GEM User Manual

Yang Chen

Contents

I. INTRODUCTION

GEM is a δf gyrokinetic Particle-in-Cell code for the study of low-frequency phenomena such as micro-turbulence and energetic particle driven Alfven waves in tokamak plasmas. These phenomena are characterized by the gyrokinetic ordering

$$\frac{e\phi}{T} \sim \frac{ev_T A_\parallel}{T} \sim \frac{k_\parallel}{k_\perp} \sim \frac{\omega}{\Omega_i} \sim \frac{\rho_i}{L_{\rm eq}} \sim \delta \ll 1 \tag{1}$$

$$k_\perp \rho_i \sim 1 \tag{2}$$

where ϕ is the fluctuating electric potential, $A_\parallel$ is the fluctuating parallel vector potential, $k_\parallel$ and $k_\perp$ are characteristic parallel and perpendicular wavenumber, respectively, Ω_i is the ion gyro-frequency, ρ_i is the ion Larmor radius, and $L_{\rm eq}$ is the equilibrium quantities scale length. This ordering is used to derive the gyrokinetic equation [1–3]. Gyrokinetic simulation started with the conventional Particle-in-Cell method [4]. The δf-method was later developed as an efficient method to reduce particle noise [5, 6]. At present both the Particle-in-Cell method and the Eulerian method are extensively used in gyrokinetic simulation of magnetic confinement fusion plasmas.

GEM started as a flux-tube code in an equilibrium with circular magnetic flux surfaces, and adiabatic electrons and electrostatic fluctuations only [7]. The first attempt at electromagnetic capabilities used a mass-less fluid electron model [8]. The split-weight scheme [9] was then used to extend the code to kinetic electrons for low-β plasmas [10]. The "cancellation problem" associated with the Ampere's equation, originally reported by J. Cummings in slab simulations [11], was solved with a control-variate method [12]. Extension to general magnetic equilibrium is accomplished in the following years [13]. The fluid electron model has also been improved and implemented in general equilibrium. At the present time GEM has the following features: radially global simulation (inside the separatrix) for arbitrary axisymmetric tokamak equilibria, with flux-tube option; multiple ion species; pitch-angle scattering collisional operator; subsonic equilibrium flows; energetic particles with a model anisotric beam distribution that reduces to the isotropic slowing-down distribution, and electromagnetic shear Alfvenic perturbations (i.e. without the parallel man=gnetic perturbation).

This manual is intended for users who are also interested in contributing to future GEM development, and contributing to the development of gyrokinetic simulation in general. In an active research area such as gyrokinetic simulation, users are also developers. Even if no new physics capabilities are needed for the problem of interest, in order to gain new physics understanding, it is likely that new diagnostic capabilities will be needed, To develop the necessary diagnostic capabilities, many details of the code, such as the coordinate system and how physical quantities are represented in the code, must be made accessible to the user/developer. The primary purpose of this manual is to provide a basic description of the source code, including the algorithms, main computational routines and data arrays. It is hoped that these information will be sufficient for anybody to write any desired diagnostic routines. these information is, of course, necessary for developing new physics capabilities, or alternative algorithms that solve the same problem.

There are plenty of new physics capabilities to be added. The use of field-line-following coordinates excludes the magnetic axis from simulation, and it is necessary to switch to other coordinates in the vicinity of the magnetic axis. The field solvers (for the gyrokinetic Poisson equation and the Ampere's equation) need to be modified accordingly. The parallel magnetic field perturbation needs to be included. Sonic level equilibrium flows need to be treated systematically, to include for instance the drift motion due to the centrifugal force and the Coriolis force. More accurate collision operator (beyond the Lorentz operator) will be needed for some applications. Nonlinear terms in the field equations, e.g. the nonlinear ion polarization effect, might be needed to study inverse turbulence cascading to low-n modes. Parallel nonlinearity, neglected in the Frieman-Chen gyrokinetic equation, might be needed for accurate nonlinear dynamics, especially for electrons. It should be noted that all present gyrokinetic codes solve the kinetic equation consistently only to first order accuracy. A complete set of gyrokinetic/Maxwell system of equations that are second order accurate have not been derived in the most general case. While some of the formally higher order terms (such as parallel nonlinearity) can be implemented in principle, questions regarding high order effects in general have to be answered by going beyond gyrokinetics, i.e. by solving the original 6-D Fokker-Planck equation. A model based on Lorentz ions and gyrokinetic electrons has been proposed [14] and the algorithm for implementing the model is being developed.

On the algorithm side, much can be done to improve the code performance by using, for instance, MPI/OpenMP or MPI/OpenACC hybrid programming. The mixed variable method of Mishchenko *et. al.* [15] shall be implemented to ease the requirement on the number of electron markers for the study of low-n modes. Perhaps the mixed variable approach can be combined with the split-weight scheme to further improve the numerical stability. The coarse-graining procedure [14] needs to be optimized for better efficiency.

Even if a user does not need new diagnostic capabilities, and the user is not interested in any project that requires new physics capabilities, he/she is encouraged to understand GEM to certain detail. The reason is the following. The

biggest difficulty in simulation is not to learn how to set up and run the code, but how to assess the validity of the simulation result. Cross-code benchmarking is essential for establishing the validity of the core algorithm. But such benchmarking or code verification can only be done for a small set of well-known, often simplified problems. New problems frequently are in regimes for which the code has not been benchmarked. In such a situation, confidence in the simulation result can only be gained by convergence tests. Some convergence tests are intimately tied to the core algorithm, and test results can only be understood after the algorithm is understood. For this purpose GEM's core algorithm, the split-weight control-variate scheme for kinetic electrons, is explained in detail in the manual, together with a description of the associated computational routines. The scheme has been described in great detail in the two papers, JCP03 [12] and JCP07 [13].

II. THE GYROKINETIC-MAXWELL SYSTEM OF EQUATIONS

The gyrokinetic equation for the distribution function $f(\mathbf{R}, \mu, v_\parallel)$ of any species is [16]

$$\frac{\partial f}{\partial t} + \mathbf{v}_G \cdot \nabla f + \dot{v}_\parallel \frac{\partial f}{\partial v_\parallel} = C(f), \tag{3}$$

where $C(f)$ is a collision operator. The guiding center drift is

$$\mathbf{v}_G = v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{\mu}{qB_\parallel^*}\mathbf{b}\times\nabla B + \frac{1}{B_\parallel^*}\mathbf{b}\times\nabla\left\langle\phi\right\rangle \tag{4}$$

with

$$\mathbf{B}^* = \mathbf{B}_0 + \frac{m}{q}v_\parallel \nabla\times\mathbf{b} + \delta\mathbf{B}_\perp \tag{5}$$

$$B_\parallel^* = B_0 + \frac{mv_\parallel}{q}\mathbf{b}\cdot\nabla\times\mathbf{b} \tag{6}$$

Here $\mathbf{b}$ is the unit vector along the equilibrium magnetic field $\mathbf{B}_0$, m is the mass, q is the charge. Species index is suppressed. The perturbed electric field is given by $\delta\mathbf{E} = -\nabla\phi - \frac{\partial A_\parallel}{\partial t}\mathbf{b}$ and the perturbed magnetic field is given by $\delta\mathbf{B}_\perp = \nabla\times A_\parallel\mathbf{b}$. $\langle\ldots\rangle$ denotes gyro-average. The parallel acceleration is

$$\dot{v}_\parallel \equiv \frac{dv_\parallel}{dt} = -\frac{1}{m}\frac{\mathbf{B}^*}{B_\parallel^*}\cdot\left(\mu\nabla B + q\nabla_\perp\left\langle\phi\right\rangle\right) + \frac{q}{m}\left(-\mathbf{b}\cdot\nabla\left\langle\phi\right\rangle - \frac{\partial\left\langle A_\parallel\right\rangle}{\partial t}\right) \tag{7}$$

The guiding center velocity contains the parallel streaming velocity $v_\parallel\mathbf{b}$, the curvature drift which is contained in the $\nabla\times\mathbf{b}$ term, the grad-B drift, the E×B drift $\mathbf{v}_E = \langle\mathbf{E}\rangle\times\mathbf{b}/B$, and the magnetic fluttering motion $v_\parallel\left\langle\delta\mathbf{B}_\perp\right\rangle/B_0$,

$$\mathbf{v}_{\mathrm{curv}} = \frac{mv_\parallel^2}{qB_\parallel^*}\mathbf{b}\times(\mathbf{b}\cdot\nabla\mathbf{b}) \tag{8}$$

$$\mathbf{v}_{\nabla\mathrm{B}} = \frac{\mu}{qB_\parallel^*}\mathbf{b}\times\nabla B \tag{9}$$

The sum of the curvature drift and the ∇B-drift will be denoted by $\mathbf{v}_D$, which is approximately

$$\mathbf{v}_D \approx \frac{v_\parallel^2 + v_\perp^2/2}{\Omega B^2}\mathbf{B}\times\nabla B, \tag{10}$$

and we frequently uses the following simplified expression for the guiding-center velocity

$$\mathbf{v}_G \approx v_\parallel\mathbf{b} + \mathbf{v}_D + \mathbf{v}_E + v_\parallel\frac{\left\langle\delta\mathbf{B}_\perp\right\rangle}{B} \tag{11}$$

The electric potential is determined by the quasi-neutrality condition

$$-\sum_s q_s n_{\mathrm{ps}} = \sum_s q_s\overline{\delta n_{is}} - e\delta n_e \tag{12}$$

The ion polarization density is

$$n_p = -\sum_{\mathrm{k}_\perp}\frac{qn_0}{T}\left(1-\Gamma_0(b)\right)\phi_{\mathrm{k}_\perp}\exp\left(\iota k_x x + \iota k_y y\right) \tag{13}$$

with $\iota = \sqrt{-1}$, $b = k_\perp^2 v_T^2/\Omega^2$, $T = mv_T^2$, all defined at the position where n_p is evaluated. This expression is derived from the integral expression

$$n_p = -\int\frac{q}{T}\left(\phi(\mathbf{x}) - \left\langle\phi(\mathbf{R},\mu,v_\parallel,\gamma)\right\rangle\right)f_M d\mathbf{v} \tag{14}$$

by assuming a local Fourier expansion in the perpendicular plane. Here γ is the gyro-angle and f_M is the Maxwellian distribution

$$f_M(\mathbf{R}, \varepsilon_k) = \frac{n(\mathbf{R})}{(2\pi)^{3/2} v_T^3} \exp\left(-\varepsilon_k / T(\mathbf{R})\right), \tag{15}$$

with $\varepsilon_k = mv^2/2$. The thermal speed is defined by $mv_T^2 = T(\mathbf{R})$. In general we will denote the equilibrium distribution function by f_0, which might deviate from the Maxwellian by a small amount, except for energetic particles. (When there is a sonic flow, the thermal ion and electron distribution in the local moving frame must be close to Maxwellian.) Most of the time $f_0 = f_M$ is understood, but we will use f_0 sometimes for generality. The total distribution is decomposed as $f = f_M + \delta f$. The perturbed ion density $\overline{\delta n_i}$ is given by

$$\overline{\delta n_i}(\mathbf{x}) = \int \delta f(\mathbf{R}, \mu, v_\parallel, \gamma) \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) \mathcal{J} d\mathbf{R} d\mu dv_\parallel d\gamma. \tag{16}$$

In Eq. 16, $\boldsymbol{\rho}$ is the gyro-radius vector leading from the guiding center $\mathbf{R}$ to a point $\mathbf{x}$ on the gyro-ring. Electrons are treated as drift-kinetic, $\boldsymbol{\rho} = 0$. $\mathcal{J}$ is the Jacobian of the coordinate transformation $(\mathbf{x}, \mathbf{v}) \to (\mathbf{R}, \mu, v_\parallel, \gamma)$,

$$d\mathbf{x} d\mathbf{v} = \mathcal{J} d\mathbf{R} d\mu dv_\parallel d\gamma. \tag{17}$$

Discretization of the quasi-neutrality equation is based on the spectral form [13, 17]. The vector potential $A_\parallel$ is determined by the parallel Ampere's equation

$$-\nabla_\perp^2 A_\parallel = \mu_0 \left(\sum_s \delta j_{\parallel \mathrm{is}} + \delta j_{\parallel \mathrm{e}} \right), \tag{18}$$

the ion current is

$$\delta j_{\parallel \mathrm{i}}(\mathbf{x}) = \int q v_\parallel \delta f(\mathbf{R}, \mu, v_\parallel, \gamma) \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) \mathcal{J} d\mathbf{R} d\mu dv_\parallel d\gamma. \tag{19}$$

Eq. 3, Eq. 12 and Eq. 18 are a complete set of equations describing low-freqnency small-amplitude fluctuations in a magnetized plasmas. We will postpone the treatment of an equilibrium flow and the parallel magnetic field perturbation, $\delta B_\parallel$, to the future.

III. δf PARTICLE-IN-CELL METHOD

Neglecting collisions, substituting $f = f_M + \delta f$ into Eq. 3, one obtains the following nonlinear gyrokinetic equation,

$$\frac{\partial \delta f}{\partial t} + \mathbf{v}_G \cdot \nabla \delta f + \dot{v}_\parallel \frac{\partial \delta f}{\partial v_\parallel} = -\left(\left\langle \mathbf{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0} \right\rangle \right) \cdot \nabla f_M - \dot{\varepsilon}_k \frac{\partial f_M}{\partial \varepsilon_k}, \tag{20}$$

where $\varepsilon_k = \mu B + m v_\parallel^2/2$ and

$$\dot{\varepsilon}_k = q \mathbf{v}_G \cdot \left\langle -\nabla \phi - \frac{\partial A_\parallel}{\partial t} \mathbf{b} \right\rangle. \tag{21}$$

On the right-hand-side (RHS) of Eq. 20, $f_M(\mathbf{R}, \varepsilon_k)$ is considered as a function of the two independent variables $\mathbf{R}$ and ε_k. The gradient operator acts on $\mathbf{R}$ with ε_k fixed, and the derivative w.r.t. ε_k is taken holding $\mathbf{R}$ fixed. On the left-hand-side (LHS), δf is a function of the independent variables $(\mathbf{R}, \mu, v_\parallel, t)$, and the partial derivatives are understood accordingly. In a tokamak the local Maxwellian distribution f_M is not an exact solution of the equilibrium kinetic equation. On the RHS of Eq. 20 zeroth-order terms such as $\mathbf{v}_D \cdot \nabla f_M$ are discarded. Such terms would be needed (as well as collisions) if one is to study neo-classical transport.

