Implementation Notes for Gyrokinetic Particle Pusher

Norman M. Cao

January 27, 2026

1 Equations of motion and geometry

1.1 Gyrokinetic characteristic equations (electrostatic)

We aim to push particles in electrostatic gyrokinetics. The equations of motion are given by

$$B_\parallel^* \dot{\mathbf{R}} = \frac{1}{q}\hat{\mathbf{b}} \times \nabla H + v_\parallel \mathbf{B}^* \tag{1a}$$

$$B_\parallel^* \dot{p}_\parallel = -\mathbf{B}^* \cdot \nabla H \tag{1b}$$

$$\dot{\mu} = 0 \tag{1c}$$

With some definitions

$$\hat{\mathbf{b}} := \mathbf{B}/B \tag{2a}$$

$$\mathbf{B}^* := \mathbf{B} + \nabla \times (p_\parallel \hat{\mathbf{b}}/q) \tag{2b}$$

$$B_\parallel^* := \hat{\mathbf{b}} \cdot \mathbf{B} \tag{2c}$$

$$H = p_\parallel^2/2m + \mu B + q\mathcal{J}[\Phi] \tag{3a}$$

$$v_\parallel := \partial_{p_\parallel} H = p_\parallel/m \tag{3b}$$

and m, q are the species mass and charge respectively.

1.2 Gyrokinetic characteristic equations (electromagnetic)

$$B_\parallel^* \dot{\mathbf{R}} = \frac{1}{q_s}\hat{\mathbf{b}} \times \frac{\partial H}{\partial \mathbf{R}} + \frac{\mathbf{B}^*}{m_s}\frac{\partial H}{\partial u_\parallel} \tag{4a}$$

$$B_\parallel^* \dot{u}_\parallel = -\frac{\mathbf{B}^*}{m_s} \cdot \frac{\partial H}{\partial \mathbf{R}} - \frac{q_s B_\parallel^*}{m_s}\frac{\partial A_\parallel^{(s)}}{\partial t} \tag{4b}$$

$$H = \frac{1}{2m_s}(m_s u_\parallel - q_s A_\parallel^{(h)})^2 + \mu B + q_s \Phi \tag{5a}$$

$$\frac{\partial H}{\partial \mathbf{R}} = \mu \nabla B + q_s \nabla \Phi - \frac{q_s}{m_s}(m_s u_\parallel - q_s A_\parallel^{(h)})\nabla A_\parallel^{(h)} \tag{5b}$$

$$\frac{\partial H}{\partial u_\parallel} = m_s u_\parallel - q_s A_\parallel^{(h)} \tag{5c}$$

$$\mathbf{B}^* = \mathbf{B} + \nabla \times ([m_s u_\parallel / q_s + A_\parallel^{(s)}]\hat{\mathbf{b}}) \tag{5d}$$

In the current implementation, we only include the Hamiltonian portion $A_\parallel^{(h)}$ in the particle pusher. Note that the generalized potential $\Phi \approx \phi - \frac{m_s}{2q_s}|\mathbf{v}_E|^2$ where $\mathbf{v}_E = \frac{\hat{\mathbf{b}} \times \nabla \phi}{B}$ is also written here for completeness, although we do not use it.

1.3 Cylindrical coordinates

We primarily work in a right-handed (R, φ, Z) cylindrical coordinate system. Thus, φ points into the (R, Z) plane. Recall that for a path $\mathbf{R} = (R(t), \varphi(t), Z(t))$, we have that

$$\dot{\mathbf{R}} = \dot{R}\hat{\mathbf{R}} + \dot{Z}\hat{\mathbf{Z}} + R\dot{\varphi}\hat{\boldsymbol{\varphi}}$$
$$\ddot{\mathbf{R}} = (\ddot{R} - R\dot{\varphi}^2)\hat{\mathbf{R}} + \ddot{Z}\hat{\mathbf{Z}} + (R\ddot{\varphi} + 2\dot{R}\dot{\varphi})\hat{\boldsymbol{\varphi}}$$

