LLNL-JRNL-735522

Kinetic Simulation of Collisional Magnetized Plasmas with Semi-Implicit Time Integration

Debojyoti Ghosh · **Mikhail A. Dorf** ·
Milo R. Dorr · **Jeffrey A. F. Hittinger**

Abstract Plasmas with varying collisionalities occur in many applications, such as tokamak edge regions, where the flows are characterized by significant variations in density and temperature. While a kinetic model is necessary for weakly-collisional high-temperature plasmas, high collisionality in colder regions render the equations numerically stiff due to disparate time scales. In this paper, we propose an implicit-explicit algorithm for such cases, where the collisional term is integrated implicitly in time, while the advective term is integrated explicitly in time, thus allowing time step sizes that are comparable to the advective time scales. This partitioning results in a more efficient algorithm than those using explicit time integrators, where the

D. Ghosh
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, 94550
E-mail: ghosh5@llnl.gov

M. A. Dorf
Physics Division, Lawrence Livermore National Laboratory, Livermore, CA, 94550
E-mail: dorf1@llnl.gov

M. R. Dorr
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, 94550
E-mail: dorr1@llnl.gov

J. A. F. Hittinger
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, 94550
E-mail: hittinger1@llnl.gov

time step sizes are constrained by the stiff collisional time scales. We implement semi-implicit additive Runge-Kutta methods in COGENT, a finite-volume gyrokinetic code for mapped, multiblock grids and test the accuracy, convergence, and computational cost of these semi-implicit methods for test cases with highly-collisional plasmas.

1 Introduction

The purpose of this paper is to describe the application and performance of a semi-implicit time integration algorithm for the solution of a system of Vlasov-Fokker-Planck equations, motivated by the goal of simulating the edge plasma region of tokamak fusion reactors. Plasma dynamics in the tokamak edge region is an unsteady multiscale phenomenon, characterized by a large range of spatial and temporal scales due to the density and temperature variations. Figure 1 shows the cross-section of a typical tokamak fusion reactor. The geometry is defined by the magnetic flux surfaces that contain the plasma, and the core and the edge regions are marked. Within the edge region, as the temperature decreases from the hot near-core region to the cold outer-edge region, there are three scale regimes. The hot and dense plasma in the inner edge region adjacent to the core plasma is weakly collisional, and the mean free paths of the particles are significantly larger than the density and temperature gradient length scales [20,26,25]. Near the separatrix that separates the closed magnetic field lines from the open ones, the plasma is moderately collisional, and the mean free paths are comparable to the density and temperature gradient scales. At the outer edge (near the material surfaces), the cold plasma is strongly collisional. In this region, the particle mean free paths are significantly smaller than the density and temperature length scales.

Due to the weak collisionality near the core, a kinetic description with an appropriate collision model is required to model accurately the perturbations to the velocity distribution from the Maxwellian distribution. The Vlasov-Fokker-Planck (VFP) equation governs the evolution of the distribution function of each charged particle in the position and velocity space [80]. In the presence of a strong, externally-applied magnetic field, the ionized particles gyrate around the magnetic field lines. In the context of tokamaks, the radius of this gyromotion (gyroradius) is much smaller than the characteristic length scales. The gyrokinetic VFP equation represents the dynamics of the particles averaged over the gyromotion [58,20], i.e., it describes the motion of the guiding center of the particles. It is thus expressed in terms of the gyrocentric coordinates (parallel and perpendicular to the magnetic field lines) and does not contain phase-dependent terms. As a result, one velocity dimension and the fast time scale of the gyromotion is removed. The drift-kinetic model is the long wavelength limit of the gyrokinetic model, where turbulent length scales (comparable to the gyroradius) are neglected. We consider the drift-kinetic VFP equation in this paper.

Numerical algorithms to solve the gyrokinetic equation can be classified into three families. The Lagrangian particle-in-cell (PIC) approach [14,38,39,18,16,17] solves
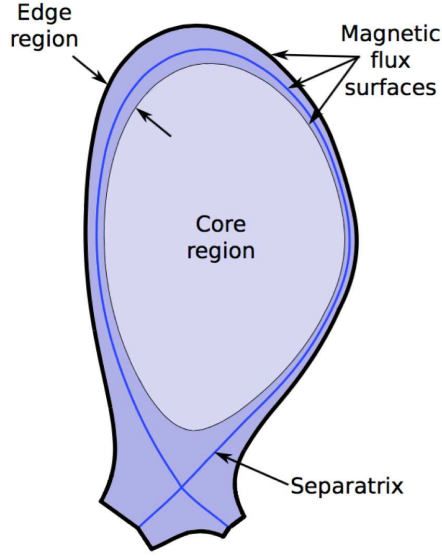
Fig. 1 Cross-section of a tokamak fusion reactor. The edge region is shaded in dark blue while the core region is shaded in light blue. The geometry is defined by magnetic flux surfaces. The separatrix (blue line) separates the open flux surfaces outside it from the closed flux surfaces inside it.

the equations of motion for "superparticles" in the Lagrangian frame. The primary drawback is that the number of superparticles needed to control the numerical noise is very large [80, 32]. Eulerian methods [42, 77, 84, 20, 85, 83] solve the VFP equation on a fixed phase space grid, where the phase space comprises the spatial position and velocity coordinates. Semi-Lagrangian approaches [36, 73, 22, 70, 71] follow characteristics, either tracing backward or forward in time, requiring interpolation of the solution from or to a fixed phase-space grid. In this paper, we adopt the Eulerian approach that allows the use of advanced numerical algorithms developed by the computational fluid dynamics (CFD) community.

The gyrokinetic VFP equation is a parabolic partial differential equation (PDE) that comprises an advective Vlasov term and the Fokker-Planck collision term. The Vlasov term describes incompressible flow in the phase space [80, 37], i.e., the evolution of the distribution function by the gyrokinetic velocity of the particles and their acceleration parallel to the magnetic field. The Fokker-Planck term describes the collisional relaxation of the distribution function to the Maxwellian distribution [80, 72] and acts in the velocity space only. The difference in the time scales of these two terms varies significantly in the edge region due to the variation in the collisionality. The weakly and moderately collisional plasma at the inner edge and the separatrix, respectively, exhibits collisional time scales much smaller or comparable to the Vlasov time scales, while the strongly-collisional plasma at the outer edge exhibits collisional time scales much smaller than the Vlasov time scales. Consequently, the VFP equation exhibits disparate scales and is numerically stiff. Explicit time integration

methods are inefficient because the time step size is constrained by the collisional time scale. Implicit time integration methods, on the other hand, are unconditionally linearly stable, but they require solutions to a linear or nonlinear system of equations. Linearly implicit [61,8,59,55,56,60,11] and nonlinearly implicit methods [57,28, 13,79,78] have been applied to the Fokker-Planck operator in isolation, and several implicit algorithms have also been proposed for the VFP equation [29,65,50,81]. In our context, the accurate resolution of the unsteady dynamics requires time steps comparable to the Vlasov time scales, and thus a fully-implicit approach is inefficient.

Semi-implicit or implicit-explicit (IMEX) time integration methods [1,66,48] allow the partitioning of the right-hand-side (RHS) into two parts: the stiff component integrated implicitly, comprising time scales faster than those of interest, and the nonstiff component integrated explicitly, comprising the slower scales. Such methods have been applied successfully to many multiscale applications [27,35,33,44, 52,47,76,82]. In the context of the Vlasov equation, semi-implicit approaches have been proposed [19,10,41,62] that partition the Vlasov term and integrate the parallel advection implicitly in time.

In this paper, we propose a semi-implicit algorithm for the gyrokinetic VFP equation, where we integrate the Vlasov term explicitly while the stiff Fokker-Planck term is integrated implicitly to allow time steps that are comparable to the Vlasov time scales. The Fokker-Planck term represents a nonlinear advection-diffusion operator that can be expressed in two forms: the Landau form [53] and the Rosenbluth form [72]. Although the two forms are theoretically equivalent, they differ in their implications on the implicit numerical solution of this term. The Landau form expresses the advection and diffusion coefficients as direct integrals and is well-suited for conservative numerical methods [15,49,57,28,61,7,4]. However, when integrated implicitly, the integral form results in a dense, nonlinear system of equations. Naive algorithms scale as $O\left(N^2\right)$, where N is the number of velocity space grid points, whereas fast algorithms [31,8,2,6] scale as $O\left(N \log N\right)$. The Rosenbluth form, on the other hand, relates the advection and diffusion coefficients to the distribution function through Poisson equations for the Rosenbluth potentials in the velocity space. Thus, each evaluation of the Fokker-Planck term requires the solution to the Poisson equations, which scales as $O\left(N\right)$, assuming that an efficient Poisson solver is available. The primary difficulty is the need for solving the Poisson equations, defined on an infinite velocity space, on the truncated numerical domain, and several approaches have been proposed [45,3,64,54,67,55,13]. An additional difficulty with the Rosenbluth form is the difficulty in enforcing mass, momentum, and energy conservation [12, 79], and modifications are necessary to ensure energy and momentum are conserved to round-off errors. In this paper, we consider the Rosenbluth form of the Fokker-Planck term.

Our semi-implicit approach is implemented in COGENT [24,26,25], a high-order finite-volume code that solves the gyrokinetic VFP equations on mapped, multiblock grids representing complex geometries. We implement multistage, conservative additive Runge-Kutta (ARK) methods [48], where the resulting nonlinear system of equations for the implicit stages are solved using the Jacobian-free Newton-Krylov approach [51]. We investigate the performance of the ARK methods for test problems representative of the tokamak edge and compare it to that of the explicit Runge-Kutta

(RK) methods for VFP problems. In particular, we verify the accuracy and convergence of the ARK methods and their computational efficiency with respect to the RK methods. Although the preconditioning that we have implemented reduces the cost of the implicit solve significantly, our approach is a preliminary effort, and the implementation of better preconditioning techniques will be addressed in the future.

The outline of the paper is as follows. Section 2 introduces the gyrokinetic VFP equations and the nondimensionalization used in our implementation. Section 3 describes the high-order finite-volume method that COGENT uses to discretize the equations in space. The time integration methods are discussed in Section 4. The algorithm verification through two test cases is presented in Section 5, and Section 6 summarizes the contributions of this paper.

2 Governing Equations

The plasma dynamics at the tokamak edge are described by the full-f gyrokinetic VFP equation [37, 26]. In this paper, we consider a single species, and the axisymmetric governing equation is expressed in the nondimensional form as

$$\frac{\partial B_{\parallel}^* f}{\partial t} + \nabla_{\mathbf{R}} \cdot \left(\dot{\mathbf{R}} B_{\parallel}^* f \right) + \frac{\partial}{\partial v_{\parallel}} \left(\dot{v}_{\parallel} B_{\parallel}^* f \right) = c_f(f), \tag{1}$$

where $f \equiv f\left(\mathbf{R}, v_{\parallel}, \mu, t\right)$ is the distribution function defined on the phase space $\left(\mathbf{R}, v_{\parallel}, \mu\right)$. $\mathbf{R} \equiv (r, \theta)$ is the spatial gyrocenter position vector in the configuration space with r as the radial coordinate and θ as the poloidal coordinate, $v_{\parallel}$ is the velocity parallel to the externally applied magnetic field $\mathbf{B}$, and $\mu = m v_{\perp}^2 / |\mathbf{B}|$ is the magnetic moment. The configuration space $\mathbf{R}$ is two-dimensional, and (1) is a four-dimensional (2D–2V) PDE. $\nabla_{\mathbf{R}}$ denotes the divergence operator in the configuration space. In the long wavelength limit, the gyrokinetic model reduces to the drift-kinetic model, and the Vlasov velocity and parallel acceleration are given by

$$\dot{\mathbf{R}} \equiv \dot{\mathbf{R}}\left(\mathbf{R}, v_{\parallel}, \mu, t\right) = \frac{1}{B_{\parallel}^*} \left[v_{\parallel} \mathbf{B}^* + \frac{\lambda_a}{Z} \hat{\mathbf{b}} \times \left(Z\mathbf{E} + \frac{\mu}{2} \nabla_{\mathbf{R}} B \right) \right], \tag{2a}$$

$$\dot{v}_{\parallel} \equiv \dot{v}_{\parallel}\left(\mathbf{R}, v_{\parallel}, \mu, t\right) = -\frac{1}{m B_{\parallel}^*} \mathbf{B}^* \cdot \left(Z\mathbf{E} + \frac{\mu}{2} \nabla_{\mathbf{R}} B \right), \tag{2b}$$

where m and Z are the mass and ionization state, respectively, λ_a is the Larmor number (ratio of the gyroradius to the characteristic length scale), $\hat{\mathbf{b}} = \mathbf{B}/|\mathbf{B}|$ and $B = |\mathbf{B}|$ are the unit vector along the magnetic field and magnitude of the magnetic field, and

$$\mathbf{B}^* \equiv \mathbf{B}^*\left(\mathbf{R}, v_{\parallel}\right) = \mathbf{B} + \lambda_a \frac{m v_{\parallel}}{Z} \nabla \times \hat{\mathbf{b}}. \tag{3}$$

$B_{\parallel}^* = \mathbf{B}^* \cdot \hat{\mathbf{b}}$ is the Jacobian of the transformation from the lab frame to the gyrocentric coordinates. In this paper, we consider cases with uniform magnetic field, and thus $\mathbf{B}^* \equiv \mathbf{B}$. The electric field is $\mathbf{E} = -\nabla_{\mathbf{R}} \phi$, where ϕ is the electrostatic potential. In this paper, we consider cases where $\phi = \phi\left(\mathbf{R}\right)$ is specified; however, COGENT

generally computes a self-consistent electrostatic potential from the species charge densities [24, 26] where a Boltzman or vorticity model is assumed for the electrons.

The Rosenbluth form of the Fokker-Planck collision term for a single species is expressed as [25]

$$c(f) = v_c \nabla_{(v_\parallel, \mu)} \cdot \overrightarrow{\Gamma}, \tag{4a}$$

$$\overrightarrow{\Gamma} = \overrightarrow{\sigma} f + \overleftrightarrow{\kappa} \nabla_{(v_\parallel, \mu)} f, \tag{4b}$$

where v_c is the collision frequency, and the coefficients are defined as follows:

$$\overrightarrow{\sigma} = \begin{bmatrix} \sigma_{v_\parallel} \\ \sigma_\mu \end{bmatrix}, \overleftrightarrow{\kappa} = \begin{bmatrix} \kappa_{v_\parallel v_\parallel} & \kappa_{v_\parallel \mu} \\ \kappa_{\mu v_\parallel} & \kappa_{\mu\mu} \end{bmatrix}, \tag{5}$$

where

$$\sigma_{v_\parallel} = \frac{\partial \varphi}{\partial v_\parallel}, \quad \sigma_\mu = 4\mu \left(\frac{m}{B}\right) \frac{\partial \varphi}{\partial \mu}, \quad \kappa_{v_\parallel v_\parallel} = -\frac{\partial^2 \rho}{\partial v_\parallel^2},$$

$$\kappa_{v_\parallel \mu} = \kappa_{\mu v_\parallel} = -4\mu \left(\frac{m}{B}\right) \frac{\partial^2 \rho}{\partial v_\parallel \partial \mu}, \quad \kappa_{\mu\mu} = -8\mu \left(\frac{m}{B}\right)^2 \left[2\mu \frac{\partial^2 \rho}{\partial \mu^2} + \frac{\partial \rho}{\partial \mu}\right]. \tag{6}$$

The operator $\nabla_{(v_\parallel, \mu)}$ denotes the gradient operator in the velocity space $(\partial/\partial v_\parallel, \partial/\partial \mu)$. The Rosenbluth potentials, φ and ρ, are related to the distribution function through the following Poisson equations in the two-dimensional velocity space $(v_\parallel, \mu)$:

$$\frac{\partial^2 \varphi}{\partial v_\parallel^2} + \frac{m}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \varphi}{\partial \mu}\right) = f, \tag{7a}$$

$$\frac{\partial^2 \rho}{\partial v_\parallel^2} + \frac{m}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \rho}{\partial \mu}\right) = \varphi. \tag{7b}$$

The Maxwellian distribution function, defined as

$$f_{\mathrm{M}}(\mathbf{R}, v_\parallel, \mu) = \frac{n(\mathbf{R})}{\sqrt{\pi}} \left(\frac{m}{2T(\mathbf{R})}\right)^{\frac{3}{2}} \exp\left(-\frac{m\{v_\parallel - \overline{v_\parallel}(\mathbf{R})\}^2 + \mu B(\mathbf{R})}{2T(\mathbf{R})}\right), \tag{8}$$

is an exact solution of the Fokker-Planck collision operator ($c(f_{\mathrm{M}}) = 0$), where the temperature $T(\mathbf{R})$ is given by

$$T(\mathbf{R}) = \frac{1}{n(\mathbf{R})} \int \int \frac{1}{3} \left[(v_\parallel - \overline{v_\parallel}(\mathbf{R}))^2 + \mu B\right] f(\mathbf{R}, v_\parallel, \mu) B_\parallel^*(\mathbf{R}, v_\parallel) dv_\parallel d\mu \tag{9}$$

and

$$n(\mathbf{R}) = \frac{1}{m} \int \int f(\mathbf{R}, v_\parallel, \mu) B_\parallel^*(\mathbf{R}, v_\parallel) dv_\parallel d\mu, \tag{10}$$

$$\overline{v_\parallel}(\mathbf{R}) = \frac{1}{mn(\mathbf{R})} \int \int v_\parallel f(\mathbf{R}, v_\parallel, \mu) B_\parallel^*(\mathbf{R}, v_\parallel) dv_\parallel d\mu \tag{11}$$

are the number density and average parallel velocity, respectively.

