Summary of Gkeyll's gyrokinetic solver

The Gkeyll team

July 6, 2023

This document is: A high-level summary of the gyrokinetic solver in `Gkeyll`, intended as a reference for (potential) users and developers.

This document is NOT: a detailed description or justification of the model or the algorithms; more in-depth explanations are found in papers listed in the references. It also does not discuss or list the things that are missing.

This document may also be "live", evolving with the code. Also, keep in mind that some features exist in branches of the code, hence some combinations of features may not be readily available but could be easily realized.

1 Collisionless model

`Gkeyll` time-integrates a long-wavelength full-f gyrokinetic model in the symplectic form, evolving the distribution function $f_s(t, \mathbf{x}, v_\parallel, \mu)$ of a species s over time t, guiding center position $\mathbf{x} = (x, y, z)$, parallel (to the equilibrium magnetic field $\mathbf{B}_0 = B_0\,\hat{\mathbf{b}}$) velocity $v_\parallel = \hat{\mathbf{b}} \cdot \mathbf{v}$, and magnetic moment $\mu = m_s v_\perp/(2B_0^2)$. Given a suitable initial condition (IC), the evolution of f_s is governed by the equation [1]

$$\frac{\partial \mathcal{J} f_s}{\partial t} + \boldsymbol{\nabla} \cdot (\mathcal{J} \dot{\mathbf{x}} f_s) + \frac{\partial}{\partial v_\parallel}\left(\mathcal{J} \dot{v}_\parallel{}^H f_s\right) - \frac{\partial}{\partial v_\parallel}\left(\mathcal{J}\frac{q_s}{m_s}\frac{\partial A_\parallel}{\partial t} f_s\right) = \mathcal{J}\mathcal{C}_s + \mathcal{J}\mathcal{S}_s + \mathcal{J}\mathcal{D}_s. \tag{1}$$

The right side of this equation includes contributions due to collisions ($\mathcal{J}\mathcal{C}_s$), sources ($\mathcal{J}\mathcal{S}_s$) and ad-hoc diffusion ($\mathcal{J}\mathcal{D}_s$), described in sections 3-4, 5 and 8, respectively. A factor of $\mathcal{J} = B_\parallel^*$, the Jacobian of the gyrocenter coordinate transformation, appears throughout. The parallel component of the effective magnetic field is $B_\parallel^* = \hat{\mathbf{b}} \cdot \mathbf{B}^*$, where $\mathbf{B}^* = \mathbf{B}_0 + (m_s v_\parallel/q_s)\boldsymbol{\nabla} \times \hat{\mathbf{b}} + \delta\mathbf{B}$ and $\delta\mathbf{B} = \boldsymbol{\nabla} \times A_\parallel \hat{\mathbf{b}}$ is the perpendicular magnetic perturbation given in terms of the magnetic vector potential $A_\parallel$. However, we assume $\hat{\mathbf{b}} \cdot \boldsymbol{\nabla} \times \hat{\mathbf{b}} \approx 0$ such that $B_\parallel^* \approx B_0$.

The collisionless terms on the left side of equation 1 are given as a function of the phase-space speeds or characteristics

$$\dot{\mathbf{x}} = \{\mathbf{x}, H\}, \qquad \dot{v}_\parallel{}^H = \{v_\parallel, H\}, \tag{2}$$

defined using the Poisson bracket

$$\{F, G\} = \frac{\mathbf{B}^*}{m_s B_\parallel^*}\left(\nabla F \frac{\partial G}{\partial v_\parallel} - \frac{\partial F}{\partial v_\parallel}\nabla G\right) - \frac{\hat{\mathbf{b}}}{q_s B_\parallel^*} \times \nabla F \cdot \nabla G. \tag{3}$$

In equation 2 H is the particle Hamiltonian

$$H = \frac{1}{2} m_s v_\parallel^2 + \mu B_0 + q_s \phi - \frac{1}{2} m_s \left|\mathbf{v}_\mathbf{E}\right|^2, \tag{4}$$

which contains a potential energy term depending on the electrostatic potential ϕ. This Hamiltonian also contains a higher-order term with the contribution from the $E \times B$ drift $\mathbf{v}_\mathbf{E} = \hat{\mathbf{b}} \times \boldsymbol{\nabla}_\perp \phi / B_0$.

The potentials ϕ and $A_\parallel$ obey a pair of elliptic equations or boundary value problems (BVPs) known as the Poisson (or quasineutrality) equation

$$-\boldsymbol{\nabla}\cdot\sum_s \frac{n_s m_s}{B_0^2}\boldsymbol{\nabla}_\perp\phi = \sum_s q_s\int \mathrm{d}\mathbf{w}\,\mathcal{J}f_s, \tag{5}$$

and Ampere's equation

$$-\nabla_\perp^2 A_\parallel = \mu_0\sum_s q_s\int \mathrm{d}\mathbf{w}\,v_\parallel\mathcal{J}f_s, \tag{6}$$

with $\mathrm{d}\mathbf{w} = (2\pi/m_s)\,\mathrm{d}v_\parallel\,\mathrm{d}\mu$ and the number density defined as $n_s = \int \mathrm{d}\mathbf{w}\,\mathcal{J}f_s$. Poisson's equation is here written with a nonuniform and time-dependent polarization density weight (n_s in left-side of equation 5), which requires the higher-order $E\times B$ term in the Hamiltonian (equation 4) in order to conserve energy; however most simulations typically use a time-independent polarization density weight ($n_s\to n_{s0}(\mathbf{x})$ or $n_s\to n_{s0}=$ const.), in which case the $E\times B$ term the Hamiltonian is dropped. Note that rather than solving this equation we instead solve [1]

$$\left(-\nabla_\perp^2 + \sum_s \frac{\mu_0 q_s^2}{m_s}\int \mathrm{d}\mathbf{w}\,\mathcal{J}f_s\right)\frac{\partial A_\parallel}{\partial t} = \mu_0\sum_s q_s\int \mathrm{d}\mathbf{w}\,v_\parallel\frac{\partial\left(\mathcal{J}f_s\right)^*}{\partial t}, \tag{7}$$

to compute $\partial_t A_\parallel$ as needed in equation 1, and $(\partial_t\mathcal{J}f_s)^*$ symbolizes the time rate of change of f_s due to all terms in equation 1 except for the $\partial_t A_\parallel$ term. Equations 5-7, as well as equation 1, require suitable boundary conditions (BCs); they are described in section 7 (and those arising from neutral interactions are epxlained in section 4). Such system is discretized with a combination of discontinuous Galerkin (DG) and continuous finite element (FEM) methods as described in section 9. Lastly, the model is cast in a field-aligned coordinate system with a flexible interface to support a variety of coordinates as well as 1D (along z only) and 2D (along x-y or x-z only), see section 2.

