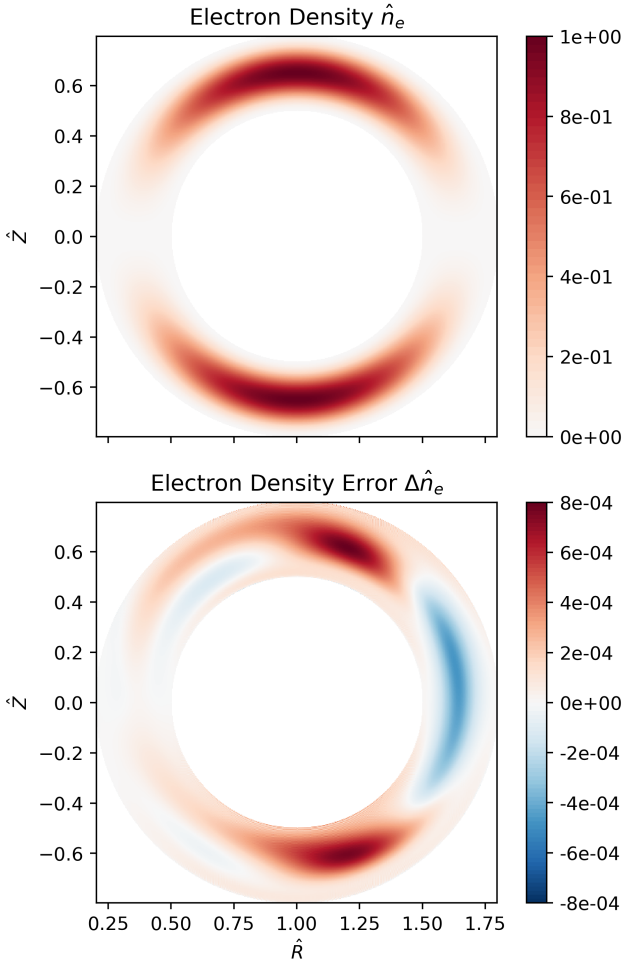

Figure 4: Figure showing the initial electron density $\hat{n}_e(\hat{t}=0)$ (top) and the error $\Delta\hat{n}_e = \hat{n}_e(\hat{t}=1) - \hat{n}_{e,\text{MMS}}(\hat{t}=1)$ of the final electron density (bottom) in the R,Z plane at $\hat{\varphi}=\pi$ for the MMS run in toroidal geometry using 80 poloidal planes.

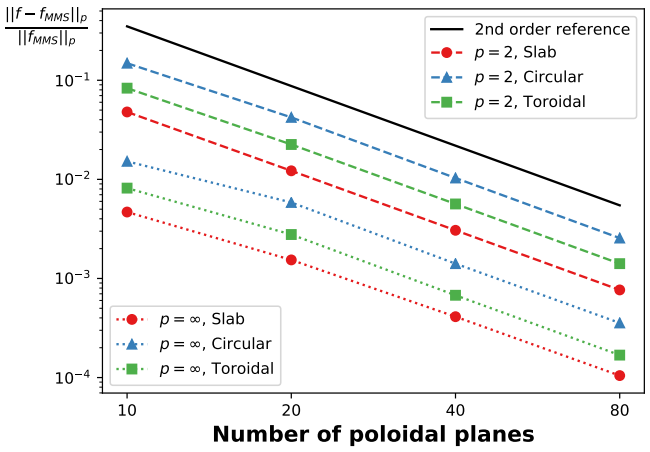

Figure 5: Figure displaying the relative L_2 (dashed) and L_∞ (dotted) error of the numerical solution in the MMS test as a function of the resolution. The resolution is represented by the number of poloidal planes. The complete resolutions are specified in Table 3. The test is performed for a slab, circular, and toroidal geometry (different markers, colors). For reference a line representing second order convergence is drawn in black. The numerical scheme converges to second order for all geometries.

