AIAA 2017-4016

AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
48th AIAA Plasmadynamics and Lasers Conference

Implicit–Explicit Time Integration for the Vlasov–Fokker–Planck Equations

Debojyoti Ghosh,[*] Mikhail A. Dorf,[†] Jeffrey A. F. Hittinger,[‡] and Milo R. Dorr[§]

Lawrence Livermore National Laboratory, Livermore, CA, 94550, United States

This paper proposes a semi-implicit time integration method for the Vlasov–Fokker–Planck equations to simulate the dynamics of tokamak edge plasmas. The plasma is cold and highly collisional near the edge, and the collisional time scales are significantly faster than the advective time scales. Explicit time integration is inefficient because the time step is limited by the collisional time scale. High-order conservative additive Runge-Kutta methods are used to integrate the Vlasov term explicitly and the collision term implicitly, and this allows time steps comparable to the advective time scale. The semi-implicit approach is implemented in COGENT, a high-order finite-volume code that solves the gyrokinetic equations on mapped, multiblock grids. Test problems representative of the tokamak edge are used to verify the algorithm and to compare the computational cost with explicit time integration.

I. Introduction

The dynamics of plasmas at the tokamak–type fusion reactor edge are characterized by a large range of spatial and temporal scales. Adjacent to the hot core region, the plasma is weakly collisional,[1–3] and the accurate modeling of the transport processes requires a kinetic simulation with a detailed collisional model. On the other hand, the cold edge region is strongly collisional with much smaller mean free paths and collisional time scales. The Vlasov–Fokker–Planck (VFP) equation for each charged species governs the evolution of its distribution function in the phase space comprising the spatial position and velocity coordinates.[4] The gyrokinetic model averages these equations over the particle gyromotion (the fastest time scale) and thus eliminates one velocity dimension.[1,5] These equations form a system of parabolic partial differential equations (PDEs), and three families of numerical methods have been developed to solve them: Lagrangian particle-in-cell (PIC) methods,[6–11] semi-Lagrangian methods,[12] and Eulerian methods.[1,13–17] The PIC method solves the equations of motion for “superparticles” in the Lagrangian frame; however, one drawback of this approach is that the required number of superparticles may be very large to control the numerical noise.[2,4,18] The semi-Lagrangian approach alleviates this drawback by using a grid-based approach: the Vlasov equations are integrated along their characteristics, and a high-order interpolation is used to compute the distribution function at grid points.[12] This paper focuses on the Eulerian methods that solve the PDE by discretizing it over a phase space grid, which allows for the use of advanced numerical algorithms from the Computational Fluid Dynamics (CFD) community.

The VFP equation comprises a Vlasov operator that describes incompressible flow in phase space[4,19] and a Fokker–Planck collision term that describes the collisional relaxation of the distribution functions.[4,20] The latter is expressed as a nonlinear, integro-differential advection–diffusion operator where the advective and diffusive coefficients are integral functions of the distribution function. The numerical implementation of this term raises two concerns:[21] (i) the conservation of mass, momentum, and energy in the discrete form and (ii) numerical stiffness. The Fokker–Planck operator can be expressed in two different but analytically equivalent forms: the Fokker–Planck–Landau form[22] and the Fokker–Planck–Rosenbluth form.[20] The Landau form expresses the advection and diffusion coefficients of the Fokker–Planck operator as direct integrals, and this

[*] Postdoctoral Research Staff Member, Center for Applied Scientific Computing, ghosh5@llnl.gov, and AIAA Member.
[†] Research Staff Member, Physics Division, dorf1@llnl.gov.
[‡] Group Leader, Scientific Computing Group, Center for Applied Scientific Computing, hittinger1@llnl.gov.
[§] Computational Scientist, Center for Applied Scientific Computing, dorr1@llnl.gov.

form is well-suited for conservative numerical methods.[23–29] However, the integral form of the coefficients results in dense, nonlinear systems of equations that are computationally expensive to solve. While the cost of a naive algorithm would scale as $\mathcal{O}\left(N^2\right)$ (where N is the number of grid points), fast methods[30–33] have been proposed where the cost scales as $\mathcal{O}\left(N \log N\right)$.

The Fokker–Planck–Rosenbluth form expresses the relationship between the advection–diffusion coefficients and the distribution function through Poisson equations for the Rosenbluth potentials. Assuming that a fast solver for the Poisson solve is available, the computational expense of evaluating the collision term scales as $\mathcal{O}\left(N\right)$. There have been several approaches in the literature to solving the Poisson equation on an infinite domain in the velocity space, including the James–Lackner method,[34] the decomposition of the distribution function with Legendre polynomials of the parallel velocity coordinate,[35–37] a spectral method involving a fast Fourier transform in the parallel velocity dimension,[38] a residual formulation,[39] and multigrid methods.[40] In this paper, the Rosenbluth form is used, and a second-order finite-difference discretization of the Poisson equations is implemented and solved iteratively.[3] A drawback of the Fokker–Planck–Rosenbluth form is the difficulty in enforcing mass, momentum, and energy conservation in the discretized form of the collision term. If discretized by a conservative numerical scheme, mass is conserved; however, additional steps are required to conserve momentum and energy to the level of round-off errors.[21,41] These considerations are beyond the scope of this paper, which concerns the efficient treatment of the fast time scales.

Collisional time scales are significantly smaller than the advective time scales in the cold edge region. The spatially discretized Fokker–Planck operator is numerically stiff, and, when coupled with the Vlasov operator, the overall system exhibits significant scale separation. Explicit time integration methods are therefore inefficient, since their time steps are constrained by the collisional time scale for linear stability. Linearly implicit methods[27,31,39,42–45] and fully (nonlinearly) implicit methods[21,25,26,40,46] have been proposed for the Fokker–Planck operator; however, these methods treat the collision term in isolation. Several algorithms have also been proposed that apply fully implicit time integration methods to the VFP equation.[47–50]

Semi–implicit or implicit–explicit (IMEX) time integration methods were developed[51–53] for multiscale simulations where the stiff components are integrated implicitly in time, while the nonstiff components are integrated explicitly. The time step for the resulting algorithm is constrained by the time scales of the nonstiff components, and larger time steps are possible compared with an explicit method. Such an approach is advantageous when the stiff terms are relatively inexpensive to compute compared with the nonstiff terms; thus an IMEX formulation will result in a nonlinear or linear system of equations that is significantly less expensive to solve than that arising from a fully implicit method. IMEX methods have been successfully applied to multiscale systems, for example, atmospheric flows.[54–57] In the context of gyrokinetic algorithms, operator-split semi–implicit algorithms have been proposed[58–61] where the Vlasov term is partitioned and where parallel advection is integrated implicitly in time.

In this paper, we propose an IMEX method for the VFP equation, where the Vlasov term is integrated explicitly in time and where the Fokker–Planck term is integrated implicitly. A semi–implicit algorithm is chosen instead of a fully implicit one to resolve unsteady advective processes accurately; thus, the time step will be of the same order as the Vlasov time scale. Semi–implicit, multistage additive Runge-Kutta (ARK) methods[53] are implemented in COGENT,[2,3,62] a finite-volume mapped multiblock gyrokinetic code for complex geometries. The nonlinear system of equations arising from the implicit integration of the Fokker–Planck term is solved using the "Jacobian-free" Newton-Krylov method.[63] We verify the implementation of the IMEX methods on test problems that are representative of tokamak edge plasma dynamics and compare their performance with explicit Runge–Kutta methods.

The paper is organized as follows. Sec. II describes the VFP equation. Sec. III discusses the numerical methodology; specifically, Sec. III C discusses the implementation of the ARK time integration methods. Numerical results and verifications are presented in Sec. IV, and conclusions are summarized in Sec. V.