2 Geometry

The model in equations 1-7 is cast in a field-aligned coordinate system in which z is the coordinate along the background magnetic field and x and y are coordinates normal to z. The background magnetic field is then written in Clebsch-like form [2]

$$\mathbf{B}_0 = \mathcal{C}(x)\nabla x\times\nabla y. \tag{8}$$

Usually we pick x to be a radial (not necessarily radius) coordinate and y the coordinate normal to x and z. Note that this may be a non-orthogonal coordinate system, and that Cartesian (as was used for LAPD simulations [3]) and helical (as was used for NSTX [4, 1] and HELIMAK [5]) coordinate systems are subsets of such field-aligned coordinates [1].

Transforming equations 1-7 from a Cartesian (or physical $\mathbf{X} = (X, Y, Z)$) coordinates to field-aligned (or computational $\mathbf{x} = (x, y, z)$) coordinates introduces metric coefficients and an additional Jacobian J. Rather than reproducing here all the equations describing the system in field-aligned computational coordinates, we point the reader to section 5.1 of [2], specifically equations 5.32-5.37, 5.40-5.51. Having implemented the equations in non-orthogonal, curvilinear field-aligned form, a framework is needed to compute the geometric factors like metric coefficients and J. We do this at the input file level, where the user can and must provide a computational-to-physical mapping (`mapc2p`). This is a function $\mathbf{X}(\mathbf{x})$ of lab/Cartesian guiding center coordinates $\mathbf{X}$ in terms of the computational/field-aligned coordinates $\mathbf{x}$. Three examples of these mappings are:

[1] Note that the specific helical mapping used in early NSTX and HELIMAK simulations does not exactly conform to this framework [2].

- Cartesian mapping: this is the simplext mapping, consisting of

$$X, Y, Z = x, y, z. \tag{9}$$

- Helical magnetic field: For a helical magnetic field that winds around the Z axis [2], choose z to be the vertical coordinate $z = Z$ or, said another way, z is the projection of the location along the field line onto the vertical axis. We also choose $x = R$ and y is the coordinate normal to x and z, hence

$$\begin{aligned} X &= x \cos\left(\frac{y}{x_0} + \frac{B_\varphi z}{B_v x}\right), \\ Y &= x \sin\left(\frac{y}{x_0} + \frac{B_\varphi z}{B_v x}\right), \\ Z &= z, \end{aligned} \tag{10}$$

  where x_0 is the x-coordinate in the center of the domain, and B_φ and B_v are the toroidal and vertical components of the background magnetic field, respectively.

- Tokamak flux-tube: Consider a background magnetic field represented as $\mathbf{B}_0 = RB_\varphi \nabla\varphi + \nabla\psi \times \nabla\varphi$, where R is the major radius, φ the toroidal angle, and ψ the poloidal flux. One can define a coordinate (ψ, χ, φ) system in which the magnetic field lines appear as straight lines with slope $q(\psi) = \mathbf{B}_0 \cdot \nabla\varphi / (\mathbf{B}_0 \cdot \nabla\chi)$, where χ is a poloidal-like angle. We can the defined the field-aligned coordinates

$$x = \psi - \psi_0, \qquad y = C_y \left(q\chi - \varphi\right) - y_0, \qquad z = \chi, \tag{11}$$

  with $C_y = r_0/q_0$ a normalization constant in terms of the minor radius (r_0) and the safety factor (q_0) in the center of the flux-tube such that $\mathbf{B}_0 = C_y^{-1} \nabla x \times \nabla y$. For a circular cross-section tokamak with a flux-tube centered at R_0, given the functions $q(\psi)$ and $\psi(r) = x$ one can create the `mapc2` function as

$$\begin{aligned} r &= \psi^{-1}(x), \qquad \varphi = q(x) z - C_y^{-1} y, \qquad \epsilon = x/R_0, \\ \cos\theta &= (\cos z - \epsilon)/(1 - \epsilon \cos z), \qquad \sin\theta = [\sqrt{1-\epsilon^2}/(1 - \epsilon\cos z)] \sin z, \\ R &= R_0 + x\cos\theta, \\ X &= R\cos\varphi, \qquad Y = R\sin\varphi, \qquad Z = x\sin\theta, \end{aligned} \tag{12}$$

  where ψ^{-1} stands for the inverse of the function $\psi(x)$. Note that this is just one possible choice for field-aligned coordinates, and many others can be used.

Given a `mapc2p` function that converts computational (field-aligned) into physical (Cartesian) coordinates, `Gkeyll`all the necessary geometric quantities (e.g. metric coefficients) automatically. For example, in field-aligned coordinates the divergence (appearing in the second term of equation 1) becomes [2]

$$\boldsymbol{\nabla} \cdot \mathbf{F} = \frac{1}{J} \sum_i \frac{\partial}{\partial x^i} \left(J F^i\right), \tag{13}$$

where $x^i \in \{x, y, z\}$, and thus the position-space Jacobian $J = \sqrt{\det g_{ij}}$ is needed. So `Gkeyll` computes the entries of the metric tensor

$$g_{ij} = \frac{\partial \mathbf{X}}{\partial x^i} \cdot \frac{\partial \mathbf{X}}{\partial x^j}, \tag{14}$$

using automatic differentiation to differentiate the mapping (`mapc2p`) $\mathbf{X}(\mathbf{x})$ given by the user.