5.2. *Screw pinch simulations*

In the following, we present a physical benchmark where we measure growth rates of an ion temperature gradient mode in a screw pinch geomety. We use a setup similar to the work in [42, 43]. We model the screw pinch with the circular geometry described in the last section with a constant q profile $q(r) = q_0$. We simulate two different cases. In the first case we choose $q_0 \to \infty$ corresponding to a slab like geometry[1] and in the second case we choose a finite q. This allows us to compare the performance of ordinary field-aligned coordinates in the slab case to the locally field-aligned coordinate system with interpolation for the finite q case. We choose the reference length parameters $L_{\text{ref}} = 7.74 \cdot 10^{-1}$ m, $B_{\text{ref}} = 1$ T, $m_{\text{ref}} = 1$ u and $T_{\text{ref}} = 1$ keV such that the ion gyroradius $\rho_{\text{ref}} = 3.23 \cdot 10^{-3}$ m and $L_{\text{ref}}/\rho_{\text{ref}} = 239$. The simulation box ranges from $\hat{r}_{\text{min}} = 4.17 \cdot 10^{-3}$ to $\hat{r}_{\text{max}} = 6.05 \cdot 10^{-2}$ spanning over $14.5\rho_{\text{ref}}$. In velocity space $(\hat{v}_\parallel, \hat{\mu})$ we choose a domain size of $[-4, 4] \times [0, 16]$ and in the toroidal angle φ of $[0, 2\pi)$. We use a realistic mass ratio with $m_e = 1/3600\, m_i$.

We initialize the simulation with local Maxwellians

$$\hat{F}_{0\sigma}(\hat{r}, \theta, \varphi, \hat{v}_\parallel, \hat{\mu}) = \frac{\hat{n}_{0\sigma}(\hat{r})}{\left(\pi \hat{T}_{0\sigma}(\hat{r})\right)^{3/2}} e^{-\frac{\hat{v}_\parallel^2 + \hat{\mu}\hat{B}_0}{\hat{T}_{0\sigma}(\hat{r})}}, \tag{45}$$

and given temperature $\hat{T}_{0\sigma}$ and density profiles $\hat{n}_{0\sigma}$. Maxwellians are an exact equilibrium of the gyrokinetic Vlasov equation in the given geometry and thus provide a stable initial condition. In order to study growth rates we initialize a mode as a perturbation of the Maxwellian

$$\hat{f}_\sigma(\hat{t} = 0, \hat{r}, \theta, \varphi, \hat{v}_\parallel, \hat{\mu}) = \hat{F}_{0\sigma}(\hat{r}, \theta, \varphi, \hat{v}_\parallel, \hat{\mu}) \times \left(1 + \epsilon \exp\left(-\left(\frac{\hat{r} - \hat{r}_{\text{mid}}}{\delta \hat{r}_{\text{mode}}}\right)^2\right) \cos\left(m\theta + n\varphi\right)\right). \tag{46}$$

We choose the profiles $\hat{T}_{0i}$, $\hat{T}_{0e}$, $\hat{n}_{0i}$ and $\hat{n}_{0e}$, appearing in the Maxwellian distribution function according to

$$P(\hat{r}) = C_P \exp\left(-\kappa_P\, \delta\, \hat{r}_P \tanh\left(\frac{\hat{r} - \hat{r}_{\text{mid}}}{\delta \hat{r}_P}\right)\right). \tag{47}$$

The parameters for the profile and the initial condition are given in Table 4. The initial density profile is chosen to be constant by setting $\kappa_n = 0$. The profiles are depicted in Figure 6.

In order to compare the growth rates to analytical predictions we use an approximate dispersion relation derived in [43, Appendix]. The dispersion relation has been derived with the assumption of adiabatic electron response and large aspect ratio, i.e. $r/(qR_0) \to 0$. Both approximations are valid for the parameters chosen and hence we expect the numerical results to be close to the prediction by the dispersion relation. In order to present the dispersion relation we need to define a single Fourier mode of $\hat{\phi}$:

$$\hat{\phi} = \hat{\phi}_{m,n,\hat{\omega}} \exp\left(i(m\theta + n\varphi - \hat{\omega}\hat{t})\right). \tag{48}$$

[1] Numerically we implement $q_0 \to \infty$ by setting $q_0 = 10^8$.

$\hat{\omega}$	n_{pol}	n_{tor}	n_{rad}	q_0	α	m_i	m_e
2π	1	1	1	0.86	2.12	1	1

Table 2: Table displaying the parameters used for the method of manufactured solutions.

N_φ	$\Delta\hat{x} = \Delta\hat{z}$	$N_{v_\parallel}$	N_μ	$\Delta\hat{t}$
10	0.025	10	5	0.01
20	0.0125	20	10	0.005
40	0.00625	40	20	0.0025
80	0.003125	80	40	0.00125

Table 3: Table displaying the different resolutions used for the method of manufactured solutions.

ϵ	$1.00 \cdot 10^{-4}$
$\hat{r}_{\text{mid}}$	$3.05 \cdot 10^{-2}$
$\delta\hat{r}_{\text{mode}} =$	$5.90 \cdot 10^{-3}$
$C_{n_i} = C_{n_e}$	$1.65 \cdot 10^{1}$
$C_{T_i} = C_{T_e}$	1.00
$\delta\hat{r}_{T_i} = \delta\hat{r}_{T_e}$	$6.05 \cdot 10^{-3}$
$\delta\hat{r}_{n_i} = \delta\hat{r}_{n_e}$	$1.21 \cdot 10^{-2}$
$\kappa_{T_i} = \kappa_{T_e}$	$6.62 \cdot 10^{1}$
$\kappa_{n_i} = \kappa_{n_e}$	0

Table 4: Table defining the parameters for the screw pinch profile for ions and electrons specified in Equation (47).