In a conventional PIC simulation, the distribution f is sampled by a large number of particles moving along the trajectories according to

$$\frac{d\mathbf{R}}{dt} = \mathbf{v}_G \tag{22}$$

$$\frac{dv_{||}}{dt} = \dot{v}_\parallel \tag{23}$$

$$\frac{d\mu}{dt} = 0 \tag{24}$$

In the δf-PIC method, each particle has a weight defined as $w = \delta f / f$ at the particle phase space coordinate [6], and the weight evolves according to

$$\frac{dw}{dt} = -\frac{1}{f_M} \left(\left\langle \mathbf{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0} \right\rangle \cdot \nabla f_M + \dot{\varepsilon}_k \frac{\partial f_M}{\partial \varepsilon_k} \right) \tag{25}$$

However, the appearance of the inductive parallel electric field $\partial A_\parallel / \partial t$, in $\dot{\varepsilon}_k$, leads to numerical difficulty. If this quantity is calculated with explicit finite difference in time, then the simulation is numerically unstable. Implicit method is needed, and is used in Eulerian codes such as GYRO [18] and gs2 [19, 20]. Implementation of an implicit method in a PIC code is more complicated than in an Eulerian code. A common method to avoid this difficulty is to use the parallel canonical momentum as a velocity coordinate [21],

$$p_\parallel = v_\parallel + \frac{q}{m} \langle A_\parallel \rangle. \tag{26}$$

The evolution equation for $p_\parallel$ is

$$\frac{dp_\parallel}{dt} = -\frac{1}{m} \frac{\mathbf{B}^*}{B_\parallel^*} \cdot (\mu \nabla B + q \nabla_\perp \langle \phi \rangle) - \frac{q}{m} \mathbf{b} \cdot \nabla \langle \phi \rangle + \frac{q}{m} \mathbf{v}_G \cdot \nabla \langle A_\parallel \rangle. \tag{27}$$

The time derivative in $\dot{v}_\parallel$ is eliminated in $\dot{p}_\parallel$. To eliminate $\partial A_\parallel / \partial t$ in the weight equation, it is necessary to decompose the distribution as $f = f_M(p_\parallel) + \delta f$, where $f_M(p_\parallel$ is

$$f_M(\mathbf{R}, \varepsilon_p) = \frac{n(\mathbf{R})}{(2\pi)^{3/2} v_T^3} \exp\left(-\varepsilon_p / T(\mathbf{R})\right), \tag{28}$$

and $\varepsilon_p = \mu B + m p_\parallel^2/2$. That is, $f_M(p_\parallel)$ is the Maxwellian in $p_\parallel$. The weight is now defined as $w = \delta f / f_M(p_\parallel)$ and evolves according to

$$\frac{dw}{dt} = -\frac{1}{f_M} \left(\left\langle \mathbf{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0} \right\rangle \cdot \nabla f_M + \dot{\varepsilon}_p \frac{\partial f_M(p_\parallel)}{\partial \varepsilon_p} \right) \tag{29}$$

with

$$\begin{aligned}\dot{\varepsilon}_p &= -q\mathbf{v}_G\cdot\nabla\langle\phi\rangle - \frac{q}{m}\langle A_\parallel\rangle\,\mathbf{b}\cdot\mu\nabla B + qp_\parallel\mathbf{v}_G\cdot\nabla\langle A_\parallel\rangle \\ &\quad -\frac{q}{m}\langle A_\parallel\rangle\left(\frac{\delta\mathbf{B}_\perp}{B}\cdot(q\nabla_\perp\langle\phi\rangle+\mu\nabla B)+q\mathbf{b}\cdot\nabla\langle\phi\rangle\right)\end{aligned} \tag{30}$$

The second line of Eq. 30 is of $\mathcal{O}(\delta^3)$ in the gyrokinetic ordering, and should be neglected for consistency. The perturbation terms in the parallel acceleration in Eq. 27 (denoted $\delta\dot{p}_\parallel$) contributes a nonlinear term of the form in the GK equation

$$\delta\dot{p}_\parallel\frac{\partial\delta f}{\partial p_\parallel} \tag{31}$$

and is also $\mathcal{O}(\delta^3)$. Notice the appearance of mass in the denomenator. These terms might be important for electrons. We keep them in the above equation for future consideration. Unless otherwise stated, parallel nonlinearity and all other $\mathcal{O}(\delta^3)$ effects in the gyrokinetic equation will be neglected. With all $\mathcal{O}(\delta^3)$ terms neglected, the transformation from $v_\parallel$ to $p_\parallel$ can be view as a transformation in the distribution, with the velocity coordinate unchanged [18],

$$\delta f_p = \delta f_v + \frac{q}{T}v_\parallel A_\parallel f_M(v_\parallel) \tag{32}$$

where $\delta f_p = f - f_M(p_\parallel)$ and $\delta f_v = f - f_M(v_\parallel)$. Notice that we have suppressed the subscript above.

A. The split-weight Control-variate algorithm

The dicretized form of the weight evolution equation is

$$\frac{dw}{dt} = -\frac{1}{g_0}\left(\left\langle\mathbf{v}_E + v_\parallel\frac{\delta\mathbf{B}_\perp}{B_0}\right\rangle\cdot\nabla f_M + \dot{\varepsilon}_p\frac{\partial f_M}{\partial\varepsilon_p}\right) \tag{33}$$

where g_0 is the loaded marker particle distribution,

$$g_0 = \frac{1}{(2\pi)^{3/2}T_g^{3/2}}\exp\left(-(\mu B + mp_\parallel^2/2)/T_g\right). \tag{34}$$

The marker distribution at the particle phase-space location needs to be known either exactly or to a good approximation [22, 23]. In collisionless simulations the marker distribution is constant along the trajectory, and this fact can be used to derive an exact equation for the weight [6]. When there are collisions and collisions are implemented with Monte-Carlo method, the marker distribution cannot be known exactly. We have chosen to load the particles uniform in space. The marker temperature can be radially varying. In general we use a constant T_g for ions, and the physical electron temperature profile for the electrons.

The numerical representation of g_0 is

$$\begin{aligned}\tilde{g} &= \frac{V}{N_p}\sum_j\delta(\mathbf{x}-\mathbf{x}_j)\delta(\mathbf{v}-\mathbf{v}_j) \\ &= \frac{V}{N_p}\sum_{j,l}\frac{1}{N}\frac{1}{\mathcal{J}}\delta(\mathbf{R}-\mathbf{R}_j)\delta(\mu-\mu_j)\delta(v_\parallel-v_{\parallel\mathrm{j}})\delta(\gamma-\gamma_{jl}),\end{aligned}$$

where the tilde "$\sim$" represents the numerical form, and the subscript l denotes N-point gyro-averaging. The appearance of the δ-function might be disturbing in a numerical representation. All quantities needed in the simulation (charge density, current etc.) will be velocity integrals (in the case of drift-kinetic electrons) or phase-space integrals (in the case of gyrokinetic ions), and the δ-functions in μ, $v_\parallel$ and γ will disappear upon integration. The delta-function in real space should be understood as a short notation for some function that represents the finite-particle-shape in Particle-in-Cell method. In GEM, the particle shape is

$$\delta(\mathbf{x}-\mathbf{x}_j)\approx\frac{1}{J(x,z)}\frac{1}{\Delta x\Delta y\Delta z}S_{1D}\left(\frac{x-x_j}{\Delta x}\right)S_{1D}\left(\frac{y-y_j}{\Delta y}\right)S_{1D}\left(\frac{z-z_j}{\Delta z}\right) \tag{35}$$

$$S_{1D}(x) = \begin{cases} 1-|x| & |x| \leqslant 1 \\ 0 & |x| > 1 \end{cases} \tag{36}$$

Numerical respresentation of δf and δn are

$$\widetilde{\delta f} = \frac{V}{N_p}\sum_{j,l}\frac{1}{N}\frac{1}{\mathcal{J}} w_j \delta(\mathbf{R}-\mathbf{R}_j)\delta(\mu-\mu_j)\delta(v_\parallel - v_{\parallel \mathrm{j}})\delta(\gamma-\gamma_{\mathrm{jl}}), \tag{37}$$

$$\widetilde{\delta n} = \frac{V}{N_p}\sum_{j,l}\frac{1}{N} w_j \delta(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{jl}), \tag{38}$$

where $\boldsymbol{\rho}_{jl}$ is the vector from guiding center to the l-th point on the gyro-ring. The numerical representation of the parallel current due to δf is

$$\widetilde{\delta j_\parallel} = \frac{V}{N_p} q \sum_{j,l}\frac{1}{N} v_{\parallel \mathrm{j}} w_j \delta(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{jl}) \tag{39}$$

In the split-weight scheme, a new electron weight is defined as

$$w_e = w - \varepsilon_g \frac{e\phi}{T_e}\frac{f_M}{g_0}, \tag{40}$$

where ε_g is in principle a free parameter. The adiabatic electron response corresponds to $\varepsilon_g = 1$. The weight w_e evolves according to

$$\begin{aligned} \frac{dw_e}{dt} \;=\; & \left[\left(\mathbf{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0}\right)\cdot\boldsymbol{\kappa} - \frac{e}{T_e}\Big(- \mathbf{v}_G\cdot\nabla\phi + \dot{v}_{\parallel 0}A_\parallel + v_{\parallel \mathrm{j}}\mathbf{v}_G\cdot\nabla A_\parallel\Big) - \right. \\ & \left. -\varepsilon_g\frac{e}{T_e}(\dot{\phi}(\mathbf{x}_j) + \mathbf{v}_G\cdot\nabla\phi|_{\mathbf{x}_j})\right]\frac{f_M}{g_0} \end{aligned} \tag{41}$$

where $\boldsymbol{\kappa} \equiv -\nabla f_M/f_M$ and $\dot{v}_{\parallel \mathrm{j}} = -\mu \mathbf{b}\cdot\nabla B_0/m$.

For ions, the time evolution equation for w_p is written as

$$\dot{w}_i = \left[\left\langle \mathbf{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0}\right\rangle\cdot\boldsymbol{\kappa} + \frac{q}{T}\left(-\mathbf{v}_G\cdot\nabla\langle\phi\rangle + \dot{v}_\parallel\langle A_\parallel\rangle + v_\parallel \mathbf{v}_G\cdot\nabla\langle A_\parallel\rangle\right)\right]\frac{f_M}{g_0} \tag{42}$$

In Eq. 41,$\dot{\phi} = \partial\phi/\partial t$ is obtained from the vorticity equation, which is the time partial derivative of the quasi-neutrality equation,

$$-qn_p = q\delta\bar{n}_i - e\delta n_e \tag{43}$$

We have assumed a single ion species for simplicity. The time derivative is,

$$-q\dot{n}_p = q\frac{\partial}{\partial t}\overline{\delta n_i} - e\frac{\partial}{\partial t}\delta n_e, \tag{44}$$

which is the vorticity equation and this equation is also used in the hybrid model of GEM. Since n_p is linear in ϕ, its time derivative is linear in $\dot{\phi}$. Discretization of the vorticity equation follows from that of the quasi-neutrality equation. The $\overline{\delta n_i}$ and $\partial\overline{\delta n_i}/\partial t$ terms appearing in quasi-neutrality equation and vorticity equation, respectively, are calculated in GEM as

$$\widetilde{\delta n_i} = \frac{V}{N_p}\sum_{j,l}\frac{1}{N} w_j \delta\left(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{\mathrm{jl}}\right) \tag{45}$$

$$\frac{\partial}{\partial t}\widetilde{\delta n_i} \;=\; \frac{V}{N_p}\sum_{j,l}\frac{1}{N}\dot{w}_j\delta(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{\mathrm{jl}}) - \nabla\cdot\frac{V}{N_p}\sum_{j,l}\frac{1}{N} w_j \mathbf{v}_{\mathrm{Gj}}\delta(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{\mathrm{jl}}) \tag{46}$$

In GEM, the first term is stored in dnidt(:, :, :) calculated in subroutine jie() and drdt(), the second term is calculated in jie(); $\nabla\cdot(\ldots)$ is calculated in drdt(). In the field-aligned coordinates (x,y,z), with a (spatial) Jacobian $\mathcal{J}_s$, $d^3\mathbf{x}=\mathcal{J}_s dx\,dy\,dz$, the divergence is calculated as

$$\nabla\cdot\mathbf{A}=\frac{1}{\mathcal{J}_s}\frac{\partial}{\partial x}(\mathcal{J}_s\mathbf{A}\cdot\nabla x)+\frac{1}{\mathcal{J}_s}\frac{\partial}{\partial y}(\mathcal{J}_s\mathbf{A}\cdot\nabla y)+\frac{1}{\mathcal{J}_s}\frac{\partial}{\partial z}(\mathcal{J}_s\mathbf{A}\cdot\nabla z) \tag{47}$$