The equations in the preceding discussion involve non-dimensional variables. The non-dimensionalization is derived by

$$
t = \frac{\tilde{t}}{t_{\text{ref}}}, \quad v_{\|} = \frac{\tilde{v}_{\|}}{v_{\text{ref}}}, \quad n = \frac{\tilde{n}}{n_{\text{ref}}}, \quad m = \frac{\tilde{m}}{m_{\text{ref}}}, \quad f = \frac{\tilde{f}}{f_{\text{ref}}}, \quad T = \frac{\tilde{T}}{T_{\text{ref}}},
$$

$$
B = \frac{\tilde{B}}{B_{\text{ref}}}, \quad \phi = \frac{\tilde{\phi}}{\phi_{\text{ref}}}, \quad \mu = \frac{\tilde{\mu}}{\mu_{\text{ref}}}, \quad x,y,L = \frac{\tilde{x},\tilde{y},\tilde{L}}{L_{\text{ref}}},
$$

where $(\tilde{\cdot})$ is the physical (dimensional) quantity and $(\cdot)_{\text{ref}}$ are the reference quantities. The Larmor number and the collision frequency are expressed as

$$
\lambda_a = \frac{m_{\text{ref}}v_{\text{ref}}}{eL_{\text{ref}}B_{\text{ref}}}, \qquad \nu_c = \frac{f_{\text{ref}}t_{\text{ref}}e^4}{m_{\text{ref}}^2}\Lambda_c \left(\frac{4\pi Z^2}{m} \right)^2,
$$

respectively, where Λ_c is the Coulomb logarithm. The primitive reference parameters are n_{ref} (number density), T_{ref} (temperature), L_{ref} (length), m_{ref} (mass), and B_{ref} (magnetic field); the derived reference quantities are

$$
v_{\text{ref}} = \sqrt{\frac{T_{\text{ref}}}{m_{\text{ref}}}} \text{ (thermal speed)}, \quad t_{\text{ref}} = \frac{L_{\text{ref}}}{v_{\text{ref}}} \text{ (transit time)},
$$

$$
\mu_{\text{ref}} = \frac{T_{\text{ref}}}{2B_{\text{ref}}} \text{ (magnetic moment)}, \quad f_{\text{ref}} = \frac{n_{\text{ref}}}{\pi v_{\text{ref}}^3} \text{ (distribution function)},
$$

$$
\phi_{\text{ref}} = \frac{T_{\text{ref}}}{e} \text{ (potential)};
$$

and e is the elementary charge.

3 Spatial Discretization

Equation (1) is a parabolic PDE, which we discretize in space using a fourth-order finite-volume method on a mapped grid [21,63]. The computational domain is defined as a four-dimensional hypercube of unit length in each dimension,

$$
\Omega = \{\boldsymbol{\xi} : 0 \le \xi_d \le 1, 1 \le d \le D\}; \quad \xi_d = \boldsymbol{\xi} \cdot \mathbf{e}_d, \tag{12}
$$

partitioned by a uniform grid with a computational cell defined as

$$
\omega_{\mathbf{i}} = \prod_{d=1}^{D} \left[\left(\mathbf{i} - \frac{1}{2}\mathbf{e}_d \right) h, \left(\mathbf{i} + \frac{1}{2}\mathbf{e}_d \right) h \right], \tag{13}
$$

where $D = 4$ is the number of spatial dimensions, $\mathbf{e}_d$ is the unit vector along dimension d, and $\mathbf{i}$ is an integer vector representing a four-dimensional grid index. The transformation $\mathbf{x} = \mathbf{X}(\boldsymbol{\xi})$ defines the mapping between a vector in the physical space

$\mathbf{x} \equiv (\mathbf{R}, v_\parallel, \mu)$ and a vector in the computational space $\boldsymbol{\xi}$. Equation (1) is integrated over a physical cell $\mathbf{X}(\omega_\mathbf{i})$ to yield the integral form

$$\frac{\partial}{\partial t} \left(\int_{\mathbf{X}(\omega_\mathbf{i})} B_\parallel^* f d\mathbf{x} \right) = \int_{\mathbf{X}(\omega_\mathbf{i})} \mathscr{V}\left(B_\parallel^* f \right) d\mathbf{x} + \int_{\mathbf{X}(\omega_\mathbf{i})} \mathscr{C}(f) d\mathbf{x}, \qquad (14)$$

where the Vlasov and collision terms are

$$\mathscr{V}\left(B_\parallel^* f \right) \equiv - \left[\nabla_\mathbf{R} \cdot \left(\dot{\mathbf{R}} B_\parallel^* f \right) + \frac{\partial}{\partial v_\parallel} \left(\dot{v}_\parallel B_\parallel^* f \right) \right], \quad \mathscr{C}(f) = c_f(f). \qquad (15)$$

Defining the computational cell-averaged solution as

$$\bar{f}_\mathbf{i} = \frac{1}{\underline{\omega_\mathbf{i}}} \int_{\underline{\omega_\mathbf{i}}} B_\parallel^* f J d\boldsymbol{\xi}; \quad J \equiv \left| \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right|, \qquad (16)$$

where $\underline{\omega_\mathbf{i}}$ is the volume of the computational cell $\omega_\mathbf{i}$, the physical cell-averaged solution is

$$\check{f}_\mathbf{i} = \frac{1}{\underline{\mathbf{X}(\omega_\mathbf{i})}} \int_{\mathbf{X}(\omega_\mathbf{i})} B_\parallel^* f d\mathbf{x} = \left(\int_{\mathbf{V}_\mathbf{i}} J d\boldsymbol{\xi} \right)^{-1} \int_{\omega_\mathbf{i}} B_\parallel^* f J d\boldsymbol{\xi} = \bar{J}_\mathbf{i}^{-1} \bar{f}_\mathbf{i}, \text{ where } \bar{J}_\mathbf{i} = \frac{1}{\underline{\omega_\mathbf{i}}} \int_{\underline{\omega_\mathbf{i}}} J d\boldsymbol{\xi}. \tag{17}$$

The Vlasov and collision terms in (14) can be written in the divergence form as follows:

$$\mathscr{V}\left(B_\parallel^* f \right) = \nabla_\mathbf{x} \cdot \mathbf{V}\left(B_\parallel^* f \right); \quad \mathbf{V}\left(B_\parallel^* f \right) = - \left[\left(\dot{\mathbf{R}} \cdot \hat{\mathbf{r}} \right) B_\parallel^* f, \ \left(\dot{\mathbf{R}} \cdot \hat{\boldsymbol{\theta}} \right) B_\parallel^* f, \ \dot{v}_\parallel B_\parallel^* f, \ 0 \right]^\mathrm{T}, \tag{18a}$$

$$\mathscr{C}(f) = \nabla_\mathbf{x} \cdot \mathbf{C}(f); \quad \mathbf{C}(f) = v_c \left[0, \ 0, \ \overrightarrow{\Gamma} \cdot \hat{\mathbf{v}}_\parallel, \ \overrightarrow{\Gamma} \cdot \hat{\boldsymbol{\mu}} \right]^\mathrm{T}, \tag{18b}$$

where $\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\mathbf{v}}_\parallel, \hat{\boldsymbol{\mu}}$ are the unit vectors along the $r, \theta, v_\parallel, \mu$ coordinates, respectively, and $\nabla_\mathbf{x}$ denotes the divergence operator in the physical space. Using the divergence theorem, (14) is expressed as

$$\frac{\partial \bar{f}_\mathbf{i}}{\partial t} = \frac{1}{\underline{\omega_\mathbf{i}}} \left[\int_{\partial \mathbf{X}(\omega_\mathbf{i})} \mathbf{V}\left(B_\parallel^* f \right) \cdot \hat{\mathbf{n}} ds + \int_{\partial \mathbf{X}(\omega_\mathbf{i})} \mathbf{C}(f) \cdot \hat{\mathbf{n}} ds \right]$$

$$= \frac{1}{\underline{\omega_\mathbf{i}}} \sum_{d=1}^{D} \left[\int_{A_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}} \mathbf{N}^\mathrm{T}(\mathbf{V}+\mathbf{C}) dA_{\boldsymbol{\xi}} - \int_{A_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}} \mathbf{N}^\mathrm{T}(\mathbf{V}+\mathbf{C}) dA_{\boldsymbol{\xi}} \right], \qquad (19)$$

where $\hat{\mathbf{n}}$ is the outward normal for $\partial \mathbf{X}(\omega_{\mathbf{i}})$, $\mathbf{N} = J\nabla_{\mathbf{x}}\boldsymbol{\xi}$, and $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ denote the faces along dimension d of the cell $\omega_{\mathbf{i}}$. The face-averaged Vlasov and collision fluxes on a given face are defined as

$$\left\langle \hat{\mathbf{V}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle = \frac{1}{\left| A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right|} \int_{A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}} \mathbf{N}^{\mathrm{T}}\mathbf{V}dA_{\boldsymbol{\xi}}; \quad \left\langle \hat{V}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle = \left\langle \hat{\mathbf{V}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle \cdot e_d, \tag{20a}$$

$$\left\langle \hat{\mathbf{C}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle = \frac{1}{\left| A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right|} \int_{A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}} \mathbf{N}^{\mathrm{T}}\mathbf{C}dA_{\boldsymbol{\xi}}; \quad \left\langle \hat{C}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle = \left\langle \hat{\mathbf{C}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right\rangle \cdot e_d. \tag{20b}$$

The computational cell is a hypercube with length h in each dimension with the volume as $\underline{\omega_{\mathbf{i}}} = h^D$, and the area of each face as $\left| A^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \right| = h^{D-1}$, $\forall d$. Thus, (19) can be written as

$$\frac{\partial \bar{f}_{\mathbf{i}}}{\partial t} = \frac{1}{h}\sum_{d=1}^{D}\left[\left(\left\langle \hat{V}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{V}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle\right) + \left(\left\langle \hat{C}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{C}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle\right)\right]. \tag{21}$$

Defining the solution vector as consisting of the cell-averaged distribution function at all the computational cells in the grid, (21) is expressed for the entire domain as a system of ordinary differential equations (ODEs) in time,

$$\frac{d\bar{\mathbf{f}}}{dt} = \hat{\mathscr{V}}\left(\bar{\mathbf{f}}\right) + \hat{\mathscr{C}}\left(\bar{\mathbf{f}}\right); \quad \bar{\mathbf{f}} = \left[\bar{f}_{\mathbf{i}}\right], \quad \mathbf{i} \in \left\{\mathbf{j} : \omega_{\mathbf{j}} \in \Omega\right\}; \tag{22a}$$

$$\hat{\mathscr{V}}\left(\bar{f}_{\mathbf{i}}\right) = \frac{1}{h}\sum_{d=1}^{D}\left(\left\langle \hat{V}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{V}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle\right) = \mathscr{V}\left(B_{\parallel}^* f\right) + \mathscr{O}\left(\Delta\mathbf{R}^p, \Delta v_{\parallel}^p, \Delta\mu^p\right), \tag{22b}$$

$$\hat{\mathscr{C}}\left(\bar{f}_{\mathbf{i}}\right) = \frac{1}{h}\sum_{d=1}^{D}\left(\left\langle \hat{C}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{C}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle\right) = \mathscr{C}\left(f\right) + \mathscr{O}\left(\Delta\mathbf{R}^q, \Delta v_{\parallel}^q, \Delta\mu^q\right), \tag{22c}$$

where p, q are the orders of the schemes used to discretize the Vlasov and collision terms, respectively.

Equation (22) requires the computation of the face-averaged Vlasov and collisional fluxes defined by (20), for which we use the discretization described in [21]. The following relationships between the cell-centered and face-centered quantities and cell-averaged and face-averaged quantities to fourth order will be used in the subsequent discussion:

$$\langle u \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = u_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \frac{h^2}{24}\sum_{\substack{d'=1 \\ d'\neq d}}^{D} \frac{\partial^2 u}{\partial\xi_{d'}^2}\bigg|_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \mathscr{O}\left(h^4\right), \tag{23a}$$

$$\bar{u}_{\mathbf{i}} = u_{\mathbf{i}} + \frac{h^2}{24}\sum_{d=1}^{D} \frac{\partial^2 u}{\partial\xi_d^2}\bigg|_{\boldsymbol{\xi}_{\mathbf{i}}} + \mathscr{O}\left(h^4\right), \tag{23b}$$

for an arbitrary variable u, where $\langle u \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}$ denotes the face-averaged value, $\bar{u}_{\mathbf{i}}$ denotes the cell-averaged value, and $u_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}$ and $u_{\mathbf{i}}$ denote the face-centered and cell-centered values, respectively. We note that it is sufficient to compute the second derivatives in the RHS of (23) to second order using centered finite-differences since they are multiplied by h^2. The face-averaged fluxes in (22) are computed to fourth order as

$$
\hat{V}^d_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} = \sum_{s=1}^{D} \langle N^s_d \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \langle V^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}
$$
$$
+ h^2 \sum_{s=1}^{D} \left\{ \mathbf{G}_0^{\perp,d} \langle N^s_d \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \right\} \cdot \left\{ \mathbf{G}_0^{\perp,d} \langle V^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \right\} + \mathscr{O}\left(h^4\right), \qquad (24a)
$$

$$
\hat{C}^d_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} = \sum_{s=1}^{D} \langle N^s_d \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \langle C^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}
$$
$$
+ h^2 \sum_{s=1}^{D} \left\{ \mathbf{G}_0^{\perp,d} \langle N^s_d \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \right\} \cdot \left\{ \mathbf{G}_0^{\perp,d} \langle C^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \right\} + \mathscr{O}\left(h^4\right), \qquad (24b)
$$

where $\mathbf{G}_0^{\perp,d} \approx \nabla_{\boldsymbol{\xi}} - \mathbf{e}^d \frac{\partial}{\partial \xi_d}$ and $\langle N^s_d \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}; s = 1, \cdots, D$ are the column vectors of the face-averaged metric quantities [21].