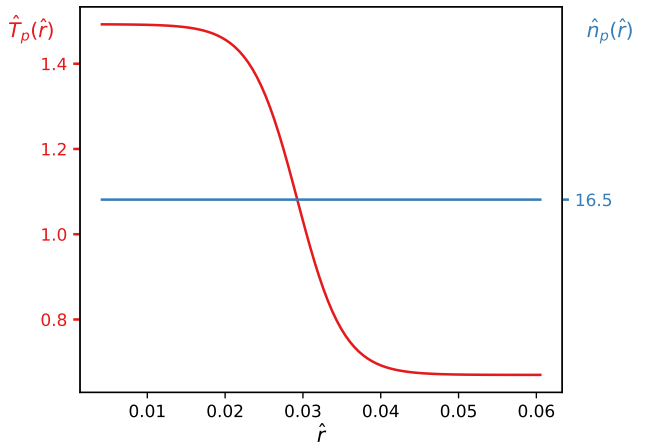

Figure 6: Figure showing the normalized initial temperature and density profile. The profile is the same for ions and electrons.

The dispersion relation reads [43, Appendix]

$$
\begin{aligned}
&-\left(\frac{\rho_{\text{ref}}}{L_{\text{ref}}}\right)^2\left[\frac{\partial^2}{\partial\hat{r}^2}+\left(\frac{1}{\hat{r}}+\frac{1}{\hat{n}_0}\frac{\partial\hat{n}_0}{\partial\hat{r}}\right)\frac{\partial}{\partial\hat{r}}-\frac{m^2}{\hat{r}^2}\right]\hat{\phi}_{m,n,\hat{\omega}}(\hat{r}) = \\
&\left\{-\frac{1}{\hat{T}_{0i}}\left(1+zZ(z)\right)-\frac{1}{\hat{T}_{0e}}+\frac{m}{\hat{k}^*\hat{r}\hat{B}_0 L}\frac{\rho_{\text{ref}}}{L_{\text{ref}}}\right. \\
&\left.\left[Z(z)\left(\frac{1}{\hat{n}_0}\frac{\partial\hat{n}_0}{\partial\hat{r}}-\frac{1}{2\hat{T}_{0i}}\frac{\partial\hat{T}_{0i}}{\partial\hat{r}}\right)+z(1+zZ(z))\frac{1}{\hat{T}_{0i}}\frac{\partial\hat{T}_{0i}}{\partial\hat{r}}\right]\right\} \\
&\times\hat{\phi}_{m,n,\hat{\omega}}(\hat{r}),
\end{aligned}
\tag{49}
$$

with $z = \hat{\omega}/\hat{k}^*$, $\hat{k}^* = \hat{k}_\parallel\sqrt{2\hat{T}_{0i}}$ and the parallel wave number $\hat{k}_\parallel = (m/q+n)b_\varphi$. Furthermore, we used the plasma dispersion function

$$
\begin{aligned}
Z(u) &= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{\exp(-x^2)}{x-u}\,\mathrm{d}x = \mathrm{i}\sqrt{\pi}\exp(-u^2)(1+\operatorname{erf}(\mathrm{i}u)), \\
\operatorname{erf}(x) &= \frac{2}{\sqrt{\pi}}\int_0^x\exp(-t^2)\,\mathrm{d}t\,.
\end{aligned}
\tag{50}
$$

We describe how the dispersion relation is solved in Appendix A. The dispersion relation, and thus the eigenvalues $\hat{\omega}$, are a function of the poloidal mode number m and the parallel wave number $\hat{k}_\parallel$. Hence, the eigenvalues $\hat{\omega}$ do not depend on q explicitly. For a given poloidal mode number m we can select two toroidal mode numbers n_{slab} and n_{screw}, for the $q = \infty$ and $q \neq \infty$ case respectively, such that the parallel wave numbers and, as a consequence, the growth rates are equal. For the slab case, the poloidal magnetic field vanishes and the parallel wave number reads $\hat{k}_{\parallel\text{slab}} = n_{\text{slab}}$. For the $q \neq \infty$ case we choose the toroidal mode number n_{screw} such that $k_{\parallel\text{screw}} = k_{\parallel\text{slab}}$, i.e.