For electrons

$$\delta n_e=\frac{V}{N_p}\sum_j\left(w_j+\varepsilon_g\frac{e\phi(\mathbf{x}_j)}{T_e}\frac{f_M}{g_0}\right)\delta(\mathbf{x}-\mathbf{R}_j) \tag{48}$$

$$\begin{aligned}\frac{\partial\delta n_e}{\partial t} &= \frac{V}{N_p}\sum_j\left[\left(\mathbf{v}_E+v_\parallel\frac{\delta\mathbf{B}_\perp}{B_0}\right)\cdot\boldsymbol{\kappa}-\frac{e}{T_e}\left(-\mathbf{v}_G\cdot\nabla\phi+\dot{v}_\parallel A_\parallel+v_\parallel\mathbf{v}_G\cdot\nabla A_\parallel\right)\right]\frac{f_M}{g_0}\delta(\mathbf{x}-\mathbf{R}_j)\\ &\quad -\frac{V}{N_p}\nabla\cdot\sum_j\left(w_j+\varepsilon_g\frac{e\phi}{T_e}\frac{f_M}{g_0}\right)\mathbf{v}_G\delta(\mathbf{x}-\mathbf{R}_j)\end{aligned} \tag{49}$$

The $\dot{v}_\parallel A_\parallel$ term and the $v_\parallel\mathbf{v}_G\cdot\nabla A_\parallel$ term must be combined to give (explained in JCP07)

$$\begin{aligned}\frac{\partial\delta n_e}{\partial t} &= \frac{V}{N_p}\sum_j\left[\left(\mathbf{v}_E+v_\parallel\frac{\delta\mathbf{B}_\perp}{B_0}\right)\cdot\boldsymbol{\kappa}+\frac{e}{T_e}\mathbf{v}_G\cdot\nabla\phi\right]\frac{f_M}{g_0}\delta(\mathbf{x}-\mathbf{R}_j)\\ &\quad -\frac{V}{N_p}\nabla\cdot\sum_j\frac{e}{T_e}v_\parallel\mathbf{v}_G A_\parallel\frac{f_M}{g_0}\delta(\mathbf{x}-\mathbf{R}_j)-\frac{V}{N_p}\nabla\cdot\sum_j\left(w_j+\varepsilon_g\frac{e\phi_j}{T_e}\frac{f_M}{g_0}\right)\mathbf{v}_{\mathrm{G}j}\delta(\mathbf{x}-\mathbf{R}_j)\end{aligned} \tag{50}$$

In the 2nd term above, $\mathbf{v}_G=v_{||}\mathbf{b}+\ldots$, only $v_{||}\mathbf{b}$ is needed. Contribution from all other terms in $\mathbf{v}_G$ vanishes. This term can be analytically evaluated, but must be evaluated with the marker particles for cancellation with part of the 3rd term that involves $v_\parallel$ in $\mathbf{v}_G$. The array deposited for this term is upa0() calculated in the subroutine jpar0(), and the divergence is calculated in drdt(). Other terms on the RHS of Eq. 50 are calculated similar to ions, i.e. divided between jie() and drdt(). Notice that the corresponding terms for the ions are not explicitly shown in Eq. 46. The ion terms are m_e/m_i smaller than the electron term and often neglected, or treated analytically (in both the Ampere's equation and the vorticity equation. These analytically given ion terms are indicated with the parameter "isiap").

For the quasi-neutrality equation itsel, the contribution to the electron density from the split-weight term portion in the electron weight can be analytically evaluated and is

$$\sum_j\varepsilon_g\frac{e\phi}{T_e}\frac{f_M}{g0}\delta(\mathbf{x}-\mathbf{R}_j)=\varepsilon_g n_e\frac{e\phi}{T_e} \tag{51}$$

It is found numerically that there is no need to treat this contribution fully kinetically with an iterative procedure, as in the Ampere's equation.

We not turn to the Ampere's equation. The perturbed parallel flow is

$$\frac{\delta j_\parallel}{q}=\int v_\parallel\left[f_0(p_\parallel)+\delta f-f_0(v_\parallel)\right]d\mathbf{v} \tag{52}$$

An equilibrium distribution f_0 written in terms of $p_\parallel$ gives rise to a perturbed parallel current (JCP03)

$$\begin{aligned}\frac{\delta j_\parallel(f_0)}{q}\equiv\int f_0(p_\parallel)v_\parallel d\mathbf{v} &\approx \int\left[f_0(v_\parallel)+\frac{q}{m}\langle A_\parallel\rangle f_0'(v_\parallel)+\frac{1}{2}\left(\frac{q}{m}\langle A_\parallel\rangle\right)^2 f_0''(v_\parallel)\right]v_\parallel d\mathbf{v}\\ &= \int\frac{q}{m}\langle A_\parallel\rangle v_\parallel f_0'(v_\parallel)d\mathbf{v}\end{aligned} \tag{53}$$

This is true for arbitrary function even in $v_\parallel$, including the Maxwellian. For the Maxwellian, $f_M'=-(mv_\parallel/T)f_M$,

$$\frac{\delta j_\parallel(f_0)}{q}=-\frac{q}{m}n_0A_\parallel \tag{54}$$

in the drift-kinetic limit, i.e. for electrons. For ions this current is not important for high-n modes, and can be treated in the drift-kinetic limit flr low-n modes.

As explained in JCP03, this analytic form should not be used, otherwise a cancellation problem arises. Two numerical errors, that arising from a finite number of particles and tha arising from finite grid sizes, enter the contribution to the parallel current from δf, and must be similarly introduced into the calculation of $\delta j_\parallel(f_0)$.

To introduce the same discretization effect in this current, we write the perturbed current for electrons in the form

$$-\frac{1}{e}\delta j_{\parallel \mathrm{e}} = \frac{V}{N_p}\sum_j \left(w_e + \varepsilon_g \frac{e\phi}{T_e}\frac{f_M}{g_0} + e v_\parallel \frac{A_\parallel}{T_e}\frac{f_M}{g_0}\right)_j v_{\parallel j}\delta(\mathbf{x}-\mathbf{R}_j) \tag{55}$$

The expression in the parenthesis is simply $(f - F_M(v_\parallel))/g_0$. The 3rd term corresponds to $\delta j_{\parallel \mathrm{e}}(f_0)$ of Eq. 54 for electrons. The second term on the RHS vanishes because it is (in the large particle limit) the integral of an odd function.

The ion parallel current is given by

$$\frac{1}{q}\delta j_{\parallel \mathrm{i}} = \frac{V}{N_p}\sum_j \frac{1}{N}\sum_l \left(w_i - q v_\parallel \frac{\langle A_\parallel\rangle}{T_i}\frac{f_M}{g_0}\right)_j v_{\parallel j}\delta(\mathbf{x}-\mathbf{R}_j-\boldsymbol{\rho}_{\mathrm{j}l}) \tag{56}$$

We have obtained numerical expressions for all the quantities on the RHS of field equations. To proceed, it will be convenient to introduce how physical variables are made dimensionless - normalization - in GEM. Equations can then be given in dimensionless form that can be more readily indentified in the codes.

B. Normalization

Table I lists the units for physical quantities in GEM. Units for temperature, density and the magnetic field can be arbitrary chosen. Other quantities such as the potentials are accordingly chosen so that many equations take familiar forms. For instance, the chosen unit for the electric field allows one to write the electrostatic relation $\mathbf{E} = -\nabla\phi$ in dimensionless form, unchanged from the dimensional form. In fact, the quasi-neutrality equation and the vorticity equation remain the same after normalization. The Ampere's equation is an exception. Define $\beta_u = \mu_0 n_u T_u/B_u^2$, the dimensionless Ampere's equation is

$$-\nabla_\perp^2 A_\parallel = \beta_u \delta j_\parallel \tag{57}$$

Resistivity is not used in the kinetic electron model. It is used in the fluid electron model where an Ohm's equation is used to determine the parallel electric field.

Flux-tube is special, the length unit is chosen as 1/1000 of major radius, i.e. in this unit $R_0 = 1000$.

Temperature	T_u	arbitrary
Number density	n_u	arbitrary
Magnetic field	B_u	$\sim B_{\mathrm{axis}}$
Particle mass	m_u	proton mass
Particle charge	q_u	e
Velocity	v_u	$\sqrt{T_u/m_u}$
Length	x_u	$m_u v_u/q_u B_u$
Angular frequency	ω_u	eB_u/m_u
Time	t_u	$1/\omega_u$
Electric current	j_u	$en_u v_u$
Vector potential	A_u	T_u/ev_u
Electric potential	Φ_u	T_u/e
Electric field	E_u	ϕ_u/x_u
Resistivity	η_u	B_u/en_u

Table I: Units for physical quantities

C. Iterative method for the Ampere's equation

The discretized Ampere's equation in dimensionless form is

$$-\nabla_\perp^2 A_\parallel = \beta_u \left[\delta j_{\parallel \mathrm{i}} + \delta j_{\parallel \mathrm{e}}\right] \tag{58}$$

with the ion and electron current given by Eq. 56 and Eq. 55, respectively. We have not discussed the discretization of the $\nabla_\perp^2$-operator (see Section ?), but it is obvious that, due to the appearance of the unknown $A_\parallel$ in Eq. 55, the resulting equation for $A_\parallel$ will be immensely complicated. Viewed as a linear problem $MA_\parallel = R$, the matrix M depends on the particle phase-space coordinates that evolve in time. Inverting such a matrix each time step will be prohibitively slow. Iterative schemes have to be used. First we note that the ion current in Eq. 56 that is proportional to $A_\parallel$ is smaller than the corresponding electron term by $\sim m_e/m_i$, can be neglected for high-n modes, and approximated by Eq. 54 for low-n modes. It can then be moved to the LFS and readily included in the matrix M. For the electron counterpart, we add the analytic form to both sides and iterate on the difference between the analytic form and the discrete form,

$$\begin{aligned} -\nabla_\perp^2 A_\parallel^{k+1} + \beta_u \frac{m_u}{m_e} n_{0\mathrm{e}} A_\parallel^{k+1} \;=\; & \beta_u \left[\delta j_{\parallel \mathrm{i}} + \delta j_{\parallel \mathrm{e}}^*\right] \\ & + \beta_u \left[\frac{m_u}{m_e} n_{0\mathrm{e}} A_\parallel^k - \frac{V}{N_p} \sum A_\parallel^k(\mathbf{R}_j) \frac{v_{\parallel \mathrm{j}}^2}{T_e} \left(\frac{f_M}{g_0}\right)_j \delta(\mathbf{x} - \mathbf{R}_j)\right], \end{aligned} \tag{59}$$

where $\delta j_{\parallel \mathrm{e}}^*$ represents the electron current excluding $\delta j_{\parallel \mathrm{e}}(f_0)$, i.e.

$$-\frac{1}{e}\delta j_{\parallel \mathrm{e}}^* = \frac{V}{N_p} \sum_j w_{\mathrm{ej}} v_{\parallel j} \delta(\mathbf{x} - \mathbf{R}_j) \tag{60}$$

Here k isthe iteration index. Assuming the electron number is sufficient, the iterative procedure usually converges in a few steps.

D. Main structure of GEM

A Particle-in-Cell code consists of two main components, particle pushing and field solving. Field quantities (ϕ, $A_\parallel$, $\mathbf{E}$ and $\mathbf{B}$ etc.) are represented on the spatial grids. To evlove particles, fields at the particle location needs to be computed from the field values on the nearby grids. This process is called *gathering* in PIC. To solve the field equations on the grids, charge density and current need to be accumulated on the grids from the particles, and this is called *deposition*. The simulation starts from an initial condition, which specifies the coordinates of all particles and their initial weights, and this is accomplished by particle loading. GEM uses the second order Runge-Kutta method for particle pushing, hence each time step consists of two sub-steps, a prediction step in which particles are pushed to $t + \triangle t/2$ using the field values at t, and a correction step in which particles are pushed from t to $t + \triangle t$ using the field values at $t + \triangle t/2$. Main subroutines are listed in Table II.

ppush	prediction ion push
cpush	correction ion push
pint	prediction electron push
cint	correction electron push
loadi	ion loader
ldel	electron loader
grid1	deposit density and current of ions and elctrons
jie	deposit quantities used in vorticity equation
drdt	calculate RHS of vorticity
dpdt	solve vorticity for $\partial\Phi/\partial t$
gkps	solve gyrokinetic poisson equation
ezmap	solve Ampere's equation
jpar0	calculate $\sum_j v_\parallel^2 A_\parallel(\mathbf{R}_j)\delta(\mathbf{x}-\mathbf{R}_j)$ in Eq. 59

Table II: Main subroutines

IV. GENERAL EQUILIBRIUM AND THE FIELD-ALIGNED COORDINATES

We use (r, θ, ζ) to denote the toroidal coordinates. r is a minor-radius-like flux surface label, θ is poloidal-angle-like but in general only implicitly specified. Looking horizontally at a tokamak, in the poloidal cross-section to the right, θ increases counter-clock-wise. Looking from above the tokamak, ζ increases clock-wise. A general axisymmetric equilibrium magnetic field is given by

$$\begin{aligned} \mathbf{B} &= \frac{f(\psi)}{R}\hat{\zeta} + \nabla\zeta \times \nabla\psi \\ &= q(\psi)\nabla\psi \times \nabla\theta_f + \nabla\zeta \times \nabla\psi, \end{aligned} \tag{61}$$

where θ_f is the straight field line flux coordinate. The safety factor is $q(\psi) = d\chi/d\psi = \mathbf{B}\cdot\nabla\zeta/\mathbf{B}\cdot\nabla\theta_f$, $2\pi\chi$ and $2\pi\psi$ are the toroidal and poloidal magnetic fluxes, respectively. The safety factor profile $q(r)$ is a flux label. It is the poloidal average of the local field line tilt,

$$\hat{q}(r,\theta) = \frac{\mathbf{B}\cdot\nabla\zeta}{\mathbf{B}\cdot\nabla\theta} \tag{62}$$

$$q(r) = \frac{1}{2\pi}\oint \hat{q} d\theta \tag{63}$$

Eq. 63 can be written as

$$2\pi = \oint \frac{f}{Rq\psi'(r)} \frac{1}{\hat{\zeta}\cdot\nabla r\times\nabla\theta} d\theta \tag{64}$$

which can be used to determine $\psi'(r)$ once the flux-surface shape and $f(\psi)$ is given. Since (r, θ, ζ) is right-handed, $\hat{\zeta}\cdot\nabla r\times\nabla\theta > 0$, this equation implies

$$\frac{f}{q\psi'} > 0 \tag{65}$$

The directions of the toroidal field $\mathbf{B}_T$, the poloidal field $\mathbf{B}_p$ and the toroidal plasma current I_p are determined by the signs of f and q, as shown in Table III. This sign relationship is consistent with the often used large aspect ratio expression, $q \approx rB_T/RB_\theta$.