II. Governing Equations

The governing equations are the full–f gyrokinetic model[2,19] expressed as

$$\frac{\partial \tilde{f}_\alpha}{\partial t} + \nabla_{\mathbf{R}} \cdot \left(\dot{\mathbf{R}}_\alpha \tilde{f}_\alpha\right) + \frac{\partial}{\partial v_\parallel}\left(\dot{v}_{\parallel,\alpha} \tilde{f}_\alpha\right) = \sum_{\beta=1}^{N_s} c^{\mathrm{FP}}_\beta \left(\tilde{f}_\alpha\right), \quad \alpha = 1, \cdots, N_s, \tag{1}$$

where α, β denote the charged species indices, N_s is the number of species, $\tilde{f}_\alpha = B^*_{\parallel,\alpha} f_\alpha$, and

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

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

are effective velocities and accelerations in the configuration and velocity space, respectively. The distribution function f_α is defined on the phase space $\left(\mathbf{R}, v_\parallel, \mu\right)$, where $\mathbf{R} \equiv (r, \theta)$ 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 parallel velocity, and $\mu = m_\alpha v_\perp^2 / 2B$ is the magnetic moment. COGENT is currently implemented for a two-dimensional configuration space, so Eq. (1) represents a four-dimensional (2D–2V) PDE. The species mass and charge state are given by m_α and Z_α, respectively, and e is the electron charge. The magnetic field is denoted by $\mathbf{B} = B\hat{\mathbf{b}}$, where B and $\hat{\mathbf{b}}$ are its magnitude and a unit vector along $\mathbf{B}$, respectively, and where

$$\mathbf{B}^*_\alpha \equiv \mathbf{B}^*_\alpha \left(\mathbf{R}, v_\parallel\right) = \mathbf{B} + \frac{m_\alpha v_\parallel}{Z_\alpha e} \nabla \times \hat{\mathbf{b}}; \;\; B^*_{\parallel,\alpha} = \mathbf{B}^*_\alpha \cdot \hat{\mathbf{b}} \tag{3}$$

is the Jacobian of the transformation from the lab frame to the gyrocentric coordinates. The electric field is $\mathbf{E} = -\nabla_{\mathbf{R}} \phi$, where ϕ is the electrostatic potential that is computed using some form of Maxwell's equations. In COGENT, the long wavelength approximation is used to calculate the electrostatic potential from the species charge densities,[2,62] where a Boltzmann model is used for the electrons.[2,64] However, in the test problems presented in this paper, the electric field is explicitly specified as $\mathbf{E} \equiv \mathbf{E}\left(\mathbf{R}\right)$.

The Fokker–Planck collision operator $c^{\rm FP}_\beta \left(\tilde{f}_\alpha\right)$ in Eq. (1) is implemented in the Rosenbluth form,[3] which is given by

$$c^{\rm FP}_\beta \left(\tilde{f}_\alpha\right) = \lambda_c \left(\frac{4\pi Z_\alpha Z_\beta e^2}{m_\alpha} \right)^2 \left(\frac{\partial \Gamma_{v_\parallel}}{\partial v_\parallel} + \frac{\partial \Gamma_\mu}{\partial \mu} \right), \tag{4}$$

where λ_c is the Coulomb logarithm and

$$\Gamma_{v_\parallel} = \sigma_{\beta,v_\parallel} f_\alpha + \kappa_{\beta,v_\parallel v_\parallel} \frac{\partial f_\alpha}{\partial v_\parallel} + \kappa_{\beta,v_\parallel \mu} \frac{\partial f_\alpha}{\partial \mu}, \tag{5a}$$

$$\Gamma_\mu = \sigma_{\beta,\mu} f_\alpha + \kappa_{\beta,\mu v_\parallel} \frac{\partial f_\alpha}{\partial v_\parallel} + \kappa_{\beta,\mu\mu} \frac{\partial f_\alpha}{\partial \mu}, \tag{5b}$$

with the coefficients defined as follows:

$$\begin{aligned} &\sigma_{\beta,v_\parallel} = \frac{\partial \varphi_\beta}{\partial v_\parallel}, \quad \sigma_{\beta,\mu} = 2\mu \frac{m_\beta}{B} \frac{\partial \varphi_\beta}{\partial \mu}, \\ &\kappa_{\beta,v_\parallel v_\parallel} = -\frac{\partial^2 \varrho_\beta}{\partial v_\parallel^2}, \; \kappa_{\beta,v_\parallel \mu} = \kappa_{\beta,\mu v_\parallel} = -2\mu \frac{m_\beta}{B} \frac{\partial^2 \varrho_\beta}{\partial v_\parallel \partial \mu}, \; \kappa_{\beta,\mu\mu} = -2\mu \left(\frac{m_\beta}{B}\right)^2 \left[2\mu \frac{\partial^2 \varrho_\beta}{\partial \mu^2} + \frac{\partial \varrho_\beta}{\partial \mu} \right]. \end{aligned} \tag{6}$$

The Rosenbluth potentials, φ_β and ϱ_β, are related to the distribution function through the following Poisson equations in the two-dimensional velocity space $\left(v_\parallel, \mu\right)$:

$$\frac{\partial^2 \varphi_\beta}{\partial v_\parallel^2} + \frac{m_\beta}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \varphi_\beta}{\partial \mu} \right) = f_\beta, \tag{7a}$$

$$\frac{\partial^2 \varrho_\beta}{\partial v_\parallel^2} + \frac{m_\beta}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \varrho_\beta}{\partial \mu} \right) = \varphi_\beta. \tag{7b}$$

In this paper, we only consider a single species ($N_s = 1$), and the subscripts α, β are dropped in the subsequent discussions.

The number density and temperature are defined as

$$\rho\left(\mathbf{R}\right) = \frac{2\pi B\left(\mathbf{R}\right)}{m}\int f\left(\mathbf{R}, v_\parallel, \mu\right) dv_\parallel d\mu, \tag{8}$$

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

respectively, where

$$\overline{v_\parallel} \equiv \overline{v_\parallel}\left(\mathbf{R}\right) = \frac{1}{\rho}\frac{2\pi B\left(\mathbf{R}\right)}{m}\int v_\parallel f\left(\mathbf{R}, v_\parallel, \mu\right) dv_\parallel d\mu \tag{10}$$

is the average parallel velocity. The Maxwellian distribution function is defined as

$$f_{\mathrm{M}}\left(\mathbf{R}, v_\parallel, \mu\right) = \left(\frac{m}{2\pi T\left(\mathbf{R}\right)}\right)^{\frac{3}{2}} \exp\left(\frac{m v_\parallel^2 + 2\mu B\left(\mathbf{R}\right)}{2T\left(\mathbf{R}\right)}\right). \tag{11}$$

These quantities are used in the description of the test cases and in the analysis of the numerical results obtained in Sec. IV.

III. Numerical Methodology

COGENT implements a high-order finite-volume method on mapped, multiblock grids for the discretization of the spatial derivatives in Eq. (1).[65,66] Although we present examples that involve Cartesian grids in this paper, our algorithm is implemented for general geometries that can be represented by multiblock curvilinear grids. We define a four-dimensional Cartesian domain of unit length in each dimension,

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

where $\mathbf{e}_d$ is the unit vector along dimension d, and $D = 4$ is the total number of dimensions. A uniform grid is used to partition Ω with a computational cell defined as

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

where $\mathbf{i}$ is an integer vector representing a four-dimensional grid index. The nonsingular mapping $\mathbf{x} = \mathbf{X}\left(\boldsymbol{\xi}\right)$ defines the transformation of the unit Cartesian domain to the physical domain, where $\mathbf{x} \equiv \left(\mathbf{R}, v_\parallel, \mu\right)$. Integrating Eq. (1) over the computational cell $\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)$, we get

$$\frac{\partial}{\partial t}\left(\int\limits_{\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)} \tilde{f} d\mathbf{x}\right) = \int\limits_{\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)} \mathcal{V}\left(\tilde{f}\right) d\mathbf{x} + \int\limits_{\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)} \mathcal{C}\left(\tilde{f}\right) d\mathbf{x}, \tag{14}$$

where the Vlasov and collision terms are

$$\mathcal{V}\left(\tilde{f}\right) \equiv -\left[\nabla_{\mathbf{R}}\cdot\left(\dot{\mathbf{R}}\tilde{f}\right) + \frac{\partial}{\partial v_\parallel}\left(\dot{v}_\parallel \tilde{f}\right)\right], \ \ \mathcal{C}\left(\tilde{f}\right) = c^{\mathrm{FP}}\left(\tilde{f}\right). \tag{15}$$