Equation (24a) requires computing $\langle V^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}, s = 1, \cdots, D$. Writing each component of four-dimensional Vlasov term as an advection operator,

$$
V^s = a^s f, \quad s = 1, \cdots, D; \quad a^s = \begin{cases} \left(\dot{\mathbf{R}} \cdot \hat{r}\right) & , s = d_r \\ \left(\dot{\mathbf{R}} \cdot \hat{\theta}\right) & , s = d_\theta \\ \dot{v}_{\parallel} & , s = d_{v_{\parallel}} \\ 0 & , s = d_\mu \end{cases}, \qquad (25)
$$

where $d_r, d_\theta, d_{v_{\parallel}}, d_\mu$ are the spatial dimensions corresponding to $r, \theta, v_{\parallel}, \mu$ coordinates, respectively, the face-averaged flux $\langle V^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}$ is computed to fourth order from the discrete convolution as

$$
\langle V^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} = \langle a^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} \left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} + \frac{h^2}{12} \sum_{d' \neq d} \left(\frac{\partial a^s}{\partial \xi_{d'}} \frac{\partial \bar{\bar{f}}}{\partial \xi_{d'}} \right)_{\boldsymbol{\xi}_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}} + \mathscr{O}\left(h^4\right). \qquad (26)
$$

The face-averaged advective coefficients $\langle a^s \rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}$ are computed by evaluating $\dot{\mathbf{R}}$ and $\dot{v}_{\parallel}$ at the face centers using (2) and transforming them to face-averaged quantities using (23a). The cell-averaged distribution function in the computational space $\bar{\bar{f}}_{\mathbf{i}}$ is defined and related to $\bar{f}_{\mathbf{i}}$ as follows:

$$
\bar{\bar{f}}_{\mathbf{i}} = \frac{1}{V_{\mathbf{i}}} \int_{V_{\mathbf{i}}} B^*_{\parallel} f \, d\boldsymbol{\xi} \quad \Rightarrow \bar{\bar{f}}_{\mathbf{i}} = J_{\mathbf{i}}^{-1} \left[\bar{f}_{\mathbf{i}} - \frac{h^2}{12} \nabla_{\xi} \bar{f} \cdot \nabla_{\xi} J \right] + \mathscr{O}\left(h^4\right). \qquad (27)
$$

The derivatives $\partial/\partial \xi_{d'}$ in (26) and ∇_{ξ} in (27) are computed using second-order central finite-differences since they are multiplied by h^2. Finally, the face-averaged distribution function in the computational space $\left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d}$ is computed from cell-averaged

values using the fifth-order weighted essentially nonoscillatory (WENO) scheme [46] with upwinding based on the sign of $\langle a^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$. Thus, the algorithm to compute $\hat{\mathscr{V}}(\bar{\mathbf{f}})$ in (22a) is fourth-order accurate, and $p = 4$ in (22b).

Equation (24b) requires the computation of the face-averaged collision flux $\langle C^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$. We note that the velocity space grid is Cartesian in our drift-kinetic model and independent of the configuration space grid; thus, $\langle C^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = 0$ if $s \neq d$, and (24b) only needs $\langle C^d \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$. The face-averaged flux is obtained from the face-centered collision flux

$$
C^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \begin{cases} \Gamma^{v_\parallel}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_{v_\parallel}} , & d = d_{v_\parallel} \\ \Gamma^{\mu}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_\mu} , & d = d_\mu \\ 0 , & \text{otherwise} \end{cases} \quad ; \quad \begin{aligned} \Gamma^{v_\parallel}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_{v_\parallel}} &= \overrightarrow{\Gamma}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_{v_\parallel}} \cdot \hat{\mathbf{v}}_\parallel, \\ \Gamma^{\mu}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_\mu} &= \overrightarrow{\Gamma}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_\mu} \cdot \hat{\boldsymbol{\mu}} \end{aligned} \tag{28}
$$

using (23a). Equation (4a) involves derivatives in the velocity space $(v_\parallel, \mu)$ only, and to simplify the subsequent discussion, two-dimensional grid indices (k, l) will be used, such that

$$
u_{k+p,l+q} \equiv u_{\mathbf{i}+p\mathbf{e}_{v_\parallel}+q\mathbf{e}_\mu}; \quad k = \mathbf{i} \cdot \mathbf{e}_{v_\parallel}, \quad l = \mathbf{i} \cdot \mathbf{e}_\mu \tag{29}
$$

for an arbitrary grid variable u and $p, q \in \mathbb{Z}$. The computation of $\Gamma^{v_\parallel}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_{v_\parallel}}$ is described in the following paragraphs and can be extended trivially to computing $\Gamma^{\mu}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_\mu}$.

Evaluating (4b) at the face $\mathbf{i} + \frac{1}{2}\mathbf{e}_{v_\parallel}$, the Fokker-Planck collision flux along the $v_\parallel$ dimension is given by

$$
\Gamma^{v_\parallel}_{k+\frac{1}{2},l} \equiv \Gamma^{v_\parallel}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{v_\parallel}} = \left[\sigma_{v_\parallel} f + \kappa_{v_\parallel v_\parallel} \frac{\partial f}{\partial v_\parallel} + \kappa_{v_\parallel \mu} \frac{\partial f}{\partial \mu} \right]_{k+\frac{1}{2},l}. \tag{30}
$$

The face-centered advective term is computed as

$$
\left[\sigma_{v_\parallel} f \right]_{k+\frac{1}{2},l} = \sigma_{v_\parallel,k+\frac{1}{2},l} f_{k+\frac{1}{2},l}, \tag{31}
$$

where the distribution function at the interface $f_{k+\frac{1}{2},l}$ is computed using the fifth-order upwind interpolation method based on the sign of $\sigma_{v_\parallel,k+\frac{1}{2},l}$, i.e.,

$$
\sigma_{v_\parallel,k+\frac{1}{2},l} < 0,
$$
$$
f_{k+\frac{1}{2},l} = \frac{1}{30} f_{k-2,l} - \frac{13}{60} f_{k-1,l} + \frac{47}{60} f_{k,l} + \frac{27}{60} f_{k+1,l} - \frac{1}{20} f_{k+2,l}, \tag{32a}
$$
$$
\sigma_{v_\parallel,k+\frac{1}{2},l} \geq 0,
$$
$$
f_{k+\frac{1}{2},l} = \frac{1}{30} f_{k+3,l} - \frac{13}{60} f_{k+2,l} + \frac{47}{60} f_{k+1,l} + \frac{27}{60} f_{k,l} - \frac{1}{20} f_{k-1,l}. \tag{32b}
$$

Note that the advective term is on the RHS of (1), and a negative value of $\sigma_{v_\parallel,k+\frac{1}{2},l}$ implies a right-moving wave. The cell-centered values of the distribution function

$f_{k,l}$ are obtained from its cell-averaged values using (23b). The diffusion terms are computed as follows:

$$\left[\kappa_{v_\parallel v_\parallel} \frac{\partial f}{\partial v_\parallel} \right]_{k+\frac{1}{2},l} = \kappa_{v_\parallel v_\parallel, k+\frac{1}{2},l} \left. \frac{\partial f}{\partial v_\parallel} \right|_{k+\frac{1}{2},l}, \tag{33}$$

and

$$\left[\kappa_{v_\parallel \mu} \frac{\partial f}{\partial \mu} \right]_{k+\frac{1}{2},l} = \frac{1}{2} \left(\kappa_{v_\parallel \mu, k+\frac{1}{2},l-\frac{1}{2}} \left. \frac{\partial f}{\partial \mu} \right|_{k+\frac{1}{2},l-\frac{1}{2}} + \kappa_{v_\parallel \mu, k+\frac{1}{2},l+\frac{1}{2}} \left. \frac{\partial f}{\partial \mu} \right|_{k+\frac{1}{2},l+\frac{1}{2}} \right), \tag{34}$$

where

$$\left. \frac{\partial f}{\partial \mu} \right|_{k+\frac{1}{2},l\pm\frac{1}{2}} = \frac{1}{2} \left(\left. \frac{\partial f}{\partial \mu} \right|_{k,l\pm\frac{1}{2}} + \left. \frac{\partial f}{\partial \mu} \right|_{k+1,l\pm\frac{1}{2}} \right). \tag{35}$$

The derivatives at the cell faces in (33) and (35) are computed using fourth-order central differences:

$$\left. \frac{\partial f}{\partial v_\parallel} \right|_{k+\frac{1}{2},l} = \frac{1}{\Delta v_\parallel} \left(\frac{1}{24} f_{k-1,l} - \frac{9}{8} f_{k,l} + \frac{9}{8} f_{k+1,l} - \frac{1}{24} f_{k+2,l} \right), \tag{36}$$

and similarly for $\partial f / \partial \mu|_{k,l+\frac{1}{2}}$. The advective and diffusive coefficients in (31), (33), and (34) are related to the Rosenbluth potentials through (6). Appendix A describes the computation of the Rosenbluth potentials from the distribution function by discretizing and solving (7), which is defined on the infinite velocity domain but is solved on a truncated numerical domain. Our current implementation is second-order accurate, and therefore, the coefficients in (31), (33), and (34) are calculated by discretizing (6) with second-order finite differences:

$$\sigma_{v_\parallel, k+\frac{1}{2},l} = \frac{1}{\Delta v_\parallel} \left(-\varphi_{k,l} + \varphi_{k+1,l} \right), \tag{37a}$$

$$\kappa_{v_\parallel v_\parallel, k+\frac{1}{2},l} = -\frac{1}{2} \left[\frac{\rho_{k-1,l} - 2\rho_{k,l} + \rho_{k+1,l}}{\Delta v_\parallel^2} + \frac{\rho_{k,l} - 2\rho_{k+1,l} + \rho_{k+2,l}}{\Delta v_\parallel^2} \right], \tag{37b}$$

$$\kappa_{v_\parallel \mu, k+\frac{1}{2},l+\frac{1}{2}} = -\left(\frac{2\mu_{l+\frac{1}{2}} m}{B} \right) \frac{\rho_{k,l} - \rho_{k+1,l} - \rho_{k,l+1} + \rho_{k+1,l+1}}{\Delta v_\parallel \Delta \mu}. \tag{37c}$$

The magnetic field magnitude B in (37c) varies in the configuration space only. Overall, the algorithm to compute $\hat{\mathscr{C}} \left(\bar{\mathbf{f}} \right)$ in (22a) is second-order accurate and $q = 2$ in (22c).

Mass and Energy Conservation The Fokker-Planck collision term conserves mass, momentum, and energy in its analytical form, and it is important that these quantities are discretely conserved as well. Mass conservation is expressed as

$$\int_{\Omega_{\mathbf{v}}} c(f)\, d\mathbf{v} = 0 \Rightarrow \int_{\partial\Omega_{\mathbf{v}}} \overrightarrow{\Gamma} \cdot d\mathbf{s} = 0 \tag{38}$$

where $\Omega_{\mathbf{v}} \equiv \left[-v_{\|,\max}, v_{\|,\max}\right] \times [0, \mu_{\max}]$ is the velocity space domain, and $\mathbf{v} \equiv \left(v_{\|}, \mu\right)$ is the velocity space coordinate. Equation (38) is satisfied if the normal flux $\overrightarrow{\Gamma} \cdot d\mathbf{s}$ is zero over the entire velocity space boundary, and this is enforced by setting the coefficients $\sigma_{v_{\|}}$, σ_{μ}, $\kappa_{v_{\|}v_{\|}}$, $\kappa_{\mu\mu}$, and $\kappa_{v_{\|}\mu}$ at the velocity boundaries to zero. Energy conservation is expressed as

$$\mathscr{E}_{\{c(f)\}}(\mathbf{R}) \equiv \int_{\Omega_{\mathbf{v}}} c(f)\left(v_{\|}^2 + \mu B\right) d\mathbf{v} = 0, \tag{39}$$

and is enforced using the approach described in [79], adapted for Cartesian velocity coordinates. Let

$$c^{(a)}(f) = v_c \nabla_{\left(v_{\|},\mu\right)} \cdot \left[\overrightarrow{\sigma} f\right], \quad c^{(d)}(f) = v_c \nabla_{\left(v_{\|},\mu\right)} \cdot \left[\overleftrightarrow{\kappa} \nabla_{\left(v_{\|},\mu\right)} f\right], \tag{40}$$

and let $\mathscr{\hat{C}}^{(a)}\left(\bar{\mathbf{f}}\right)$ and $\mathscr{\hat{C}}^{(d)}\left(\bar{\mathbf{f}}\right)$ be the corresponding spatially discretized terms. Equation (39) implies

$$\mathscr{E}_{\left\{c^{(d)}(f)\right\}}(\mathbf{R}) = -\mathscr{E}_{\left\{c^{(a)}(f)\right\}}(\mathbf{R}). \tag{41}$$

We define

$$v_{\mathscr{E}}(\mathbf{R}) = -\mathscr{E}_{\left\{\mathscr{\hat{C}}^{(a)}\left(\bar{\mathbf{f}}\right)\right\}}(\mathbf{R}) \big/ \mathscr{E}_{\left\{\mathscr{\hat{C}}^{(d)}\left(\bar{\mathbf{f}}\right)\right\}}(\mathbf{R}), \tag{42}$$

and modify (4) as

$$c(f) = v_c \nabla_{\left(v_{\|},\mu\right)} \cdot \overrightarrow{\Gamma}, \quad \overrightarrow{\Gamma} = \overrightarrow{\sigma} f + v_{\mathscr{E}}(\mathbf{R}) \overleftrightarrow{\kappa} \nabla_{\left(v_{\|},\mu\right)} f \tag{43}$$

We note that $v_{\mathscr{E}}(\mathbf{R}) \to -1$ as $\Delta v_{\|}, \Delta\mu \to 0$ for a consistent discretization of the collision term, and this factor counteracts the energy imbalance in the collision term resulting from the discretization errors. Equation (43) is discretized as decribed in the preceding discussions. Discrete enforcement of momentum conservation is also discussed in [79]; however, the numerical test cases considered in this paper have negligible average velocities and yielded stable and convergent results without momentum conservation enforcement. In our implementation, discrete energy conservation is necessary to preserve a Maxwellian solution; without the modification described by (43), unphysical cooling of the Maxwellian solution is observed that is alleviated by refining the grid in the velocity space. This behavior is similar to that reported in [79].
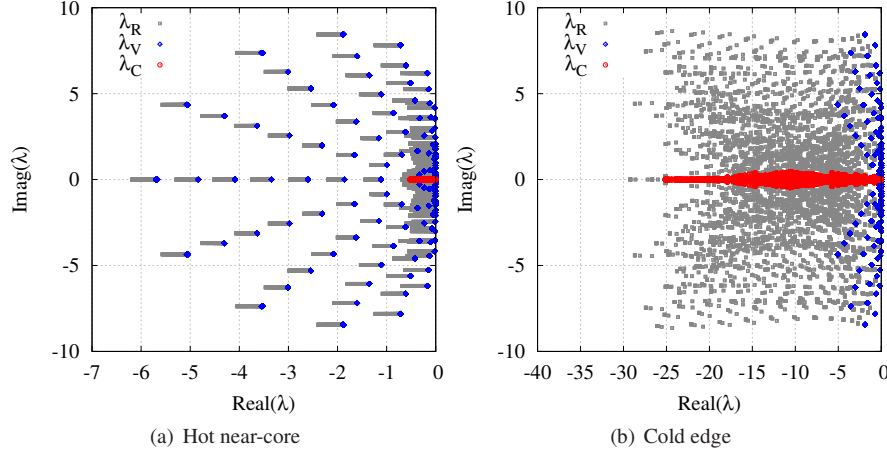
Fig. 2 Eigenvalues of the RHS of (22a) (λ_R) and its constituent terms – the Vlasov (λ_V) and collision terms (λ_C). (a) In the hot near-core region, the collisional eigenvalues (red) are much smaller than the Vlasov eigenvalues (blue). (b) However, in the cold edge region, the former are much larger in magnitude than the latter. Note the change in the horizontal scale.

4 Time Integration

Equation (22a) represents a system of ODEs in time that are solved using high-order, multistage explicit and IMEX time integrators in COGENT. The plasma density and temperature vary significantly in the edge region along the radial direction, and as a result, the stiffness of (22a) changes substantially from the inner edge region (adjacent to the core) to the outer edge. We use a simple problem to analyze the eigenvalues of the RHS of (22a). The two-dimensional (in space) Cartesian domain is $\mathbf{R} \equiv (r, \theta) \equiv (x, y)$ with specified density and temperature and a Maxwellian distribution function. The periodic domain is $0 \leq x, y < 1$, where $x = \tilde{x}/L_{\text{ref}}, y = \tilde{y}/L_{\text{ref}}$ are the non-dimensional spatial coordinates; $\tilde{x}, \tilde{y}$ are the physical coordinates; and $L_{\text{ref}} = 1\,\text{m}$ is the reference length. A uniform density $n = 1$ is specified, and the temperature is specified as

$$T = 1 + 0.1 \cos(2\pi y), \tag{44}$$

where n and T are nondimensional quantities. The magnetic field is $(B_x, B_y, B_z) = (0, 0.2, 2)$ where z is normal to the plane of the domain, and the reference magnetic field is $B_{\text{ref}} = 1\,\text{T}$. The reference mass is specified as the proton mass, i.e., $m_{\text{ref}} = 1.6726 \times 10^{-24}\,\text{g}$. We consider two values for the reference density and temperature, corresponding to two extremes of the edge region: the hot near-core region with $n_{\text{ref}} = 10^{20}\,\text{m}^{-3}$, $T_{\text{ref}} = 500\,\text{eV}$, and the cold edge region with $n_{\text{ref}} = 10^{19}\,\text{m}^{-3}$, $T_{\text{ref}} = 20\,\text{eV}$. These values are based on measurements in the edge region of a DIII-D tokamak [69].