Typically we will work with the orthonormal basis of unit vectors $(\hat{\mathbf{R}}, \hat{\boldsymbol{\varphi}}, \hat{\mathbf{Z}})$. From the above expressions, the velocity acts on coordinates as

$$\dot{R} = \dot{\mathbf{R}} \cdot \hat{\mathbf{R}} \qquad \dot{Z} = \dot{\mathbf{R}} \cdot \hat{\mathbf{Z}} \qquad \dot{\varphi} = (\dot{\mathbf{R}} \cdot \hat{\boldsymbol{\varphi}})/R$$

then, the acceleration acts on components of the velocity vector (in the orthonormal basis) as

$$(\dot{\mathbf{R}} \cdot \hat{\mathbf{R}})' = \ddot{\mathbf{R}} \cdot \hat{\mathbf{R}} + R\dot{\varphi}^2$$
$$(\dot{\mathbf{R}} \cdot \hat{\mathbf{Z}})' = \ddot{\mathbf{R}} \cdot \hat{\mathbf{Z}}$$
$$(\dot{\mathbf{R}} \cdot \hat{\boldsymbol{\varphi}})' = \dot{R}\dot{\varphi} + R\ddot{\varphi} = \ddot{\mathbf{R}} \cdot \hat{\boldsymbol{\varphi}} - \dot{R}\dot{\varphi}$$

1.4 Magnetic field

We use the following representation of the magnetic field and current

$$\mathbf{B} = F(\psi)\nabla\varphi + \nabla\varphi \times \nabla\psi = \frac{F(\psi)\hat{\boldsymbol{\varphi}} + \hat{\boldsymbol{\varphi}} \times \nabla\psi}{R}$$
$$\nabla \times \mathbf{B} = F'\nabla\psi \times \nabla\varphi + \nabla \times (\nabla\varphi \times \nabla\psi)$$

where ψ is the poloidal flux. Note the second term in the current can be evaluated using an in-plane curl

It's useful to have the following representations for certain terms in the gyrokinetic equation:

$$\nabla B = \frac{\nabla(RB) - B\nabla R}{R}$$
$$2RB\nabla(RB) = \nabla(R^2B^2) = \nabla(F^2 + |\nabla\psi|^2) = 2F'\nabla\psi + 2\,\mathrm{Hess}[\psi]\nabla\psi$$
$$\nabla \times \hat{\mathbf{b}} = \frac{B(\nabla \times \mathbf{B}) - (\nabla B) \times \mathbf{B}}{B^2}$$

note that these can be written purely in terms of analytic derivatives of $\psi(R, Z)$ and $F(\psi)$.

2 Eikonal Representation of Fields

2.1 Flux Coordinates

First, we define the flux coordinates that we adopt in order to represent modes. We use ψ as the radial coordinate and φ as the toroidal coordinate. Now, we define a binormal coordinate θ by a "UV mapping" strategy. Namely, we define fields $(u(R, Z), v(R, Z))$ such that $\theta = \arctan(v/u)$. These UV fields are defined to satisfy the following:

1. $u = v = 0$ at the magnetic axis
2. $\nabla\theta = 0$ at the two primary magnetic X-points
3. At each magnetic null, the Hessian $H[\psi]$ is congruent to a diagonal matrix D with entries ± 1, $H[\psi] = SDS^T$; this is Sylvester's law of inertia. We constrain the Hessian $H[\theta] = SAS^T$ where $A_{11} = A_{22} = 0$ and $A_{12} = A_{21} = a$.

These help ensure that θ acts as a good "conjugate coordinate" to ψ.

2.2 Clebsch Representation of Magnetic Field

Consider a Clebsch representation of the magnetic field

$$\mathbf{B} = \nabla\psi \times \nabla\alpha$$

Any function $S(\psi, \alpha)$ automatically satisfies $\mathbf{B} \cdot \nabla S = 0$. We can choose $\alpha = \alpha_\Phi(\psi, \theta) - \varphi$. Note that α_Φ is the component related to the toroidal flux.

To compute α_Φ, notice that

$$\mathbf{B} = \frac{\partial \alpha_\Phi}{\partial \theta} \nabla\psi \times \nabla\theta + \nabla\varphi \times \nabla\psi$$

In other words,

$$\frac{\partial \alpha_\Phi}{\partial \theta} = \frac{\mathbf{B} \cdot \nabla\varphi}{(\nabla\psi \times \nabla\theta) \cdot \nabla\varphi}$$

Thus, by integrating along curves of constant ψ and varying θ, it is possible to compute α_Φ.