We define the computational cell-averaged conserved quantity as

$$\bar{f}_{\mathbf{i}} = \frac{1}{\mathrm{Vol}\left(\mathbb{V}_{\mathbf{i}}\right)}\int\limits_{\mathbb{V}_{\mathbf{i}}} \tilde{f} J d\boldsymbol{\xi}; \quad J \equiv \left|\frac{\partial\mathbf{x}}{\partial\boldsymbol{\xi}}\right|, \tag{16}$$

where $\mathrm{Vol}\left(\mathbb{V}_{\mathbf{i}}\right)$ is the volume of the computational cell $\mathbb{V}_{\mathbf{i}}$, such that the physical cell-averaged solution is

$$\breve{f}_{\mathbf{i}} = \frac{1}{\mathrm{Vol}\left(\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)\right)}\int\limits_{\mathbf{X}\left(\mathbb{V}_{\mathbf{i}}\right)} \tilde{f} d\mathbf{x} = \left(\int\limits_{\mathbb{V}_{\mathbf{i}}} J d\boldsymbol{\xi}\right)^{-1}\int\limits_{\mathbb{V}_{\mathbf{i}}} \tilde{f} J d\boldsymbol{\xi} = \bar{J}_{\mathbf{i}}^{-1}\bar{f}_{\mathbf{i}}, \tag{17}$$

and note that

$$\mathcal{V}\left(\tilde{f}\right) = \nabla_{\mathbf{x}}\cdot\mathbf{V}\left(\tilde{f}\right);\quad \mathbf{V}\left(\tilde{f}\right) = -\left[\begin{array}{ccc}\dot{\mathbf{R}}\tilde{f} & \dot{v}_{\parallel}\tilde{f} & 0\end{array}\right]^{\mathrm{T}}, \tag{18a}$$

$$\text{and}\;\; \mathcal{C}\left(\tilde{f}\right) = \nabla_{\mathbf{x}}\cdot\mathbf{C}\left(\tilde{f}\right);\quad \mathbf{C}\left(\tilde{f}\right) = \lambda_c\left(\frac{4\pi Z^2 e^2}{m}\right)^2\left[\begin{array}{cccc}0 & 0 & \Gamma_{v_{\parallel}} & \Gamma_{\mu}\end{array}\right]^{\mathrm{T}}. \tag{18b}$$

Therefore, using the divergence theorem, we obtain

$$\begin{aligned}\frac{\partial\bar{f}_{\mathbf{i}}}{\partial t} &= \frac{1}{\mathrm{Vol}\left(\mathbb{V}_{\mathbf{i}}\right)}\left[\int\limits_{\partial\mathbf{X}(\mathbb{V}_{\mathbf{i}})}\mathbf{V}\left(\tilde{f}\right)\cdot\hat{\mathbf{n}}d\mathbf{s} + \int\limits_{\partial\mathbf{X}(\mathbb{V}_{\mathbf{i}})}\mathbf{C}\left(\tilde{f}\right)\cdot\hat{\mathbf{n}}d\mathbf{s}\right]\\ &= \frac{1}{\mathrm{Vol}\left(\mathbb{V}_{\mathbf{i}}\right)}\sum_{d=1}^{D}\left[\int\limits_{A^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}}\mathbf{N}^{\mathrm{T}}\mathbf{V}dA_{\boldsymbol{\xi}} - \int\limits_{A^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}\mathbf{N}^{\mathrm{T}}\mathbf{V}dA_{\boldsymbol{\xi}} + \int\limits_{A^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}}\mathbf{N}^{\mathrm{T}}\mathbf{C}dA_{\boldsymbol{\xi}} - \int\limits_{A^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}\mathbf{N}^{\mathrm{T}}\mathbf{C}dA_{\boldsymbol{\xi}}\right],\end{aligned} \tag{19}$$

where $\hat{\mathbf{n}}$ is the outward normal, and $\mathbf{N} = J\nabla_{\mathbf{x}}\boldsymbol{\xi}$. The spatially discretized equation can be written for the i-th computational cell as

$$\frac{\partial\bar{f}_{\mathbf{i}}}{\partial t} = \frac{1}{h}\sum_{d=1}^{D}\left[\left(V^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - V^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right) + \left(C^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - C^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right)\right];\quad V^d = \mathbf{V}\cdot\mathbf{e}_d, C^d = \mathbf{C}\cdot\mathbf{e}_d, \tag{20}$$

where, due to our computational cell being a Cartesian hypercube with length h in each dimension, $\mathrm{Vol}\left(\mathbb{V}_{\mathbf{i}}\right) = h^D$, and the area of each face is $A^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = h^{D-1}\;\forall d$. Defining the solution vector as consisting of the cell-averaged distribution function at the computational cells,

$$\bar{\mathbf{f}} = \left[\bar{f}_{\mathbf{i}}\right],\quad \mathbf{i}\in\mathcal{G};\quad \mathcal{G} = \{\mathbf{j} : \mathbb{V}_{\mathbf{j}}\in\Omega\}, \tag{21}$$

Eq. (20) can be expressed for the entire computational domain as a system of ordinary differential equations (ODEs) in time,

$$\frac{d\bar{\mathbf{f}}}{dt} = \hat{\mathcal{V}}\left(\bar{\mathbf{f}}\right) + \hat{\mathcal{C}}\left(\bar{\mathbf{f}}\right). \tag{22}$$

Here, $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ denote the spatially discretized Vlasov and collision terms:

$$\hat{\mathcal{V}}\left(\bar{f}_{\mathbf{i}}\right) = \frac{1}{h}\sum_{d=1}^{D}\left(V^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - V^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right) = \mathcal{V}\left(\tilde{f}\right) + \mathcal{O}\left(\Delta\mathbf{R}^p, \Delta v^p_{\parallel}, \Delta\mu^p\right), \tag{23a}$$

$$\hat{\mathcal{C}}\left(\bar{f}_{\mathbf{i}}\right) = \frac{1}{h}\sum_{d=1}^{D}\left(C^d_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - C^d_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right) = \mathcal{C}\left(\tilde{f}\right) + \mathcal{O}\left(\Delta\mathbf{R}^q, \Delta v^q_{\parallel}, \Delta\mu^q\right), \tag{23b}$$

where p, q are the order of the schemes used to discretize the Vlasov and collision terms, respectively. In the current implementation, $p = 4$ but $q = 2$ (this will be explained subsequently).

In the following subsection, we describe the computation of the Vlasov and collision terms at the cell faces ($V^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ and $C^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$) in Eq. (23), while in Sec. III C, we discuss the methods used to evolve Eq. (22) in time.