$$
\begin{aligned}
n_{\text{screw}} &= k_{\parallel\text{slab}} - \frac{b_\theta m}{r}, \\
&= n_{\text{slab}} - \frac{m}{q}\,.
\end{aligned}
\tag{51}
$$

Having two setups with the same growth rates allows us to compare the performance of field aligned coordinates in the $q = \infty$ case, to the locally field aligned coordinates in the $q \neq \infty$ case.

q	m	n	$\hat{k}_\parallel$
∞	5	1	1
∞	10	1	1
∞	15	1	1
∞	20	1	1

q	m	n	$\hat{k}_\parallel$
5/3	5	−2	1
10/3	10	−2	1
15/3	15	−2	1
20/3	20	−2	1

Table 5: Table displaying the different initial mode configurations for the simulation. On the left, the mode numbers for the $q = \infty$ case and on the right, the mode numbers for the $q \neq \infty$ case are shown. The poloidal and toroidal mode numbers are constructed such that the parallel mode number $k_\parallel$ is equal to one in all cases.

The q factors and mode numbers chosen for the simulation are shown in Table 5. They are selected such that a broad range of modes and q factors are tested.

5.2.1. *Growth rates*

We measure the growth rate from the L_2 norm of the electrostatic potential. The L_2 norm is calculated numerically with the trapezoidal quadrature rule similar to the charge density in the quasi-neutrality equation. We determine the growth rate by performing a linear fit of $\log\|\hat{\phi}\|$ between the times $\hat{t} = 1$ and $\hat{t} = 3.5$. This is shown exemplarily in Fig. 7.

We use a resolution of $\Delta\hat{x} = \Delta\hat{z} = 6.46\cdot10^{-4}$ corresponding to $0.2\,\rho_{\text{ref}}$, $n_\varphi = 16$, $n_{v_\parallel} = 256$, $n_\mu = 16$ and a timestep of $\Delta t = 1.4\cdot10^{-3}\,t_{\text{ref}} = 3.5\text{ns}$. The timestep is constant throughout the simulation and limited by the parallel electron streaming. We study the convergence of the simulation at the example of the $q = 15/3$ case. Upon doubling the resolution in each dimension, $\Delta\hat{x}\to0.5\Delta\hat{x}$, $\Delta\hat{z}\to0.5\Delta\hat{z}$, $n_\varphi\to2n_\varphi$, $n_{v_\parallel}\to2n_{v_\parallel}$ and $n_\mu\to2n_\mu$, the growth rate of the $q = 15/3$ mode changes less than 1%. Therefore, we assume a 1% error on the growth rates obtained from our simulations.

The growth rates obtained from the simulation as well as the solution of the approximate dispersion relation are shown in Fig. 8. The simulations for $q = \infty$ and $q \neq \infty$ agree well with each other within the 1% error margin, as predicted by the dispersion relation. Furthermore, the numerical results are close to the solution of the dispersion relation. They agree within the 1% numerical error margin for the poloidal mode numbers $m = 15$ and $m = 20$. For the poloidal mode numbers $m = 5$ and $m = 10$ the measured growth rates are slightly higher than growth rates obtained from the dispersion relation. As discussed above, the dispersion relation approximates the growth rates and thus exact agreement is not expected.

5.3. *Energy and particle conservation*

In order to study the behaviour of the nonlinear phase of the simulation we investigate the conservation of particle number and energy during the onset of turbulence. We choose the same screw pinch setup as presented in the last section and run the $q = 15/3$ case. We use a resolution of $\Delta\hat{x} = \Delta\hat{z} = 3.23\cdot10^{-4}$, corresponding to $0.1\,\rho_{\text{ref}}$, $n_\varphi = 32$, $n_{v_\parallel} = 64$ and $n_\mu = 16$. We run the simulation until $\hat{t} = 10$, when the turbulence starts to hit the wall. The chosen Dirichlet boundary conditions allow an energy flux through the domain boundaries, which becomes effective when eddies hit the wall. As an example the ion density

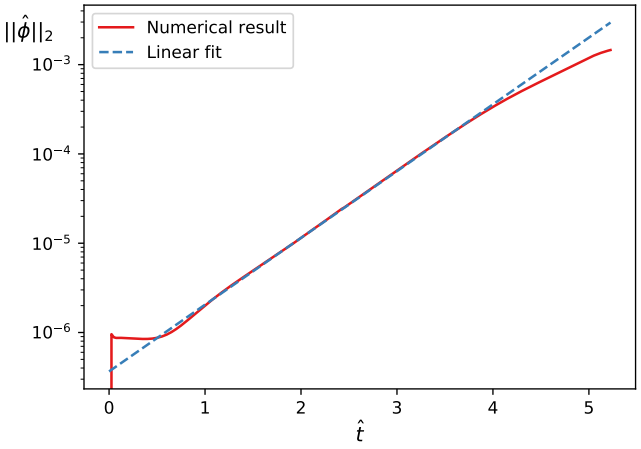

Figure 7: Figure displaying the L_2 norm of the electrostatic potential as a function of time. The growth rate is obtained by performing a linear fit of $\log ||\hat{\phi}||_2$ between the times $\hat{t} = 1$ and $\hat{t} = 3.5$. The fit is drawn in a blue dashed line. Shown are the results from the simulation with $q = 15/3$ and $m = 15$.