We make a brief remark about singularities. Let $\mathbf{x} = (R - R_0, Z - Z_0)$, and let $\boldsymbol{\xi} = S^T\mathbf{x}$. Near poloidal magnetic field nulls, we have

$$\psi \approx \frac{1}{2}\mathbf{x}^T H[\psi]\mathbf{x} = \frac{1}{2}\boldsymbol{\xi}^T D\boldsymbol{\xi}$$
$$\theta \approx \frac{1}{2}\mathbf{x}^T H[\theta]\mathbf{x} = \frac{1}{2}\boldsymbol{\xi}^T A\boldsymbol{\xi}$$

2.3 Eikonal Ansatz

We take an eikonal ansatz of the form

$$\delta\phi = \exp(i(\omega t - n\alpha))g(\alpha_\Phi, \psi) \tag{6}$$

In the JAX version, automatic differentiation is used to compute gradients of $\delta\phi$.

3 Monte Carlo Solution of Transport Equations

In this section, we discuss the strategy for using Monte Carlo in combination with the Feynman-Kac formula to solve certain transport equations.

3.1 Problem Setup

First, we discuss our strategy for dealing with the collision operator. Let's start with the nonlinear Landau collision operator, which is a Fokker-Planck operator of the form

$$C_{ab} = \frac{\partial}{\partial \mathbf{v}_a} \cdot \left[\mathbf{B}_{ab} f_a - \mathbf{D}_{ab} \cdot \frac{\partial f_a}{\partial \mathbf{v}_a}\right] \tag{7}$$

See `https://farside.ph.utexas.edu/teaching/plasma/Plasma/node38.html` for definitions of $\mathbf{B}_{ab}, \mathbf{D}_{ab}$ (note we have absorbed the mass factor into these definitions).

The gyrokinetic Vlasov equation can be written in conservative form

$$\frac{\partial (B_\parallel^* f_a)}{\partial t} + \nabla \cdot (\dot{\mathbf{R}} B_\parallel^* f_a) + \frac{\partial (\dot{u}_\parallel B_\parallel^* f_a)}{\partial u_\parallel} = \sum_b B_\parallel^* C_{ab}^{gk}[f_a, f_b] \tag{8}$$

where C_{ab}^{gk} is the gyroaveraged collision operator. We assume that we can write this in Fokker-Planck form.

The key strategy we take is to ignore the conservation properties of the collision operator, and simply treat it as a velocity-space diffusion operator. Similar to how the fields are not computed self-consistently from the particles, the collisional drag and diffusion are not computed self-consistently from the particle distributions. Energy/momentum is not conserved in the test particle picture even in the collisionless case, as they exchange energy with the fields. However, if the time-dependent fields were known, then the test particle picture would be exact.

Let's generalize this situation a bit. Suppose $F_s(z)$ satisfies the Fokker-Planck equation

$$\frac{\partial F_s}{\partial t} + \frac{\partial}{\partial \mathbf{z}} \cdot \left[\mathbf{b}_s[F]F_s - \mathbf{D}_s[F] \cdot \frac{\partial F_s}{\partial \mathbf{z}}\right] = 0$$

where $\mathbf{z}$ is the phase space coordinate, and $\mathbf{b}_s, \mathbf{D}_s$ are the drift and diffusion coefficients respectively, depending on $F = F_1, ..., F_n$. Supposing that this nonlinear Fokker-Planck equation satisfies some (possibly nonlinear) conservation law $\frac{\mathrm{d}}{\mathrm{d}t}Q[F] = 0$, we can consider

$$\frac{\partial G_s}{\partial t} + \frac{\partial}{\partial \mathbf{z}} \cdot \left[\mathbf{b}_s[H]G_s - \mathbf{D}_s[H] \cdot \frac{\partial G_s}{\partial \mathbf{z}}\right] = 0$$

While in general we no longer expect $\frac{\mathrm{d}}{\mathrm{d}t}Q[G] = 0$, the conservation law will still hold if $H = G$. Thus, we can think about solving just the transport part of the equation with a fixed background H, and potentially iterating to self-consistency.

3.2 Periodic Solutions to Transport Equations

We now focus on

$$\frac{\partial G}{\partial t} + \frac{\partial}{\partial \mathbf{z}} \cdot \left[\mathbf{b}(\mathbf{z},t)G - \mathbf{D}(\mathbf{z},t) \cdot \frac{\partial G}{\partial \mathbf{z}}\right] = \frac{\partial G}{\partial t} - \mathcal{L}(t)[G] = S(\mathbf{z},t) \tag{9}$$

subject to the boundary conditions

$$G = g(\mathbf{z},t) \text{ on } \partial\Omega$$

where $\mathbf{b}(\mathbf{z},t), \mathbf{D}(\mathbf{z},t), S(\mathbf{z},t), g(\mathbf{z},t)$ are periodic in time with period T. We want to find a periodic solution $G(\mathbf{z},t+T) = G(\mathbf{z},t)$.