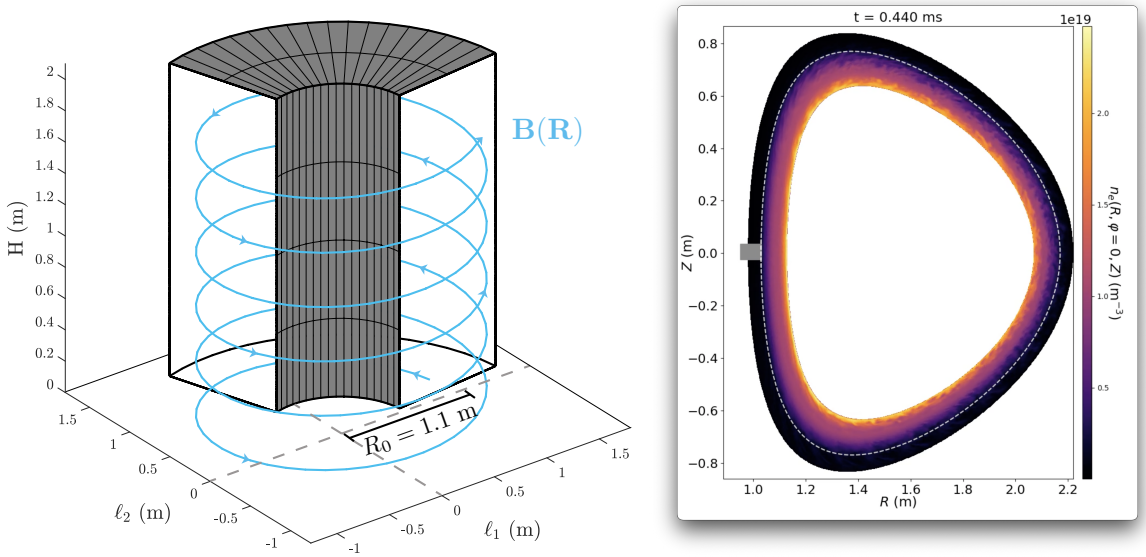

Figure 1: Left: Helical magnetic field. Right: projection onto a poloidal plane of a DIII-D simulation of an inboard-limited plasma that spans regions with open and closed field lines.

Significant flexibility is allowed in the definition of the `mapc2p`) function $\mathbf{X}(\mathbf{x})$. The main restriction is that it is field-aligned, with z being a coordinate parametrizing the location along the field line. Furthermore, the defintion of the magnetic field (and its amplitude $B_0 = B_0(\mathbf{x})$) have to be consistent with the chosen coordinate system. For example, the coordinate mapping in equations 11-12 is used in conjunction with the following function for the amplitude of $\mathbf{B}_0$:

$$B_t = B_{\text{ref}} \frac{R_0}{R}, \qquad B_p = B_t \frac{\epsilon}{q\sqrt{1-\epsilon^2}}, \qquad B_0 = \sqrt{B_t^2 + B_p^2}. \tag{15}$$

Other quantities computed once the determinant is obtained are the covariant components of $\hat{\mathbf{b}}$ as $b_i = g_{iz}/\sqrt{g_{zz}}$.

2.1 In simpler terms, what does this mean?

We presently only support the specification of one $\mathbf{X}(\mathbf{x})$ mapping (`mapc2p`) for the whole computational domain. This allows us to do simulations, as mentioned above, with Cartesian domains, helical magnetic fields, or even flux-tubes in closed-field line regions of the tokamak (provided suitable parallel BCs described in section 7.2.1). It also makes it possible to do simulations in domains with open and closed field lines for a limited plasma. Some of these examples are illustrated in figure 1.

Note that because some of the metric quantities (e.g. the Jacobian J) of a field-aligned coordinate system diverge at the X-point of a diverted tokamak, `Gkeyll` cannot currently get too close or encompass the X-point. The user may, however, provide a $\mathbf{X}(\mathbf{x})$ mapping for a domain away from the X-point even if it corresponds to a diverted plasma.

2.2 Lower dimensions

Until now we have described the gyrokinetic solver in `Gkeyll` in the context of five-dimensional simulations: 3 position dimensions (x, y, z) and 2 velocity dimensions $(v_\parallel, \mu)$. It is however possible to do lower dimensional simulations, either for the purpose of development or to perform lower-fidelity simulations for scientific reasons. This is done by ignoring the variation along disregarded dimensions, or in the case of axisymmetric GK simulations, using the 3D solver and using a single cell in one direction.

Some of these lower-dimensional options are:

- 1x1v: these are simulations in $(z, v_\parallel)$ space, and are typically only done for development purposes.
- 1x2v: simulations in $(z, v_\parallel, \mu)$ space. May be used for development, but we have also used them to study parallel dynamics in LTX and the Wisconsin High-temperature superconducting Axisymmetric Mirror (WHAM).
- 2x2v: simulations in $(x, y, v_\parallel, \mu)$ space. So far we have only used these for validation and development.
- Axisymmetric tokamak simulations: In tokamak simulations x is usually a radial coodinate, z a field aligned coordinate that when mapped to the poloidal plan spans the whole poloidal angle range, and y a binormal coordinate. When we wish to assume perfect axisymmetry, we may ignore the variation along y and effectively run $(x, z, v_\parallel, \mu)$ simulations. Presently, instead of running a 4D simulation, we use the 5D solver with 1 cell in y. Provided that the ICs, the metric quantities and that the drifts have no variation in y (they shouldn't), the the solution will stay homogeneous in y within that 1 cell, making y essentially ignorable. In the future we may change the code to make these axisymmetric simulations genuinely 4D.

3 Plasma collisions

3.1 Collision operators

We can incorporate the effect of intra- and inter-particle plasma collisions via a term on the right-side of equation 1 like

$$\frac{\partial \mathcal{J} f_s}{\partial t} + \cdots = \mathcal{J}\mathcal{C}_s = \mathcal{J} \sum_r C[f_s, f_r]. \tag{16}$$

Presently we have two options for the collision operator $C[f_s, f_r]$. The first option is to use a BGK operator

$$C[f_s, f_r] = \nu_{sr} \left(f_{Msr} - f_s \right), \tag{17}$$

where f_{Msr} is a Maxwellian distribution

$$f_{Msr} = \frac{n_s}{\left(2\pi v_{tsr}^2\right)^{d_v/2}} \exp\left[-\frac{\left(v_\parallel - u_{\parallel sr}\right)^2 + 2\mu B_0/m_s}{2 v_{tsr}^2} \right] \tag{18}$$

constructed with the intermediate parallel drift speed $u_{\parallel sr}$ and thermal speed v_{tsr}, and in terms of the number of physical velocity degrees of freedom accounted for d_v. These two are constructed such that momentum and energy are conserved, and matching the momentum and thermal relaxation rates of the Landau-Fokker-Planck operator [6, 7]. This however is an insufficient set of conditions, hence a free (user-input) parameter $\beta > -1$ remains, and the intermediate speeds are

$$\begin{aligned} u_{\parallel sr} &= u_{\parallel s} - \frac{\alpha_E}{2} \frac{m_s + m_r}{m_s n_s \nu_{sr}} \left(u_{\parallel s} - u_{\parallel r} \right), \\ v_{tsr}^2 &= v_{ts}^2 - \frac{1}{d_v} \frac{\alpha_E}{m_s n_s \nu_{sr}} \left[d_v \left(m_s v_{ts}^2 - m_r v_{tr}^2 \right) - m_r \left(u_{\parallel s} - u_{\parallel r} \right)^2 \right. \\ &\quad \left. + 4 \frac{\alpha_E}{m_s n_s \nu_{sr}} \left(m_s + m_r \right)^2 \left(u_{\parallel s} - u_{\parallel r} \right)^2 \right], \end{aligned} \tag{19}$$

which use

$$\alpha_E = \frac{2 m_s n_s \nu_{sr}\, m_r n_r \nu_{rs}}{m_s n_s \nu_{sr} + m_r n_r \nu_{rs}} \frac{1+\beta}{m_s + m_r}. \tag{20}$$