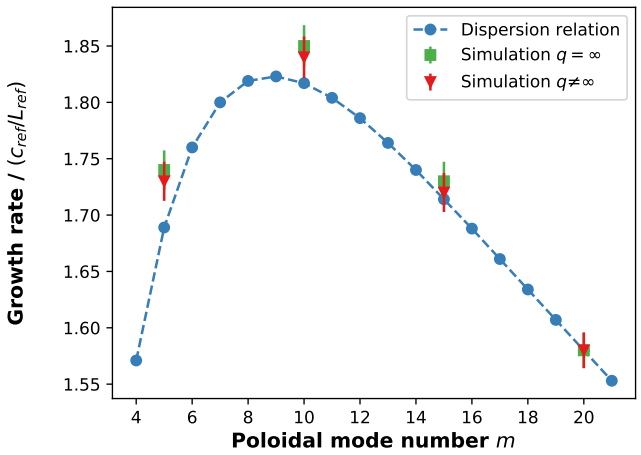

Figure 8: Figure displaying the growth rates obtained from the approximate dispersion relation and the simulations with different q factors and mode numbers specified in Table 5. Vertical lines through the markers represent the error of the simulation. The measured growth rates are close to the prediction from the dispersion relation.

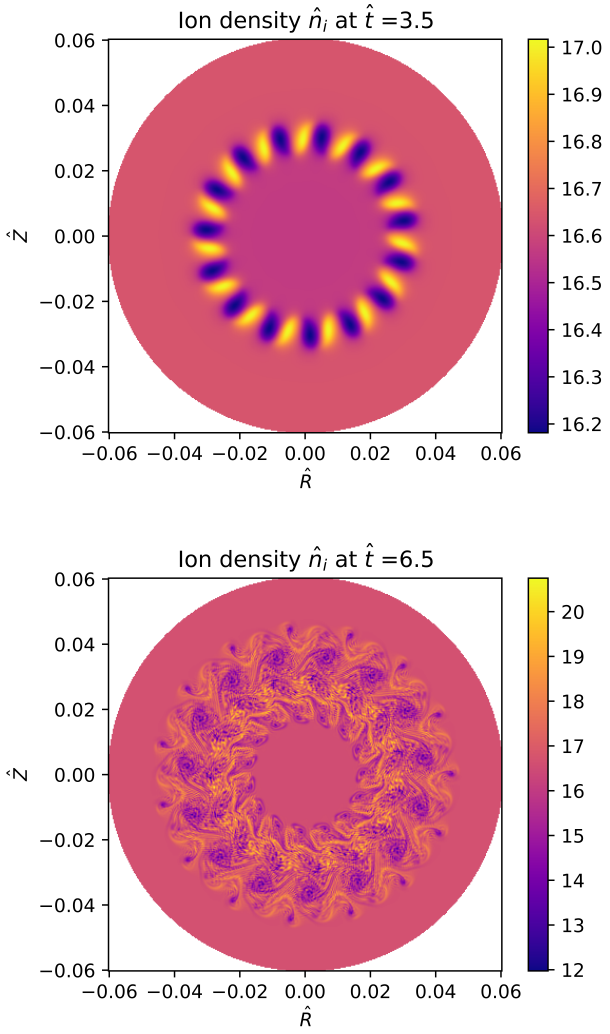

Figure 9: Figure displaying the ion density in the poloidal plane located at $\varphi = \pi$ in the linear phase at $\hat{t} = 3.5$ and the nonlinear phase at $\hat{t} = 6.5$ of a screw pinch simulation with $q = 15/3$. The simulation is initialized with the temperature and density profile shown in Fig. 6 and perturbed with an unstable $m = 15$, $n = -2$ mode.

is depicted in the linear and nonlinear phase in Fig. 9. The energy of the ions and electrons is shown in Fig. 10. The number of particles is conserved up to a precision of 10^{-7}. The energy is conserved up to a precision of 10^{-4}. In the linear phase the conservation is improved. As soon as the nonlinear phase starts, energy gets transferred between electrons and ions.