A. Flux Calculation

Equation (20) requires the approximation of face-averaged $V^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ and $C^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, and these are computed in the physical space to fourth-order accuracy as[65]

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

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

where $\mathbf{G}_0^{\perp,d} \approx \nabla_{\boldsymbol{\xi}} - \mathbf{e}^d \frac{\partial}{\partial \xi_d}$; $\langle N_d^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}; s = 1, \cdots, D$ are the column vectors of the face-averaged metric quantities,[65] and

$$\langle (\cdot) \rangle_\Gamma = \frac{1}{\text{Area}\,(\Gamma)} \int\limits_\Gamma (\cdot)\, d\Gamma \tag{25}$$

denotes the average of $(\cdot)$ over the face Γ. Equation (24a) requires the computation of the face-averaged $\langle V^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}, s = 1, \cdots, D$ in the computational space. The Vlasov flux term is of the form

$$\mathbf{V}\left(\tilde{f}\right) = \mathbf{a}\tilde{f}, \quad \mathbf{a} = -\left[\begin{array}{ccc} \dot{\mathbf{R}} & \dot{v}_\parallel & 0 \end{array}\right] \quad \Rightarrow V^d = a^d \tilde{f}, \quad d = 1, \cdots, D, \tag{26}$$

and the face-averaged flux is computed from the discrete convolution as

$$\langle V^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \langle a^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \frac{h^2}{12} \sum_{d' \neq d} \left(\frac{\partial a^s}{\partial \xi_{d'}} \frac{\partial \bar{\bar{f}}}{\partial \xi_{d'}} \right)_{\boldsymbol{\xi}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}} + \mathcal{O}\left(h^4\right), \tag{27}$$

where $\bar{\bar{f}}$ is the cell-averaged distribution function in the computational space,

$$\bar{\bar{f}}_{\mathbf{i}} = \frac{1}{\mathbb{V}_{\mathbf{i}}} \int\limits_{\mathbb{V}_{\mathbf{i}}} \tilde{f} d\boldsymbol{\xi} \quad \Rightarrow \bar{\bar{f}}_{\mathbf{i}} = J_{\mathbf{i}}^{-1} \left[\bar{f}_{\mathbf{i}} - \frac{h^2}{12} \nabla_\xi \bar{f} \cdot \nabla_\xi J \right] + \mathcal{O}\left(h^4\right), \tag{28}$$

and where $\nabla_{\boldsymbol{\xi}} \tilde{f}$ is computed to second-order accuracy at the cell centers as

$$\left(\nabla_{\boldsymbol{\xi}} \tilde{f}\right)_{\mathbf{i}}^d = \frac{1}{2h} \left[\frac{\bar{f}_{\mathbf{i}+\mathbf{e}_d}}{\bar{\bar{J}}_{\mathbf{i}+\mathbf{e}_d}} - \frac{\bar{f}_{\mathbf{i}-\mathbf{e}_d}}{\bar{\bar{J}}_{\mathbf{i}-\mathbf{e}_d}} \right] + \mathcal{O}\left(h^2\right). \tag{29}$$

The evaluation of $\langle a^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ requires the evaluation of $\left\langle \dot{\mathbf{R}} \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ and $\left\langle \dot{v}_\parallel \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, which are obtained by evaluating Eq. (2) at the face centers and by computing the face-averaged quantities as

$$\langle (\cdot) \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = (\cdot)_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \frac{h^2}{24} \sum_{d' \neq d} \left. \frac{\partial^2 (\cdot)}{\partial \xi_{d'}^2} \right|_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \mathcal{O}\left(h^4\right). \tag{30}$$

Finally, the derivatives $\partial (\cdot) / \partial \xi_d'$ in Eqs. (27) and (30) are computed to second-order accuracy using central finite-differences, and $\left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ is computed using the fifth-order weighted essentially nonoscillatory (WENO) scheme[67] with upwinding based on the sign of $\langle a^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$.

Equation (24b) requires the computation of the face-averaged $\langle C^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}, s = 1, \cdots, D$, in the computational space,

$$\langle C^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \lambda_c \left(\frac{4\pi Z^2 e^2}{m} \right)^2 \begin{cases} \left\langle \Gamma_{v_\parallel} \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}, & s = d_{v_\parallel}, \\ \left\langle \Gamma_\mu \right\rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}, & s = d_\mu, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

The face-averaged collisional fluxes are computed from the cell-averaged collision fluxes to fourth-order accuracy as

$$\left\langle \Gamma_{(\cdot)} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{h} \left[\frac{7}{12} \left(\bar{\Gamma}_{(\cdot),\mathbf{i}} + \bar{\Gamma}_{(\cdot),\mathbf{i}+\mathbf{e}_d} \right) - \frac{1}{12} \left(\bar{\Gamma}_{(\cdot),\mathbf{i}+2\mathbf{e}_d} + \bar{\Gamma}_{(\cdot),\mathbf{i}-\mathbf{e}_d} \right) \right] + \mathcal{O}\left(h^4\right), \tag{32}$$

and similarly for $\left\langle \Gamma_{(\cdot)} \right\rangle_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}$. The cell-averaged collisional fluxes $\bar{\Gamma}_{(\cdot),\mathbf{i}}$ are computed from their cell-centered values as

$$\bar{\Gamma}_{(\cdot),\mathbf{i}} = \Gamma_{(\cdot),\mathbf{i}} + \frac{h^2}{24} \sum_{d=1}^{D} \left. \frac{\partial^2 \Gamma_{(\cdot)}}{\partial \xi_d^2} \right|_{\boldsymbol{\xi}_{\mathbf{i}}} + \mathcal{O}\left(h^4\right). \tag{33}$$

The cell-centered collisional fluxes have the following form:

$$\Gamma_{v_\parallel,\mathbf{i}} = \sigma_{v_\parallel,\mathbf{i}} f_{\mathbf{i}} + \kappa_{v_\parallel v_\parallel,\mathbf{i}} \left.\frac{\partial f}{\partial v_\parallel}\right|_{\mathbf{i}} + \kappa_{v_\parallel \mu,\mathbf{i}} \left.\frac{\partial f}{\partial \mu}\right|_{\mathbf{i}}, \tag{34a}$$

$$\Gamma_{\mu,\mathbf{i}} = \sigma_{\mu,\mathbf{i}} f_{\mathbf{i}} + \kappa_{\mu v_\parallel,\mathbf{i}} \left.\frac{\partial f}{\partial v_\parallel}\right|_{\mathbf{i}} + \kappa_{\mu\mu,\mathbf{i}} \left.\frac{\partial f}{\partial \mu}\right|_{\mathbf{i}}. \tag{34b}$$

The cell-centered values of the advection and diffusion coefficients $\sigma_{(\cdot),\mathbf{i}}, \kappa_{(\cdot),\mathbf{i}}$ are related to the Rosenbluth potentials by Eq. (6) and are computed using fourth-order accurate finite-differences:

$$\left.\frac{\partial(\cdot)}{\partial v_\parallel}\right|_{\mathbf{i}} = \frac{(\cdot)_{\mathbf{i}-2\mathbf{e}_{d_{v_\parallel}}} - 8(\cdot)_{\mathbf{i}-\mathbf{e}_{d_{v_\parallel}}} + 8(\cdot)_{\mathbf{i}+\mathbf{e}_{d_{v_\parallel}}} - (\cdot)_{\mathbf{i}+2\mathbf{e}_{d_{v_\parallel}}}}{12\Delta v_\parallel} + \mathcal{O}\left(\Delta v_\parallel^4\right), \tag{35a}$$

$$\left.\frac{\partial(\cdot)}{\partial \mu}\right|_{\mathbf{i}} = \frac{(\cdot)_{\mathbf{i}-2\mathbf{e}_{d_\mu}} - 8(\cdot)_{\mathbf{i}-\mathbf{e}_{d_\mu}} + 8(\cdot)_{\mathbf{i}+\mathbf{e}_{d_\mu}} - (\cdot)_{\mathbf{i}+2\mathbf{e}_{d_\mu}}}{12\Delta \mu} + \mathcal{O}\left(\Delta \mu^4\right), \tag{35b}$$

$$\left.\frac{\partial^2(\cdot)}{\partial v_\parallel^2}\right|_{\mathbf{i}} = \frac{-(\cdot)_{\mathbf{i}-2\mathbf{e}_{d_{v_\parallel}}} + 16(\cdot)_{\mathbf{i}-\mathbf{e}_{d_{v_\parallel}}} - 30(\cdot)_{\mathbf{i}} + 16(\cdot)_{\mathbf{i}+\mathbf{e}_{d_{v_\parallel}}} - (\cdot)_{\mathbf{i}+2\mathbf{e}_{d_{v_\parallel}}}}{12\Delta v_\parallel^2} + \mathcal{O}\left(\Delta v_\parallel^4\right), \tag{35c}$$

$$\left.\frac{\partial^2(\cdot)}{\partial \mu^2}\right|_{\mathbf{i}} = \frac{-(\cdot)_{\mathbf{i}-2\mathbf{e}_{d_\mu}} + 16(\cdot)_{\mathbf{i}-\mathbf{e}_{d_\mu}} - 30(\cdot)_{\mathbf{i}} + 16(\cdot)_{\mathbf{i}+\mathbf{e}_{d_\mu}} - (\cdot)_{\mathbf{i}+2\mathbf{e}_{d_\mu}}}{12\Delta \mu^2} + \mathcal{O}\left(\Delta \mu^4\right), \tag{35d}$$

$$\left.\frac{\partial^2(\cdot)}{\partial v_\parallel \partial \mu}\right|_{\mathbf{i}} \equiv \left.\frac{\partial}{\partial v_\parallel}\left(\left.\frac{\partial(\cdot)}{\partial \mu}\right|_{\mathbf{i}}\right)\right|_{\mathbf{i}} + \mathcal{O}\left(\Delta v_\parallel^4, \mu^4\right). \tag{35e}$$