For intra-species collisions the intermediate speeds are simply $u_{\parallel sr} = u_{\parallel s}$ and $v_{tsr} = v_{ts}$. The primitive moments of each species are computed via discrete equivalents of

$$\begin{aligned} u_{\parallel s} M_{0s} &= M_{1s}, \\ u_{\parallel s} M_{1s} + d_v v_{ts}^2 M_{0s} &= M_{2s}, \end{aligned} \tag{21}$$

written in terms of the velocity moments

$$\begin{aligned} M_{0s} &= \int \mathrm{d}\mathbf{w}\, \mathcal{J} f_s, \\ M_{1s} &= \int \mathrm{d}\mathbf{w}\, v_\parallel \mathcal{J} f_s, \\ M_{2s} &= \int \mathrm{d}\mathbf{w}\, \left(v_\parallel^2 + 2\mu\, B_0/m_s\right) \mathcal{J} f_s. \end{aligned} \tag{22}$$

Unfortunately the BGK operator at present doesn't conserve momentum or energy. This flaw is due to the fact that the projection of equation 18 is done using Gaussian quadrature, which doesn't ensure that the moments of f_{Msr} exactly match the moments corresponding to the input $u_{\parallel sr}$ and v_{tsr}.

A second and more common option is to use a Dougherty operator [8, 9] given by

$$C[f_s, f_r] = \nu_{sr} \left\{ \frac{\partial}{\partial v_\parallel} \left[\left(v_\parallel - u_{\parallel sr}\right) + v_{tsr}^2 \frac{\partial}{\partial v_\parallel} \right] + \frac{\partial}{\partial \mu} \left(2\mu + 2 \frac{m_s v_{tsr}^2}{B_0} \mu \frac{\partial}{\partial \mu} \right) \right\} f_s. \tag{23}$$

The intermediate speeds and v_{tsr} are computed using momentum and energy conservation, and the Fokker-Planck momentum and thermal relaxation rates. Once again, the lack of additional restrictions introduces a free parameter $\beta > -1$. In continuous infinite velocity-space, these intermediate speeds are

$$\begin{aligned} u_{\parallel sr} &= u_{\parallel s} + \frac{\alpha_E}{2} \frac{m_s + m_r}{m_s n_s \nu_{sr}} \left(u_{\parallel r} - u_{\parallel s}\right), \\ v_{tsr}^2 &= v_{ts}^2 + \frac{\alpha_E}{2} \frac{1}{m_s n_s \nu_{sr}} \left[m_r v_{tr}^2 - m_s v_{ts}^2 + \frac{m_r}{d_v} \left(u_{\parallel s} - u_{\parallel r}\right)^2 \right] \end{aligned} \tag{24}$$

3.2 Collision frequencies

Both operators in section 3.1 assume that the collision frequencies are independent of velocity. Making them velocity dependent likely requires a reformulation of the algorithm in order to conserve energy discretely. In the meantime there are four options for setting the collision frequencies ν_{sr}:

1. A constant, homogeneous collision frequency $\nu_{sr} = \text{const}$ chosen by the user.
2. A constant, inhomogeneous user-input collision frequency $\nu_{sr} = \nu_{sr}(\mathbf{x})$.
3. A spatially- and time-varying collision frequency that just re-scales a normalized Spitzer-like collision frequency given by the user. For example, for intra-species collisions the user would provide $\nu_{s0} T_{s0}^{3/2}/n_{s0}$ where the subscript 0 indicates reference values, and in every time step the collision frequency is computed as $\nu_{s0}(T_{s0}^{3/2}/n_0)(T_s^{3/2}(\mathbf{x},t)/n_s(\mathbf{x},t))$.
4. A spatially- and time-varying Spitzer-like collision frequency, computed automatically in each time step using $\nu_{sr} = (n_r/m_s)(1/m_s + 1/m_r) \left\{ q_s^2 q_r^2 \log \Lambda_{sr} / [3(2\pi)^{3/2} \epsilon_0^2)] \right\} / (v_{ts}^2 + v_{tr}^2)$, where $\log \Lambda_{sr}$ is a Coulomb logarithm.

4 Neutral interactions

The collision terms on the right side of equation 1, as well as its BCs, can account for the effect of the plasma interacting with neutral species. The approach to modeling neutral interactions in `Gkeyll` is to leverage our algorithms for full kinetics [10] and model neutrals using a Vlasov equation in $\mathbf{v}_X = (v_X, v_Y, v_Z)$ Cartesian velocity space. That is, for each neutral species s we solve the equation

$$\frac{\partial f_s}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{v}_X f_s = \mathcal{S}_s + \mathcal{C}_s, \tag{25}$$

to obtain the evolution of the neutral distribution function $f_s(t, \mathbf{x}, \mathbf{v}_X)$. The neutral kinetic equation can also incorporate external sources ($\mathcal{S}_s$) specified by the user as is done for the plasma species and described in section 5. The collision term $\mathcal{C}_s$ can be chosen to account for neutral-neutral collisions (e.g. via the operators in section 3.1), but it most importantly used to incorporate the effects of interacting with the plasma species.

Because the gyrokinetic solver uses (curvilinear) field-aligned coordinates (see section 2), we transform equation 25 as is done with the gyrokinetic equation. This means that the equation that we solve is

$$\frac{\partial J f_s}{\partial t} + \sum_i \frac{\partial}{\partial x^i}\left(J v^i f_s\right) = J\mathcal{S}_s + J\mathcal{C}_s, \tag{26}$$

where $x^i \in \{x, y, z\}$ are the field-aligned coordinates and the contravariant curvilinear velocities $v^i = \mathbf{v}_X \cdot \nabla x^i = g^{ij} v_j$ are computed after obtaining the covariant components $v_i = g_{ij} v_X^j$. The transformation of velocity coordinates due to the fact that we use curvilinear position coordinates but Cartesian velocity coordinates must be accounted for in the plasma-neutral interaction terms as well. At the moment these interactions only account for ionization, charge exchange and ion recycling at the wall where field lines terminate [11].