The Jacobians of the RHS of (22a) and its constituents $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ are computed by applying finite differences on these operators, which scales as $\mathcal{O}(N^2)$ operations where N is the total number of grid points. Therefore, a very coarse grid with

$\left(N_x, N_y, N_{v_\parallel}, N_\mu \right) = (6, 16, 16, 12)$ points is used to discretize the domain for this analysis. Figure 2 shows the eigenvalues of the RHS of (22a) as well as its constituents evaluated separately:

$$\lambda_{\mathrm{R}} = \lambda \left(\left[\frac{d\hat{\mathscr{R}}\left(\bar{\mathbf{f}} \right)}{d\bar{\mathbf{f}}} \right] \right), \ \lambda_{\mathrm{V}} = \lambda \left(\left[\frac{d\hat{\mathscr{V}}\left(\bar{\mathbf{f}} \right)}{d\bar{\mathbf{f}}} \right] \right), \ \lambda_{\mathrm{C}} = \lambda \left(\left[\frac{d\hat{\mathscr{C}}\left(\bar{\mathbf{f}} \right)}{d\bar{\mathbf{f}}} \right] \right), \quad (45)$$

where $\hat{\mathscr{R}}\left(\bar{\mathbf{f}} \right) = \hat{\mathscr{V}}\left(\bar{\mathbf{f}} \right) + \hat{\mathscr{C}}\left(\bar{\mathbf{f}} \right)$ and $\lambda\left([\cdot] \right)$ are the eigenvalues of $[\cdot]$. In the hot near-core region, shown in Figure 2(a), the eigenvalues of the Vlasov term dominate the spectrum, and thus, the overall system of equations is not stiff with respect to the Vlasov time scale. However, in the cold edge region, shown in Figure 2(b), the strong collisionality results in the eigenvalues of $\hat{\mathscr{C}}$ being much larger than those of the eigenvalues of $\hat{\mathscr{V}}$. It should be noted that the magnitudes of the eigenvalues of $\hat{\mathscr{V}}$ are similar in the two figures.

The stable time step for an explicit time integration method is constrained by

$$\lambda_{\mathrm{R}} \Delta t \in \left\{ z : \mathscr{P}_{\exp}\left(z \right) \leq 1 \right\}, \quad (46)$$

where $\mathscr{P}$ represents the stability polynomial. It is evident from Figure 2 that, though an explicit method will allow time steps comparable to the Vlasov time scales in the near-core region, it will be constrained by the collisional time scales in the cold edge region. This fact motivates the use of the IMEX approach, where the collisions are integrated in time implicitly, while the Vlasov term is integrated explicitly; therefore, the stable time step for an IMEX method is constrained by

$$\lambda_{\mathrm{V}} \Delta t \in \left\{ z : \mathscr{P}_{\mathrm{IMEX}}\left(z \right) \leq 1 \right\}, \quad (47)$$

where $\mathscr{P}_{\mathrm{IMEX}}$ is the stability polynomial for the explicit component of the IMEX method. The IMEX approach allows time steps comparable to the Vlasov time scales in the entire edge region.

In this paper, we investigate high-order multistage ARK methods [48] that are expressed in the Butcher tableaux [9] form as:

$$\left(\frac{c_i \,\big|\, a_{ij}}{\,\big|\, b_j}, \frac{\tilde{c}_i \,\big|\, \tilde{a}_{ij}}{\,\big|\, \tilde{b}_j}; i, j = 1, \cdots, n_s \right), c_i = \sum_{j=1}^{n_s} a_{ij}, \tilde{c}_i = \sum_{j=1}^{n_s} \tilde{a}_{ij}, \quad (48)$$

where a_{ij}, b_j, c_i define the explicit component of the IMEX method, with $a_{ij} = 0, j \geq i$; $\tilde{a}_{ij}, \tilde{b}_j, \tilde{c}_i$ define the implicit component, with $\tilde{a}_{ij} = 0, j > i$; and n_s is the number of stages. The implicit components of the methods considered in this paper are L-stable, explicit-first-stage, single-diagonal-coefficient implicit Runge-Kutta (ESDIRK) methods, and the coefficients satisfy $\tilde{a}_{11} = 0$; $\tilde{a}_{ii} = \gamma, i = 2, \cdots, n_s$, where γ is a constant. In addition, the methods are conservative, and $b_i = \tilde{b}_i, \forall i$. A time step of the ARK methods applied to Eq. (22a) is expressed as follows:

$$\bar{\mathbf{f}}^{(i)} = \bar{\mathbf{f}}_n + \Delta t \left\{ \sum_{j=1}^{i-1} a_{ij} \hat{\mathscr{V}}\left(\bar{\mathbf{f}}^{(j)} \right) + \sum_{j=1}^{i} \tilde{a}_{ij} \hat{\mathscr{C}}\left(\bar{\mathbf{f}}^{(j)} \right) \right\}, \quad i = 1, \cdots, n_s, \quad (49a)$$

$$\bar{\mathbf{f}}_{n+1} = \bar{\mathbf{f}}_n + \Delta t \sum_{i=1}^{n_s} \left\{ b_i \hat{\mathscr{V}}\left(\bar{\mathbf{f}}^{(i)} \right) + \tilde{b}_i \hat{\mathscr{C}}\left(\bar{\mathbf{f}}^{(i)} \right) \right\}, \quad (49b)$$

where Δt is the time step, n denotes the time level ($\bar{\mathbf{f}}_n = \bar{\mathbf{f}}(t_n)$, $t_n = t_0 + n\Delta t$), and $\bar{\mathbf{f}}^{(i,j)}$ are the stage solutions. The second–order (three–stage) ("ARK2") [34], the third-order (four-stage) ("ARK3") [48], and the fourth-order (six-stage) ("ARK4") [48] are implemented.

Equation (49a) is a nonlinear system of equations for $i > 1$ that is expressed as

$$\mathscr{F}(\mathbf{y}) \equiv \alpha\mathbf{y} - \hat{\mathscr{C}}(\mathbf{y}) - \mathbf{r} = 0, \tag{50}$$

where $\mathbf{y} \equiv \bar{\mathbf{f}}^{(i)}$ is the unknown stage solution and

$$\alpha = \frac{1}{\gamma\Delta t}, \quad \mathbf{r} = \frac{1}{\gamma\Delta t}\left[\bar{\mathbf{f}}_n + \Delta t \sum_{j=1}^{i-1}\left\{a_{ij}\hat{\mathscr{V}}\left(\bar{\mathbf{f}}^{(j)}\right) + \tilde{a}_{ij}\hat{\mathscr{C}}\left(\bar{\mathbf{f}}^{(j)}\right)\right\}\right]. \tag{51}$$

We use the Jacobian–Free Newton-Krylov (JFNK) approach [51] to solve Eq. (50). The inexact Newton's method [23] is used to solve the nonlinear equation, where the initial guess is the solution of the previous stage $\mathbf{y}_0 \equiv \bar{\mathbf{f}}_0^{(i)} = \bar{\mathbf{f}}^{(i-1)}$ (note that the first stage is always explicit), and the k-th Newton step is

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \mathscr{J}(\mathbf{y}_k)^{-1}\mathscr{F}(\mathbf{y}_k); \quad \mathscr{J}(\mathbf{y}) = \frac{d\mathscr{F}(\mathbf{y})}{d\mathbf{y}} = \alpha\mathbb{I} - \frac{d\hat{\mathscr{C}}(\mathbf{y})}{d\mathbf{y}}, \tag{52}$$

where $\mathbb{I}$ is the identity matrix and $\mathscr{J}$ is the Jacobian of $\mathscr{F}$. The exit criterion is

$$\|\mathscr{F}(\mathbf{y}_k)\|_2 \leq \max(\varepsilon_r\|\mathscr{F}(\mathbf{y}_0)\|_2, \varepsilon_a) \text{ or } \|\mathscr{J}^{-1}\mathscr{F}(\mathbf{y}_k)\|_2 \leq \varepsilon_s, \tag{53}$$

where ε_a and ε_r are the absolute and relative tolerances for the Newton solver, respectively, and ε_s is the step size tolerance. Equation (52) requires the solution to the linear system of equations,

$$[\mathscr{J}(\mathbf{y}_k)]\mathbf{x} = \mathscr{F}(\mathbf{y}_k), \tag{54}$$

which is solved using the preconditioned generalized minimum residual (GMRES) method [74,75]. The GMRES solver only requires the action of the matrix $\mathscr{J}$ on a vector, which is approximated by computing the directional derivative [68],

$$\mathscr{J}(\mathbf{y}_k)\mathbf{x} \approx \frac{1}{\varepsilon}[\mathscr{F}(\mathbf{y}_k + \varepsilon\mathbf{x}) - \mathscr{F}(\mathbf{y}_k)] = \alpha\mathbf{x} - \frac{1}{\varepsilon}[\hat{\mathscr{C}}(\mathbf{y}_k + \varepsilon\mathbf{x}) - \hat{\mathscr{C}}(\mathbf{y}_k)], \tag{55}$$

where $\varepsilon = \sqrt{\varepsilon_m(1 + \|\mathbf{y}_k\|_2)}/\|\mathbf{x}\|_2$ and ε_m is machine round-off error (taken as 10^{-14} in our algorithm). The exit criterion for the GMRES solver is

$$\|\mathbf{r}_l\|_2 \leq \max(\tilde{\varepsilon}_a, \tilde{\varepsilon}_r\|\mathbf{r}_0\|_2), \quad \mathbf{r}_l = \mathscr{J}(\mathbf{y}_k)\mathbf{x}_l + \mathscr{F}(\mathbf{y}_k), \tag{56}$$

where $\mathbf{x}_l$ is the solution at the l-th GMRES iteration, and $\mathbf{r}_0 = \mathscr{F}(\mathbf{y}_k)$ since $\mathbf{x}_0 = \mathbf{0}$.

The preconditioned GMRES algorithm uses a preconditioning matrix to accelerate convergence. Equation (54) is rewritten as

$$[\mathscr{J}(\mathbf{y}_k)][\mathscr{P}(\mathbf{y}_k)][\mathscr{P}(\mathbf{y}_k)]^{-1}\mathbf{x} = \mathscr{F}(\mathbf{y}_k), \tag{57}$$

where $\mathscr{P} \approx \mathscr{J}$ is the preconditioning matrix. Although the Jacobian $\mathscr{J}$ is not assembled, the preconditioning matrix $\mathscr{P}$ is assembled and stored as a banded matrix. The preconditioner is constructed as

$$\mathscr{P}(\mathbf{y}) = \alpha \mathscr{I} - \frac{d\bar{\mathscr{C}}(\mathbf{y})}{d\mathbf{y}}, \ \bar{\mathscr{C}}(\bar{\mathbf{f}}) = \mathscr{C}(\bar{\mathbf{f}}) + \mathscr{O}\left(\Delta v_{\parallel}^r, \Delta \mu^r\right), \ r < q. \qquad (58)$$

Thus, the preconditioning matrix is defined as the Jacobian of a lower–order discretization of the collision term. In our current implementation, $\bar{\mathscr{C}}$ is constructed by reconstructing $f_{k+\frac{1}{2},l}$ in (31) with the first-order upwind method and discretizing $\partial_{v_{\parallel}} f\big|_{k+\frac{1}{2},l}$ and $\partial_{\mu} f\big|_{k,l+\frac{1}{2}}$ in (33) and (35) with second-order central differences. In addition, the conversions from cell-centered to cell-averaged solutions and vice-versa using (23) are neglected, which results in a nine-banded sparse matrix. The Gauss–Seidel method is used to invert it. Though this preconditioning approach is satisfactory for the cases presented in this paper, we are exploring more efficient approaches presented in the literature [79].

5 Results

In this section, we test the accuracy, convergence, and computational cost of the IMEX time integration methods by applying them to two cases that are representative of the tokamak edge region. The edge is characterized by a steep decrease in the plasma density and temperature in the outward radial direction [69] that results in an increase in the collisionality from the near-core region to the edge. Our current implementation is insufficient to resolve significant variations in temperature due to the use of a fixed velocity grid in the entire configuration space domain; implementation of adaptive velocity grids based on the local thermal velocity will be pursued in future studies. We investigate our time integration methods for constant and varying collisionality by considering two Cartesian, two-dimensional heat transport problems where $\mathbf{R} \equiv (x,y)$. Both of these problems involve the thermal equilibration of an initially sinusoidal temperature profile while a specified electrostatic potential drives the flow. The first problem is essentially one-dimensional with zero gradients along the x dimension, and the collisionality does not vary significantly through the domain. The second problem introduces significantly varying collisionality along the x dimension (similar to the variation at the tokamak edge) by specifying an appropriate density profile. The performance of the IMEX methods is analyzed for both these problems and compared with that of the explicit RK methods.

The cases and the results in this section are described in terms of the nondimensional variables described in Section 2, unless noted otherwise. The reference quantities (dimensional) and their values are

$$\begin{aligned}
&L_{\text{ref}} = 1\,\text{m}, &&n_{\text{ref}} = 10^{20}\,\text{m}^{-3}, &&T_{\text{ref}} = 20\,\text{eV}, \\
&m_{\text{ref}} = 1.6726 \times 10^{-24}\,\text{g}, &&B_{\text{ref}} = 1\,\text{T}, &&v_{\text{ref}} = 4.377 \times 10^6\,\text{cm/s}, \\
&t_{\text{ref}} = 2.3 \times 10^{-5}\,\text{s}, &&\mu_{\text{ref}} = 1.602 \times 10^{-15}\,\text{g}\,\text{cm}^2\,\text{s}^{-2}\,\text{G}^{-1} &&\phi_{\text{ref}} = 20\,\text{V},
\end{aligned}$$

$$(59)$$

and the nondimensional values in the subsequent discussions should be multiplied by the corresponding reference values in (59) to calculate the dimensional values.

COGENT is implemented for distributed-memory parallel computations using the MPI library, and the simulations discussed in this paper are run on a Linux cluster where each node has two Intel Xeon 8-core processors with 2.6 GHz clock speed and 32 GB of memory.

5.1 Case 1: Uniform Collisionality

This case simulates the parallel heat transport over a two-dimensional (in configuration space) slab where the collisionality does not vary significantly throughout the domain. The configuration space domain is $0 \leq x, y \leq L$, where $L = 1$, and the velocity space domain is $-3.5 \leq v_{\parallel} \leq 3.5$, $0 \leq \mu \leq 10$. The initial solution is the Maxwellian distribution in the velocity space with uniform density $n(x, y, t = 0) = 1$, and the initial temperature profile is specified as

$$T(x, y, t = 0) \equiv T_0(y) = 1 + 0.1 \cos(2\pi y). \tag{60}$$

A steady electrostatic potential $\phi(x, y, t) = -0.1 \cos(2\pi y)$ and a constant magnetic field $B_y = 0.2, B_z = 2$ are applied throughout the simulation. Periodic boundary conditions are applied at all boundaries. Flow gradients along x are zero at all times in the simulation, and this case is essentially one-dimensional. The species charge and mass correspond to ionized hydrogen. The collision time is defined as

$$\tilde{\tau} = \frac{3}{4} \sqrt{\frac{\tilde{m}\tilde{T}^3}{\pi}} \frac{1}{\tilde{n}e^4 \lambda_c} = 2.4 \times 10^{-6}\,\text{s} \Rightarrow \tau = \frac{\tilde{\tau}}{t_{\text{ref}}} = 0.104, \tag{61}$$

where $(\tilde{\cdot})$ are dimensional quantities and τ is the nondimensional collision time. The Coulomb logarithm is specified as $\lambda_c = 11$ [40]. The collisional mean free path is $\lambda = v_{\text{th}}\tau$, where $v_{\text{th}} = \sqrt{2T/m}$ is the thermal velocity, and the characteristic length scale is $k_{\parallel}^{-1} = (B/B_y)(L/2\pi)$; consequently, the ratio of the particle mean free path to the characteristic length scale is

$$k_{\parallel}\lambda = 0.065, \tag{62}$$

and therefore, the plasma is highly collisional for this case.