6. Summary and outlook

We developed a full-f, continuum gyrokinetic turbulence code based on a locally field-aligned coordinate system that is free of coordinate singularities and allows simulations ranging from the core to the scrape-off layer of a magnetic confinement fusion device.

We started this work by presenting the gyrokinetic model, implemented in the code. The model incorporates both dynamic ions and electrons and is energy conservative on the

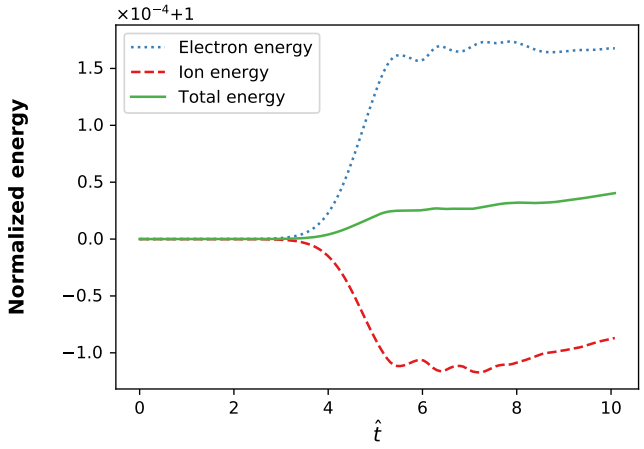

Figure 10: Figure displaying the normalized energy $E(t)/E(0)$ for electrons and ions of a simulation in screw pinch geometry with $q = 15/3$. The simulation is initialized with the temperature profile shown in Figure 6 and perturbed with an unstable $m = 15$, $n = -2$ mode. The simulation developes turbulence at approximately $\hat{t} = 4$.

continuous level. We introduced the locally field-aligned coordinate system following the flux-coordinate independent approach and expressed the differential operators appearing in the gyrokinetic Vlasov-Poisson equation in the new coordinate system. The operators were discretized with a finite difference scheme and the remaining inital value problem was solved with a fourth order Runge-Kutta method. Furthermore, we presented careful tests of the code on closed magnetic flux surfaces. First we used the method of manufactured solutions to verify that the numerical scheme converges with second order accuracy. The tests were performed for a slab, circular, and toroidal geometry to be comprehensive. Second we measured growth rates of an ion temperature gradient mode in screw pinch geometry and compared them to analytical predictions. Finally, we investigated the conservation of particles and energy during a turbulence simulation.

We conclude that the presented locally field-aligned coordinate system, together with the chosen numerical scheme work well in a gyrokinetic code.

As a next step we will implement boundary conditions to treat open magnetic field lines enabling us perform simulations in the diverted geometry of a complete fusion device. Further, we plan to improve the model by implementing a collision operator and use the geometric multigrid solver provided by the `GRILLIX` code to solve a nonlinear quasi-neutrality equation.

Acknowledgements. We would like to thank Thomas Body, Alberto Bottino, Tilman Dannert, Tobias Görler, Omar Maj, Stefan Possanner, Erik Sonnendrücker, and Wladimir Zholobenko for stimulating and encouraging discussions. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A. Numerical solution of the dispersion relation

The dispersion relation in Eq. (49) poses a nonlinear, differential eigenvalue problem that has to be solved for the radial electrostatic potential $\hat{\phi}_{m,n,\hat{\omega}}$ and the complex eigenvalue $\hat{\omega}$ simultaneously. As we are not aware of methods to solve the equation analytically we employ a numerical method. First we discretize the differential operators to obtain an ordinary nonlinear eigenvalue problem. Similar to Section 4.2 we choose fourth order centered finite differences to discretize the r derivative. At the boundary, we implement Dirichlet boundary conditions and set $\hat{\phi}_{m,n,\hat{\omega}}(\hat{r}_{\min}) = \hat{\phi}_{m,n,\hat{\omega}}(\hat{r}_{\max}) = 0$. We choose 128 points for the discretization.

The remaining nonlinear eigenvalue problem can be solved by calculating the roots of the characteristic polynomial. We use a determinant free method based on the Newton iteration described in [44, pp. 28–29]. As an initial guess for the Newton iteration we choose $\hat{\omega}_{\text{init}} = -2+4i$. The calculation is done with `Mathematica` [41].

References

[1] P. Stangeby, The Plasma Boundary of Magnetic Fusion Devices, Series in Plasma Physics and Fluid Dynamics, Taylor & Francis, 2000.