4.1 Ionization

We model the effect of electron-impact ionization ($e^- + n \rightarrow i^+ + 2e^- - E_{iz}$) on the neutral distribution via

$$\mathcal{C}_n^{iz} = -n_e f_n \left\langle \sigma v_e \right\rangle. \tag{27}$$

In this term $\langle \sigma v_e \rangle$ is the average reactivity given by the fit

$$\left\langle \sigma v_e \right\rangle (h) = A \frac{1 + P\, h^{1/2}}{X + h} h^K e^{-h} \times 10^{-6}, \tag{28}$$

where $h = E_{iz}/T_e$ depends on the ionization energy E_{iz} and $\{A, P, X, K\}$ are fitting parameters. This model is accompanied by corresponding collision terms in the ion and electron gyrokinetic equations

$$\begin{aligned} \mathcal{C}_i^{iz} &= n_e f_{Mi}(n_n, u_{\parallel n}, v_{tn}^2) \left\langle \sigma v_e \right\rangle, \\ \mathcal{C}_e^{iz} &= n_n \left\langle \sigma v_e \right\rangle \left(2 f_{Me}\left(n_e, u_{\parallel n}, v_{t,iz}^2\right) - f_e\right). \end{aligned} \tag{29}$$

These terms require constructing Maxwellians $f_{Ms}(n_r, u_{\parallel r}, v_{tr}^2)$ on the velocity grid of species s given the moments $n_r, u_{\parallel r}, v_{tr}^2$, and $v_{t,iz}^2 = v_{te}^2/2 - E_{iz}/(3m_e)$ is the squared thermal speed of resulting lower energy electrons. Note that since the neutrals are defined on $\mathbf{v}_X$ velocity space, the neutral parallel speed used to project the Maxwellian onto the ion or electron grid (e.g. $f_{Mi}(n_n, u_{\parallel n}, v_{tn}^2)$) is computed using

$$n_n u_{\parallel n} = \left(\int \mathrm{d}^3\mathbf{v}_X \ \mathbf{v}_X f_n\right) \cdot \hat{\mathbf{b}}_X = n_n \mathbf{u}_n \cdot \hat{\mathbf{b}}_X, \tag{30}$$

with $\hat{\mathbf{b}}_X = (b_X, b_Y, b_Z)$ written in terms of the Cartesian components of $\hat{\mathbf{b}}$ computed from $b_{Xi} = g_{ij}b^j$ given the contravariant components of $\hat{\mathbf{b}}$: $b^i = g^{ij}b_j$. The thermal speed used to compute a field-aligned Maxwellian with neutral moments is then computed using

$$n_n u_{\parallel n}^2 + 3n_n v_{tn}^2 = M_{2n} = \int \mathrm{d}^3\mathbf{v}_X \; v_X^2 f_n \tag{31}$$

4.2 Charge exchange

The charge exchange process between ions and neutrals of equal mass is simulated via the terms

$$\begin{aligned} \mathcal{C}_n^{cx} &= -\sigma_{cx} V_{cx} \left(n_i f_n - n_n f_{Mn}\left(n_i, u_{\parallel i}, v_{ti}\right)\right), \\ \mathcal{C}_i^{cx} &= \quad \sigma_{cx} V_{cx} \left(n_i f_{Mi}\left(n_n, u_{\parallel n}, v_{tn}\right) - n_n f_i\right). \end{aligned} \tag{32}$$

The charge exchange cross section is computed with a fit; for example, for H-H$^+$ reactions, we use

$$\sigma_{cx} = 1.12 \times 10^{-18} - 7.15 \times 10^{-20} \ln V_{cx}. \tag{33}$$

The relative effective velocity is computed using

$$V_{cx} = \frac{4}{\pi}\left(v_{ti}^2 + v_{tn}^2\right) + \left| u_{\parallel i}\hat{\mathbf{b}}_X - \mathbf{u}_n \right|^2. \tag{34}$$

Lastly, when creating a Maxwellian on the neutral velocity grid with ion primitive moments, we use

$$f_{Mn}(n_i, u_{\parallel i}, v_{ti}) = \frac{n_i}{\left(2\pi v_{ti}^2\right)^{3/2}} \exp\left[-\frac{\left(\mathbf{v}_X - u_{\parallel i}\hat{\mathbf{b}}_X\right)^2}{2v_{ti}^2} \right]. \tag{35}$$

4.3 Wall recycling

A model of ion recycling at the walls where the equilibrium magnetic field lines terminate (in the SOL) may be used. This model sets a BC for the neutrals at the z-boundaries as a function of the incoming ion flux. For example, at the lower z-boundary we set (**MF 2023/04/25: I have some suspicion that these BCs need to be modified in the field-aligned coordinate case**)

$$f_n(x, y, z = z_{\min}, \mathbf{v}_X) = \alpha_b f_n(x, y, z = z_{\min}, v_X, v_Y, -v_Z) + C_{\mathrm{rec}} f_{Mn}\left(1, 0, T_{n,\mathrm{rec}}/m_n\right), \tag{36}$$

where α_b is the reflection coefficient and the unit, non-drifting Maxwellian is created with the user-specified temperature (e.g. Frack-Condon temperatures). The amplitude of this Maxwellian (C_{rec}) is set such that the flux of incoming neutrals due to this BC is equal to a (recycling) fraction of the flux of ions impinging on the wall, that is C_{rec} is given by

$$C_{\mathrm{rec}} \int_0^{v_{Z,\max}} \int_{v_{Y,\min}}^{v_{Y,\max}} \int_{v_{X,\min}}^{v_{X,\max}} \mathrm{d}^3\mathbf{v}_X f_{Mn}\left(1, 0, T_{n,\mathrm{rec}}/m_n\right) = \alpha_{\mathrm{rec}} \int_0^{\mu_{\max}} \int_{\mu_{\min}}^{0} \mathrm{d}v_\parallel \mathrm{d}\mu \, v_\parallel \mathcal{J} f_i(z = z_{\min}) \tag{37}$$

with α_{rec} a user-specified recycling coefficient.

5 Sources

The user can specify a time- and (phase) space- dependent source term $\mathcal{S}_s$ via the input file. There is significant flexibility in the form of this source, and some variance in what the intended purpose may be. Three examples of how this source term has been used in the past are:

- Mimic cross-field transport from the core to the SOL: in SOL-only simulations we have used a Maxwellian source limited to the outboard midplane like

$$\mathcal{S}_s = \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(z-\mu_z)^2}{2\sigma_z^2}\right] \frac{N_s}{\left(2\pi T_{\mathrm{src},s}/m_s\right)^2} \exp\left[-\frac{v_\parallel^2 + 2\mu B_0/m_s}{2T_{\mathrm{src},s}/m_s}\right] \tag{38}$$

  given user-defined locations ($\mu_{x,z}$) and extents ($\sigma_{x,z}$) along x and z, respectively, and a user-defined density source rate (N_s) and temperature (T_{src}), to model the plasma entering the SOL due to transport from the core across the LCFS.