The first trick we do is to convert the inhomogeneous boundary conditions to homogeneous ones. Let $G = U + V$ where V is a periodic function which satisfies the boundary conditions, but not necessarily the PDE. Then, U satisfies the PDE, but with homogeneous boundary conditions and a modified (but still time-periodic) source term, which we also call S. Thus, we focus on solving the homogeneous Dirichlet boundary condition case.

Let $\mathcal{P}_s^t$ be the evolution operator taking the solution of the homogeneous Dirichlet problem from time s to time t. Starting with initial data $U(0)$, we have (via Duhamel's principle)

$$U(t) = \mathcal{P}_0^t[U(0)] + \int_0^t \mathcal{P}_\tau^t[S(\tau)]\,\mathrm{d}\tau$$

To enforce periodicity, we require $U(0) = U(T)$, so we can write

$$(\mathcal{I} - \mathcal{P}_0^T)U(0) = \int_0^T \mathcal{P}_\tau^T[S(\tau)]\,\mathrm{d}\tau \tag{10}$$

This equation allows us to solve for the initial data $U(0)$ which result in a periodic solution.

Note that formally, $(\mathcal{I} - A)^{-1} = \sum_{m=0}^\infty A^m$ for any operator A with spectral radius less than 1. This gives an explicit formal series representation of the solution:

$$\begin{aligned} U(t) &= \mathcal{P}_0^t \sum_{m=0}^\infty (\mathcal{P}_0^T)^m \left[\int_0^T \mathcal{P}_\tau^T[S(\tau)]\,\mathrm{d}\tau\right] \\ &= \int_0^T \sum_{m=0}^\infty \mathcal{P}_\tau^{t+(m+1)T}[S(\tau)]\,\mathrm{d}\tau \end{aligned}$$

Now, for kinetic problems, we are interested in computing temporally-averaged spatially-weighted velocity moments of G; these will take the form of linear functionals of G that look like

$$\langle \eta, G \rangle = \int_0^T \int_\Omega \eta(\mathbf{z}, t) G(\mathbf{z}, t) \, \mathrm{d}\mathbf{z} \, \mathrm{d}t$$

Consider the adjoint to the Fokker-Planck equation, i.e. the Kolmogorov backward equation:

$$-\frac{\partial w}{\partial t} - \mathbf{b}(\mathbf{z}, t) \cdot \frac{\partial w}{\partial \mathbf{z}} - \frac{\partial}{\partial \mathbf{z}} \cdot \left[\mathbf{D}(\mathbf{z}, t) \cdot \frac{\partial w}{\partial \mathbf{z}} \right] = \eta(\mathbf{z}, t) \tag{11}$$

with spatial boundary conditions $w = 0$ on $\partial\Omega$, and periodic in time $w(0) = w(T)$. Observe that

$$\begin{aligned}
\langle \eta, G \rangle &= \int_0^T \int_\Omega \left[-\frac{\partial w}{\partial t} - \mathbf{b} \cdot \frac{\partial w}{\partial \mathbf{z}} - \frac{\partial}{\partial \mathbf{z}} \cdot \left(\mathbf{D} \cdot \frac{\partial w}{\partial \mathbf{z}} \right) \right] G \, \mathrm{d}\mathbf{z} \, \mathrm{d}t \\
&= \int_0^T \int_\Omega w \left[\frac{\partial G}{\partial t} + \frac{\partial}{\partial \mathbf{z}} \cdot \left(\mathbf{b} G - \mathbf{D} \cdot \frac{\partial G}{\partial \mathbf{z}} \right) \right] \mathrm{d}\mathbf{z} \, \mathrm{d}t \\
&= \int_0^T \int_\Omega w S \, \mathrm{d}\mathbf{z} \, \mathrm{d}t = \langle w, S \rangle
\end{aligned}$$

In other words, solving the adjoint equation allows us to compute the desired moments of G without explicitly computing G itself.