Equation (34) requires the cell-centered distribution function, which is obtained from the computational cell-averaged conserved variable $\bar{f}_i$ as

$$f_i = \frac{1}{B_\parallel^*}\left(\bar{f}_i - \frac{h^2}{24}\sum_{d=1}^{D} \left.\frac{\partial^2 f}{\partial \xi_d^2}\right|_{\boldsymbol{\xi}_{\mathbf{i}}}\right) + \mathcal{O}\left(h^4\right), \tag{36}$$

and the $v_\parallel-$ and $\mu-$derivatives of the distribution function are computed as fourth-order centered finite-differences given by Eqs. (35a) and (35b).

B. Rosenbluth Potentials

The advection and diffusion coefficients, $\sigma_{(\cdot)}$ and $\kappa_{(\cdot)}$, are computed from the Rosenbluth potentials, φ and ϱ, using the finite-difference approximations given by Eq. (35). The Rosenbluth potentials are obtained by solving Eq. (7) numerically. Equation (7) represents Poisson equations on an infinite velocity space $\Omega_{\mathbf{v},\infty} \equiv [-\infty,\infty] \times [0,\infty]$; however they are solved on a finite domain $\Omega_{\mathbf{v}} \equiv \left[-v_{\parallel,\max}, v_{\parallel,\max}\right] \times [0,\mu_{\max}]$. The algorithm implemented in COGENT to solve Eq. (7) was previously reported[3] and is summarized in this section. The boundary values can be computed using the Green's function method as

$$\varphi\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = \frac{1}{4\pi}\int\limits_{\Omega_\infty} \frac{f(\mathbf{v}')}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|} d\mathbf{v}', \tag{37a}$$

$$\varrho\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = \frac{1}{8\pi}\int\limits_{\Omega_\infty} \left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right| f(\mathbf{v}')\, d\mathbf{v}', \tag{37b}$$

where $\mathbf{v}_{\partial\Omega_{\mathbf{v}}}$ is the velocity vector at the computational domain boundary $\partial\Omega_{\mathbf{v}}$ and where Ω_∞ is the infinite three-dimensional velocity space. The direct computation of Eq. (37) is very expensive, and an asymptotic method[3] is used. The Green's function is expanded as[68]

$$\frac{1}{|\mathbf{v}-\mathbf{v}'|} = 4\pi\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{1}{2l+1}\frac{v_<^l}{v_>^{l+1}} Y_{lm}^*(\theta',\psi')\, Y_{lm}(\theta,\psi), \tag{38}$$

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,

$$\varphi\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = -\frac{1}{4\pi}\sum_{l=0}^{\infty}\frac{h_l}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}}P_l\left(\cos\theta\right); \quad h_l = \frac{2\pi B}{m}\int\limits_{\Omega_{\mathbf{v}}} f\left(v_{\parallel}, \mu\right) v^l P_l\left(\cos\theta\right) dv_{\parallel} d\mu, \tag{39}$$

where P_l denotes the Legendre polynomials, $v = |\mathbf{v}|$ is the particle speed, and $f\left(v_{\parallel}, \mu : v > \min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|\right) = 0$.

Similar to Eq. (37a), the second Rosenbluth potential ϱ can be expressed on the domain boundary as

$$\varrho\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = \frac{1}{4\pi}\int\limits_{\Omega_{\infty}}\frac{\varphi\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\mathbf{v}'; \tag{40}$$

however, the analysis for φ above does not apply to ϱ because the assumption that $\varphi\left(v_{\parallel}, \mu : v > \min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|\right) = 0$ (corresponding to $f\left(v_{\parallel}, \mu : v > \min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|\right) = 0$) is not true. Thus, Eq. (40) is decomposed as

$$\frac{1}{4\pi}\int\limits_{\Omega_{\infty}}\frac{\varphi\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\mathbf{v}' = \int\limits_{0}^{\min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|} v^2 dv \int\limits_0^{\pi} d\theta \int\limits_0^{2\pi}\frac{\hat{\varphi}\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\psi + \int\limits_{0}^{\min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|} v^2 dv \int\limits_0^{\pi} d\theta \int\limits_0^{2\pi}\frac{\tilde{\varphi}\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\psi, \tag{41}$$

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

$$\tilde{\varphi}\left(\mathbf{v}\right) = -\frac{1}{4\pi}\sum_{l=0}^{\infty}\frac{h_l}{|\mathbf{v}|^{l+1}}P_l\left(\cos\theta\right). \tag{42}$$

Therefore,

$$\int\limits_{0}^{\min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|} v^2 dv \int\limits_0^{\pi} d\theta \int\limits_0^{2\pi}\frac{\hat{\varphi}\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\psi = \sum_{l=0}^{\infty}\frac{g_l}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}}P_l\left(\cos\theta\right); \quad g_l = \frac{2\pi B}{m}\int\limits_{\Omega_{\mathbf{v}}}\hat{\hat{\varphi}}\left(v_{\parallel}, \mu\right) v^l P_l\left(\cos\theta\right) dv_{\parallel} d\mu, \tag{43}$$

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

$$\int\limits_{0}^{\min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|} v^2 dv \int\limits_0^{\pi} d\theta \int\limits_0^{2\pi}\frac{\tilde{\varphi}\left(\mathbf{v}'\right)}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}} - \mathbf{v}'\right|}d\psi = -\sum_{l=0}^{\infty}\frac{h_l P_l\left(\cos\theta\right)}{2l+1}\left(\frac{1}{2\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l-1}} - \frac{1}{2\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}} + \frac{1}{\left(2l-1\right)\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l-1}}\right). \tag{44}$$

These manipulations result in the following expression for the second Rosenbluth potential at the computational domain boundary:

$$\varrho\left(\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right) = -\frac{1}{4\pi}\sum_{l=0}^{\infty}\left[\frac{g_l}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}} - \frac{h_l}{2\left(2l+1\right)}\left(\frac{2l+1}{2l-1}\frac{}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l-1}} - \frac{\left\{\min\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|\right\}^2}{\left|\mathbf{v}_{\partial\Omega_{\mathbf{v}}}\right|^{l+1}}\right)\right]P_l\left(\cos\theta\right). \tag{45}$$

The Poisson equations for the Rosenbluth potentials, Eq. (7), are thus solved with the boundary conditions given by Eqs. (39) and (45). Evaluation of the integral in Eq. (43) introduces cut-cell issues,[3] which can be solved to second order accuracy using linear interpolations. Therefore, the current implementation uses second-order central finite differences to discretize the derivatives in Eq. (7), and the computation of the collision term is, overall, second-order accurate ($q = 2$). The fourth-order accurate implementation of this algorithm is non-trivial and will be investigated in the future. The resulting system of equations is solved using the conjugate gradient method with a structured multigrid preconditioner from the *hypre* library.[69]

C. Time Integration

Equation (22) is a multiscale ODE with a nonstiff Vlasov term and a stiff collision term, i.e.,

$$\left|\lambda\left[\frac{\partial\hat{\mathcal{V}}\left(\bar{\mathbf{f}}\right)}{\partial\bar{\mathbf{f}}}\right]\right| \ll \left|\lambda\left[\frac{\partial\hat{\mathcal{C}}\left(\bar{\mathbf{f}}\right)}{\partial\bar{\mathbf{f}}}\right]\right|, \tag{46}$$

where λ denotes the eigenvalues. Explicit time integration methods are inefficient because the time step is constrained by the eigenvalues of the stiff collision operator. We therefore adopt a semi–implicit time integration method where the Vlasov term $\hat{\mathcal{V}}$ is integrated explicitly in time while the collision term $\hat{\mathcal{C}}$ is integrated implicitly in time. Consequently, the time step of the resulting algorithm is constrained by the eigenvalues of the nonstiff Vlasov term, and significantly larger time steps are possible compared with an explicit time integration method. An alternative is a fully implicit method with unconditional stability; however, to resolve unsteady advective processes accurately, time steps must be comparable to the Vlasov time scales. Thus, implicit integration of the Vlasov term would yield no additional benefits.