$f>0,\ q>0$	$\mathbf{B}_T \parallel \hat{\zeta}$	$I_p \parallel \hat{\zeta}$	$\mathbf{B}_p \parallel \hat{\theta}$
$f>0,\ q<0$	$\mathbf{B}_T \parallel \hat{\zeta}$	$-I_p \parallel \hat{\zeta}$	$-\mathbf{B}_p \parallel \hat{\theta}$
$f<0,\ q>0$	$-\mathbf{B}_T \parallel \hat{\zeta}$	$-I_p \parallel \hat{\zeta}$	$-\mathbf{B}_p \parallel \hat{\theta}$
$f<0,\ q<0$	$-\mathbf{B}_T \parallel \hat{\zeta}$	$I_p \parallel \hat{\zeta}$	$\mathbf{B}_p \parallel \hat{\theta}$

Table III: Direction of magnetic field and plasma current

The straight-field-line poloidal angle is

$$\theta_f = \int_0^\theta \frac{f}{Rq} \frac{1}{\hat{\zeta}\cdot\nabla\psi\times\nabla\theta} d\theta \tag{66}$$

Let (R, Z, ζ) be the cylindrical coordinates, with R the major radius, $Z = 0$ at the location of the magnetic axis. The flux surface shape is specified by $R(r,\theta)$ and $Z(r,\theta)$. Various geometrical factors, such as ∇r, $\nabla\theta$, $\nabla r\cdot\nabla\theta$, $\nabla r\times\nabla\theta$, will be needed.

$$\begin{cases} \nabla R = \hat{R} = (\partial_r R)\nabla r + (\partial_\theta R)\nabla\theta \\ \nabla Z = \hat{Z} = (\partial_r Z)\nabla r + (\partial_\theta Z)\nabla\theta \end{cases}, \tag{67}$$

where $\partial_r R \equiv \partial R(r,\theta)/\partial r$, etc. Eq. 67 can be solved to give

$$\nabla r = \frac{1}{\partial_r R\partial_\theta Z - \partial_r Z\partial_\theta R}\left(\hat{R}\partial_\theta Z - \hat{Z}\partial_\theta R\right) \tag{68}$$

$$\nabla\theta = \frac{1}{\partial_r R\partial_\theta Z - \partial_r Z\partial_\theta R}\left(-\hat{R}\partial_r Z + \hat{Z}\partial_r R\right) \tag{69}$$

$$\nabla r\cdot\nabla\theta = -\frac{\partial_\theta Z\partial_r Z + \partial_\theta R\partial_r R}{(\partial_r R\partial_\theta Z - \partial_r Z\partial_\theta R)^2} \tag{70}$$

$$\nabla r\times\nabla\theta = \frac{1}{\partial_r R\partial_\theta Z - \partial_r Z\partial_\theta R}\hat{\zeta}. \tag{71}$$

We denote the major radius of the magnetic axis by $R_{\rm axis}$, and the major radius of the two points a flux surface intersets the mid-plane (the horizontal plane containing the magnetic axis) by R_- and R_+, $R_- < R_+$. The coordinate r is an arbitrary flux label, and can be chosen to be any of $R_{\rm axis} - R_-$, $R_+ - R_{\rm axis}$ and $(R_+ - R_-)/2$. The poloidal angle θ is also arbitrary. In the most general case the surface shapes $R(r,\theta)$ and $Z(r,\theta)$ are provided in the form of 2-D arrays radius(0:nr, 0:ntheta) and hght(0:nr, 0:ntheta), with nr and ntheta the number of grids in r and θ respectively. The geometrical interpretation of (r,θ) might not be provided, but implicit in the data.

The Miller model for surface shape is often used,

$$\begin{cases} R = R_0(r) + r\cos[\theta + (\sin^{-1}\delta(r))\sin\theta] \\ Z = \kappa(r) r\sin\theta. \end{cases} \tag{72}$$

In this case the shape is specified by the three 1D profiles, the Shafronov shift profile $R_0(r)$, the elongation profile $\kappa(r)$ and the triangularity profile $\delta(r)$.

A. The field-aligned coordinates

The field-aligned coordinates (or field-line-following coordinates) are defined by,

$$\begin{cases} x = r - r_0 \\ y = \frac{r_0}{q_0}\left(\int_0^\theta \hat{q}(r,\theta')d\theta' - \zeta\right) = \frac{r_0}{q_0}(q\theta_f - \zeta) \\ z = q_0 R_0\theta \end{cases} \tag{73}$$

The coefficients in the definition of y and z are such that, in the large aspect ratio, local approximation, y and z are proper lengths, with $|\nabla y| = 1$ and $\mathbf{b}\cdot\nabla z = 1$. Here q_0 is arbitrary, usually chosen to be the safety factor at the center of the simulation domain. The major radius R_0 in Eq. 73 is also arbitrary, usually chosen to be $R_0 = R_{\rm axis}$. It is readily proved that $\mathbf{B}\cdot\nabla x = \mathbf{B}\cdot\nabla y = 0$, that is, (x,y) are field line labels. Varying z while holding (x,y) fixed is to follow the field line, and varying y while holding (x,z) fixed is to varyζ with (r,θ) fixed. Equilibrium quantities are independent of y, hence y is the symmetry direction in this coordinates.

The Jacobian $\mathcal{J}_s$ of this coordinate system is given by

$$\frac{1}{\mathcal{J}_s} = \nabla z\cdot\nabla x\times\nabla y = \frac{r_0 R_0}{R}|\nabla r\times\nabla\theta| \tag{74}$$

Equilibrium variables are shown in Table IV.

radius(0:nr, 0:ntheta)	$R(r,\theta)$
hght(0:nr, 0:ntheta)	$Z(r,\theta)$
sf(0:nr)	safety factor profile $q(r)$
psip(0:nr)	$\psi'(r)$
psi(0:nr)	$\psi(r)$
f(0:nr)	poloidal plasma current profile $f(r)$
gr(0:nr, 0:ntheta)	$\|\nabla r\|$
gth(0:nr, 0:ntheta)	$\|\nabla\theta\|$
grdgt(0:nr,0:ntheta)	$\nabla r\cdot\nabla\theta$
grcgt(0:nr, 0:ntheta)	$\hat{\zeta}\cdot\nabla r\times\nabla\theta$
qhat(0:nr, 0:ntheta)	$\hat{q}(r,\theta)$
bfld(0:nr, 0:ntheta)	$\|B(r,\theta)\|$
dbdr(0:nr, 0:ntheta)	$\partial\,\|B(r,\theta)\|/\partial r$
dbdth(0:nr, 0:ntheta)	$\partial\,\|B(r,\theta)\|/\partial\theta$
thflx(0:nr, 0:ntheta)	$\theta_f(r,\theta)$
jacob(0:nr, 0:ntheta)	$\mathcal{J}_s(r,\theta)$
yfn(0:nr, 0:ntheta)	$\frac{r_0}{q_0}q(r)\theta_f$
dydr(0:nr, 0:ntheta)	$\frac{r_0}{q_0}q(r)\partial\theta_f/\partial r$
gxdgy(0:nr, 0:ntheta)	$\nabla x\cdot\nabla y$
xn0s(0:nsm, 0:nr)	ion density profile $n_s(r)$
capns(0:nsm, 0:nr)	$-n_s'(r)/n_s$
xn0e(0:nr)	electron density profile $n_e(r)$
capne(0:nr)	$-n_e'(r)/n_e$
t0s(0:nsm, 0:nr)	ion temperature profile
capts(0:nsm, 0:nr)	$-T_s'(r)/T_s$
t0e(0:nr)	electron temperature profile
capte(0:nr)	$-T_e'(r)/T_e$

Table IV: Equilibrium arrays

B. Domain decomposition along z

The z-dimension is the primary domain decomposition in the code, with one MPI process (and clones) assigned to a grid-cell in $z\in[z_k,z_{k+1}]$. Here $z_k:\ k=0,1,\ldots,N_z$ are the equally spaced grid points. Particles in each z-grid are distributed among ntube MPI-processes (clones), so that total number of MPI-processes is $N_z$ntube. Due to particle free streaming along the field line, marker particles distribute uniformly along the field line in real space, which means there are more particles per poloidal angle at $\theta=0$ than at $\theta=\pi$. For a tokamak with $R_{\rm maj}/a\sim 3$, this amounts to almost a factor of two difference between the number of particles in a $\theta=0$ process and that in a $\theta=\pi$ process. This problem can be solved by choosing the domain size in z to be dependent on z and nonuniform grid size in z. Alternatively, we can redefine the z coordinate such that equal-sized domains in the new coordinate have equal number of particles.

Assuming particles are loaded uniform in space, the number of marker particles in $(\theta,\theta+\triangle\theta)$ is proportional to

$$\bar{\mathcal{J}}_s(\theta)=\int_{r_{\rm in}}^{r_{\rm out}}\mathcal{J}_s(r,\theta)\,dr, \tag{75}$$

where $r_{\rm in}$ and $r_{\rm out}$ are the inner and outer simulation boundaries. This number depends on θ for fixed $\triangle\theta$, or fixed $\triangle z=q_0R_0\triangle\theta$. We can define $z'(\theta)$ so that the number of particles in $(z',z'+\triangle z')$ is independent of θ. This is accomplished by choosing

$$\frac{dz'}{dz}=\frac{\bar{\mathcal{J}}_s(\theta)}{\oint\bar{\mathcal{J}}_s(\theta)\,d\theta}\equiv 1/\mathsf{jfn}(\theta). \tag{76}$$

with the boundary condition $z'(\theta) = z(\theta)$ at $\theta = \pm\pi$. This choice of z' also makes the parallel box size invariant, $z'(\theta = \pi) - z'(\theta = -\pi) = 2\pi q_0 R_{\rm maj}$. The 1D array jfn(0:ntheta) is calculated in equil. The Jacobian for (x, y, z') is jacob $*$ jfn, and $dz'/dz = 1/$jfn. The effect of the additional coordinate transformation $z \to z'$ enters the code through the array jfn. When jfn is set to unity, $z' = z$, and particles loading is not balanced, but of course the simulation results should not change.

"Equilibrium arrays" in Table IV are defined on the grids $(r_i, \theta_j) : i = 0, 1, \ldots, N_r; j = 0, 1, \ldots, N_\theta$. Additional equilibrium arrays are defined on the grids $(x_i, z_k) : i = 0, 1, \ldots, N_r; k = 0, 1$ for each process. These arrays are calculated in the initialization subroutine init. jfn appears in those arrays that are affected by using z', for instance, bdgrzn $\equiv \mathbf{b} \cdot \nabla z'$. The mapping $\theta \to z'$ is denoted by zfnth and the inverse function is thfnz. These mappings also use jfn.

C. Toroidal Boundary Condition

The simulation domain is a rectangle, $(x, y, z) \in [0, L_x] \times [0, L_y] \times [0, L_z]$, with $L_x = r_{\rm out} - r_{\rm in}$, $L_z = 2\pi \mid q_0 \mid R_0$. The box size L_y is usually chosen to correspond to a wedge of tokamak, $L_y = 2\pi r_0 / \mid q_0 \mid /$lymult, corresponding to a fundamental toroidal mode number of $n =$ lymult allowed in the simulation. The end points $z = 0$ and $z = L_z$ correspond to the high field side $\theta = \pm\pi$. To map the domain $r_{\rm in} \leqslant r \leqslant r_{\rm out}$, $-\pi \leqslant \theta < \pi$ and $0 \leqslant \zeta < 2\pi$, into the rectangle, the definition of the field-aligned coordinates needs to be modified,

$$\begin{cases} x = r - r_0 + L_x/2 \\ y = \mathrm{mod}\left(\frac{r_0}{q_0}\left(\int_0^\theta \hat{q}(r,\theta')d\theta' - \zeta\right), L_y\right) \\ z = q_0 R_0 \left(\theta + \pi q_0 / \mid q_0 \mid\right) \end{cases} \tag{77}$$

where $\mathrm{mod}(x, 2\pi)$ means adding or sutracting some multiple of 2π to x until the result is in $[0, 2\pi)$. For explicitness we continue to use z as the parallel coordinate, z' is understood implicitly.

The inverse for ζ is

$$\zeta = \mathrm{mod}\left(\int_0^\theta \hat{q}(r,\theta')d\theta' - \frac{q_0}{r_0}y, 2\pi/\text{lymult}\right) \tag{78}$$

Physical quantities are periodic in (θ, ζ), which mean periodicity in y, but not periodic in z. There is a shift in y as the toroidal coordinate is changed from $(\theta = -\pi, \zeta)$ to $(\theta = \pi, \zeta)$,

$$y(r, \theta = -\pi, \zeta) = \mathrm{mod}\left[\frac{r_0}{q_0}\left(\int_0^{-\pi} \hat{q}(r,\theta')d\theta' - \zeta\right), L_y\right] \tag{79}$$

$$\begin{aligned} y(r, \theta = \pi, \zeta) &= \mathrm{mod}\left[\frac{r_0}{q_0}\left(\int_0^{\pi} \hat{q}(r,\theta')d\theta' - \zeta\right), L_y\right] \\ &= \mathrm{mod}\left[\frac{r_0}{q_0}\left(\int_0^{-\pi} \hat{q}(r,\theta')d\theta' - \zeta + 2\pi q(r)\right), L_y\right] \\ &= \mathrm{mod}\left[y(r, \theta = -\pi, \zeta) + 2\pi\frac{r_0}{q_0}q(r), L_y\right] \end{aligned} \tag{80}$$

Particles moving outside of the simulation box in z will be relocated at the other end, with its y-coordinate shifted by $2\pi r_0 q(r)/q_0$. For any field variable array, only the values at the two $x - y$ planes (corresponding to z_k and z_{k+1}) are stored. A y-grid at $z = 0$ will map to a y value at $z = Lz$ that in general will not coincide with a grid. Linear interpolation is used to obtain the value of a field variable from neighboring grid points. The weights used for linear interpolation are calculated in subroutine weight.