The solutions discussed in this section are obtained on a grid with $\left(N_x, N_y, N_{v_{\parallel}}, N_{\mu}\right) = (6, 64, 36, 24)$ points. Figure 3 shows the evolution of the density and temperature in time. This solution is obtained with the ARK4 method with a time step of $\Delta t = 0.05$ that results in a Vlasov CFL of $\sigma_V \sim 1.1$ and a collision CFL of $\sigma_C \sim 12.9$, defined as follows:

$$\sigma_V = \max\left\{\frac{\left(\dot{\mathbf{R}} \cdot \hat{\mathbf{x}}\right)\Delta t}{\Delta x}, \frac{\left(\dot{\mathbf{R}} \cdot \hat{\mathbf{y}}\right)\Delta t}{\Delta y}, \frac{\dot{v}_{\parallel}\Delta t}{\Delta v_{\parallel}}\right\}, \tag{63a}$$

$$\sigma_C = \max\left\{\frac{\sigma_{v_{\parallel}}\Delta t}{\Delta v_{\parallel}}, \frac{\sigma_{\mu}\Delta t}{\Delta \mu}, \frac{\kappa_{v_{\parallel}v_{\parallel}}\Delta t}{\Delta v_{\parallel}^2}, \frac{\kappa_{\mu\mu}\Delta t}{\Delta \mu^2}, \frac{\kappa_{v_{\parallel}\mu}\Delta t}{\Delta v_{\parallel}\Delta \mu}\right\}. \tag{63b}$$
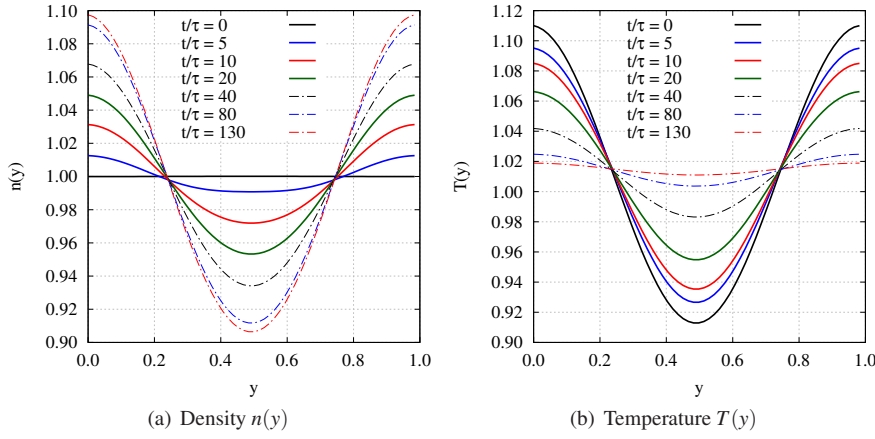
(a) Density $n(y)$ (b) Temperature $T(y)$

Fig. 3 Evolution of density and temperature in time. The constant initial density assumes a cosine-like profile to balance the specified electrostatic potential. Thermal equilibration drives the temperature to a constant value.
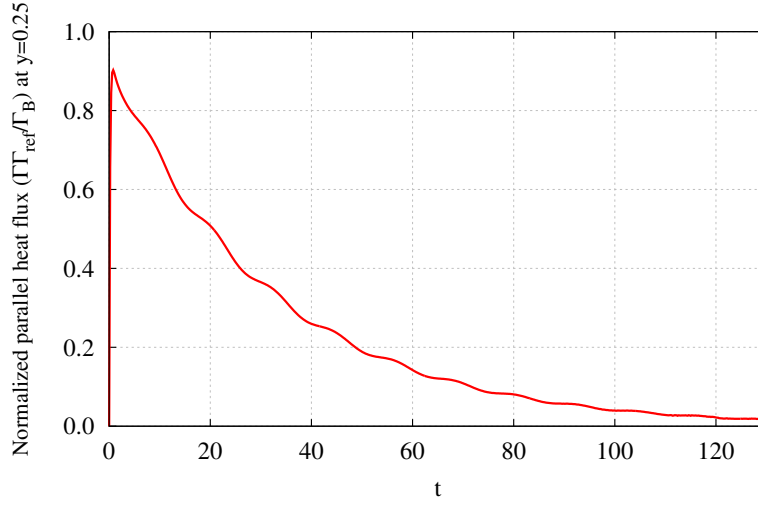

Fig. 4 Parallel heat flux as a function of time at $y = 0.25$: The heat flux attains its "quasi-steady" value consistent with the temperature gradient at an instant in time. As the temperature equilibriates, it decays to zero.

The simulation is run until a final time of $t_f = 130$ on 192 cores (12 nodes) with 8 MPI ranks along y, 6 ranks along $v_\parallel$, and 4 ranks along μ. The specified electrostatic potential forces the initially constant density to assume a sinusoidal profile while thermal equilibration drives the temperature to a constant value. These processes occur on the slow transport time scale ($\tau_T \approx mL^2/2\tau T$). Figure 4 shows the normalized parallel heat flux ($\Gamma\Gamma_{\text{ref}}/\Gamma_B$) at $y = 0.25$ as a function of time, where $\Gamma_B = 1.87 \times 10^8 \, \text{g s}^{-3}$

is the heat flux computed by the Braginskii model [5],

$$\Gamma\left(\mathbf{R}\right) = \int\int\left\{v_{\parallel} - \overline{v_{\parallel}}\left(\mathbf{R}\right)\right\}\frac{\left\{v_{\parallel} - \overline{v_{\parallel}}\left(\mathbf{R}\right)\right\}^2 + \mu B}{2}f\left(\mathbf{R}, v_{\parallel}, \mu\right)B_{\parallel}^*\left(\mathbf{R}, v_{\parallel}\right)dv_{\parallel}d\mu \tag{64}$$

is the nondimensional parallel heat flux, and

$$\Gamma_{\text{ref}} = n_{\text{ref}}T_{\text{ref}}^{\frac{3}{2}}m_{\text{ref}}^{-\frac{1}{2}} = 1.4 \times 10^{10}\,\text{g}\,\text{s}^{-3} \tag{65}$$

is the reference heat flux. At a given time, the collisions drive the heat flux to a value consistent with the local temperature gradient at the collisional time scale τ. As the temperature assumes a constant value, the heat flux decays to zero, which happens on the transport time scale. Thus, this problem demonstrates the disparate time scales resulting from a highly-collisional plasma.

Figure 5(a) shows the L_2 norm of the time integration error as a function of the collisional and Vlasov CFL numbers defined by (63) for solutions obtained at a final time of $t_f = 0.1$. The errors for the ARK methods are compared with those for explicit second-order (RK2a), third-order (RK3), and fourth-order (RK4) RK methods. The error is defined as

$$\mathbf{f}_{\varepsilon} = \bar{\mathbf{f}} - \bar{\mathbf{f}}_{\text{RK4},\Delta t=10^{-5}}, \tag{66}$$

where $\bar{\mathbf{f}}$ is the numerical solution and $\bar{\mathbf{f}}_{\text{RK4},\Delta t=10^{-5}}$ is the reference solution obtained with the RK4 method with a time step of $\Delta t = 10^{-5}$ (corresponding to $\sigma_V \sim 2.2 \times 10^{-4}$, $\sigma_C \sim 2.4 \times 10^{-3}$). The relative tolerance for the Newton solver is $\varepsilon_r = 10^{-10}$ for all the solutions, and the absolute tolerance ε_a is specified as the maximum value $\varepsilon_a^{\text{max}}$ for which the error in solving the nonlinear system is less than the truncation error of the time integration method for a given Δt. Thus, it is a function of the method and the time step size and is defined as

$$\varepsilon_a^{\text{max}} = \max\left(\varepsilon_\alpha\right) \quad \text{such that} \quad \|\mathbf{f}_{\varepsilon,\varepsilon_a}\|_2 = \|\mathbf{f}_{\varepsilon,10^{-10}}\|_2, \tag{67}$$

where $\mathbf{f}_{\varepsilon,\varepsilon_a}$ is the error defined by (66) for a solution obtained with an absolute tolerance of ε_a, and $\mathbf{f}_{\varepsilon,10^{-10}}$ is the error for a solution obtained with an absolute tolerance of 10^{-10}. As an example, Figure 5(b) shows the L_2 norm of the error as a function of the absolute tolerance ε_a for the ARK4 method at three values of Δt. A tolerance $\varepsilon_a^{\text{max}} \sim 10^{-2}$ is sufficient for the largest time step $\Delta t = 0.04$, the intermediate time step $\Delta t = 0.004$ requires a tolerance of $\varepsilon_a^{\text{max}} \sim 10^{-5}$, and the smallest time step $\Delta t = 0.0004$ requires $\varepsilon_a^{\text{max}} \sim 10^{-9}$. This approach avoids solving the nonlinear system of equations to a higher accuracy than required by the time integration method for a given Δt. The relative tolerance for the GMRES solver is specified as $\tilde{\varepsilon}_r = 10^{-4}$, and the absolute tolerance is specified as $\tilde{\varepsilon}_a = \varepsilon_a^{\text{max}}/10$ for all the simulations. Figure 5(a) shows that all the methods implemented converge at their theoretical orders. The errors for a given method are shown at all CFL numbers for which the method is stable. The explicit RK methods are unstable beyond a collisional CFL of $\sigma_C \sim 0.5$ that corresponds to a Vlasov CFL of $\sigma_V \sim 0.04$ since they are constrained by the
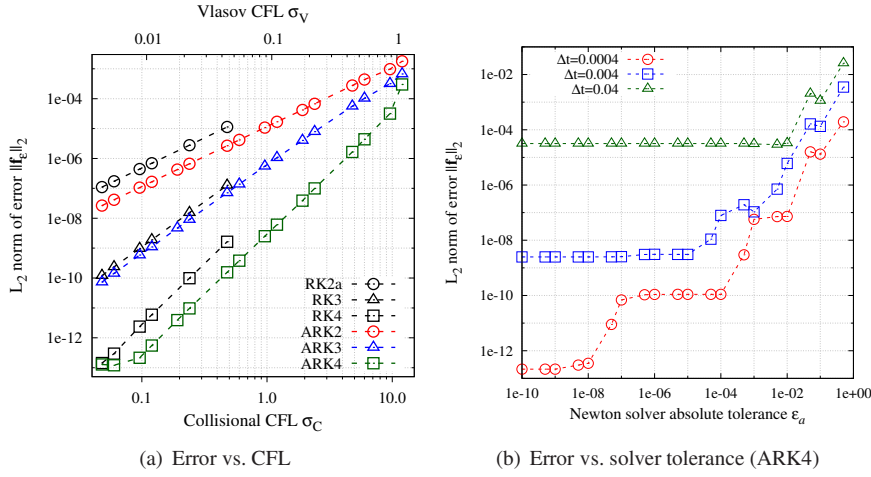
(a) Error vs. CFL

(b) Error vs. solver tolerance (ARK4)

Fig. 5 Accuracy and convergence: (a) The L_2 norm of the error as a function of the collisional and Vlasov CFL numbers. The time integration methods converge at their theoretical orders of convergence. (b) The error as a function of the Newton solver absolute tolerance for ARK4 at three values of Δt. A higher Δt allows a more relaxed tolerance without increasing the truncation error.

(a) Error vs. number of function calls
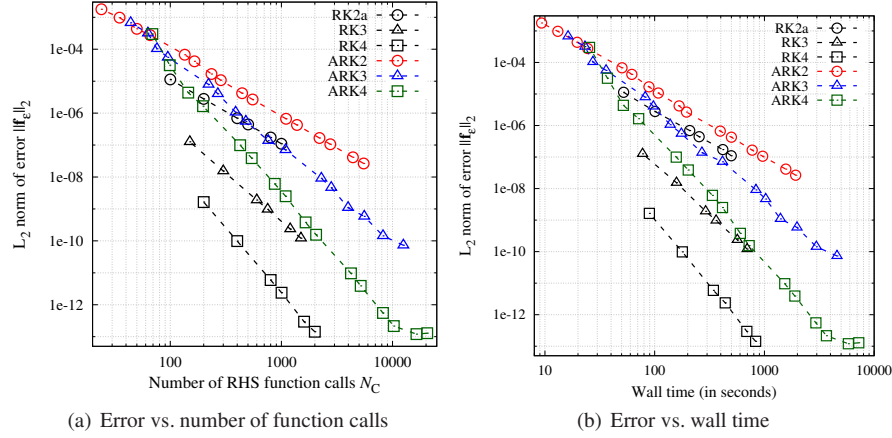
(b) Error vs. wall time

Fig. 6 Computational cost: (a) The error as a function of the number of calls to the collision operator $\mathscr{C}$ and (b) the error as a function of the wall time in seconds. The IMEX approach allows significantly faster stable solutions compared with explicit time integration.

collisional time scale. The ARK methods allow stable time steps comparable to the Vlasov time scales; the maximum Vlasov CFL is $\sigma_V \sim 1.1$, corresponding to a collisional CFL of $\sigma_C \sim 12$.

The computational cost of the ARK methods are compared with that of the explicit RK methods for the solutions shown in Figure 5(a). Figure 6(a) shows the L_2 norm of the truncation error $\mathbf{f}_\varepsilon$ as a function of the number of calls to the collision

Table 1 Computational cost with and without preconditioning for solutions obtained with the ARK4 method at a final time of $t_f = 130$. The preconditioner is effective at reducing the cost of the linear solver at higher CFL numbers.

Δt	σ_V	σ_C	N_{GMRES}		N_C		Wall time (seconds)	
			No PC	With PC	No PC	With PC	No PC	With PC
ARK4								
0.01	0.2	2.4	$180,568$	$188,974$	$388,408$	$397,023$	1.4×10^5	1.5×10^5
0.025	0.6	6.1	$73,734$	$59,096$	$156,935$	$142,297$	5.7×10^4	5.3×10^4
0.04	0.9	9.7	$54,321$	$25,852$	$103,801$	$75,249$	3.7×10^4	2.7×10^4
0.05	1.1	12.1	$49,905$	$21,755$	$89,544$	$61,298$	3.3×10^4	2.3×10^4
RK4								
0.002	0.04	0.50		—		$260,000$		1.1×10^5

operator $\widehat{\mathscr{C}}\left(\bar{\mathbf{f}}\right)$. It is defined as

$$N_C = N_T \times n_s \tag{68}$$

for an explicit RK time integrator, where $N_T = t_f / \Delta t$ is the total number of time steps, and n_s is the number of stages in the RK methods. Since we use the JFNK approach to solve the nonlinear system of equations in the ARK time integrators, it is defined as

$$N_C = N_T \times n_s + N_{\text{Newton}} + N_{\text{GMRES}} \tag{69}$$

for the ARK methods, where N_{Newton} is the total number of Newton iterations and N_{GMRES} is the total number of preconditioned GMRES iterations. The time step Δt (and consequently, the collisional and Vlasov CFL numbers) increases as one moves along a curve from bottom-right to the top-left. In the region where the explicit methods are stable, they are more efficient, and this is expected because of the additional burden on the ARK methods to solve an nonlinear system of equations. However, the ARK methods allow stable solutions at a significantly lower cost at higher CFL numbers where the explicit methods are unstable. The metric shown in Figure 6(a) does not reflect the cost of assembling and solving the preconditioner. The Gauss-Seidel method is used to invert the preconditioner, and 80 iterations are required to solve it to sufficient accuracy. Figure 6(b) shows truncation error $\|\bar{\mathbf{f}}_\varepsilon\|_2$ as a function of the the wall time (in seconds), which includes the additional cost of preconditioning the system. The simulations are run on 96 cores (6 nodes) with 8 MPI ranks along y, 6 ranks along $v_\parallel$, and 2 ranks along μ. A similar trend is observed, and the ARK methods allow for significantly faster stable time step. For example, the fastest stable solution with ARK2 is ~ 6 times faster than that with RK2a, while the fastest stable solution with ARK4 is ~ 4 times faster than that with RK4.