3.3 Monte Carlo Estimation

The solution to the Kolmogorov backward equation can be represented as an expectation over stochastic trajectories. For $t > s$, let $\mathbf{Z}_t(\mathbf{z}, s)$ satisfy the foward-time SDE

$$\begin{gathered}
\mathrm{d}\mathbf{Z}_t = \tilde{\mathbf{b}}(\mathbf{Z}_t, t) \, \mathrm{d}t + \sigma(\mathbf{Z}_t, t) \cdot \mathrm{d}\mathbf{W}_t \\
\mathbf{Z}_s(\mathbf{z}, s) = \mathbf{z}
\end{gathered} \tag{12}$$

Then, we have the Feynman-Kac representation (TODO: check sign)

$$w(\mathbf{z}, s) = \mathbb{E}[w(\mathbf{Z}_T(\mathbf{z}, s), T)] + \mathbb{E}\left[\int_s^T \eta(\mathbf{Z}_t(\mathbf{z}, s), t) \, \mathrm{d}t \right]$$

4 C^1 interpolation on unstructured meshes

4.1 Field line tracing

Let $\vec{R}_B(R, Z; \varphi) = (R_B(...), Z_B(...))$ be the motion of the field-line trace starting at (R, Z) on a poloidal plane, parameterized by the toroidal angle φ moved along the field line.

$\vec{R}_B$ satisfies the ODE

$$\frac{\mathrm{d}\vec{R}_B}{\mathrm{d}\varphi} = \vec{b}_p \circ \vec{R}_B := \left. \frac{R\mathbf{B}_p}{B_t} \right|_{\vec{R}_B} = \left. \frac{R\hat{\boldsymbol{\varphi}} \times \nabla\psi}{F(\psi)} \right|_{\vec{R}_B}; \qquad \vec{R}_B(R, Z; 0) = (R, Z)$$

This ODE is essentially a reparameterization of the magnetic field line ODEs with φ as time. For $\frac{\mathrm{d}}{\mathrm{d}\varphi}$ we think of (R, Z) as being parameters. These ODEs have an associated variational equation

$$\frac{\mathrm{d}[D\vec{R}_B]}{\mathrm{d}\varphi} = ([D\vec{b}_p] \circ \vec{R}_B)[D\vec{R}_B]; \qquad D\vec{R}_B(R, Z; 0) = I_{2\times 2}$$

here we think of D as the differential in (R, Z) with φ as a parameter, that is:

$$D\vec{b}_p = \begin{bmatrix} \partial_R(\vec{b}_p \cdot \hat{\mathbf{R}}) & \partial_Z(\vec{b}_p \cdot \hat{\mathbf{R}}) \\ \partial_R(\vec{b}_p \cdot \hat{\mathbf{Z}}) & \partial_Z(\vec{b}_p \cdot \hat{\mathbf{Z}}) \end{bmatrix}$$

$$D\vec{R}_B = \begin{bmatrix} \partial_R R_B & \partial_Z R_B \\ \partial_R Z_B & \partial_Z Z_B \end{bmatrix}$$

4.2 Field-aligned interpolation

Suppose we are trying to interpolate $\phi(R, \varphi, Z)$ knowing its values on some equally spaced poloidal planes $\phi_i(R, Z)$. This can be accomplished by

$$\phi(R, \varphi, Z) = \sum_i p_i(\varphi)\phi_i(\vec{R}_B(R, Z; \varphi_i - \varphi))$$

here p_i are some piecewise polynomial basis functions. We can compute its gradient by

$$\nabla\phi = \sum_i \left[p'(\phi_i \circ \vec{R}_B)\nabla\varphi + p_i\nabla(\phi_i \circ \vec{R}_B)\right]$$

Using the chain rule,

$$\begin{aligned} \nabla(\phi_i \circ \vec{R}_B) &= \begin{bmatrix}\nabla R & \nabla Z\end{bmatrix} [D(\phi_i \circ \vec{R}_B)]^T - \nabla\varphi\left(\frac{\mathrm{d}(\phi_i \circ \vec{R}_B)}{\mathrm{d}\varphi}\right) \\ &= \begin{bmatrix}\hat{\mathbf{R}} & \hat{\mathbf{Z}}\end{bmatrix} [D\vec{R}_B]^T[[\nabla\phi_i] \circ \vec{R}_B] - \frac{\hat{\boldsymbol{\varphi}}}{R}\left([[\nabla\phi_i] \circ \vec{R}_B] \cdot \frac{\mathrm{d}\vec{R}_B}{\mathrm{d}\varphi}\right) \end{aligned}$$

Note that it's possible to show that $\mathbf{B} \cdot \nabla(\phi_i \circ \vec{R}_B) = 0$.