V. DISCRETIZATION SCHEME

A. Time integration

Particle coordinates $(x, y, z, p_\parallel)$ and the weight w are evolved in time with the 2nd order Runge-Kutta method. Write the equation of motion for y, for instance, as

$$\frac{dy}{dt} = f(Z, t), \tag{81}$$

with $Z = \{\mathbf{R}, \mu, v_\parallel, w\}$. Dependence on the electromagnetic fields is understood, and the explicit dependence on t comes primarily from the fields' dependence on time. A time step consists of a prediction step and a correction step,

$$y^{n+1/2} = y^n + \frac{\Delta t}{2} f(Z^n, t^n) \tag{82}$$

$$y^{n+1} = y^n + \Delta t \cdot f(Z^{n+1/2}, t^{n+1/2}) \tag{83}$$

Equation of motion for (x, y, z) is derived from the guiding center velocity

$$\mathbf{v}_G = v_\parallel \mathbf{b} + \frac{v_\parallel^2 + v_\perp^2/2}{\Omega B^2} \mathbf{B} \times \nabla B + \frac{\mathbf{E} \times \mathbf{b}}{B} + v_\parallel \frac{\delta \mathbf{B}_\perp}{B} \tag{84}$$

$$\begin{aligned} \frac{dx}{dt} &= \mathbf{v}_G \cdot \nabla x \\ \frac{dy}{dt} &= \mathbf{v}_G \cdot \nabla y \\ \frac{dz}{dt} &= \mathbf{v}_G \cdot \nabla z \end{aligned} \tag{85}$$

All equilibrium quantities, $q(r)$, $\hat{q}(r, \theta)$, $B(r, \theta)$, $\partial B/\partial r$, etc. are 2D or 1D arrays on (r, θ) grids. Linear interpolation is used to obtain values at particle location.

The guiding center motion is given by,

$$\mathbf{v}_D \cdot \nabla x = -\frac{e_B}{B^2} \frac{f}{R} \frac{\partial B}{\partial \theta} \left| \nabla r \times \nabla \theta \right| \tag{86}$$

$$\mathbf{v}_D \cdot \nabla y \approx e_B \frac{f}{R} \left| \nabla r \times \nabla \theta \right| \left(-\frac{\partial y}{\partial r} \frac{\partial B}{\partial \theta} + \hat{s} \frac{\partial B}{\partial r} \right) \tag{87}$$

$$v_\parallel \mathbf{b} \cdot \nabla z = \frac{v_\parallel q_0 R_{\mathrm{maj}}}{BR} \Psi'(r) \left| \nabla r \times \nabla \theta \right| \tag{88}$$

$$\mathbf{v}_E \cdot \nabla x = \hat{s} \frac{\langle E_y \rangle}{B^2} \frac{f}{R} \left| \nabla r \times \nabla \theta \right| \tag{89}$$

$$\mathbf{v}_E \cdot \nabla y = -\hat{s} \frac{\langle E_x \rangle}{B^2} \frac{f}{R} \left| \nabla r \times \nabla \theta \right| \tag{90}$$

$$\mathbf{v}_E \cdot \nabla z \approx 0 \tag{91}$$

$$v_\parallel \frac{\delta \mathbf{B}_\perp}{B} \cdot \nabla x = \frac{v_\parallel}{B} \frac{\partial A_\parallel}{\partial y} \mathbf{b} \cdot \nabla x \times \nabla y \tag{92}$$

$$v_\parallel \frac{\delta \mathbf{B}_\perp}{B} \cdot \nabla y = -\frac{v_\parallel}{B} \frac{\partial A_\parallel}{\partial x} \mathbf{b} \cdot \nabla x \times \nabla y \tag{93}$$

$$v_\parallel \frac{\delta \mathbf{B}_\perp}{B} \cdot \nabla z \approx 0 \tag{94}$$

with $e_B = m(v_\parallel^2 + v_\perp^2/2)/qB$, $\hat{s}(r, \theta) = \frac{r_0}{q_0} \hat{q}$. Notice that the arrays names ex, ey, delbx, delby refer to (signed) partial derivatives of ϕ and $A_\parallel$, e.g. $\mathsf{ex} = -\frac{\partial \phi}{\partial x}$, $\mathsf{ey} = -\frac{\partial \phi}{\partial y}$ and $\mathsf{ez} = -\frac{\partial \phi}{\partial z}$.

The mirror force is given by

$$\dot{v}_\parallel = -\frac{\mu}{m} \mathbf{b} \cdot \nabla B = -\frac{\mu}{mRB} \Psi'(r) \frac{\partial B}{\partial \theta} \left| \nabla r \times \nabla \theta \right| \tag{95}$$

Quantities such as $\frac{\partial y}{\partial r}$, $\frac{\partial B}{\partial r}$, $\frac{\partial B}{\partial \theta}$ and $|\nabla r \times \nabla \theta|$ are all computed and saved as numerical tables.

Gyro-averaging for the perturbed quantities $\langle E_x \rangle$, etc., is done using the four-point averaging scheme [24]. The location of the four points, $\mathbf{r}(x + \triangle x_l, y + \triangle y_l)$, $l = (1, 2, 3, 4)$, are obtained from

$$\mathbf{r}(x + \triangle x_l, y + \triangle y_l) - \mathbf{r}(x, y) \approx \frac{\partial \mathbf{r}}{\partial x} \triangle x_l + \frac{\partial \mathbf{r}}{\partial y} \triangle y_l = \boldsymbol{\rho}_l. \tag{96}$$

Here $\boldsymbol{\rho}_l$ is the vector leading from the guiding center to the particle position on its gyro-ring. Setting $\boldsymbol{\rho}_1 = \rho \hat{\mathbf{r}}$, $\boldsymbol{\rho}_2 = -\rho \hat{\mathbf{r}}$, $\boldsymbol{\rho}_3 = \rho \mathbf{b} \times \hat{\mathbf{r}}$ and $\boldsymbol{\rho}_4 = -\rho \mathbf{b} \times \hat{\mathbf{r}}$, with $\hat{\mathbf{r}} = \nabla r / |\nabla r|$, one obtains,

$$\triangle x_1 = \rho |\nabla r| \tag{97}$$
$$\triangle y_1 = \rho \nabla x \cdot \nabla y / |\nabla r| \tag{98}$$
$$\triangle x_2 = -\triangle x_1 \tag{99}$$
$$\triangle y_2 = -\triangle y_1 \tag{100}$$
$$\triangle x_3 = 0 \tag{101}$$
$$\triangle y_3 = \frac{\rho}{B |\nabla r|} \left(\hat{s} \frac{f}{R} |\nabla r \times \nabla \theta| + \frac{r_0 \Psi'}{q_0 R^2} |\nabla r|^2 \right) \tag{102}$$
$$\triangle x_4 = 0 \tag{103}$$
$$\triangle y_4 = -\triangle y_3 \tag{104}$$

B. Collisions

a Lorentzian operator for electrons, $C(f_e) = C_L(f_e)$, with

$$C_L(f_e) = \nu_e \frac{1}{2} \frac{\partial}{\partial \lambda} (1 - \lambda^2) \frac{\partial}{\partial \lambda} f_e \tag{105}$$

where $\lambda = v_\parallel / v$ is the pitch angle parameter. ν_e is the collision frequency,

$$\nu_e = \frac{n_{0e} e^4 \ln \Lambda}{4\pi \epsilon_0^2 m_e^2 v^3} \left(Z_{\text{eff}} + H_{\text{ee}} \left(\sqrt{m_e v^2 / 2 T_{0e}} \right) \right) \tag{106}$$

with $H_{\text{ee}}(x) = \frac{e^{-x^2}}{\sqrt{\pi} x} + (1 - \frac{1}{2x^2}) \text{erf}(x)$.

$$C_L(f_e) = C_L(f_{0e}(p_\parallel)) - C_L(\epsilon_g \phi \frac{\partial f_{0e}}{\partial \varepsilon_e}) + C_L(h). \tag{107}$$

The ϵ_g term is nonlinear and will be neglected. The first term is given by,

$$C_L(f_{0e}(p_\parallel)) = -\tau \nu_e A_\parallel f_{0e}, \tag{108}$$

which is implemented as an additional term in the electron weight equation. The third term on the RHS of Eq. 107 is implemented using the Monte-Carlo method. After both the predictor and the corrector step, a random change to the pitch angle variable λ is carried out, with the average amount of change determined by collision frequency and the time step,

$$\lambda_{\text{new}} = \lambda_{\text{old}} (1 - \nu_e \delta t) \pm \left[\left(1 - \lambda_{\text{old}}^2 \right) \nu_e \delta t \right]^{1/2}, \tag{109}$$

where $\pm$ means equal probability of $+$ or $-$[25]. $\delta t = \triangle t$ for corrector step and $\delta t = 2 \triangle t$ for predictor step, $\triangle t$ is the time step of the simulation.

C. Particle sorting and recycle

Particles moving outside of $[0, L_y]$ are relocated according to strict periodicity, i.e. $y = \mathsf{modulo}(y, L_y)$. Particles moving beyond the z-boundaries are similarly relocated, but with a shift in y according to the toroidal boundary

condition. In flux-tube simulations the boundary condition in x is also strictly periodic. In global simulations, particles moving beyong the x-boundaries are relocated following their equilibrium trajectories. In up-down symmetric equilibrium, this is achieved by setting z to the opposite value, with other coordinates unchanged. The particle weight can be kept unchanged, which allows profile relaxation at the boundary, or set to zero, which prohibits profile relaxation.

At the end of each push step (predictor or corrector), the boundary conditions are used to relocate those particles moving out of the simulation box. Particles are then sorted into the correct MPI process using the library pputil, according to the z-coordinate.

D. Ampere's Equation

The Laplacian operator in the field-aligned coordinates is given by

$$\nabla_\perp^2 f \approx \frac{\partial^2 f}{\partial^2 x} \left|\nabla x\right|^2 + 2\frac{\partial^2 f}{\partial x \partial y}\nabla x \cdot \nabla y + \frac{\partial^2 f}{\partial^2 y}\left|\nabla y\right|^2 \tag{110}$$

The vector potential is assumed to satisfy the Fixed boundary condition $A_\parallel(x=0) = A_\parallel(x=L_x) = 0$, and can be expanded in sine series. The forward sine transform is given by

$$b_m = \frac{2}{N}\sum_{j=0}^{N-1} f(x_j)\sin\left(\frac{\mathrm{j}m\,\pi}{N}\right), \quad m = 0, 1, \ldots N-1 \tag{111}$$

and the backward transform

$$f(x_j) = \sum_{m=0}^{N-1} b_m \sin\left(\frac{\mathrm{j}m\,\pi}{N}\right) \tag{112}$$

Assume

$$A_\parallel(x,y) = \sum_{k_y} e^{\iota k_y \cdot y} \sum_{m=1}^{N-1} b(m,k_y)\sin\frac{m\pi x}{L_x} \tag{113}$$

with

$$b(m,k_y) = \frac{2}{N}\sum_{i=0}^{N-1} A_\parallel(k_y,x_i)\sin\frac{\mathrm{i}m\,\pi}{N} \tag{114}$$

The Laplacian operator becomes

$$-\nabla_\perp^2 A_\parallel(x_i,y) = \sum_{k_y} e^{\iota k_y y}\sum_{i'=1}^{N-1} M(i,i',k_y)A_\parallel(k_y,x_i') \tag{115}$$

with the matrix element

$$M(k_y,i,i') = \frac{2}{N}\sum_{m=1}^{N-1}\left\{\left[\left(\frac{m\pi}{L_x}\right)^2 \cdot \left|\nabla x\right|^2 + k_y^2\left|\nabla y\right|^2\right]\sin\frac{i'm\pi}{N}\sin\frac{\mathrm{i}m\,\pi}{N}\right. \tag{116}$$

$$\left. - \mathrm{i}k_y \cdot \frac{m\pi}{L_x}\cdot 2\nabla x\cdot\nabla y\cdot\sin\frac{i'm\pi}{N}\cos\frac{\mathrm{i}m\,\pi}{N}\right\} \tag{117}$$

The z-coordinate enters the Laplacian as a parameter and is suppressed. The factors ∇x and ∇y are to be evaluated at (x_i, z). The actual matrix to be inverted contains the second term on the LHS of Eq. 59. This matrix is constructed and inverted using L-R decomposition at the beginning of the program, when the subroutine ezamp is called for the first time. In the code the mode k_y is indexed with j, the matrix column is indexed with ix, and ikx is used as the summation index over the sine components, m in the above equation.

ezamp only computes the solution of Eq. 59 given the RHS. The iteration scheme in Eq. 59 is coded in subroutine ampere, with the number of iterations specified with the variable iter. The summation over the particles on the RHS of Eq. 59 is calculated in subroutine jpar0.