- Mimic ∇B fluxes: in annulus simulations of the tokamak edge there are ∇B drifts across the inner radial (lower x) boundary. At some location along z these drifts take particles away from our domain, and at other z locations they introduce particles into our domain (depending on the direction of $\mathbf{B}_0$, the species charge, and the coordinate mapping used). If absorbing BCs are used on that boundary, we don't naturally reintroduce these particles. But we could instead approximately account for the particles that should drift back into our domain (from the core) via volumetric sources localized near the inner radial boundary. For example, one form used in the past was

$$\mathcal{S}_s = \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{|z|^{3/2}}{2}\right] \sin(kz) f_{Ms}(N_s, 0, T_{\mathrm{src}}/m_s), \tag{39}$$

  for some parameter k and choosing N_s such that the volume integral of this source equals the particle loss rate due to ∇B fluxes at the radial boundaries, for example:

$$\int \mathrm{d}^3\mathbf{x}\, \mathcal{S}_s = \int \mathrm{d}y\, \mathrm{d}z\, J\boldsymbol{\Gamma}_{s,\nabla B}\Big|_{x=x_{\min}} \approx \int \mathrm{d}y\, \mathrm{d}z\, J\frac{n_s T_s}{q_s B_0^2}\hat{\mathbf{b}} \times \nabla B_0\Big|_{x=x_{\min}}. \tag{40}$$

- Mimic neutrals in plasma-only simulations: Self-consistently simulating 6D neutrals in order to incorporate their effect on the plasma can be expensive. If a prior simulation with self-consistent neutrals has been done in a similar parameter regime, one can craft source terms $\mathcal{S}_s$ that approximate the neutrals-related collision terms and recycling sources in that simulation. These sources can then be used in the gyrokinetic equations of a new simulation without self-consistent neutrals.

6 Initial conditions

As with sources, there is significant run-time flexibility for crafting ICs within the user input file. In principle the user can specify nearly any function of phase space, or use the Lua I/O functions to read ICs from a file. Aside from that, `Gkeyll` has pre-packaged options for gyrokinetic simulations that wish to start with Maxwellians, as described below.

6.1 Local Maxwellians

The user can provide the particle number density ($n_s(\mathbf{x})$), parallel speed ($u_{\parallel s}(\mathbf{x})$) and temperature ($T_s(\mathbf{x}) = m_s v_{ts}^2$) so that `Gkeyll` initializes the simulation with an isotropic Maxwellian

$$f_s(t=0,\mathbf{x},v_\parallel,\mu) = \begin{cases} \left[n_s/\left(2\pi v_{ts}^2\right)^{1/2}\right] \exp\left[-\left(v_\parallel - u_{\parallel s}\right)^2/(2v_{ts}^2)\right] & 1v \\ \left[n_s/\left(2\pi v_{ts}^2\right)^{3/2}\right] \exp\left\{-\left[\left(v_\parallel - u_{\parallel s}\right)^2 + 2\mu B_0/m_s\right]/(2v_{ts}^2)\right\} & 2v. \end{cases} \tag{41}$$

It is also possible to request bi-Maxwellians, which require the user to provide separate parallel ($T_{\parallel s}(\mathbf{x})$) and perpendicular ($T_{\perp s}(\mathbf{x})$) temperatures and initialize with

$$f_s(t=0,\mathbf{x},v_\parallel,\mu) = \frac{n_s}{\left(2\pi T_{\parallel s}^{1/3} T_{\perp s}^{2/3}\right)^{3/2}} \exp\left[-\frac{\left(v_\parallel - u_{\parallel s}\right)^2}{2T_{\parallel s}/m_s} - \frac{2\mu B_0/m_s}{2T_{\perp s}/m_s}\right]. \tag{42}$$

6.2 Canonical Maxwellians

In tokamak simulations, a "local maxwellian" initial distribution function $f(\psi, v_\parallel, \mu)$ is not a true equilibrium because it is a function of ψ, the poloidal flux, which is not a constant of motion [12]. ∇B drifts cause vertical charge separation (ion and electron banana widths are different, etc), which will result in a large radial electric field and strong $E \times B$ flows which prevent turbulence. In order to resolve this (get rid of this charge separation), a parallel flow (like the Pfirsch-Schluter current) is necessary. The local Maxwellian, f_{LM} is defined as

$$f_{LM}(\psi, v_\parallel, \mu) = \frac{n(\psi)}{(2\pi T(\psi)/m_s)^{3/2}} e^{-E/T(\psi)} \tag{43}$$

where $E = m_s v_\parallel^2/2 + \mu B_0$ and $n(\psi)$, $T(\psi)$ are the desired density and temperature profiles.

A "canonical Maxwellian" is a function of the toroidal momentum $P_\varphi = m_s R v_\phi + q_s \psi$, a constant of motion, instead of ψ, so it is a true equilibrium. It includes a parallel flow to balance the aforementioned charge separation. For a canonical Maxwellian, temperature and density profiles are no longer well-defined and could be very different from the desired profiles. To minimize the difference between the desired density and temperature profiles and the effective profiles obtained by taking moments of the canonical Maxwellian as well as to minimize the parallel flow, we can add a correction term which is a function only of the constants of the motion to P_φ. The correction term is an estimate of the average radial excursion of the particle orbits from a flux surface.

$$\hat{P}_\varphi = q_s\psi + m_s R v_\phi - sgn(v_\parallel) m_s \sqrt{\frac{2}{m}} R_0 \sqrt{E - \mu B_{max}}\,\Theta(E - \mu B_{max}) \tag{44}$$

where $\hat{P}_\varphi$ is the corrected toroidal momentum. The canonical maxwellian is then

$$f_{CM}(\hat{P}_\varphi, v_\parallel, \mu) = \frac{n(\hat{P}_\varphi)}{(2\pi T(\hat{P}_\varphi)/m_s)^{3/2}} e^{-E/T(\hat{P}_\varphi)} \tag{45}$$

In practice, ψ may not be used as the actual radial coordinate x for simulations. In this case it makes sense to invert $\psi(x)$ to get a relationship $x = \hat{x}(\psi)$ [13]. The canonical radial coordinate is then $\bar{x} = \hat{x}(\hat{P}_\varphi)$, and the canonical Maxwellian is

$$f_{CM}(\bar{x}, v_\parallel, \mu) = \frac{n(\bar{x})}{(2\pi T(\bar{x})/m_s)^{3/2}} e^{-E/T(\bar{x})} \tag{46}$$

7 Boundary conditions

In velocity space `Gkeyll` generally uses zero-flux BCs; this ensures that number of particles, momentum density and energy are conserved. A slight departure from this BC takes place in the Dougherty collision operator, although that is also formulated in a way that conserves particle number, momentum and energy density. On position-space boundaries `Gkeyll` can use periodic, absorbing or reflecting BCs. More complex BCs for specific problems and geometries are also available, as described below.