The performance of the preconditioner is assessed for the ARK4 method. Table 1 shows the number of GMRES iterations (N_{GMRES}), number of function calls to $\widehat{\mathscr{C}}$ (N_C), and the wall time for solutions obtained with various time steps Δt at final time of $t_f = 130$. The simulations are run with the preconditioner ("With PC") and without it ("No PC") on 192 cores (12 nodes) with 8 MPI ranks along y, 6 ranks along $v_\parallel$, and 4 ranks along μ. As a reference, the number of function calls and the wall
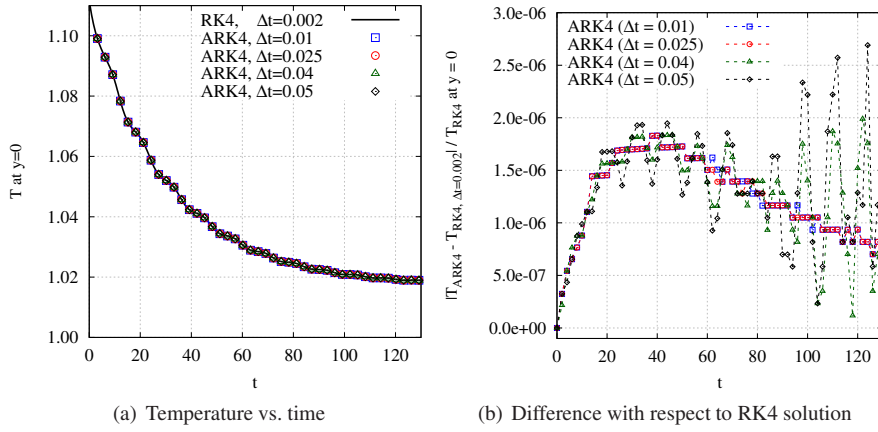
Fig. 7 (a) Temperature T at $y = 0$ as a function of time for the solutions reported in Table 1. The solutions obtained with the semi-implicit ARK4 method agree well with that obtained with the explicit RK4 method. (b) The normalized difference between the ARK4 and RK4 solutions.

time are reported for the explicit RK4 with the largest stable time step $\Delta t = 0.002$; this represents the fastest stable solution with the RK4 method. The preconditioner is ineffective at the lower CFL numbers considered; at $\sigma_C = 2.4$, the number of preconditioned GMRES iterations is slightly larger than the number of unpreconditioned GMRES iterations, while at $\sigma_C = 6.1$, the number of preconditioned iterations is slightly less than the number of unpreconditioned iterations. At the higher CFL numbers $\sigma_C = 9.7, 12.1$, the preconditioner reduces the number of GMRES iterations by factors of ~ 2.1 and ~ 2.3, respectively. The number of function calls N_C includes the calls from the Newton solver and the time integrator, and therefore, the effect of the preconditioner is less pronounced when considering this metric. At the highest stable time step for the ARK4 method ($\Delta t = 0.05$), the number of function calls for the preconditioned simulation is ~ 1.5 times smaller than that for the unpreconditioned simulation. In terms of the wall time, the preconditioned simulation is ~ 1.4 times faster than the unpreconditioned simulation, and this shows that the construction and inversion of the preconditioner is relatively cheap even though 80 Gauss-Seidel iterations are required. Overall, this test case does not result in a very ill-conditioned system; at the highest collisional CFL number considered, the unpreconditioned simulation requires only ~ 4 GMRES iterations on average ($N_{\mathrm{GMRES}}/(N_T \times n_s^{\mathrm{imp}})$), where $n_s^{\mathrm{imp}} = 5$ is the number of implicit stages for ARK4 and the nonlinearity is sufficiently weak that only one Newton iteration is observed for each implicit stage). Figure 7(a) shows the evolution of the temperature T at $y = 0$ for the simulations in Table 1. The solutions obtained with the semi-implicit approach at high collisional CFL numbers agree well with the solution obtained with the explicit RK4 method with a time step constrained by the collisional time scale. Figure 7(b) shows the normalized difference between the ARK4 solutions (T_{ARK4}) and the fastest RK4 solution ($T_{\mathrm{RK4},\Delta t=0.002}$). The errors are larger for larger time steps, and they grow with time as expected; the

(a) Initial density $n_0(x)$ (b) Ratio of mean free path to length scale $k_\parallel \lambda$
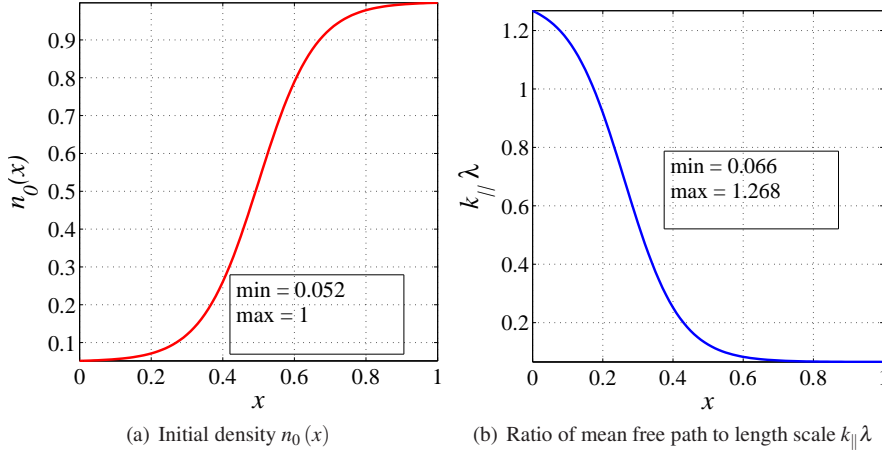
Fig. 8 The initial density variation along x for the varying collisionality case and the consequent collisionality variation. The plasma is weakly collisional for $x \to 0$ while it is highly collisional for $x \to 1$.

global (in time) error at a given time is an accumulation of the local truncation error for all preceding time steps.

5.2 Case 2: Varying Collisionality

This case is similar to the previous case, except that a significant variation in the density, and therefore, the collisionality, is introduced along x. The initial density is specified as

$$n(x,y,t=0) \equiv n_0(x) = 1 - \delta + \delta \tanh(2\pi x - \pi), \tag{70}$$

where $\delta = 0.475$. The problem setup is identical to the previous case in all other aspects. Figure 8 shows the initial density and the ratio of the mean free path to the characteristic length scale $k_\parallel \lambda$. The mean free path is comparable to the characteristic length scale near $x = 0$, and the plasma is weakly collisional. On the other hand, the mean free path is much smaller than the length scale near $x = 1$, and the plasma is strongly collisional. The collisional time scales are thus comparable to the Vlasov time scales as $x \to 0$ but are much smaller as $x \to 1$. This variation in the collisionality is representative of the radial variation in the tokamak edge region [69]. The plasma conditions as $x \to 1$ correspond to the previous test case. This case is solved on a grid with $\left(N_x, N_y, N_{v_\parallel}, N_\mu \right) = (32, 32, 36, 24)$ points in the subsequent discussions.

Figure 9 shows the cross-sectional temperature along y at $x = 0.1$ (weakly collisional) and $x = 0.9$ (strongly collisional). The solutions are obtained with the ARK4 method at a time step of $\Delta t = 0.1$, corresponding to a Vlasov CFL number of $\sigma_V \approx 1.1$ and a collisional CFL number of $\sigma_C \approx 26$. The simulation is run until a final time of $t_f = 130$ on 384 cores (24 nodes) with 4 MPI ranks along x, y, and μ and 6 MPI ranks

along $v_\parallel$. The dynamics are similar to the previous test case; however, the varying collisionality results in varying thermal equilibration and density diffusion rates. Near $x = 0$, the collisional and Vlasov time scales are comparable while near $x = 1$, the collisional time scale is much smaller than the Vlasov time scale. In the region where the plasma is weakly collisional ($x \to 0$), the temperature equilibrates much faster while the equilibration is much slower in the region where the plasma is strongly collisional ($x \to 1$). Figure 10 shows the evolution of the normalized parallel heat flux ($\Gamma\Gamma_{\mathrm{ref}}/\Gamma_B$) at $y = 0.25$ and $x = 0.1, 0.9$. At $x = 0.1$, the heat flux decays to zero much faster than at $x = 0.9$ due to the faster equilibration.

Figure 11(a) shows the L_2 norm of the time integration error as a function of the Vlasov and collisional CFL numbers defined by (63) for solutions at a final time of $t_f = 0.2$. The errors are computed with respect to a reference solution obtained with the RK4 method and a time step of $\Delta t = 2 \times 10^{-5}$, corresponding to a Vlasov CFL
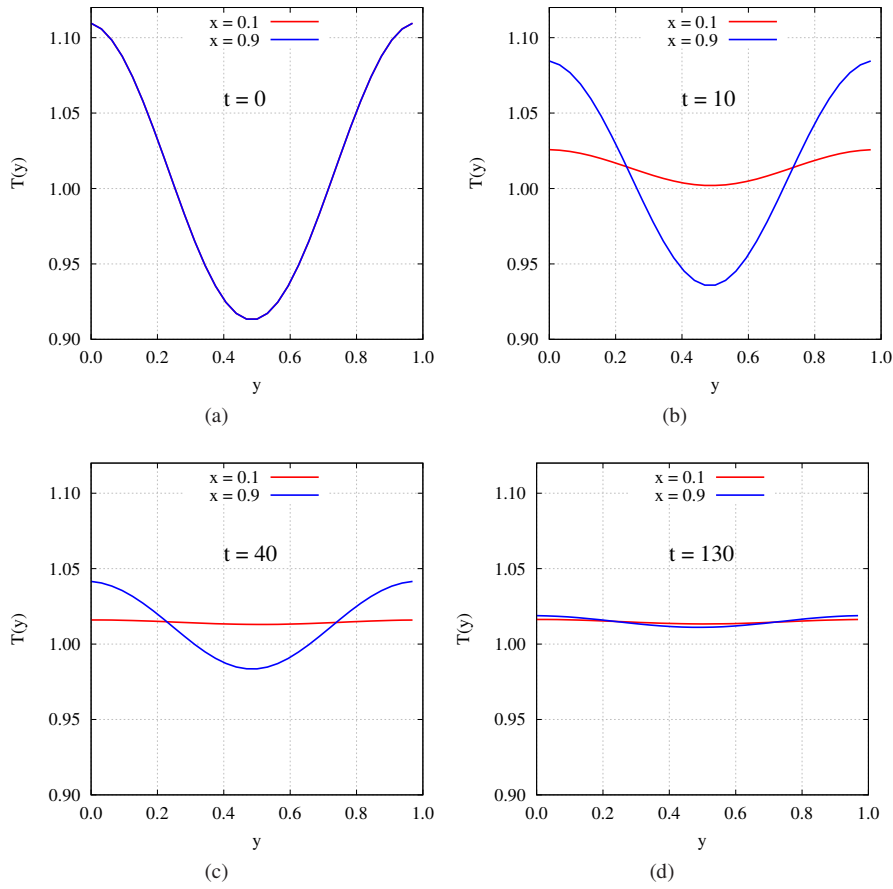

Fig. 9 Cross-sectional temperature $T(y)$ at $x = 0.1, 0.9$ at four different solution times. The temperature equilibrates faster in the low-collisionality region ($x = 0.1$) than in the high-collisionality region ($x = 0.9$).
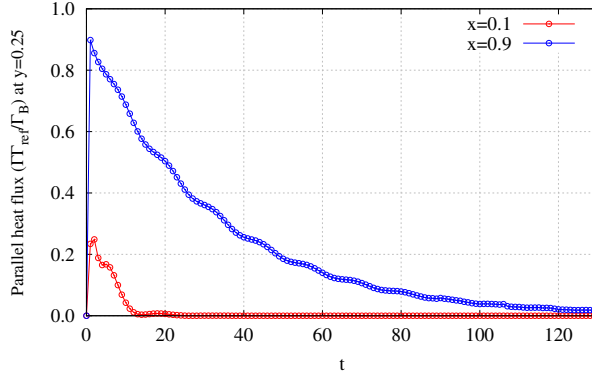
Fig. 10 The time evolution of the parallel heat flux at $y = 0.25$ and $x = 0.1, 0.9$. The dynamics are qualitatively similar to those in Fig. 4. In the low collisionality region ($x = 0.1$), thermal equilibration is faster and the heat flux decays to zero faster than in the high collisionality region ($x = 0.9$).

number of $\sigma_V \approx 2.2 \times 10^{-4}$ and a collisional CFL number of $\sigma_C \approx 4.7 \times 10^{-3}$. Thus, the fastest collisional time scale is ~ 21 times faster than the Vlasov time scale for this test case. The relative and absolute tolerances for the Newton and GMRES solvers are specified in the same way as described for the previous test case. All the methods considered converge at their theoretical orders of convergence. The maximum stable time step for the explicit RK methods correspond to the collisional CFL number of $\sigma_C \approx 0.5$ because they are constrained by the collisional time scales. The semi-implicit methods are stable until a collisional CFL of $\sigma_C \approx 25$ that corresponds to a Vlasov CFL of $\sigma_V \approx 1.1$; thus, they are constrained by the Vlasov time scale.

Figure 11(b) shows the error as a function of the number of calls (N_C) to the collision operator $\mathscr{C}(\bar{\mathbf{f}})$ defined through (68) and (69) for the explicit RK and semi-implicit ARK methods, respectively. The ARK methods allow significantly less expensive stable solutions; for example, the fastest stable solution with ARK2 is ~ 7 times less expensive than that with RK2a, and the fastest stable solution with ARK4 is ~ 6 times less expensive than that with RK4. Figure 11(c) shows the error as a function of the wall time. The simulations are run on 192 cores with 4 MPI ranks along x and y, 6 ranks along $v_\parallel$, and 2 ranks along μ. The wall times include the cost of assembling and inverting the preconditioning matrix (80 Gauss-Seidel iterations), and similar trends are observed. Thus, the additional cost of the preconditioner is insignificant despite 80 Gauss-Seidel iterations being required to solve it to sufficient accuracy.

Table 2 compares the computational cost of the fastest stable solution with the ARK4 method, both with and without the preconditioner, as well as the fastest stable RK4 solution. The preconditioner reduces the total number of GMRES iterations by a factor of ~ 3. The unpreconditioned simulation requires an average of ~ 5 GMRES iterations in each implicit stage, while the preconditioned simulation requires an average of ~ 1.8 GMRES iterations. The nonlinearity was sufficiently weak that the Newton solver was observed to converge in one iteration for these simulations.
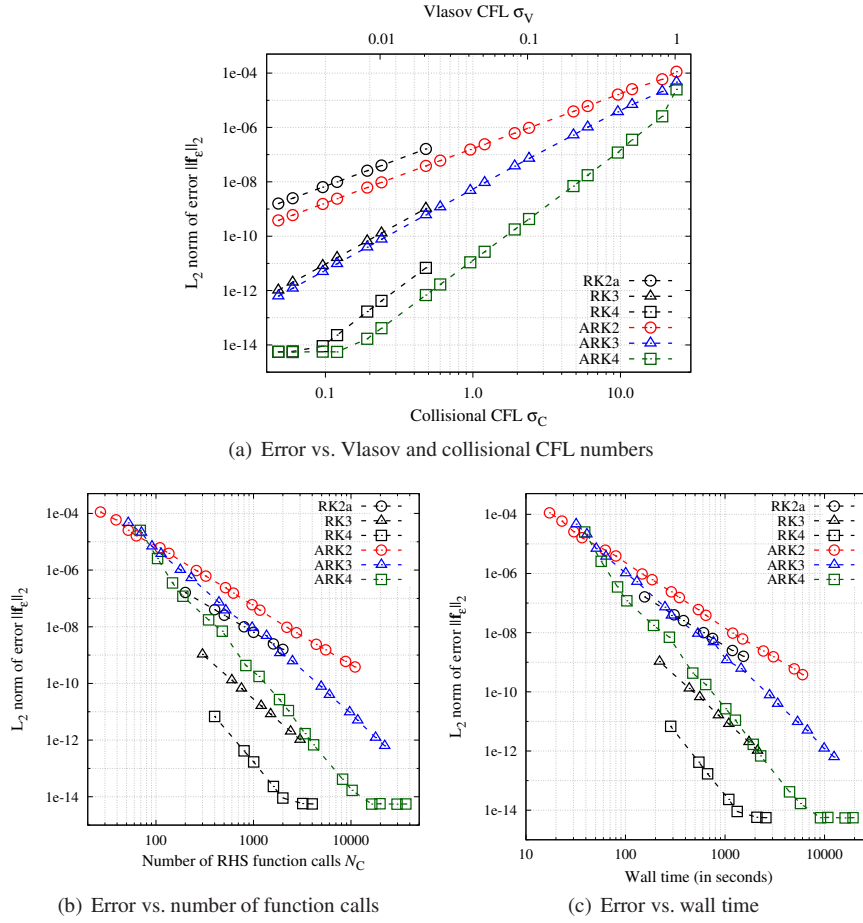
(a) Error vs. Vlasov and collisional CFL numbers

(b) Error vs. number of function calls

(c) Error vs. wall time

Fig. 11 Convergence and work-precision: (a) The error as a function of the Vlasov and collisional CFL numbers for explicit RK and semi-implicit ARK methods. All the methods converge at their theoretical orders. (b) The error as a function of the number of calls to the collision operator $\hat{\mathscr{C}}$, and (c) the error as a function of the wall time in seconds. The ARK methods allow significantly faster stable solutions.