E. Poisson Equation

For a ϕ represented in perpendicular Fourier modes,

$$\phi(\mathbf{x}) = \sum_{\mathbf{k}_\perp} e^{\iota \mathbf{k}_\perp \cdot \mathbf{x}} \phi_{\mathbf{k}_\perp} \tag{118}$$

the ion polarization density is

$$n_p = -\frac{q n_0(r)}{T(r)} \sum_{\mathbf{k}_\perp} e^{\iota \mathbf{k}_\perp \cdot \mathbf{x}} \phi_{\mathbf{k}_\perp} [1 - \Gamma_0(b)], \tag{119}$$

with $b = k_\perp^2 v_T^2/\Omega_i^2$. This expression of n_p is drived from local Fourier expansion. For a sine series in x, it is necessary to express it in the terms of Fourier series,

$$\begin{aligned} \phi(k_y, x) &= \sum_{m=1}^{N_x-1} b(m, k_y) \sin\left(\frac{m\pi x}{L_x}\right) \\ &= \sum_{m=1}^{N_x-1} b(m, k_y) \frac{1}{2\iota} \left(e^{\mathrm{im}\,\pi x/L_x} - e^{-\mathrm{im}\,\pi x/L_x}\right) \end{aligned}$$

Corresponding to each k_y and sine component, there are two $k_\perp$'s given by

$$\begin{aligned} k_\pm^2 &= \left(\frac{m\pi}{L_x}\right)^2 |\nabla r|^2 + k_y^2 \left[\left(\frac{\partial y}{\partial r}\right)^2 |\nabla r|^2 + \left(\frac{r_0}{q_0}\hat{q}|\nabla\theta|\right)^2 + 2\frac{\partial y}{\partial r}\frac{r_0}{q_0}\hat{q}\nabla r\cdot\nabla\theta\right] \\ &\quad \pm 2\frac{m\pi}{L_x} k_y \nabla x \cdot \nabla y \end{aligned}$$

hence

$$-n_p = \frac{q n_0(r)}{T(r)} \sum_{k_y} \sum_{m=1}^{N_x-1} \frac{1}{2\iota} b(m, k_y) \cdot \left\{[1 - \Gamma_0(b_+)]\, e^{\mathrm{i}k_y\, y} \cdot e^{\mathrm{im}\,\pi x/L_x} - [1 - \Gamma_0(b_-)]\, e^{\mathrm{i}k_y\, y} \cdot e^{-\mathrm{im}\,\pi x/L_x}\right\}$$

with $b_\pm = k_\pm^2 v_T^2/\Omega_i^2$. Substituting

$$b(m, k_y) = \frac{2}{N_x} \sum_{i=1}^{N_x-1} \phi(k_y, x_i) \sin\frac{\mathrm{im}\,\pi}{N_x} \tag{120}$$

into this expression we obtain

$$\begin{aligned} -n_p &= \frac{q n_0(r)}{T(r)} \sum_{k_y} e^{\iota k_y y} \sum_{i'=0}^{N_x-1} \phi(k_y, x_{i'}) \cdot \frac{1}{\iota \mathrm{N_x}} \sum_{m=1}^{N_x-1} \sin\frac{i' m\pi}{N_x} \\ &\quad \{(1 - \Gamma_0(b_+)) e^{\mathrm{im}\,\pi x_i/L_x} - (1 - \Gamma_0(b\_) e^{-\mathrm{im}\,\pi x_i/L_x}\} \end{aligned}$$

This is the discretized n_p at (x_i, y). In the code, m is replaced with ikx, i' replaced with ix, and $\Gamma_0(b_+)$ is saved in the array element gamb1(ikx, j, i, k), with j being the index for k_y, i the radial grid index, $k = 0, 1$ the z-grid index.

F. The vorticity equation

Once the right-hand-side of the vorticity equation is obtained on the grids, the equation is discretized similar to the Poisson equation. The same matrix for the ion polarization density is used. Unlike the Poisson equation, the mass term proportional to ε_g is absent in the matrix. The potentials ϕ and $A_\parallel$ are already available for solving the vorticity equation. All the quantities needed to compute $\partial\delta n_i/\partial t$ and $\partial\delta n_e/\partial t$ are deposited in the subroutine jie. Computationally, jie is one of the most intensive subroutines in the code. It consists of loops over all ions, and a loop over electrons. The contribution due to the evolution of the weights are deposited into the arrays dnidt and dnedt, for

ions and electrons respectively. The ion flows in the three directions are collected in jionx, jiony and jion, respectively. The electron flows are collected in jpex, jpey and jpar. Deposition for the part of dnedt,

$$-\frac{V}{N_p}\nabla\cdot\sum_j \frac{e}{T_e}v_\parallel \mathbf{v}_G A_\parallel \frac{f_M}{g_0}\delta(\mathbf{x}-\mathbf{R}_j) \tag{121}$$

is calculated in the subroutine jpar0, in the process of solving the Ampere equation. These quantities are used to calculate the RHS of the vorticity equation in subroutine drdt. The quantity $\dot{\phi}$ is finally solved for in subroutine dpdt.

G. Approximation in GEM field equation

The $\nabla_\perp^2$ operator is defined as $\nabla_\perp^2 = \nabla^2 - \nabla_\parallel^2$

$$\nabla\phi = \frac{\partial\phi}{\partial x}\nabla x + \frac{\partial\phi}{\partial y}\nabla y + \frac{\partial\phi}{\partial z}\nabla z \tag{122}$$

$$\begin{aligned}
\nabla^2\phi &= \frac{\partial^2\phi}{\partial x^2}|\nabla x|^2 + 2\frac{\partial^2\phi}{\partial x\partial y}\nabla x\cdot\nabla y + \frac{\partial^2\phi}{\partial y^2}|\nabla y|^2 \longrightarrow \text{what's in GEM} \\
&= \frac{\partial\phi}{\partial x}\nabla\cdot\nabla x + \frac{\partial\phi}{\partial y}\nabla\cdot\nabla y \longrightarrow \text{neglected} \\
&= \frac{\partial^2\phi}{\partial x\partial z}\nabla x\cdot\nabla z + \frac{\partial^2\phi}{\partial y\partial z}\nabla y\cdot\nabla z + \nabla\cdot\left(\frac{\partial\phi}{\partial z}\nabla z\right) \longrightarrow \text{neglected}
\end{aligned}$$

For ordering purpose we can take $k_x \sim k_y \sim m/r$. In the high-n limit $m \sim nq \to \infty$, only term quadratic in $k_\perp$ are kept.

For $n \sim m \sim 1$, this is clearly a poor approximation. For instance

$$\nabla\cdot\left(\frac{\partial\phi}{\partial z}\nabla z\right) = \frac{\partial^2\phi}{\partial z^2}|\nabla z|^2 + \frac{\partial^2\phi}{\partial x\partial z}\nabla x\cdot\nabla z + \frac{\partial^2\phi}{\partial y\partial z}\nabla y\cdot\nabla z + \frac{\partial\phi}{\partial z}\nabla\cdot\nabla z \tag{123}$$

Note $z = q_0 R_0\theta, \nabla\theta \sim \frac{1}{r}$, then

$$\frac{\partial^2\phi}{\partial z^2}|\nabla z|^2 \sim \frac{\phi}{(q_0R_0)^2}|\nabla(q_0R_0\theta)|^2 \sim \frac{\phi}{r^2}$$

which competes with $\partial^2\phi/\partial y^2|\nabla y|^2 \sim m^2\phi/r^2$ for small m.

Note that the $\nabla_\parallel^2$ contribution in $\nabla_\perp^2 = \nabla^2 - \nabla_\parallel^2$ is always small, can be neglected for any n.

VI. INPUT PARAMETERS AND MAIN VARIABLES

A. Input parameters

The following variables are contained in the input file gem.in. Variables in the input file not listed here should be ignored.

itube: $= 1$ indicates flux-tube simulation, $= 0$ for global simulation. At present the flux-tube model is implemented only with the Miller equilibrium model. If there is an equilibrium flow, in principle fixed boundary condition in radius should be used.
iperi: indicator for radial boundary condition in field solvers. iperi $= 1$ indicates periodic boundary condition, iperi $= 0$ indicates fixed boundary.
iperidf: indicator for particle radial boundary condition. iperidf $= 1$ for periodic boundary condition in x.
ibunit: rarely used.
mimp: $= m_{\rm ion}/m_{proton}$, the main ion mass.
mcmp: $m_{impurity}/m_{proton}$, the "impurity" mass.
chgi: the charge number of main ion.
chgc: the charge number of impurity ion.
Rovera: for itube $= 1$, aspect ratio R_0/a.
elon0: for itube $= 1$, elongation κ.
selon0: for itube $= 1$, $r\kappa'/\kappa$.
tria0: for itube $= 1$, triangularity.
stria0: for itube $= 1$, $r\delta'/\delta$.
rmaj0p: for itube $= 1$, Shafranov shift R_0'.
q0: for itube $= 1$, safety factor.
shat0: for itube $= 1$, rq'/q.
teti: for itube $= 1$, T_e/T_i.
tcti: for itube $= 1$, $T_{impurity}/T_i$.
rhoia: for itube $= 1$, $\rho^* = \left(m_i\sqrt{T_e/m_i}/eB_0\right)/a$.
Rovlni: for itube $= 1$, $-\frac{d}{dr}ln(n_i(r))$.
Rovlti: for itube $= 1$, $-\frac{d}{dr}ln(T_i(r))$.
imx: number of grid cells in x.
jmx: number of grid cells in y.
kmx: number of grid cells in z.
mmx: maximum number of particles per ion specie per process, which should satisfy mmx $\geqslant 1.5 \times$ mm1/numprocs. All ion species have the same number of particles. The total number of MPI processes is numprocs $=$ ntube $*$ kmx.
mmxe: maximum number of electrons per process.
nmx: maximum number of time steps allowed over multiple restarted runs.
nm: the number of time steps in the present run or restart.
nsmx: maximum number of ion species.
nsm: actual number of ion species.
ntube: number of processes (clones) at each z-grid.
lxa: L_x/a
lymult: $L_y = 2\pi r_0/(|\,q_0\,| *$lymult). The lowest nonzero toroidal mode number included in the simulation.
delra, delri, delre, delrn: these parameters are used to smooth the density/temperature gradient profiles near the radial boundaries.
nrst, eprs: the coarse-graining-procedure is done every nrst time steps if eprs > 0. Always set eprs=0 for a new case. Use eprs > 0 only when running into problems in nonlinear simulations.
dt: time step
xshape, yshape, zshape: Parameters that control the hyper-Gaussian filtering in nonlinear simulations.
iput: if $= 1$, dump files in /dump will be written if iput=1, for restart
iget: $= 0$ indicates an initial run. $= 1$ indicates restarting a job by reading data from the './dump/' directory
ision: Only for some special simplified problems ions are not important. In that case set ision=0. Ions are then not loaded and pushed.
peritr: $= 1$ indicates to use the correct twisted boundary condition. The flux-tube model can be reduced to slab by setting peritr=0.
llk: for some diagnostics

mlk: to select toroidal mode. Usually mlk=1 for a single mode simulation.
onemd: $= 1$ indicates only one mode is retained in the simulation in a linear run.
iflut: iflut=0 removes the magnetic fluttering nonlinearity,$\delta B \cdot \nabla \delta f$, in the electron drift-kinetic equation.
nplot: indicates the number of time steps to write out field arrays for visualization.
xnplot: the number of time steps to screen output.
nzcrt: Sometimes the long wavelength zonal flows cause numerical problems, especially in global (non-flux-tube) runs. Zonal components with $k_x <$ nzcrt $* \pi / L_x$ are removed.
npze npzi npzc npzb: Parameters that control how much particles are allowed in the phase-space before they are relocated to their initial phase coordinates. Only used in nonlinear non-flux-tube simulations.
isft, mynf, frmax, ifskp: parameters for mode frequency diagnostics. isft=1 indicates it is a post-run analysis of the mode frequency.
iphbf, iapbf, idpbf: parameters that control whether the blending function filtering (PHYSICS OF PLASMAS 17, 102504 2010) is used. By default, these are all set to zero.
amp: The initial amplitude for particle weight, e.g. weight is asigned the value amp multiplied by a random number.
tor: If tor=0 (and with some changes in the code) one can reduce the code to a shear-less slab code, no toroidal effects. This is useful for algorithm development.
fradi: The fraction of adiabatic electrons in the Poisson equation. It is equal to the split-weight parameter isg: for kinetic electrons. fradi $= 1$ for adiabatic electrons.
isg: the split-weight parameter for kinetic electrons. The default value is isg $= 1$.
kxcut: after field-solving, components with radial wavenumber larger than kxcut are set to zero.
kycut: after field-solving, components with toroidal wavenumber larger than kycut are set to zero.
roa: the simulation domain is centered around r0=r0a*a
vpp, vt0, yd0, vexbsw, vparsw, mach, gamma_E: these are all related to equilibrium flows.
c4: a parameter used to control filtering in x-y plane. Similar to xshape, yshape. Will be re-designed.
ifluid: $= 1$ for kinetic electrons, $= 0$ for adiabatic electrons.
amie: $m_{\text{proton}}/m_{\text{electron}}$
rneu: the electron collision frequency, measured in eB/m_p.
rneui: ion collision frequency. By default it is zero.
beta: the normalization constant $\mu_0 n_u T_u / B_u^2$, but check equil and subroutine init in gem_main.
nonlin, nonline: $= 1$ for a linear run, $= 0$ for a linear run.
ipara: controls whether parallel nonlinearity is included. Simulations with ipara=0 are sometimes called partially linearized simulations.
vwidth vwidthe isuni: Parameters introduced when testing uniform loading in velocity space. Might be used in the future, but not used for Maxwellian loading (isuni=0).
vcut: particles with $mv^2/2 > v_{\text{cut}} T$ are not counted in deposition.
mm1: total number of ions per species.
mm2: total number of electrons.