7.1 Sheath entrance

In the scrape-off layer `Gkeyll`'s z-domain ends at the sheath entrance, since the sheaths are not quasineutral, which is assumed in gyrokinetics. At the sheath entrance we impose conducting sheath BCs [14]. These BCs obtain the sheath potential $\phi_s = \phi(z = z_{\min}, z_{\max})$ from the solution to the gyrokinetic Poisson equation 5. Given the sheath potential, and assuming a wall potential ϕ_w, we reflect all the particles with a parallel velocity that cannot overcome this potential hill, i.e. $m_s v_\parallel^2/2 \leq -q_s(\phi_s - \phi_w)$. For the typical case of $\phi_s > \phi_w$, this means that ions free-stream to and are absorbed by the wall, as are high energy electrons with $m_e v_\parallel^2/2 > e(\phi_s - \phi_w)$, while a population of low energy electrons is reflected by the potential drop across the sheath (see figure 2).

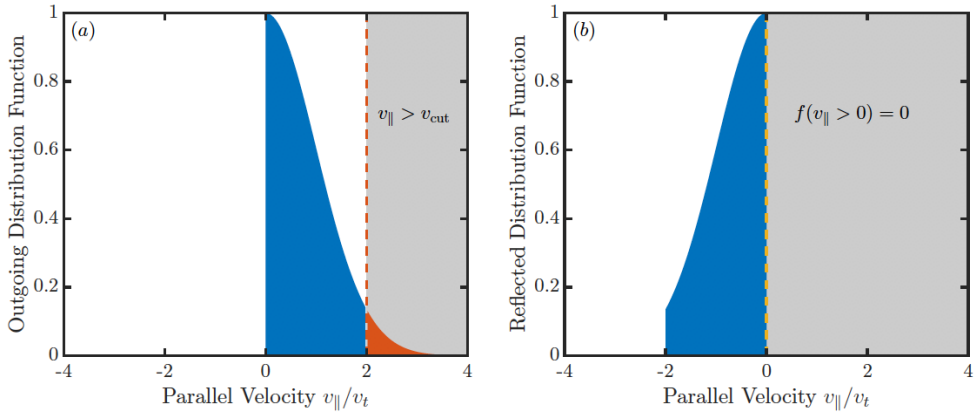

Figure 2: Schematic of the outgoing (a) and reflected (b) electron populations at the upper z-boundary where conducting sheath BCs are applied. Electrons with $m_e v_\parallel^2/2 > e(\phi_s - \phi_w)$ (orange) are absorbed by the wall.

Insulating or logical sheath BCs have been implemented in earlier versions of `Gkeyll` [14], but have fallen out of favor due to lower robustness and possibly weaker physical relevance.

7.2 Tokamak closed-flux region

`Gkeyll` can do simulations of the tokamak core (i.e. the closed-flux region) using a core-only flux-tube, or simulations that span part of the core and the SOL for limited (but possibly shaped) plasmas. In these cases the finite extent of our simulation domain along the z direction calls for special BCs, since these z-ends may be at physically distinct locations within the plasma (so periodic, absorbing or reflecting BCs would be inappropriate). For a similar reason, the inner radial boundary, which also lies in a place within the (actively evolving) plasma may call for special BCs. We described the options currently available below.

7.2.1 Parallel direction

The parallel BCs in the closed-flux region are the well known twist-shift BCs [15]. For the coordinate mapping in equations 11-12, this BC at the upper z-boundary is [15, 16]

$$f_s(x, y, z = z_{\max}) = f_s(x, y - L_z C_y q, z = z_{\min}), \tag{47}$$

where $L_z = z_{\max} - z_{\min}$. This is typically used in combination with periodic BCs in y, such that if the shifted coordinate $y - L_z C_y q$ lies outside of our domain, it simply wraps around to our domain (or alternatively we say it is located in another flux-tube that is an exact copy of ours).

Simulations with open and closed field lines (see figure 1(right)) are made possible by a combination of these twist-shift BCs in the closed flux-region up to the last closed-flux surface (LCFS), and sheath BCs in the open field line region past the LCFS.

7.2.2 Inner radius

As mentioned in section 5, annular simulations of the tokamak edge have to deal with the fact that there are ∇B-drifts across the inner radial boundary that may take particles away from out domain. These particles should re-enter at a different z location of our domain. To mimic this behavior one may use volumetric sources as described in section 5. An alternative is to place a (canonical) Maxwellian (see section 6.2 for a description of canonical Maxwellians) with a user-specified density and temperature in the inner radial ghost cell. This allows the ∇B drifts to naturally carry particles with a canonical Maxwellian form from the core into our domain, and avoid dangerous density holes that could form if we don't model the reentry of these ∇B-drifting particles.

7.3 Field equations

The Poisson and Ampere field equations 5-7 require BCs on the boundaries of the perpendicular plane (x, y). In the y direction the most common choice is to use periodicity. The x directions may employ either Dirichlet or Neumann BCs. Most of our simulations so far have used $\phi = 0$ and $A_{\parallel} = 0$ Dirichlet BCs along x, although some more recent simulations have started using a nonzero Dirichlet or a homogenous Neumann BC for ϕ.

8 Other bells & whistles

A few other features have been implemented for very specific types of simulations. Some of these are described below.

8.1 Ad-hoc diffusion

The user may choose to include an additional diffusion term $\mathcal{D}_s$ on the right side of the kinetic equation. This could be diffusion along position- or velocity-space. Two common uses of this functionality are:

- Add hyperdiffusion to mitigate numerical instabilities in velocity space. For this purpose the diffusion term has a form

$$J\mathcal{D}_s = J(-1)^{n-1} D_s \nabla_{\mathbf{v}}^{2k} f_s, \qquad k \in \{1, 2, 3\}, \tag{48}$$

  and D_s is a user-specified diffusion operator. Note that although it may improve stability in some cases, it does not conserve momentum or energy and can produce solutions that are overly diffusive.

- Model perpendicular transport in axisymmetric tokamak simulations: Since axisymmetric simulations don't have turbulence to drive perpendicular transport, we implemented a model of diffusive transport via

$$J\mathcal{D}_s = J \sum_i \frac{\partial}{\partial x^i} D_i(x^i) \frac{\partial f_s}{\partial x^i}, \tag{49}$$

  where $x^i \in \{x, y, z\}$ and $D_i(x^i)$ are user-defined diffusion coefficients.