4.3 Choice of basis functions

Traditional cubic spline interpolation, which minimizes 'bending' and C^2 continuity, has polynomial coefficients which are computed by solving a linear system involving all of the data points as well as boundary conditions. Instead we rely on polynomial splines which involve only the 4 points in the neighborhood of any φ, and generally enforce C^1 continuity at the nodes.

Two options are considered. The first is cubic Hermite interpolation, with an array of polynomial coefficients

$$p = \begin{bmatrix} 0 & -1/2 & 1 & -1/2 \\ 1 & 0 & -5/2 & 3/2 \\ 0 & 1/2 & 2 & -3/2 \\ 0 & 0 & -1/2 & 1/2 \end{bmatrix}$$

where $p_i(t) = \sum_{j=0}^{3} p_{ij}t^j$ (here the array entries are being 0-indexed). This scheme exactly interpolates the nodes and also enforces C^1 continuity at the nodes with a value of the derivative given by the centered difference of the adjacent two nodes.

The second is a quadratic smoothing spline,

$$p = \begin{bmatrix} 1/4 & -1/2 & 1/4 \\ 1/2 & 0 & -1/4 \\ 1/4 & 1/2 & -1/4 \\ 0 & 0 & 1/4 \end{bmatrix}$$

This scheme can be thought of as the anti-derivative of linear interpolation on the derivative, computed via centered difference, while enforcing C^1 continuity at the nodes. The quadratic dependence sacrifices exact interpolation at the nodes in exchange for a derivative with less oscillations.

Finally we remark that parallel noise seems to be non-negligible; in theoretical cases, a Lanczos filter is applied along the field line to smooth out these high-frequency parallel fluctuations.

4.4 Interpolation on poloidal planes

Interpolation on poloidal planes uses rHCT elements, which are C^1 elements that minimize a 'bending energy'. The code is essentially a fork of the matplotlib `CubicTriInterpolator`[1] with a few optimizations.

5 Extraction of Ballooning Coefficients

5.1 Straight Field-line Coordinates

On the closed flux surfaces, let θ_g be the geometric poloidal angle relative to the magnetic axis, with $\theta_g = 0$ representing the outboard midplane. Using $F(\psi) = B_t/R$, we can compute the relationship between the straight field-line angle θ in terms of θ_g by

$$\theta = \frac{1}{q(\psi)} \int_{\theta_0(\psi)}^{\theta_g} \frac{F(\psi)}{B_p(\psi, \theta_g')} \ell'(\theta_g')\, \mathrm{d}\theta_g'$$

$$q(\psi) = \frac{1}{2\pi} \int_0^{2\pi} \frac{F(\psi)}{B_p(\psi, \theta_g')} \ell'(\theta_g')\, \mathrm{d}\theta_g'$$

where $\ell'(\theta_g)$ is the derivative of the arclength of the field line along the flux surface with respect to θ_g, and $\theta_0(\psi)$ is the arbitrary offset on each flux surface where $\theta = 0$ lies. An easy choice is $\theta_0(\psi) = 0$, which results in $\theta = 0$ being the outboard midplane. This relationship is then numerically inverted to get θ.

5.2 Ballooning Transform

Moving to flux coordinates (ψ, ζ, θ) with $\zeta = \varphi$, we can always write

$$\phi(R, \varphi, Z) = \sum_{n=-\infty}^{\infty} e^{in\zeta} \phi_n(\psi, \theta)$$

We can move to the covering space $\theta \mapsto \eta$, let $\rho = nq$, and introduce the eikonal factor

$$\phi_n(\psi, \theta) = \sum_{\ell=-\infty}^{\infty} \hat{\phi}_n(\psi, \eta + 2\pi\ell)$$

$$\hat{\phi}_n(\psi, \eta) = e^{-i\rho\eta} f_n(\rho, \eta)$$

$$f_n(\rho, \eta) = \int_{-\infty}^{\infty} \mathrm{d}\theta_k\, e^{i\rho\theta_k} \tilde{f}_n(\theta_k, \theta)$$

the complete representation is then

$$\phi(R, \varphi, Z) = \sum_{n=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} \int_{-\infty}^{\infty} \mathrm{d}\theta_k\, e^{in(\zeta - q(\eta - \theta_k + 2\pi\ell))} \tilde{f}_n(\theta_k, \theta + 2\pi\ell)$$

[1] `https://matplotlib.org/stable/api/tri_api.html#matplotlib.tri.CubicTriInterpolator`

Now, if we had an exact ballooning symmetry, we would have for a fixed parameter θ_k (related to the radial wavenumber),

$$f_n(\rho,\eta) = e^{i\rho\theta_k}\tilde{F}_n(\eta)$$

This is exactly analogous to $f(x) = \tilde{F}e^{ikx}$ with k is a fixed parameter. Our goal is to find an approximation to $\tilde{f}_n$.