B. Main variables

The following variables are defined in gem_com.f90.
den: perturbed ion density on the (x, y, z) grids of dimension (0:imx,0:jmx,0:1). The first dimension is ion species index. The total ion charge density is rho.
jpar: perturbed ion parallel flow for each ion species. The total ion parallel current is jion
dene: perturbed electron density
upar: perturbed electron parallel flow
phi: electrostatic potential ϕ
ex,ey,ez: $-\partial\phi/\partial x$, $-\partial\phi/\partial y$ and $-\partial\phi/\partial z$, respectively.
apar: parallel vector potential $A_\parallel$
delbx,delby: $\partial A_\parallel/\partial y$ and $-\partial A_\parallel/\partial x$, respectively.
mu: the magnetic moment for ions
x2,y2,z2: ion coordinate at the current time level
x3,y3,z3: ion coordinate at the future time level
u2,u3: ion parallel velocity at the current time level and the future time level, respectively
w2,w3: ion weight at the current time level and the future time level, respectively
mue: the magnetic moment for electrons
x2e,y2e,z2e: electron coordinate at the current time level

x3e,y3e,z3e: electron coordinate at the future time level
u2e,u3e: electron parallel velocity at the current time level and the future time level, respectively
w2e,w3e: electron weight at the current time level and the future time level, respectively

The following are equilibrium quantities defined on the (x, z) grids, with dimension $(0 : imx)$ if it depends on r only, and dimension $(0 : imx, 0 : 1)$ if it depends on (r, θ). They are calculated in subroutine init from the corrresponding arrays in equil.
bmag: $|B|$
bdgxcgy: $\mathbf{b} \cdot \nabla x \times \nabla y$
bdgrzn: $\mathbf{b} \cdot \nabla z$
gn0e: electron density profile
gt0e: electron temperature profile
gn0s: ion density profiles
gt0i: the main ion temperature profile

The following variables are related to the toroidal boundary condition. They are calculated in subroutine weight, and used in repeated code segments to handle the shift in y across the z-boundaries, e.g. those in subroutines eqmo,enforce,enfz,drdt.
deljp, deljm, jpl,jpn,jmi,jmn,weightp,weightpn: "weighty" variables

The following variables are for particle and energy fluxes, for each particle species. The radial domain are divided into nsubd domains and fluxes are computed for each domain.
efle_es: electron heat flux due to E×B motion
efle_em: electron heat flux due to magnetic fluttering, $v_\parallel \delta \mathbf{B}_\perp / B$
pfle_es: electron particle flux due to E×B motion
pfle_em: electron particle flux due to magnetic fluttering, $v_\parallel \delta \mathbf{B}_\perp / B$
efl_es: ion heat flux due to E×B motion
efl_em: ion heat flux due to magnetic fluttering, $v_\parallel \delta \mathbf{B}_\perp / B$
pfl_es: ion particle flux due to E×B motion
pfl_em: ion particle flux due to magnetic fluttering, $v_\parallel \delta \mathbf{B}_\perp / B$

VII. THE FLUID ELECTRON MODEL

The gyrokinetic ion/fluid electron hybrid model [8, 26] is useful for the simulation of low-n modes such as the energetic particle driven Alfven waves. The model consists of the electron continuity equation, an Ohm's law that determines $E_\parallel$, and the isothermal condition for the electron temperature perturbation. The electron continuity equation is

$$\frac{\partial \delta n_e}{\partial t} + n_0 \mathbf{B}\cdot\nabla\frac{u_{\parallel \mathrm{e}}}{B} + \mathbf{v}_E\cdot\nabla n_e + \frac{1}{m_e\Omega_e B^2}\mathbf{B}\times\nabla B\cdot\nabla(\delta p_\perp + \delta p_\parallel) + \frac{2n_0}{B^3}\mathbf{B}\times\nabla B\cdot\nabla\phi = 0, \tag{124}$$

where $u_{\parallel \mathrm{e}}$ is the electron parallel flow velocity, $\mathbf{v}_E = \mathbf{E}\times\mathbf{b}/B$, $\Omega_e = eB/m_e$ is the electron gyro-frequency, $\delta p_\perp$ and $\delta p_\parallel$ are the perturbed perpendicular and parallel electron pressure, respectively.

The parallel electric field $E_\parallel$ is determined by the Ohm's equation,

$$en_0 E_\parallel - \frac{m_e}{\mu_0 e}(\nabla_\parallel\nabla_\perp^2\phi + \nabla_\perp^2 E_\parallel) = -\nabla\cdot\int m_e v_\parallel \mathbf{v}_G f_{\mathrm{e}1}\, d\mathbf{v} - \int \mu\mathbf{b}\cdot\nabla B_0 f_{\mathrm{e}1}\, d\mathbf{v} - \int m_e v_\parallel^2 \frac{\delta\mathbf{B}_\perp}{B_0}\cdot\nabla f_{\mathrm{e}0}\, d\mathbf{v} + e\int m_e v_\parallel \mathbf{v}_{\mathrm{G}\perp}\cdot\nabla\phi\frac{f_{\mathrm{e}0}}{T_e}\, d\mathbf{v}. \tag{125}$$

This equation is derived by combining the time derivative of the Ampere's equation with the $v_\parallel$-moment of the electron drift-kinetic equations [27]. Here $f_{\mathrm{e}1}$ and $f_{\mathrm{e}0}$ are the perturbed and equilibrium electron distribution, respectively. $\mathbf{v}_G = v_\parallel\mathbf{b} + \mathbf{v}_D + \mathbf{v}_E + v_\parallel\delta\mathbf{B}_\perp/B \equiv v_\parallel\mathbf{b} + \mathbf{v}_{\mathrm{G}\perp}$ is the guiding center velocity, $\mathbf{v}_D$ is the ∇B and curvature drift. In the fluid model the mirror force term, the second term on the right-hand-side (RHS), is neglected. Nonlinear terms are contained in the first and the last term on the RHS. In a mass-less fluid electron model, one takes the limit $m_e \to 0$ while holding $m_e v_\parallel^2$ and $m_e v_\perp^2$ as fixed. The mass-less fluid electron model used here differs from the previous model [8] in that, here nonzero electron mass effect is included in the second term on the left-hand-side (LHS). This Laplace term has the desirable effect of reducing perpendicular grid-scale $E_\parallel$ fluctuations. These approximations lead to the following fluid Ohm's equation,

$$en_e E_\parallel - \frac{m_e}{\mu_0 e}(\nabla_\parallel\nabla_\perp^2\phi + \nabla_\perp^2 E_\parallel) = -\tilde{\mathbf{b}}\cdot\nabla\delta p_{\parallel \mathrm{e}} - \frac{\delta\mathbf{B}_\perp}{B_0}\cdot\nabla(p_{\parallel \mathrm{e}} - en_e\phi), \tag{126}$$

where n_e is the equilibrium electron density, $p_{\parallel \mathrm{e}}$ the equilibrium electron pressure, $\tilde{\mathbf{b}} = \hat{\mathbf{b}} + \delta\mathbf{B}_\perp/B_0$ is the unit vector along the total magnetic field.

The main closure relation for this fluid-electron model is the linearized isothermal condition for electrons [28],

$$\mathbf{b}\cdot\nabla\delta T_e + \frac{\delta\mathbf{B}_\perp}{B_0}\cdot\nabla T_e = 0, \tag{127}$$

which, for MHD waves with $E_\parallel \approx 0$, is used to derive an evolution equation for δT_e [26]. Here T_e is the equilibrium electron temperature. This isothermal condition is derived from the electron drift kinetic equation assuming $\omega \ll k_\parallel v_{\mathrm{Te}}$ (v_{Te} being the electron thermal speed). It clearly breaks down for the zonal components, which have $k_\parallel = 0$.

The potential $A_\parallel$ is evolved from

$$\frac{\partial A_\parallel}{\partial t} = -\mathbf{E}_\parallel - \nabla_\parallel\phi, \tag{128}$$

while the Ampere's equation is used to obtain the electron parallel flow $u_{\parallel \mathrm{e}}$ from the $A_\parallel$ and the ion parallel current,

$$en_{0\mathrm{e}}u_{\parallel \mathrm{e}} = \delta j_{\parallel \mathrm{i}} + \frac{1}{\beta_u}\nabla_\perp^2 A_\parallel. \tag{129}$$

The electron flow can then be used in the continuity equation to evolve δn_e. The electric potential ϕ is as usual obtained from the quasi-neutrality equation. The ions are described by the gyrokinetic equation of the $v_\parallel$ formalism, as $E_\parallel$ (including the inductive component $\partial A_\parallel/\partial t$) is now given by the Ohm's law and there is no need to transform to the $p_\parallel$ formalism.

A. Closure for $\delta p_\perp$ and $\delta p_\parallel$

The perturbed electron pressure is assumed to be isotropic

$$\delta p_\perp = \delta p_\parallel = T_0 \delta n_e + n_0 \delta T_e \tag{130}$$

and δT_e is assumed to be constant along the perturbed field line,

$$\tilde{\mathbf{b}} \cdot \nabla (T_0 + \delta T_e) = 0, \tag{131}$$

which is linearized to give

$$\mathbf{b} \cdot \nabla \delta T_e = -\frac{\delta \mathbf{B}_\perp}{B_0} \cdot \nabla T_{0\mathrm{e}} \tag{132}$$

The RHS of Ohm's law (126) becomes

$$-\mathbf{b} \cdot \nabla T_{0\mathrm{e}} \delta n_e - T_{0\mathrm{e}} \frac{\delta \mathbf{B}_\perp}{B_0} \cdot \nabla n_{0\mathrm{e}} + \frac{\delta \mathbf{B}_\perp}{B_0} \cdot \nabla (e n_{0\mathrm{e}} \phi - T_{0\mathrm{e}} \delta n_e) \tag{133}$$

Inverting the magnetic differential equation Eq. 132 is difficult. For MHD waves, $\delta \mathbf{B} = \nabla \times (\boldsymbol{\xi} \times \mathbf{B}_0)$, one can show $\delta T_e = -\boldsymbol{\xi} \cdot \nabla T_{0\mathrm{e}}$. Also for MHD,

$$\frac{\partial \boldsymbol{\xi}}{\partial t} = \frac{\mathbf{E} \times \mathbf{b}}{B_0}$$

Using these relations one can derive an evolution equation for δT_e,

$$\frac{\partial \delta T_e}{\partial t} = \frac{1}{B} \frac{\partial T_e}{\partial x} \frac{\partial \phi}{\partial y} \hat{\mathbf{b}} \cdot (\nabla x \times \nabla y) \tag{134}$$

B. The vorticity approach to the quasi-neutrality equation

Instead of solving the quasi-neutrality equation for ϕ, it is sometimes advantageous to solve the vorticity equation for $\partial\phi/\partial t$, then integrate it in time to give ϕ. The GK poisson equation is then not used. This proves to be more robust. It is also more accurate for low-n modes, as the relation $\sum_s q_s n_{0s} = 0$, hence,

$$\sum_s q_s \nabla \cdot \left(\frac{\mathbf{E} \times \mathbf{b}}{B_0} n_{0s} \right) = 0$$

can be made explicit. For gyrokinetic ions, $\nabla \cdot (\mathbf{v}_E n_{0s})$ with FLR correction appears as the $\mathbf{E}_\perp$ terms in the ion weight equation,

$$-\langle \mathbf{v}_E \rangle \cdot \frac{\nabla f_0}{g} - q \langle \mathbf{E}_\perp \rangle \cdot \mathbf{v}_D \frac{\partial f_0}{\partial \varepsilon} \frac{1}{g}. \tag{135}$$

For electrons the corresponding terms are$\mathbf{v}_E \cdot \nabla n_{0\mathrm{e}} + \frac{2 n_{0\mathrm{e}}}{B^3} \mathbf{B} \times \nabla B \cdot \nabla \phi$ in the continuity equation. The following algorithm is used to enforce the linear E$\times$B effects. For ions, the $\mathbf{E}_\perp$ terms in $\dot{w}$ are calculated twice, first with FLR, then treated as drift-kinetic (no FLR). Only the difference is kept. In the electron continuity equation, the corresponding terms are dropped.

C. Energetic particles

Three energetic particle distributions have been implemented: Maxwellian, isotropic slowing-down, and anisotropic slowing-down. The isotropic slowing-down distribution is

$$f_h = \frac{n_h(r)}{v^3 + v_I^3}, \tag{136}$$

which is a special case of the anisotropic slowing-down distribution,

$$f_b = \frac{n_b(r)}{v^3 + v_I^3} \exp\left[-\frac{(\lambda - \lambda_0)^2}{(\Delta\lambda)^2}\right] \tag{137}$$

with $(\Delta\lambda)^2 = \Delta_0^2 + \frac{1}{3}(1-\lambda_0)\ln y(v)$, $y(v) = (v^3 + v_I^3)/v^3(1 + (v_I/v_b)^3)$ and $\lambda = \frac{\mu B_0}{E}$.

For the slowing-down distribution, isotropic or anisotropic, the energetic particles are loaded according to

$$g = \frac{1}{C_v}\frac{1}{v^3 + v_I^3} \approx \frac{V}{N}\sum_j \delta(\boldsymbol{x} - \boldsymbol{x}_j)\delta(\boldsymbol{v} - \boldsymbol{v}_j), \tag{138}$$

$$v_1 \leqslant v \leqslant v_c, \tag{139}$$

$$C_v = \int_{v_1}^{v_c} \frac{1}{v^3 + v_I^3} 4\pi v^2 dv. \tag{140}$$

To load this velocity distribution, we first find a set of v-grids such that

$$(v_i - v_{i-1})g(v_{i-1})v_{i-1}^2 = (v_{i+1} - v_i)g(v_i)v_i^2 \tag{141}$$

then for each new particle, randomly selects a v-grid.

The EP gyrokinetic equation is

$$\frac{\partial \delta f}{\partial t} + (v_{||}\hat{b} + \boldsymbol{v}_D + \boldsymbol{v}_E)\cdot\nabla\delta f = -\left(\boldsymbol{v}_E + v_\parallel \frac{\delta \mathbf{B}_\perp}{B_0}\right)\cdot\nabla f_0 - \dot{\varepsilon}\frac{\partial f_0}{\partial \varepsilon} \tag{142}$$

This is the same as thermal Maxwellian ions. Anisotropy leads to adiabatic response $\propto \partial f_0/\partial\varepsilon$, $\partial f_0/\partial\mu$, and corresponding polarization density to the quasi-neutrality equation. But these are all neglected. The total polarization density in the quasi-neutrality equation is assumed to be from the thermal main ions, but with a density of n_{e0}/q_i.