8.2 Adiabatic electrons (linear)

For some validation purposes we have included an option to run with linearized adiabatic electrons. This model does not evolve the electron gyrokinetic equation and simply assumes

$$n_e = \int \mathrm{d}\mathbf{w}\, \mathcal{J} f_e = n_0 \frac{e\phi}{T_{e0}} \tag{50}$$

to compute the electron contribution to the right-side of the Poisson equation 5. This effectively changes the field problem from ϕ from a Poisson to a Helmholtz problem.

8.3 Boltzmann, isothermal electrons on open field lines with insulating, logical sheath BCs

For the purposes of validation, benchmarking, and providing an additional complimentary view of a 1D system, we have implemented a solver that assumes isothermal Boltzmann electrons, quasineutrality and ambipolar particle fluxes at the sheath entrance [17]. In this model we only evolve the ion kinetic equation, from which we obtain ion particle fluxes at the sheath entrance Γ_i. Then the sheath potential is computed via

$$\phi_s = \frac{T_{e0}}{e} \ln\left(\frac{n_i v_{te0}}{\sqrt{2\pi}\Gamma_i}\right). \tag{51}$$

Having obtained this sheath entrance potential, we can combine the Boltzmann electron assumption with quasineutrality to obtain the potential throughout the domain as

$$\phi = \phi_s + \frac{T_{e0}}{e} \ln\left(\frac{n_i}{n_{is}}\right), \tag{52}$$

where n_{is} is the ion density at the sheath entrance.

9 Discretization

`Gkeyll` uses a modal discontinuous Galerkin (DG) discretization for the gyrokinetic equation, and a continuous finite element method (FEM) discretiztion to solve the field equations 5-7. Time integration is done by treating every term explicitly and using a strong stability-preserving 3rd order Runge-Kutta algorithm (SSP-RK3). Explaining these algorithms is beyond the scope of this document; the reader may refer to the works listed in the references or contact the developers directly to obtain such explanations. In particular, you may wish to see

- [1, 2] for details on the discretization of collisionless terms.
- [10, 18] for details on our DG methods.
- [8, 9] for details on our treatment of the Dougherty collision operator.
- [16] for our DG implementation of the twist-shift BCs for tokamak closed-flux regions.

Other useful details may also be found in Eric Shi [14] and Tess Bernard's [19] dissertations.

References

[1] N. R. Mandell, A. Hakim, G. W. Hammett, and M. Francisquez. Electromagnetic full-f gyrokinetics in the tokamak edge with discontinuous Galerkin methods. Journal of Plasma Physics, 86, 2020.

[2] N. R. Mandell. Magnetic Fluctuations in Gyrokinetic Simulations of Tokamak Scrape-Off Layer Turbulence. PhD thesis, Princeton University, 2021.

[3] E. L. Shi, G. W. Hammett, T. Stoltzfus-Dueck, and A. Hakim. Gyrokinetic continuum simulation of turbulence in a straight open-field-line plasma. Journal of Plasma Physics, 83(3):905830304, 2017.

[4] E. L. Shi, G. W. Hammett, T. Stoltzfus-Dueck, and A. Hakim. Full-f gyrokinetic simulation of turbulence in a helical open-field-line plasma. Physics of Plasmas, 26(1), 01 2019. 012307.

[5] T. N. Bernard, E. L. Shi, K. W. Gentle, A. Hakim, G. W. Hammett, T. Stoltzfus-Dueck, and E. I. Taylor. Gyrokinetic continuum simulations of plasma turbulence in the texas helimak. Physics of Plasmas, 26(4):042301, 2019.

[6] J. M. Greene. Improved Bhatnagar-Gross-Krook model of electron-ion collisions. The Physics of Fluids, 16(11):2022–2023, 1973.

[7] J. R. Haack, C. D. Hauck, and M. S. Murillo. A Conservative, Entropic Multispecies BGK Model. Journal of Statistical Physics, 168(4):826–856, 2017.

[8] M. Francisquez, T.N. Bernard, N.R. Mandell, G.W. Hammett, and A. Hakim. Conservative discontinuous Galerkin scheme of a gyro-averaged Dougherty collision operator. Nuclear Fusion, 60(9):096021, aug 2020.

[9] Manaure Francisquez, James Juno, Ammar Hakim, Gregory W. Hammett, and Darin R. Ernst. Improved multispecies Dougherty collisions. Journal of Plasma Physics, 88(3):905880303, 2022.

[10] J Juno, Ammar Hakim, J TenBarge, E Shi, and William Dorland. Discontinuous Galerkin algorithms for fully kinetic plasmas. J. Comput. Phys., 353:110–147, 2018.

[11] T. N. Bernard, F. D. Halpern, M. Francisquez, N. R. Mandell, J. Juno, G. W. Hammett, A. Hakim, G. J. Wilkie, and J. Guterl. Kinetic modeling of neutral transport for a continuum gyrokinetic code. Physics of Plasmas, 29(5), 05 2022. 052501.

[12] P. Angelino, A. Bottino, R. Hatzky, S. Jolliet, O. Sauter, T. M. Tran, and L. Villard. On the definition of a kinetic equilibrium in global gyrokinetic simulations. Physics of Plasmas, 13(5):052304, 2006.

[13] Guilhem Dif-Pradalier, Virginie Grandgirard, Yanick Sarazin, Xavier Garbet, and Ph Ghendrih. Defining an equilibrium state in global full-f gyrokinetic models. Communications in Nonlinear Science and Numerical Simulation, 13(1):65–71, 2008.

[14] E. Shi. Gyrokinetic continuum simulation of turbulence in open-field-line plasmas. PhD thesis, Princeton University, 2017.

[15] M. A. Beer, S. C. Cowley, and G. W. Hammett. Field-aligned coordinates for nonlinear simulations of tokamak turbulence. Physics of Plasmas, 2(7):2687–2700, 1995.

[16] Manaure Francisquez, Noah R. Mandell, Ammar Hakim, and Gregory W. Hammett. Mapped discontinuous galerkin interpolations and sheared boundary conditions, 2021.

[17] E. L. Shi, A. H. Hakim, and G. W. Hammett. A gyrokinetic one-dimensional scrape-off layer model of an edge-localized mode heat pulse. Physics of Plasmas, 22(2):022504, 2015.

[18] A. Hakim, G. Hammett, E. Shi, and N. Mandell. Discontinuous galerkin schemes for a class of hamiltonian evolution equations with applications to plasma fluid and kinetic problems, 2019.

[19] T. N. Bernard. Discontinuous Galerkin modeling of plasma turbulence in a simple magnetized torus. PhD thesis, University of Texas at Austin, 2019.