Taking $S = n(\zeta - q\eta)$, we can compute

$$\nabla(e^{iS}f_n(\rho,\eta)) = e^{iS}\left[i\nabla S f_n + \partial_q f_n \nabla q + \partial_\eta f_n \nabla\eta\right]$$
$$\nabla S = n(\nabla\zeta - q\nabla\eta - \eta\nabla q)$$

Note that in the ballooning representation, we also have

$$\tilde{S} = n(\zeta - q(\eta - \theta_k))$$
$$\nabla(e^{i\tilde{S}}\tilde{f}_n(\theta_k,\eta)) = e^{i\tilde{S}}(i\nabla\tilde{S}\tilde{f}_n + \partial_\eta\tilde{f}_n\nabla\eta)$$
$$\nabla\tilde{S} = n(\nabla\zeta - q\nabla\eta - (\eta-\theta_k)\nabla q)$$

Note furthermore we have the gyroaverage operator taking the eikonal approximation

$$\left\langle e^{i\tilde{S}}\tilde{f}_n(\theta_k,\eta)\right\rangle \approx e^{i\tilde{S}}J_0(k_\perp\rho)\tilde{f}_n(\theta_k,\eta)$$
$$\mathbf{k}_\perp = \nabla\tilde{S}$$

5.3 Up-down Symmetry

One final thing we do is to modify θ_0 to account for up-down symmetry. We compute $v_r = \mathbf{v}_D\cdot\hat{\mathbf{r}}$ and $v_y = \mathbf{v}_D\cdot\hat{\mathbf{y}}$ where $\hat{\mathbf{y}} = \hat{\mathbf{r}}\times\hat{\mathbf{b}}$ is the binormal vector. We take $\theta_0 = \operatorname{argmax}_\theta v_y$.

5.4 Approximation by Gauss-Hermite Functions

Finally, we approximate $f_n(\rho,\eta)$ (equivalently $\tilde{f}_n(\theta_k,\eta)$) with complex-amplitude Gauss-Hermite functions.

Note that naively, if we try to extract ϕ_n from the data ϕ, aliasing issues cause pollution in the toroidal mode number spectrum. Instead, we first upsample the toroidal resolution using field-aligned interpolation (rather than numerically following field lines, we use constant $\alpha = \zeta - q\theta$), perform the FFT, then truncate in frequency space. Additionally, we apply a Lanczos filter in the parallel direction to reduce high-frequency parallel noise – not sure what its source is.

To compute gradients of ϕ, note the coordinate transform

Define v_D^q, v_D^α by

$$\mathbf{v}_D\cdot\nabla = v_D^q\partial_q + v_D^\alpha\partial_\alpha = v_D^R\partial_R + v_D^Z\partial_Z + v_D^\varphi\partial_\varphi$$

Note that $v_D^\varphi = \mathbf{v}_D\cdot\hat{\boldsymbol{\varphi}}/R$.

We have the following coordinate transforms

$$q = q(\psi) \qquad \alpha = \zeta - q\theta \qquad \eta = \theta$$

$$\begin{bmatrix} \mathrm{d}q \\ \mathrm{d}\alpha \\ \mathrm{d}\eta \end{bmatrix} = \begin{bmatrix} q' & 0 & 0 \\ -q'\theta & 1 & -q \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \mathrm{d}\psi \\ \mathrm{d}\zeta \\ \mathrm{d}\theta \end{bmatrix} = \begin{bmatrix} q' & 0 & 0 \\ -q'\theta & 1 & -q \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \psi_R & 0 & \psi_Z \\ 0 & 1 & 0 \\ \theta_R & 0 & \theta_Z \end{bmatrix}\begin{bmatrix} \mathrm{d}R \\ \mathrm{d}\varphi \\ \mathrm{d}Z \end{bmatrix}$$