D. Fast particle collision operator

The energetic particle collision operator is

$$C(f_\alpha) = \frac{1}{v^2\tau_s}\frac{\partial}{\partial v}[(v^3 + v_I^3)f_\alpha(v)] + \frac{v_b^3}{v^3\tau_s}\hat{L}(f_\alpha) \tag{143}$$

where $\hat{L}$ is the Lorentz operator

$$\hat{L}(f) = \frac{1}{2}\frac{\partial}{\partial\lambda}(1-\lambda^2)\frac{\partial}{\partial\lambda} \tag{144}$$

The collision time is

$$\tau_s = \frac{3(2\pi)^{3/2}\epsilon_0^2 m_\alpha T_e^{3/2}}{Z_\alpha^2 e^4 m_e^{1/2} n_e \ln\Lambda} \tag{145}$$

and the critical velocity is given by

$$v_I = \left(\frac{n_i Z_i^2}{n_e}\cdot\frac{3\pi^{1/2}m_e}{4m_i}\right)^{1/3} v'_{T_e} \tag{146}$$

with $v'_{T_e} = (2T_e/m_e)^{1/2}$ and $v_b = (m_i/m_\alpha)^{1/3}v_I$

The pitch-angle scattering is implemented similar to the electrons. The velocity drag term is implemented by evolving the particle velocity

$$\frac{\mathrm{dv}}{\mathrm{dt}} = -\nu\left(v + \frac{V_I^3}{v^2}\right) \qquad \nu = \frac{1}{\tau_s} \tag{147}$$

As markers slow-down, they are removed at $v = v_1$, recycled with $v = v_c$ (the birth velocity). It is worth noting that the slowing-down distribution $f = \frac{1}{v^3 + V_I^3}$ is a solution of the collision operator Eq. 143, whereas a constant phase-space distribution is not.

E. The kink term

For MHD phenomena the equilibrium current gradient can be an important drive for instabilities, and should be included in the simulation. The equilibrium current is given by $\nabla \times \mathbf{B}_0 = \beta_u \mathbf{j}_0$,

$$\mathbf{b} \cdot \nabla \times \mathbf{B_0} = \frac{\mathbf{1}}{\mathbf{B_0}} \left[\nabla\zeta \times \nabla\psi \cdot \nabla \mathbf{f} \times \nabla\zeta + \frac{\mathbf{f}}{\mathbf{R}} \left(\frac{\partial \mathbf{B}_\theta}{\partial \mathbf{r}} - \frac{\partial \mathbf{B_r}}{\partial \theta} \right) |\nabla \mathbf{r} \times \nabla\theta| \right] \tag{148}$$

Assuming the parallel equilibrium currentin is all carried by the electrons, the equilibrium electron flow $u_{\parallel e0}$ can be calculated. It enters the electron continuity equation through the following term,

$$\nabla \cdot \left[n u_{\parallel e0} \left(\mathbf{b} + \frac{\delta \mathbf{B}_\perp}{\mathbf{B_0}} \right) \right] = \nabla \cdot (\delta(n_e u_{\parallel e}) \mathbf{b}) + \delta \mathbf{B}_\perp \cdot \nabla (\mathbf{n}_{e0} \mathbf{u}_{\parallel e0} / \mathbf{B_0}) \tag{149}$$

$$\begin{aligned} \delta \mathbf{B}_\perp \cdot \nabla \left(\frac{n_{e0} u_{\parallel e0}}{B_0} \right) \;=\; & -\frac{\partial A_\parallel}{\partial x} \frac{\partial}{\partial \theta} \left(\frac{n_{0e} u_{\parallel e0}}{B_0} \right) \mathbf{b} \cdot \nabla r \times \nabla\theta + \frac{\partial A_\parallel}{\partial y} \left[-\frac{\partial y}{\partial r} \frac{\partial}{\partial \theta} \left(\frac{n_{e0} u_{\parallel e0}}{B_0} \right) + \right. \\ & \left. \frac{r_0}{q_0} \hat{q} \frac{\partial}{\partial r} \left(\frac{n_{e0} u_{\parallel e0}}{B_0} \right) \right] \mathbf{b} \cdot \nabla r \times \nabla\theta \end{aligned}$$

F. Kinetic electron closure

In principle electron PIC simulation can be used to provide kinetic closure for the pressure terms in the continuity equation Eq. 124 and the Ohm's law Eq. 125. Such a closure scheme has been studied in slab [29] and toroidal flux-tube [30], and implemented in toroidal global geometry. In fact, various variants of the closure scheme have been tested. For instance, the $v_\parallel$-formalism is obtained if the Ampere's equation is solved for $A_\parallel$ and Eq.125 is solved for $E_\parallel$, and the electron density is obtained from the kinetic electrons directly. Approximate closure schemes have also been studied. All of these efforts aim to include trapped electron effects in the fluid electron model. But no effective schemes have been found. Whenever kinetic electron effects are needed the direct method should be used. The fluid electron model with the isothermal condition should be used only for low-n MHD phenomena. Nevertheless, the subroutines for kinetic electron closure are kept in the hybrid model code for further exploration.

G. Main subroutines

ppush()	main ion pusher
cpush()	ion push full step Δt
mpush()	Maxwellian EP pusher
mcush()	
bpush()	anistropic beam EP pusher
bcush()	
hpush()	isotropic slowing-down EP pusher
hcush()	
pint()	kinetic electron closure, electron pusher (is Ke=1)
cint()	electron push full step Δt
pintef()	evolving $\frac{\partial \delta n_e}{\partial t} = \ldots.$, $\frac{\partial A_\parallel}{\partial t} = -E_\parallel - \nabla_\parallel \phi$, $\frac{\partial \delta T_e}{\partial t} = \ldots.$, $\frac{\partial \phi}{\partial t} = \dot{\phi}$
cintef	
grid1()	deposition for ion density and $j_\parallel$
jie()	deposition for the RHS of vorticity equation
drdt()	calculates RHS of vorticity equation
dpdt()	solver of vorticity equation
OA jie814()	deposition for the RHS of Ohm if is Ke=1
drdt814()	calcutes the RHS of Ohm
eqmo814()	solver of Ohm
fltx()	filters. gam() called in fltx() to treat n=0 specially.
filter	
ezmap()	solves Ampere's equation backwards, form $A_\parallel$ to $U_{\parallel e}$
field()	calls grad() to get $E_x, E_y, \delta B_x, \delta B_y$, also calls eqmo814()

Table V: Main subroutines for the hybrid model

VIII. APPENDIX A: BASIC DIAGNOSTICS

In principle, all particle arrays and field variables can be written to external files at each time step, to be used for any post-simulaiton data analysis. But this is impossible on most computer systems, as it will take too much disk space. Instead, only a small subset of data are routinely saved. These include some quantities of general interest, such as the root-mean-square values of ϕ, $A_\parallel$ and particles weights, which are frequently used to monitor the quality of a simulation. With experience, it is usually possible to tell from these quantities whether the simulation is numerically unstable (due to too large a time step, or too few particles, etc.). The evolution of individual Fourier mode amplitude is also of interest. It can be used to obtain the frequency and growth rate of individual toroidal mode. The subroutines yveck and yveck1 provides example codes that perform Fourier decomposition for any quantities that are defined on the (x, y, z) grids. Fourier decomposition in the straight-field-line poloidal angle θ_f (in Table IV) can be done similar to that in subroutine pol2d. This is needed if, e.g. the radial profile of individual poloidal harmonics is to be computed.

The source file outd.f90 contains subroutines that produce data for mode structure analysis. Every nplot time steps, 2-D data of ϕ and $A_\parallel$ on the $x-y$ plane at $z = L_z/2$, and on the $x-z$ plane at $y = Ly/2$, and the entire 3-D data, are saved in the out subdirectory in the run directory. The 3-D data can be used to make contour plots in the poloidal plane, and this is done in subroutine pol2d. The mode contour plots in the $x-y$ and $x-z$ planes are useful for quick examination of the mode structure, as this does not involve transformation to laboratory coordinates.

For validation studies, particle and energy fluxes or equivalently, particle and heat diffusivities, are the most frequently used quantities to compare with experimental measurements. The volume averaged radial energy flux in a volume V_s is,

$$\begin{aligned} Q &= \frac{1}{V_s}\int \frac{1}{2}mv^2\,\delta f\,\left(\frac{\mathbf{E}\times\mathbf{b}}{B_0}+v_\parallel\frac{\delta\mathbf{B}_\perp}{B_0}\right)\cdot\frac{\nabla r}{|\nabla r|}\,d\mathbf{x}\,d\mathbf{v} \\ &= \frac{1}{V_s}\sum_{j\in V_s} w_j\frac{1}{2}mv_j^2\left(V_{\rm Er}+v_\parallel\frac{\delta B_r}{B_0}\right) \end{aligned}$$

The summation is over all particles in the volume V_s. Particle number flux Γ is defined with $mv^2/2$ replaced by 1 in the above formulae.

The particle diffusivity D is defined by

$$\Gamma = -D\frac{\partial n}{\partial r} \tag{150}$$

and the heat diffusivity χ is defined by

$$Q = -n_0\chi\frac{\partial T}{\partial r} \tag{151}$$

The gyro-Bohm particle flux $\Gamma_{\rm GB}$ (pflxgb) and the gyro-Bohm heat flux $Q_{\rm GB}$ (eflxgb) are defined by

$$\begin{aligned} \Gamma_{GB} &= n_e c_s\left(\frac{\rho_s}{a}\right)^2 \\ Q_{GB} &= n_e c_s T_e\left(\frac{\rho_s}{a}\right)^2. \end{aligned}$$

Fluxes are saved in the variables eflx_es, pfl_es, etc.. The root-mean-square values of fields and weights are written to the file plot, and fluxes are written to the file flux.

How to get the mode frequency and growth rate for a linear case?
To get mode frequency and growth rate, one usually runs linearly (nonlin=nonline=0) with only one mode retained in the simulation (onemd=1). In a nonlinear run with many modes, the linear stage is sometimes too short to obtain a good estimate of the mode frequency and growth rate. Drift waves have very low frequency, so the frequency is more difficult to obtain. In GEM, the subroutine ftcamp() and the data camp (computed in subroutineyveck1()) are used to calculate the frequency spectrum of the mode at a few radial locations, these spectrum data are written to the file freq. To measure the linear growth rate, it is better to use the flux data, for instance the main ion energy flux, rather than a measure of the field amplitude such as rms(ϕ). This is because the flux, being quadratic in perturbations, more readily picks up coherent information that is contained in a growing physical mode. Needless to say that a flux grows at twice the linear growth rate in linear simulations.

How to perform a linear mode number scan?
A scan over the toroidal mode number n is done by changing lymult, and setting onemd=mlk=1 in gem.in. The simulation will include only the fundamental mode in the simulaiton. Each toroidal mode needs a separate simulation. Including multiple modes in a single simulation is fine in principle, but then different modes will have different resolution, which might not be desirable in the mode number scan.

[1] E. Frieman and L. Chen, Phys. Fluids **25**, 502 (1982).
[2] M. Artun and W. M. Tang, Phys. Plasmas **1**, 8 (1994).
[3] H. Sugama and W. Horton, Phys. Plasmas **5**, 2560 (1998).
[4] W. W. Lee, Phys. Fluids **26**, 556 (1983).
[5] M. Kotschenreuther, Bull. Am. Phys. Soc. **34**, 2107 (1988).
[6] S. Parker and W. Lee, Phys. Fluids B **5**, 77 (1993).
[7] C. C. Kim and S. E. Parker, J. Comput. Phys. **161**, 589 (2000).
[8] Y. Chen and S. E. Parker, Phys. Plasmas **8**, 441 (2001).
[9] I. Manuilskiy and W. W. Lee, Phys. Plasmas **7**, 1381 (2000).
[10] Y. Chen and S. E. Parker, Phys. Plasmas **8**, 2095 (2001).
[11] J. Cummings, Ph.D. thesis, Plasma Physics Lab, Princeton University (1994).
[12] Y. Chen and S. E. Parker, J. Comput. Phys. **189**, 463 (2003).
[13] Y. Chen and S. E. Parker, J. Comput. Phys. **220**, 839 (2007).
[14] Y. Chen and S. E. Parker, Phys. Plasmas **14**, 082301 (2007).
[15] A. Mishchenko, A. Konies, R. Kleiber, and M. Cole, Phys. Plasmas **21**, 092110 (2014).
[16] A. J. Brizard and T. S. Hahm, Review of Modern Physics **79**, 421 (2007).
[17] Y. Chen, J. Chowdhury, S. E. Parker, and W. Wan, Phys. Plasmas **22**, 042111 (2015).
[18] J. Candy and R. Waltz, J. Comput. Phys. **186**, 545 (2003).
[19] M. Kotschenreuther, G. Rewoldt, and W. Tang, Comp. Phys. Comm. **88**, 128 (1995).
[20] W. Dorland, F. Jenko, M. Kotschenreuther, and B. Rogers, Phys. Rev. Lett. **85**, 5579 (2000).
[21] T. S. Hahm, W. W. Lee, and A. Brizard, Phys. Fluids **31**, 1940 (1988).
[22] G. Hu and J. Krommes, Phys. Plasmas **1**, 863 (1994).
[23] Y. Chen and R. B. White, Phys. Plasmas **10**, 3591 (1997).
[24] W. Lee, J. Comput. Phys. **72**, 243 (1987).
[25] A. H. Boozer and G. Kuo-Petravic, Phys. Fluids **24**, 851 (1981).
[26] Y. Chen, T. Munsat, S. E. Parker, W. W. Heidbrink, M. A. V. Zeeland, B. J. Tobias, and C. W. Domier, Phys. Plasmas **20**, 012109 (2013).
[27] J. V. W. Reynders, Ph.D. thesis, Princeton University (1992).
[28] P. Snyder and G. Hammett, Phys. Plasmas **8**, 3199 (2001).
[29] Y. Chen and S. E. Parker, Phys. Plasmas **18**, 055703 (2011).
[30] Y. Chen, S. E. Parker, W. Wan, and R. Bravenec, Phys. Plasmas **20**, 092511 (2013).