High-order multistage, semi–implicit ARK methods[53,56] can be expressed in the Butcher tableaux[70] form as:

$$\left(\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}, \begin{array}{c|c} \tilde{c}_i & \tilde{a}_{ij} \\ \hline & \tilde{b}_j \end{array}; i,j = 1,\cdots,s\right), c_i = \sum_{j=1}^{s} a_{ij}, \tilde{c}_i = \sum_{j=1}^{s}\tilde{a}_{ij}, \tag{47}$$

where a_{ij}, b_j, c_i define the explicit integrator for the nonstiff (Vlasov) term, $\tilde{a}_{ij}, \tilde{b}_j, \tilde{c}_i$ define the implicit integrator for the stiff (collision) term, and s is the number of stages. The coefficients satisfy

$$a_{ij} = 0, j \geq i; \quad \tilde{a}_{ij} = 0, j > i. \tag{48}$$

In addition, the methods implemented are conservative with ESDIRK (explicit first-stage, single-diagonal coefficient), L–stable implicit parts. Thus,

$$b_i = \tilde{b}_i, \forall i; \quad \tilde{a}_{11} = 0; \quad \tilde{a}_{ii} = \gamma, i = 2,\cdots,s, \tag{49}$$

where γ is a constant. A time step of the ARK methods applied to Eq. (22) is expressed as follows:

$$\bar{\mathbf{f}}^{(i)} = \bar{\mathbf{f}}_n + \Delta t\left\{\sum_{j=1}^{i-1} a_{ij}\hat{\mathcal{V}}\left(\bar{\mathbf{f}}^{(j)}\right) + \sum_{j=1}^{i}\tilde{a}_{ij}\hat{\mathcal{C}}\left(\bar{\mathbf{f}}^{(j)}\right)\right\}, \quad i = 1,\cdots,s, \tag{50a}$$

$$\bar{\mathbf{f}}_{n+1} = \bar{\mathbf{f}}_n + \Delta t\sum_{i=1}^{s}\left\{b_i\hat{\mathcal{V}}\left(\bar{\mathbf{f}}^{(i)}\right) + \tilde{b}_i\hat{\mathcal{C}}\left(\bar{\mathbf{f}}^{(i)}\right)\right\}, \tag{50b}$$

where Δt is the time step, n denotes the time level ($\bar{\mathbf{f}}_n = \bar{\mathbf{f}}\left(t_n\right), t_n = t_0 + n\Delta t$), and $\bar{\mathbf{f}}^{(i,j)}$ are the stage solutions. The following ARK methods are implemented in COGENT: a second–order (three–stage) method (ARK 2e),[56] a third-order (four-stage) method (ARK 3),[53] and a fourth-order (six-stage) method (ARK 4).[53]

Equation (50a) represents a nonlinear system of equations that, if $\tilde{a}_{ii} \neq 0$, can be expressed as

$$\mathcal{F}\left(\mathbf{y}\right) \equiv \alpha\mathbf{y} - \hat{\mathcal{C}}\left(\mathbf{y}\right) - \mathbf{r} = 0, \tag{51}$$

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

$$\alpha = \frac{1}{\tilde{a}_{ii}\Delta t}, \quad \mathbf{r} = \frac{1}{\tilde{a}_{ii}\Delta t}\left[\bar{\mathbf{f}}_n + \Delta t\sum_{j=1}^{i-1}\left\{a_{ij}\hat{\mathcal{V}}\left(\bar{\mathbf{f}}^{(j)}\right) + \tilde{a}_{ij}\hat{\mathcal{C}}\left(\bar{\mathbf{f}}^{(j)}\right)\right\}\right]. \tag{52}$$

A Jacobian–Free Newton-Krylov method[63] is used to solve Eq. (51), where the nonlinear equation is solved using Newton's method:[71] Given an initial guess $\mathbf{y}_0 \equiv \bar{\mathbf{f}}_0^{(i)}$, the k-th Newton iteration updates the k-th guess $\mathbf{y}_k$ as

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \mathcal{J}\left(\mathbf{y}_k\right)^{-1}\mathcal{F}\left(\mathbf{y}_k\right); \quad \mathcal{J}\left(\mathbf{y}\right) = \frac{d\mathcal{F}\left(\mathbf{y}\right)}{d\mathbf{y}}, \tag{53}$$

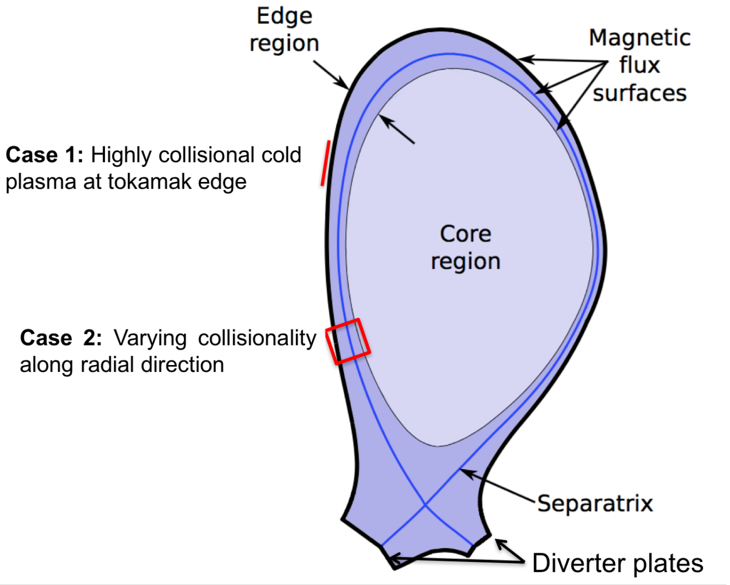

Edge region
Magnetic flux surfaces
Case 1: Highly collisional cold plasma at tokamak edge
Core region
Case 2: Varying collisionality along radial direction
Separatrix
Diverter plates

Figure 1. Illustration of the edge region of the tokamak–type fusion reactor and the correspondence between the cases presented in this paper and the tokamak edge plasma dynamics.

where $\mathcal{J}$ is the Jacobian of $\mathcal{F}$. The exit criterion for the Newton iterations is given by

$$\|\mathcal{F}\left(\mathbf{y}_k\right)\|_2 \leq \max\left(\hat{\epsilon}_r \|\mathcal{F}\left(\mathbf{y}_0\right)\|_2, \hat{\epsilon}_a\right) \text{ or } \|\mathcal{J}^{-1}\mathcal{F}\left(\mathbf{y}_k\right)\|_2 \leq \hat{\epsilon}_s, \tag{54}$$

where $\hat{\epsilon}_a$ and $\hat{\epsilon}_r$ are the absolute and relative tolerances for the Newton solver, and $\hat{\epsilon}_s$ is the step size tolerance. The initial guess for solving (50a) is taken as the solution of the previous stage $\bar{\mathbf{f}}^{(i-1)}$ if $i > 1$ or the current time step $\bar{\mathbf{f}}_n$ if $i = 1$. The Newton update, given by Eq. (53), requires the solution to the linear system of equations,

$$\left[\mathcal{J}\left(\mathbf{y}_k\right)\right]\mathbf{x} = \mathcal{F}\left(\mathbf{y}_k\right), \tag{55}$$