The total number of calls to the collision operator (N_C), which includes calls from the time integrator and the Newton solver, is reduced by a factor of ~ 1.7 by the preconditioner, while the total wall time is reduced by ~ 1.6. The difference in these two factors reflects the cost of inverting the preconditioning matrix with 80 Gauss-Seidel iterations. The preconditioned ARK4 solution is ~ 11 times faster than the fastest stable RK4 solution in terms of the wall time. These solutions are obtained on 384 cores (24 nodes) with 4 MPI ranks along x, y, and μ, and 6 ranks along $v_{\parallel}$. Figure 12 shows the cross-sectional density $n(y)$ at $x = 0.1$ and $x = 0.9$ for the two simulations in Table 2 at the final time of $t_f = 130$. The initially constant density develops a non-constant profile to balance the specified electrostatic potential. The solution obtained

Table 2 A comparison of the computational cost with and without preconditioning of the fastest stable solution with the ARK4 method at a final time of $t_f = 130$, as well as the fastest stable solution with RK4. The preconditioner is effective at reducing the cost of the linear solve.

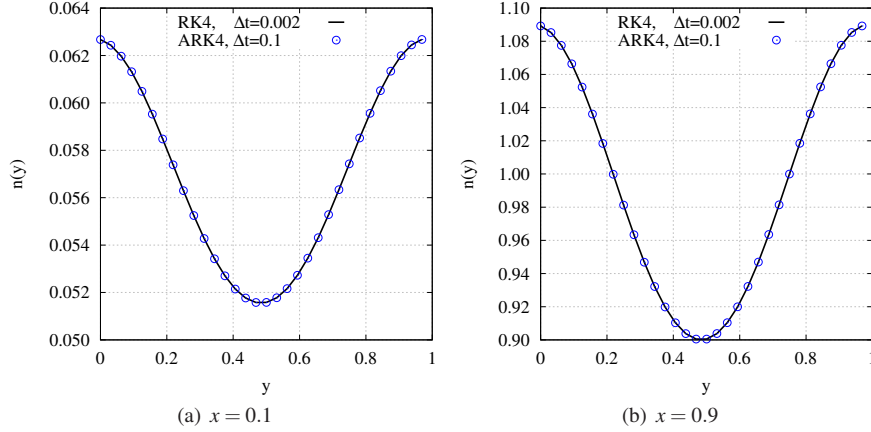| Δt	σ_V	σ_C	N_{GMRES}		N_C		Wall time (seconds)	
			No PC	With PC	No PC	With PC	No PC	With PC
ARK4								
0.1	1.1	25.8	33,131	11,535	52,551	30,852	2.5×10^4	1.6×10^4
RK4								
0.002	0.02	0.52		—		260,000		1.7×10^5

(a) $x = 0.1$

(b) $x = 0.9$

Fig. 12 Cross-sectional density $n(y)$ at two x locations corresponding to the weakly and strongly collisional regions for the simulations in Table 2 at the final simulation time of $t_f = 130$. The solutions agree well with each other.

with the semi-implicit ARK4 method agrees well with that obtained with the explicit RK4 method.

6 Conclusion

This paper describes a semi-implicit time integration algorithm for the Vlasov-Fokker-Planck equations with the motivating application area being tokamak edge plasma simulations. The dynamics in the edge region of a tokamak fusion reactor are characterized by disparate length and time scales. The hot, weakly-collisional plasma near the core requires a kinetic model; however, near the cold edge, the strong collisionality renders the kinetic model numerically stiff. Explicit time integration methods are thus too expensive for such applications. In this paper, we consider as our governing equations the drift-kinetic Vlasov equation, which is the long-wavelength limit of the gyrokinetic VFP equation, with the Fokker-Planck collision model in the Rosenbluth form.

We implement high-order, semi-implicit Additive Runge-Kutta methods where the Vlasov term is integrated explicitly in time while the Fokker-Planck term is integrated implicitly. The semi-implicit algorithms are applied to two test problems, where the ratio of the collisional time scales to the Vlasov time scales are representative of the tokamak edge region. We show that the semi-implicit time integrators allow time steps that respect the Vlasov time scales and are larger than the maximum time steps allowed by explicit time integrators, which respect the collisional time scales. Accuracy and convergence tests of the methods show that the semi-implicit methods converge at their theoretical orders. We also assess the computational cost of these methods and compare the relative cost of the semi-implicit solutions with those obtained with explicit Runge-Kutta methods. Overall, the IMEX approach results in significantly faster stable solutions. Explicit methods are more efficient when seeking solutions with high accuracy. However, the IMEX approach allows faster solutions that are less accurate (due to a higher truncation error from a larger time step size) but well-resolved; explicit methods cannot yield solutions at a comparable cost even allowing for lower accuracy since their time step sizes are constrained.

Two deficiencies of this paper are areas of active research and will be reported in future publications. A better preconditioning approach is needed, both in terms of constructing the preconditioning matrix as an approximation to the true Jacobian and implementing an efficient method to solve it. The use of fixed velocity grids restricts us to applications without severe temperature gradients, and thus, our current implementation cannot solve for true tokamak conditions where the temperature drops significantly from the hot core to the cold edge. This limitation will be addressed by using velocity renormalization.

A Computation of Rosenbluth Potentials

The Rosenbluth potentials are related to the distribution function by Poisson equations, given by (7), that are defined on an infinite velocity domain. The algorithm implemented in COGENT to solve them on a finite numerical domain was introduced in [25] and is summarized briefly in this section. The numerical velocity domain is defined as $\Omega_{\mathbf{v}} \equiv \left[-v_{\parallel,\max}, v_{\parallel,\max}\right] \times [0, \mu_{\max}]$, while the infinite velocity domain is $\Omega_{\mathbf{v},\infty} \equiv [-\infty, \infty] \times [0, \infty]$. The Green's function method is used to compute the boundary values as

$$\varphi\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = \frac{1}{4\pi} \int_{\Omega_{\mathbf{v},\infty}} \frac{f\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|} d\mathbf{v}', \tag{71a}$$

$$\rho\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = \frac{1}{8\pi} \int_{\Omega_{\mathbf{v},\infty}} \left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right| f\left(\mathbf{v}'\right) d\mathbf{v}', \tag{71b}$$

where $\mathbf{v}_{\partial\Omega_{\mathbf{v}}}$ is the velocity vector at the computational domain boundary $\partial\Omega_{\mathbf{v}}$. Direct evaluation of (71) is expensive and an asymptotic method is used. The Green's function is expanded as [43]

$$\frac{1}{\left|\mathbf{v} - \mathbf{v}'\right|} = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{v_<^l}{v_>^{l+1}} Y_{lm}^*\left(\theta', \psi'\right) Y_{lm}\left(\theta, \psi\right), \tag{72}$$

where Y_{lm} is the spherical harmonic function, $\theta = \arccos\left(v_{\parallel}/v\right)$ is the pitch angle, ψ is the gyro-angle, $v_> = \max\left(|\mathbf{v}|, |\mathbf{v}'|\right)$, and $v_< = \min\left(|\mathbf{v}|, |\mathbf{v}'|\right)$. Therefore,

$$\varphi\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = -\frac{1}{4\pi} \sum_{l=0}^{\infty} \frac{h_l}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}} P_l\left(\cos\theta\right); \quad h_l = \frac{2\pi B}{m} \int_{\Omega_{\mathbf{v}}} f\left(v_{\parallel}, \mu\right) v^l P_l\left(\cos\theta\right) dv_{\parallel} d\mu, \tag{73}$$

where P_l denotes the Legendre polynomials, $v = |\mathbf{v}|$ is the particle speed, and

$$f\left(v_\parallel, \mu : v > \min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|\right) = 0. \tag{74}$$

The second Rosenbluth potential ρ is expressed on the domain boundary in terms of the first Rosenbluth potential φ as

$$\rho\left(\mathbf{v}_{\partial\Omega_\mathbf{v}}\right) = \frac{1}{4\pi}\int_{\Omega_\infty}\frac{\varphi(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\mathbf{v}';. \tag{75}$$

Though this is similar to (71a), the above analysis does not apply because $\varphi\left(v_\parallel, \mu : v > \min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|\right) \neq 0$. Equation (75) is decomposed as

$$\frac{1}{4\pi}\int_{\Omega_\infty}\frac{\varphi(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\mathbf{v}' = \int_0^{\min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|}v^2 dv\int_0^\pi d\theta\int_0^{2\pi}\frac{\hat{\varphi}(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\psi + \int_0^{\min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|}v^2 dv\int_0^\pi d\theta\int_0^{2\pi}\frac{\tilde{\varphi}(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\psi, \tag{76}$$

where $\hat{\varphi}(\mathbf{v})$ is the numerical solution to Eq. (7a) and

$$\tilde{\varphi}(\mathbf{v}) = -\frac{1}{4\pi}\sum_{l=0}^\infty\frac{h_l}{|\mathbf{v}|^{l+1}}P_l(\cos\theta). \tag{77}$$

Therefore,

$$\int_0^{\min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|}v^2 dv\int_0^\pi d\theta\int_0^{2\pi}\frac{\hat{\varphi}(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\psi = \sum_{l=0}^\infty\frac{g_l}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l+1}}P_l(\cos\theta); \quad g_l = \frac{2\pi B}{m}\int_{\Omega_\mathbf{v}}\hat{\varphi}\left(v_\parallel, \mu\right)v^l P_l(\cos\theta)dv_\parallel d\mu, \tag{78}$$

where $\hat{\varphi}(\mathbf{v}) = \varphi(\mathbf{v}), v \leq \min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|, \hat{\varphi}(\mathbf{v}) = 0, v > \min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|$, and

$$\int_0^{\min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|}v^2 dv\int_0^\pi d\theta\int_0^{2\pi}\frac{\tilde{\varphi}(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}} - \mathbf{v}'\right|}d\psi = -\sum_{l=0}^\infty\frac{h_l P_l(\cos\theta)}{2l+1}\left(\frac{1}{2\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l-1}} - \frac{1}{2\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l+1}} + \frac{1}{(2l-1)\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l-1}}\right). \tag{79}$$

The second Rosenbluth potential is thus expressed at the computational domain boundary as

$$\rho\left(\mathbf{v}_{\partial\Omega_\mathbf{v}}\right) = -\frac{1}{4\pi}\sum_{l=0}^\infty\left[\frac{g_l}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l+1}} - \frac{h_l}{2(2l+1)}\left(\frac{2l+1}{2l-1}\frac{1}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l-1}} - \frac{\left\{\min\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|\right\}^2}{\left|\mathbf{v}_{\partial\Omega_\mathbf{v}}\right|^{l+1}}\right)\right]P_l(\cos\theta). \tag{80}$$

Equations (73) and (80) serve as the boundary conditions for (7) on the finite numerical domain Ω_v. Cut-cell issues, arising in evaluating (78), are solved by linear interpolations [25], and therefore, the current implementation uses second-order central finite differences to discretize the derivatives in Eq. (7). The fourth-order implementation will be investigated in the future. We use the conjugate gradient method with a structured multigrid preconditioner from the *hypre* library [30] to solve the discretized system of equations.

References

1. Ascher, U.M., Ruuth, S.J., Spiteri, R.J.: Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Applied Numerical Mathematics **25**(2-3), 151–167 (1997). DOI 10.1016/S0168-9274(97)00056-1

2. Banks, J.W., Brunner, S., Berger, R.L., Tran, T.M.: Vlasov simulations of electron-ion collision effects on damping of electron plasma waves. Physics of Plasmas **23**(3), 032108 (2016). DOI 10.1063/1.4943194

3. Belli, E.A., Candy, J.: Full linearized Fokker–Planck collisions in neoclassical transport simulations. Plasma Physics and Controlled Fusion **54**(1), 015,015 (2012). DOI 10.1088/0741-3335/54/1/015015

4. Berezin, Y., Khudick, V., Pekker, M.: Conservative finite-difference schemes for the Fokker-Planck equation not violating the law of an increasing entropy. Journal of Computational Physics **69**(1), 163–174 (1987). DOI 10.1016/0021-9991(87)90160-4

5. Braginskii, S.I.: Transport processes in a plasma. Reviews of Plasma Physics **1**, 205 (1965)

6. Brunner, S., Tran, T., Hittinger, J.: Numerical implementation of the non-linear Landau collision operator for Eulerian Vlasov simulations. part I: Computation of the Rosenbluth potentials. Tech. Rep. LLNL-SR-459135, Lawrence Livermore National Laboratory, Livermore, CA (2010)

7. Buet, C., Cordier, S.: Conservative and entropy decaying numerical scheme for the isotropic Fokker–Planck–Landau equation. Journal of Computational Physics **145**(1), 228–245 (1998). DOI 10.1006/jcph.1998.6015

8. Buet, C., Cordier, S., Degond, P., Lemou, M.: Fast algorithms for numerical, conservative, and entropy approximations of the Fokker–Planck–Landau equation. Journal of Computational Physics **133**(2), 310–322 (1997). DOI 10.1006/jcph.1997.5669

9. Butcher, J.: Numerical Methods for Ordinary Differential Equations. Wiley (2003)

10. Candy, J., Waltz, R.E.: Anomalous transport scaling in the DIII–D tokamak matched by supercomputer simulation. Phys. Rev. Lett. **91**, 045,001 (2003). DOI 10.1103/PhysRevLett.91.045001

11. Casanova, M., Larroche, O., Matte, J.P.: Kinetic simulation of a collisional shock wave in a plasma. Phys. Rev. Lett. **67**, 2143–2146 (1991). DOI 10.1103/PhysRevLett.67.2143

12. Chacón, L., Barnes, D.C., Knoll, D.A., Miley, G.H.: An implicit energy-conservative 2D Fokker–Planck algorithm: I. difference scheme. Journal of Computational Physics **157**(2), 618–653 (2000). DOI 10.1006/jcph.1999.6394

13. Chacón, L., Barnes, D.C., Knoll, D.A., Miley, G.H.: An implicit energy-conservative 2D Fokker–Planck algorithm: II. Jacobian-free Newton–Krylov solver. Journal of Computational Physics **157**(2), 654–682 (2000). DOI 10.1006/jcph.1999.6395

14. Chang, C.S., Ku, S.: Spontaneous rotation sources in a quiescent tokamak edge plasma. Physics of Plasmas **15**(6), 062510 (2008). DOI 10.1063/1.2937116

15. Chang, J., Cooper, G.: A practical difference scheme for Fokker-Planck equations. Journal of Computational Physics **6**(1), 1–16 (1970). DOI 10.1016/0021-9991(70)90001-X

16. Chen, G., Chacón, L.: An energy- and charge-conserving, nonlinearly implicit, electromagnetic 1d-3v Vlasov–Darwin particle-in-cell algorithm. Computer Physics Communications **185**(10), 2391–2402 (2014). DOI 10.1016/j.cpc.2014.05.010