[2] A. Loarte, B. Lipschultz, A. Kukushkin, G. Matthews, P. Stangeby, N. Asakura, G. Counsell, G. Federici, A. Kallenbach, K. Krieger, A. Mahdavi, V. Philipps, D. Reiter, J. Roth, J. Strachan, D. Whyte, R. Doerner, T. Eich, W. Fundamenski, A. Herrmann, M. Fenstermacher, P. Ghendrih, M. Groth, A. Kirschner, S. Konoshima, B. LaBombard, P. Lang, A. Leonard, P. Monier-Garbet, R. Neu, H. Pacher, B. Pegourie, R. Pitts, S. Takamura, J. Terry, E. Tsitrone, the ITPA Scrape-off Layer, D. Group, Nucl. Fusion 47 (2007) S203–S263. doi:10.1088/0029-5515/47/6/s04.

[3] A. J. Brizard, T. S. Hahm, Rev. Mod. Phys. 79 (2007) 421–468. doi:10.1103/RevModPhys.79.421.

[4] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, R. B. White, Science 281 (1998) 1835–1837. doi:10.1126/science.281.5384.1835.

[5] S. E. Parker, C. Kim, Y. Chen, Phys. Plasmas 6 (1999) 1709–1716. doi:10.1063/1.873429.

[6] F. Jenko, W. Dorland, M. Kotschenreuther, B. N. Rogers, Phys. Plasmas 7 (2000) 1904–1910. doi:10.1063/1.874014.

[7] W. Dorland, F. Jenko, M. Kotschenreuther, B. N. Rogers, Phys. Rev. Lett. 85 (2000) 5579–5582. doi:10.1103/PhysRevLett.85.5579.

[8] J. Candy, R. E. Waltz, Phys. Rev. Lett. 91 (2003) 045001. doi:10.1103/PhysRevLett.91.045001.

[9] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, J. Vaclavik, L. Villard, J. Comput. Phys. 217 (2006) 395 – 423. doi:10.1016/j.jcp.2006.01.023.

[10] Y. Idomura, M. Ida, T. Kano, N. Aiba, S. Tokuda, Comput. Phys. Comm. 179 (2008) 391 – 403. doi:10.1016/j.cpc.2008.04.005.

[11] S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T. Tran, B. Mcmillan, O. Sauter, K. Appert, Y. Idomura, L. Villard, Comput. Phys. Comm. 177 (2007) 409 – 425. doi:10.1016/j.cpc.2007.04.006.

[12] A. Peeters, Y. Camenen, F. Casson, W. Hornsby, A. Snodin, D. Strintzi, G. Szepesi, Comput. Phys. Comm. 180 (2009) 2650 – 2672. doi:10.1016/j.cpc.2009.07.001, 40 YEARS OF CPC: A celebratory issue focused on quality software for high performance, grid and novel computing architectures.

[13] T. Görler, X. Lapillonne, S. Brunner, T. Dannert, F. Jenko, F. Merz, D. Told, J. Comput. Phys. 230 (2011) 7053 – 7071. doi:10.1016/j.jcp.2011.05.034.

[14] P. Ricci, F. D. Halpern, S. Jolliet, J. Loizu, A. Mosetto, A. Fasoli, I. Furno, C. Theiler, Plasma Phys. Control. Fusion 54 (2012) 124047. doi:10.1088/0741-3335/54/12/124047.

[15] A. Stegmeir, D. Coster, O. Maj, K. Hallatschek, K. Lackner, Comput. Phys. Comm. 198 (2016) 139 – 153. doi:10.1016/j.cpc.2015.09.016.

[16] P. Tamain, H. Bufferand, G. Ciraolo, C. Colin, D. Galassi, P. Ghendrih, F. Schwander, E. Serre, J. Comput. Phys. 321 (2016) 606 – 623. doi:10.1016/j.jcp.2016.05.038.

[17] B. D. Dudson, J. Leddy, Plasma Phys. Control. Fusion 59 (2017) 054010. doi:10.1088/1361-6587/aa63d2.

[18] B. Zhu, M. Francisquez, B. N. Rogers, Comput. Phys. Comm. 232 (2018) 46 – 58. doi:10.1016/j.cpc.2018.06.002.

[19] C. S. Chang, S. Ku, P. Diamond, M. Adams, R. Barreto, Y. Chen, J. Cummings, E. D'Azevedo, G. Dif-Pradalier, S. Ethier, L. Greengard, T. S. Hahm, F. Hinton, D. Keyes, S. Klasky, Z. Lin, J. Lofstead, G. Park, S. Parker, N. Podhorszki, K. Schwan, A. Shoshani, D. Silver, M. Wolf, P. Worley, H. Weitzner, E. Yoon, D. Zorin, J. Phys.: Conf. Ser. 180 (2009) 012057. doi:10.1088/1742-6596/180/1/012057.