which is solved using the generalized minimum residual (GMRES) method.[72,73] The GMRES solver does not require the matrix $\mathcal{J}$; it only requires the definition of the action of this matrix on a vector. Thus, the Jacobian matrix is not assembled. We use a Jacobian-free approach, where the action of the Jacobian on a given vector is approximated using a directional derivative,[74]

$$\mathcal{J}\left(\mathbf{y}_k\right)\mathbf{x} \approx \frac{1}{\epsilon}\left[\mathcal{F}\left(\mathbf{y}_k + \epsilon\mathbf{x}\right) - \mathcal{F}\left(\mathbf{y}_k\right)\right] = \alpha\mathbf{x} - \frac{1}{\epsilon}\left[\hat{\mathcal{C}}\left(\mathbf{y}_k + \epsilon\mathbf{x}\right) - \hat{\mathcal{C}}\left(\mathbf{y}_k\right)\right], \quad \epsilon = \frac{\sqrt{\epsilon_m\left(1 + \|\mathbf{y}_k\|_2\right)}}{\|\mathbf{x}\|_2}, \tag{56}$$

where ϵ_m is machine round-off error (taken as 10^{-14} in our algorithm). The exit criterion for the GMRES solver is given by

$$\|\mathbf{r}_l\|_2 \leq \max\left(\tilde{\epsilon}_a, \tilde{\epsilon}_r\|\mathbf{r}_0\|_2\right), \quad \mathbf{r}_l = \mathcal{J}\left(\mathbf{y}_k\right)\mathbf{x}_l + \mathcal{F}\left(\mathbf{y}_k\right), \tag{57}$$

where $\mathbf{x}_l$ is the solution at the l-th GMRES iteration, and $\mathbf{r}_0 = \mathcal{F}\left(\mathbf{y}_k\right)$ since $\mathbf{x}_0 = \mathbf{0}$. The implementation in this paper does not include a preconditioner, and development of an efficient preconditioner will be reported in future papers.

IV. Results and Verification

In this section, we test the performance of the semi-implicit ARK methods and compare them with the explicit Runge-Kutta methods in the context of the VFP equation. In particular, we compare the fourth-order, six-stage ARK4 method with the fourth-order, four-stage RK4 method. The test problems are representative of the plasma dynamics at the tokamak edge. Figure 1 illustrates the two-dimensional cross-section of the edge region (marked by the darker shade of blue) of a tokamak–type fusion reactor. The lighter shade of blue is the hot core region. The geometry is defined by the magnetic flux surfaces; the separatrix (shown as a blue line) separates the open and closed magnetic field lines. The diverter plates are shown at the bottom. Although COGENT is capable of simulating complicated geometries with the mapped, multiblock finite-volume algorithm, we consider problems with a Cartesian geometry in this paper; thus, $\mathbf{R} \equiv (x, y)$ in Eq. (1). We consider two cases. The first case is an essentially one-dimensional (in configuration space) slab with uniform collisionality; the parameters are chosen such that the case is representative of the highly collisional cold plasma at the tokamak edge. The second case consists of a two-dimensional slab with the collisionality varying along one dimension. At one end, the collisional mean free paths are comparable to the density and temperature gradient length scales, while at the other end, the collisional mean free paths are much smaller. This variation in the collisionality is similar to the one observed in the tokamak edge region along the radial direction away from the X-point. The electrostatic potential is prescribed for both the cases, and thus, a self-consistent electric field is not computed from the charge densities.

A. Case 1: Uniform Collisionality

The first test problem simulates the plasma dynamics over a two-dimensional slab with uniform collisionality. The dynamics are essentially one-dimensional in the configuration space, simulated using a two-dimensional slab by enforcing zero gradients in the x direction. The configuration space domain is a square of length $L = 1\,\mathrm{m}$, while the velocity domain is taken as $[-3.5\bar{v}, 3.5\bar{v}] \times [0, 10\bar{\mu}]$ where $\bar{v} = 4.377 \times 10^6\,\mathrm{cm/s}$ and $\bar{\mu} = 1.602 \times 10^{-15}\,\mathrm{g\,cm^2\,s^{-2}\,G^{-1}}$ are the reference velocity and magnetic moment. The initial solution is a Maxwellian in the velocity space with uniform density $\rho(x, y, t = 0) = \rho_{\mathrm{ref}}$ and a temperature distribution given by

$$T(x, y, t = 0) \equiv T_0(y) = T_{\mathrm{ref}} \left\{ 1.0 + 0.1 \cos\left(2\pi \frac{y}{L}\right) \right\}. \tag{58}$$

The electrostatic potential is specified throughout the simulation as

$$\phi(x, y, t) = -0.1 \cos\left(2\pi \frac{y}{L}\right), \tag{59}$$

and the electric field is computed as $\mathbf{E} = -\nabla_{\mathbf{R}}\phi$. A constant external magnetic field $B_y = 0.2\,\mathrm{T}, B_z = 2\,\mathrm{T}$ is applied normal to the slab (z-dimension) and along the y dimension, representative of the magnetic field lines at the tokamak edge. The species charge and mass correspond to ionized hydrogen. Periodic boundary conditions are applied along y, while inflow/outflow boundary condition are applied along x. The reference density and temperature are specified as $\rho_{\mathrm{ref}} = 10^{20}\,\mathrm{m}^{-3}$ and $T_{\mathrm{ref}} = 20\,\mathrm{eV}$, respectively. Consequently, the ratio of the particle mean free path length to the characteristic length scale is

$$k_\parallel \lambda = 0.065;\ k_\parallel = \left(\frac{B_y}{B}\right)\left(\frac{2\pi}{L}\right)\ \text{(length scale)},\ \lambda = v_{\mathrm{th}}\tau = \sqrt{\frac{2T}{m}}\tau\ \text{(mean free path)}, \tag{60}$$

where

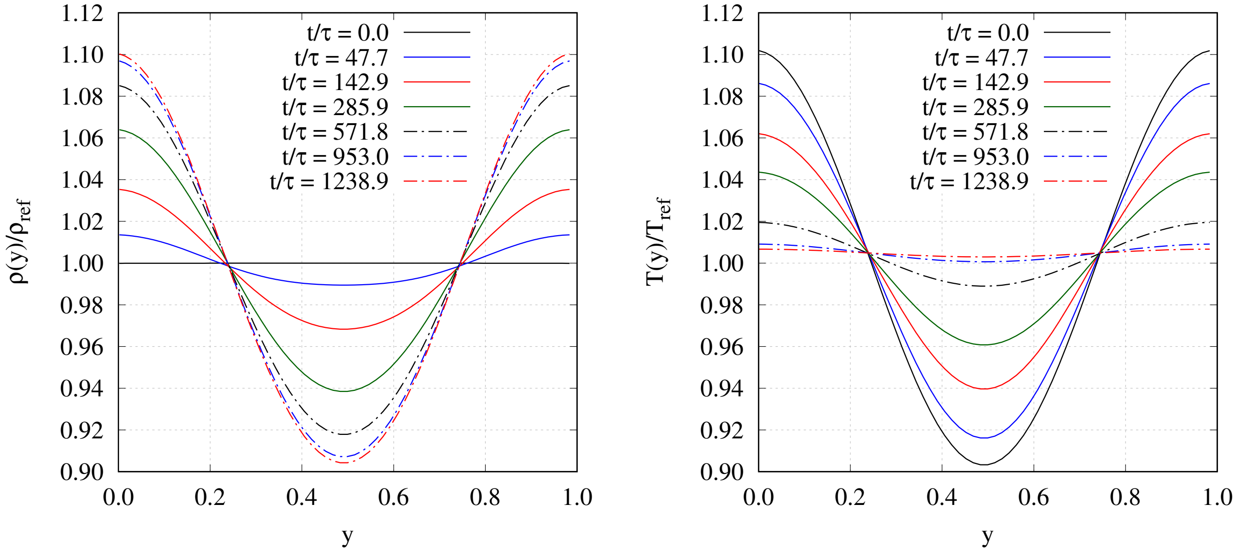
$$\tau = \frac{3\sqrt{mT^3}}{4\sqrt{\pi}\rho e^4 \lambda_c} = 2.4 \times 10^{-6}\,\mathrm{s} \quad (\lambda_c = 11) \tag{61}$$

is the collision time. Thus, the plasma is highly collisional and is representative of the cold edge of a tokamak. In the subsequent discussion, the solutions are obtained on a grid with $6\,(x) \times 64\,(y) \times 72\,\left(v_\parallel\right) \times 48\,(\mu)$ points. The solution times and time step sizes are reported in terms of the time scale defined as

$$t_{\mathrm{ref}} = \frac{L}{\sqrt{T_{\mathrm{ref}}/m_{\mathrm{ref}}}} = 2.3 \times 10^{-5}\,\mathrm{s}, \tag{62}$$

where $m_{\mathrm{ref}} = 1.6726 \times 10^{-24}\,\mathrm{g}$ is the proton mass. Thus, a simulation time or time step size of unity corresponds to a physical time of $2.3 \times 10^{-5}\,\mathrm{s}$.