17. Chen, G., Chacón, L.: A multi-dimensional, energy- and charge-conserving, nonlinearly implicit, electromagnetic Vlasov–Darwin particle-in-cell algorithm. Computer Physics Communications **197**, 73–87 (2015). DOI 10.1016/j.cpc.2015.08.008

18. Chen, G., Chacón, L., Barnes, D.: An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm. Journal of Computational Physics **230**(18), 7018–7036 (2011). DOI 10.1016/j.jcp.2011.05.031

19. Cheng, C., Knorr, G.: The integration of the Vlasov equation in configuration space. Journal of Computational Physics **22**(3), 330–351 (1976). DOI 10.1016/0021-9991(76)90053-X

20. Cohen, R.H., Xu, X.Q.: Progress in kinetic simulation of edge plasmas. Contributions to Plasma Physics **48**(1-3), 212–223 (2008). DOI 10.1002/ctpp.200810038

21. Colella, P., Dorr, M., Hittinger, J., Martin, D.: High–order, finite–volume methods in mapped coordinates. Journal of Computational Physics **230**(8), 2952–2976 (2011). DOI 10.1016/j.jcp.2010.12.044

22. Crouseilles, N., Respaud, T., Sonnendrücker, E.: A forward semi-Lagrangian method for the numerical solution of the Vlasov equation. Computer Physics Communications **180**(10), 1730–1745 (2009). DOI 10.1016/j.cpc.2009.04.024

23. Dennis, J., Schnabel, R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Society for Industrial and Applied Mathematics (1996). DOI 10.1137/1.9781611971200

24. Dorf, M.A., Cohen, R.H., Compton, J.C., Dorr, M., Rognlien, T.D., Angus, J., Krasheninnikov, S., Colella, P., Martin, D., McCorquodale, P.: Progress with the COGENT edge kinetic code: Collision operator options. Contributions to Plasma Physics **52**(5-6), 518–522 (2012). DOI 10.1002/ctpp.201210042

25. Dorf, M.A., Cohen, R.H., Dorr, M., Hittinger, J., Rognlien, T.D.: Progress with the COGENT edge kinetic code: Implementing the Fokker-Planck collision operator. Contributions to Plasma Physics **54**(4-6), 517–523 (2014). DOI 10.1002/ctpp.201410023

26. Dorf, M.A., Cohen, R.H., Dorr, M., Rognlien, T., Hittinger, J., Compton, J., Colella, P., Martin, D., McCorquodale, P.: Simulation of neoclassical transport with the continuum gyrokinetic code CO-GENT. Physics of Plasmas **20**(1), 012513 (2013). DOI 10.1063/1.4776712

27. Durran, D.R., Blossey, P.N.: Implicit–explicit multistep methods for fast-wave–slow-wave problems. Monthly Weather Review **140**(4), 1307–1325 (2012). DOI 10.1175/MWR-D-11-00088.1

28. Epperlein, E.: Implicit and conservative difference scheme for the Fokker-Planck equation. Journal of Computational Physics **112**(2), 291 – 297 (1994). DOI 10.1006/jcph.1994.1101

29. Epperlein, E.M., Rickard, G.J., Bell, A.R.: Two-dimensional nonlocal electron transport in laser-produced plasmas. Physical Review Letters **61**, 2453–2456 (1988). DOI 10.1103/PhysRevLett.61.2453

30. Falgout, R.D., Yang, U.M.: hypre: A Library of High Performance Preconditioners, pp. 632–641. Springer Berlin Heidelberg, Berlin, Heidelberg (2002). DOI 10.1007/3-540-47789-6_66

31. Filbet, F., Pareschi, L.: A numerical method for the accurate solution of the Fokker–Planck–Landau equation in the nonhomogeneous case. Journal of Computational Physics **179**(1), 1–26 (2002). DOI 10.1006/jcph.2002.7010

32. Filbet, F., Sonnendrücker, E., Bertrand, P.: Conservative numerical schemes for the Vlasov equation. Journal of Computational Physics **172**(1), 166–187 (2001). DOI 10.1006/jcph.2001.6818

33. Ghosh, D., Constantinescu, E.M.: Semi-implicit time integration of atmospheric flows with characteristic-based flux partitioning. SIAM Journal on Scientific Computing **38**(3), A1848–A1875 (2016). DOI 10.1137/15M1044369

34. Giraldo, F.X., Kelly, J.F., Constantinescu, E.: Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA). SIAM Journal on Scientific Computing **35**(5), B1162–B1194 (2013). DOI 10.1137/120876034

35. Giraldo, F.X., Restelli, M., Läuter, M.: Semi-implicit formulations of the Navier-Stokes equations: Application to nonhydrostatic atmospheric modeling. SIAM Journal on Scientific Computing **32**(6), 3394–3425 (2010). DOI 10.1137/090775889

36. Grandgirard, V., Brunetti, M., Bertrand, P., Besse, N., Garbet, X., Ghendrih, P., Manfredi, G., Sarazin, Y., Sauter, O., Sonnendrücker, E., Vaclavik, J., Villard, L.: A drift-kinetic semi-Lagrangian 4D code for ion turbulence simulation. Journal of Computational Physics **217**(2), 395–423 (2006). DOI 10.1016/j.jcp.2006.01.023

37. Hahm, T.S.: Nonlinear gyrokinetic equations for turbulence in core transport barriers. Physics of Plasmas **3**(12), 4658–4664 (1996). DOI 10.1063/1.872034

38. Heikkinen, J., Kiviniemi, T., Kurki-Suonio, T., Peeters, A., Sipilä, S.: Particle simulation of the neo-classical plasmas. Journal of Computational Physics **173**(2), 527—548 (2001). DOI 10.1006/jcph.2001.6891

39. Heikkinen, J.A., Henriksson, S., Janhunen, S., Kiviniemi, T.P., Ogando, F.: Gyrokinetic simulation of particle and heat transport in the presence of wide orbits and strong profile variations in the edge plasma. Contributions to Plasma Physics **46**(7-9), 490–495 (2006). DOI 10.1002/ctpp.200610035

40. Huba, J.D.: NRL plasma formulary. Tech. rep., Naval Research Laboratory, Washington, DC (2016)

41. Idomura, Y., Ida, M., Kano, T., Aiba, N., Tokuda, S.: Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation. Computer Physics Communications **179**(6), 391–403 (2008). DOI 10.1016/j.cpc.2008.04.005

42. Idomura, Y., Ida, M., Tokuda, S.: Conservative gyrokinetic Vlasov simulation. Communications in Nonlinear Science and Numerical Simulation **13**(1), 227–233 (2008). DOI 10.1016/j.cnsns.2007.05.015. Vlasovia 2006: The Second International Workshop on the Theory and Applications of the Vlasov Equation

43. Jackson, J.D.: Classical electrodynamics, 3rd ed. edn. Wiley, New York, NY (1999)

44. Jacobs, G., Hesthaven, J.: Implicit–explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning. Computer Physics Communications **180**(10), 1760–1767 (2009). DOI 10.1016/j.cpc.2009.05.020

45. James, R.: The solution of Poisson's equation for isolated source distributions. Journal of Computational Physics **25**(2), 71–93 (1977). DOI 10.1016/0021-9991(77)90013-4

46. Jiang, G.S., Shu, C.W.: Efficient implementation of weighted ENO schemes. Journal of Computational Physics **126**(1), 202–228 (1996). DOI 10.1006/jcph.1996.0130

47. Kadioglu, S.Y., Knoll, D.A., Lowrie, R.B., Rauenzahn, R.M.: A second order self-consistent IMEX method for radiation hydrodynamics. Journal of Computational Physics **229**(22), 8313–8332 (2010). DOI 10.1016/j.jcp.2010.07.019
48. Kennedy, C.A., Carpenter, M.H.: Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Applied Numerical Mathematics **44**(1-2), 139–181 (2003). DOI 10.1016/S0168-9274(02)00138-1
49. Kho, T.H.: Relaxation of a system of charged particles. Physics Review A **32**, 666–669 (1985). DOI 10.1103/PhysRevA.32.666
50. Kingham, R., Bell, A.: An implicit Vlasov–Fokker–Planck code to model non-local electron transport in 2–D with magnetic fields. Journal of Computational Physics **194**(1), 1–34 (2004). DOI 10.1016/j.jcp.2003.08.017
51. Knoll, D., Keyes, D.: Jacobian-free Newton–Krylov methods: a survey of approaches and applications. Journal of Computational Physics **193**(2), 357–397 (2004). DOI 10.1016/j.jcp.2003.08.010
52. Kumar, H., Mishra, S.: Entropy stable numerical schemes for two-fluid plasma equations. Journal of Scientific Computing **52**(2), 401–425 (2012). DOI 10.1007/s10915-011-9554-7
53. Landau, L.D.: The kinetic equation in the case of Coulomb interaction. Zh. Eksper. i Teoret. Fiz. **7**(2), 203–209 (1937)
54. Landreman, M., Ernst, D.R.: New velocity-space discretization for continuum kinetic calculations and Fokker–Planck collisions. Journal of Computational Physics **243**, 130–150 (2013). DOI 10.1016/j.jcp.2013.02.041
55. Larroche, O.: Kinetic simulation of a plasma collision experiment. Physics of Fluids B **5**(8), 2816–2840 (1993). DOI 10.1063/1.860670
56. Larroche, O.: Kinetic simulations of fuel ion transport in ICF target implosions. The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics **27**(2), 131–146 (2003). DOI 10.1140/epjd/e2003-00251-1
57. Larsen, E., Levermore, C., Pomraning, G., Sanderson, J.: Discretization methods for one-dimensional Fokker-Planck operators. Journal of Computational Physics **61**(3), 359 – 390 (1985). DOI 10.1016/0021-9991(85)90070-1
58. Lee, W.W.: Gyrokinetic approach in particle simulation. Physics of Fluids **26**(2), 556–562 (1983). DOI 10.1063/1.864140
59. Lemou, M.: Multipole expansions for the Fokker-Planck-Landau operator. Numerische Mathematik **78**(4), 597–618 (1998). DOI 10.1007/s002110050327
60. Lemou, M., Mieussens, L.: Fast implicit schemes for the Fokker–Planck–Landau equation. Comptes Rendus Mathematique **338**(10), 809–814 (2004). DOI 10.1016/j.crma.2004.03.010
61. Lemou, M., Mieussens, L.: Implicit schemes for the Fokker–Planck–Landau equation. SIAM Journal on Scientific Computing **27**(3), 809–830 (2005). DOI 10.1137/040609422
62. Maeyama, S., Ishizawa, A., Watanabe, T.H., Nakajima, N., Tsuji-Iio, S., Tsutsui, H.: A hybrid method of semi-Lagrangian and additive semi-implicit Runge–Kutta schemes for gyrokinetic Vlasov simulations. Computer Physics Communications **183**(9), 1986–1992 (2012). DOI 10.1016/j.cpc.2012.04.028
63. McCorquodale, P., Dorr, M., Hittinger, J., Colella, P.: High–order finite–volume methods for hyperbolic conservation laws on mapped multiblock grids. Journal of Computational Physics **288**, 181–195 (2015). DOI 10.1016/j.jcp.2015.01.006
64. McCoy, M., Mirin, A., Killeen, J.: FPPAC: A two-dimensional multispecies nonlinear Fokker-Planck package. Computer Physics Communications **24**(1), 37–61 (1981). DOI 10.1016/0010-4655(81)90105-3
65. Mousseau, V., Knoll, D.: Fully implicit kinetic solution of collisional plasmas. Journal of Computational Physics **136**(2), 308–323 (1997). DOI 10.1006/jcph.1997.5736
66. Pareschi, L., Russo, G.: Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. Journal of Scientific Computing **25**(1-2), 129–155 (2005). DOI 10.1007/BF02728986
67. Pataki, A., Greengard, L.: Fast elliptic solvers in cylindrical coordinates and the Coulomb collision operator. Journal of Computational Physics **230**(21), 7840–7852 (2011). DOI 10.1016/j.jcp.2011.07.005
68. Pernice, M., Walker, H.F.: NITSOL: A Newton iterative solver for nonlinear systems. SIAM Journal on Scientific Computing **19**(1), 302–318 (1998). DOI 10.1137/S1064827596303843
69. Porter, G.D., Isler, R., Boedo, J., Rognlien, T.D.: Detailed comparison of simulated and measured plasma profiles in the scrape-off layer and edge plasma of DIII-D. Physics of Plasmas **7**(9), 3663–3680 (2000). DOI 10.1063/1.1286509

70. Qiu, J.M., Christlieb, A.: A conservative high order semi-Lagrangian WENO method for the Vlasov equation. Journal of Computational Physics **229**(4), 1130–1149 (2010). DOI 10.1016/j.jcp.2009.10.016

71. Qiu, J.M., Shu, C.W.: Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov–Poisson system. Journal of Computational Physics **230**(23), 8386–8409 (2011). DOI 10.1016/j.jcp.2011.07.018

72. Rosenbluth, M.N., MacDonald, W.M., Judd, D.L.: Fokker-Planck equation for an inverse-square force. The Physical Review **107**, 1–6 (1957). DOI 10.1103/PhysRev.107.1

73. Rossmanith, J.A., Seal, D.C.: A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov–Poisson equations. Journal of Computational Physics **230**(16), 6203–6232 (2011). DOI 10.1016/j.jcp.2011.04.018

74. Saad, Y.: Iterative Methods for Sparse Linear Systems: Second Edition. Society for Industrial and Applied Mathematics (2003)

75. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing **7**(3), 856–869 (1986). DOI 10.1137/0907058

76. Salmi, S., Toivanen, J., von Sydow, L.: An IMEX-scheme for pricing options under stochastic volatility models with jumps. SIAM Journal on Scientific Computing **36**(5), B817–B834 (2014). DOI 10.1137/130924905

77. Scott, B.: Gyrokinetic study of the edge shear layer. Plasma Physics and Controlled Fusion **48**(5A), A387 (2006)

78. Taitano, W., Chacón, L., Simakov, A.: An adaptive, conservative 0D–2V multispecies Rosenbluth–Fokker–Planck solver for arbitrarily disparate mass and temperature regimes. Journal of Computational Physics **318**, 391–420 (2016). DOI 10.1016/j.jcp.2016.03.071

79. Taitano, W., Chacón, L., Simakov, A., Molvig, K.: A mass, momentum, and energy conserving, fully implicit, scalable algorithm for the multi-dimensional, multi-species Rosenbluth–Fokker–Planck equation. Journal of Computational Physics **297**, 357–380 (2015). DOI 10.1016/j.jcp.2015.05.025

80. Thomas, A., Tzoufras, M., Robinson, A., Kingham, R., Ridgers, C., Sherlock, M., Bell, A.: A review of Vlasov–Fokker–Planck numerical modeling of inertial confinement fusion plasma. Journal of Computational Physics **231**(3), 1051–1079 (2012). DOI 10.1016/j.jcp.2011.09.028. Special Issue: Computational Plasma Physics

81. Thomas, A.G.R., Kingham, R.J., Ridgers, C.P.: Rapid self-magnetization of laser speckles in plasmas by nonlinear anisotropic instability. New Journal of Physics **11**(3), 033,001 (2009)

82. Wolke, R., Knoth, O.: Implicit–explicit Runge–Kutta methods applied to atmospheric chemistry-transport modelling. Environmental Modelling & Software **15**(6–7), 711–719 (2000). DOI 10.1016/S1364-8152(00)00034-7

83. Xu, X., Bodi, K., Cohen, R., Krasheninnikov, S., Rognlien, T.: TEMPEST simulations of the plasma transport in a single-null tokamak geometry. Nuclear Fusion **50**(6), 064,003 (2010)

84. Xu, X., Xiong, Z., Dorr, M., Hittinger, J., Bodi, K., Candy, J., Cohen, B., Cohen, R., Colella, P., Kerbel, G., Krasheninnikov, S., Nevins, W., Qin, H., Rognlien, T., Snyder, P., Umansky, M.: Edge gyrokinetic theory and continuum simulations. Nuclear Fusion **47**(8), 809 (2007)

85. Xu, X.Q., Xiong, Z., Gao, Z., Nevins, W.M., McKee, G.R.: TEMPEST simulations of collisionless damping of the geodesic-acoustic mode in edge-plasma pedestals. Phys. Rev. Lett. **100**, 215,001 (2008). DOI 10.1103/PhysRevLett.100.215001