[20] M. A. Dorf, M. R. Dorr, J. A. Hittinger, R. H. Cohen, T. D. Rognlien, Phys. Plasmas 23 (2016) 056102. doi:10.1063/1.4943106.

[21] E. L. Shi, G. W. Hammett, T. Stoltzfus-Dueck, A. Hakim, J. Plasma Phys. 83 (2017) 905830304. doi:10.1017/S002237781700037X.

[22] Q. Pan, D. Told, E. L. Shi, G. W. Hammett, F. Jenko, Phys. Plasmas 25 (2018) 062303. doi:10.1063/1.5008895.

[23] E. Caschera, G. Dif-Pradalier, P. Ghendrih, V. Grandgirard, Y. Asahi, N. Bouzat, P. Donnel, X. Garbet, G. Latu, C. Passeron, Y. Sarazin, J. Phys.: Conf. Ser. 1125 (2018) 012006. doi:10.1088/1742-6596/1125/1/012006.

[24] M. Boesl, A. Bergmann, A. Bottino, D. Coster, E. Lanti, N. Ohana, F. Jenko, Phys. Plasmas 26 (2019) 122302. doi:10.1063/1.5121262.

[25] M. Francisquez, T. N. Bernard, B. Zhu, A. Hakim, B. N. Rogers, G. W. Hammett, Phys. Plasmas 27 (2020) 082301. doi:10.1063/5.0005333.

[26] F. Hariri, M. Ottaviani, Comput. Phys. Comm. 184 (2013) 2419 – 2429. doi:10.1016/j.cpc.2013.06.005.

[27] M. Held, M. Wiesenberger, A. Stegmeir, Comput. Phys. Comm. 199 (2016) 29 – 39. doi:10.1016/j.cpc.2015.10.009.

[28] B. Shanahan, B. Dudson, P. Hill, Plasma Phys. Control. Fusion 61 (2018) 025007. doi:10.1088/1361-6587/aaed7d.

[29] P. Paruta, P. Ricci, F. Riva, C. Wersal, C. Beadle, B. Frei, Phys. Plasmas 25 (2018) 112301. doi:10.1063/1.5047741.

[30] W. Zholobenko, T. Body, P. Manz, A. Stegmeir, B. Zhu, M. Griener, G. Conway, D. Coster, F. Jenko, Plasma. Phys. Control. F. (2021). doi:10.1088/1361-6587/abd97e, accepted manuscript.

[31] M. Dorf, M. Dorr, Contrib. to Plasma Phys. 60 (2020) e201900113. doi:10.1002/ctpp.201900113.

[32] A. Stegmeir, A. Ross, T. Body, M. Francisquez, W. Zholobenko, D. Coster, O. Maj, P. Manz, F. Jenko, B. N. Rogers, K. S. Kang, Phys. Plasmas 26 (2019) 052517. doi:10.1063/1.5089864.

[33] B. Scott, J. Smirnov, Phys. Plasmas 17 (2010) 112302. doi:10.1063/1.3507920.

[34] A. Bottino, E. Sonnendrücker, J. Plasma Phys. 81 (2015) 435810501. doi:10.1017/S0022377815000574.

[35] H. Sugama, Phys. Plasmas 7 (2000) 466–480. doi:10.1063/1.873832.

[36] M. Ottaviani, Phys. Lett. A 375 (2011) 1677 – 1685. doi:10.1016/j.physleta.2011.02.069.

[37] A. Stegmeir, D. Coster, O. Maj, K. Lackner, Contrib. Plasma Phys. 54 (2014) 549–554. doi:10.1002/ctpp.201410041.

[38] A. Arakawa, J. Comput. Phys. 1 (1966) 119–143. doi:10.1016/0021-9991(66)90015-5.

[39] Runge-Kutta and Extrapolation Methods, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993, pp. 129–353. doi:10.1007/978-3-540-78862-1.

[40] P. J. Roache, J. Fluids Eng. 124 (2001) 4–10. doi:10.1115/1.1436090.

[41] Wolfram Research, Inc., Mathematica, Version 12.1, Champaign, IL, 2020. URL: `https://www.wolfram.com/mathematica`.

[42] D. Coulette, N. Besse, J. Comput. Phys. 248 (2013) 1 – 32. doi:10.1016/j.jcp.2013.03.065.

[43] G. Latu, M. Mehrenberger, Y. Güçlü, M. Ottaviani, E. Sonnendrücker, J. Sci. Comput. 74 (2018) 1601–1650. doi:10.1007/s10915-017-0509-5.

[44] S. Güttel, F. Tisseur, Acta Numer. 26 (2017) 1–94. doi:10.1017/S0962492917000034.