This case exhibits dynamics at two time scales. The thermal diffusion of the initially sinusoidal temperature to a constant and the evolution of the initially constant density to a sinusoidal variation (to satisfy the prescribed electrostatic potential) occur on the slow transport time scale. At any given time during these



(a) Density $\rho\,(y)$

(b) Temperature $T\,(y)$

Figure 2. Cross-sectional (a) density and (b) temperature at different times. The initially constant density develops a sinusoidal profile consistent with the prescribed electrostatic potential while the initially sinusoidal temperature diffuses to a constant value.

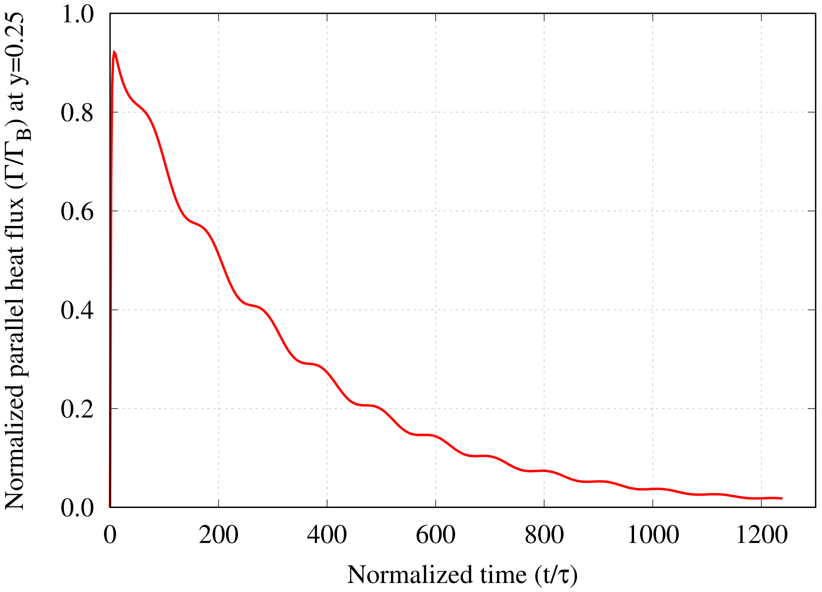


Figure 3. Parallel heat flux as a function of time at $y = 0.25\,\mathrm{m}$. The heat flux rapidly saturates to its "quasi–steady" value consistent with the initial temperature gradient on the collisional time scale. Thereafter, it gradually decays to zero as diffusion drives the temperature towards a uniform profile.

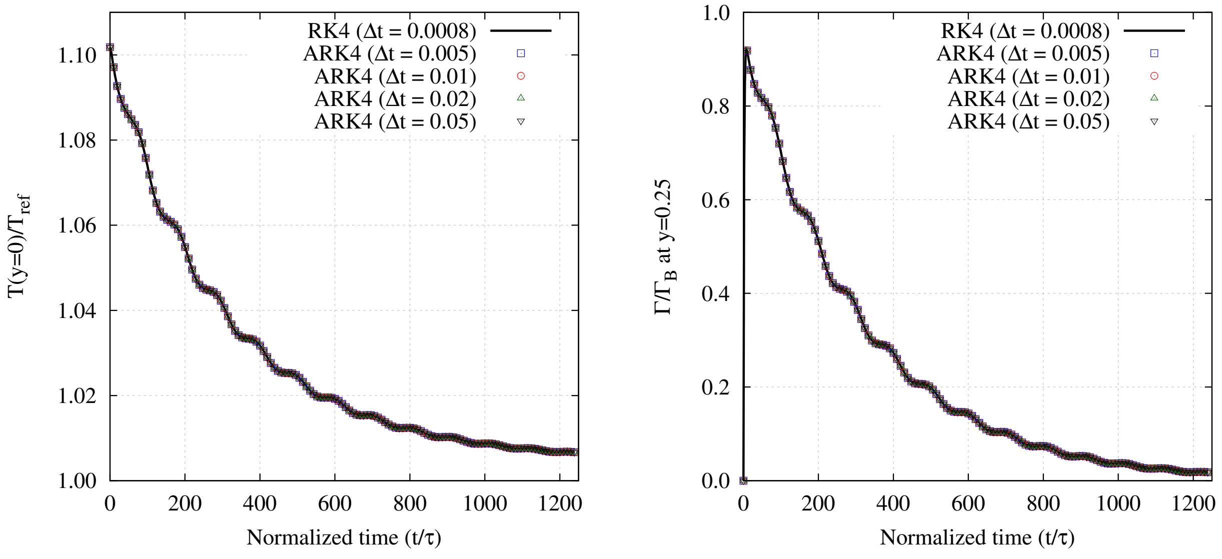
Table 1. Wall times (in seconds) and number of function calls to the right-hand-side functions $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ for the RK4 and ARK4 methods for a final simulation time of 10.0.

Name	Δt	N	$\bar{\sigma}_{\hat{\mathcal{V}}}$	$\bar{\sigma}_{\hat{\mathcal{C}}}$	η (seconds)	$\eta_{\text{min,RK4}}/\eta$	$N_{\hat{\mathcal{V}}}$	$N_{\hat{\mathcal{C}}}$
RK4	0.0008	12500	0.02	0.8	4.66×10^4	1	50000	50000
ARK4	0.005	2000	0.1	4.8	6.24×10^4	0.75	12000	93810
	0.01	1000	0.2	9.6	3.01×10^4	1.55	6000	45665
	0.02	500	0.5	19	1.73×10^4	2.69	3000	25883
	0.05	200	1.1	48	7.05×10^3	6.62	1200	10485

processes, the collisions drive the parallel heat flux at all points in the domain towards its "quasi–steady" value that is consistent with the local temperature gradients at that time. This evolution happens on the collisional time scale that is significantly faster than the advective and transport time scales. The time step of an explicit time integration method, for example RK4, is bounded by the collisional time scale, thus making it inefficient for simulating the advective and transport dynamics. The time step of a semi-implicit method is bounded by the advective time scale for linear stability since the collisional term is integrated implicitly in time. Figure 2 shows the cross-sectional density $\rho(y)$ and temperature $T(y)$ at the initial time and subsequent times until a normalized final time (t/τ) of 1238.9, where t is the simulation time. Figure 3 shows the evolution of the parallel heat flux Γ/Γ_B at $y = 0.25\,\text{m}$ as a function of time, where Γ is computed as

$$\Gamma\left(\mathbf{R}\right) = \int\int \left\{v_\parallel - \overline{v_\parallel}\left(\mathbf{R}\right)\right\} \left(\left\{v_\parallel - \overline{v_\parallel}\left(\mathbf{R}\right)\right\}^2 + \frac{2\mu B}{m}\right) f\left(\mathbf{R}, v_\parallel, \mu\right) dv_\parallel d\mu \tag{63}$$

and Γ_B is the heat flux computed by the Braginskii model.[75] The solutions are obtained using the ARK4 time integration method at a time step size of 0.05.

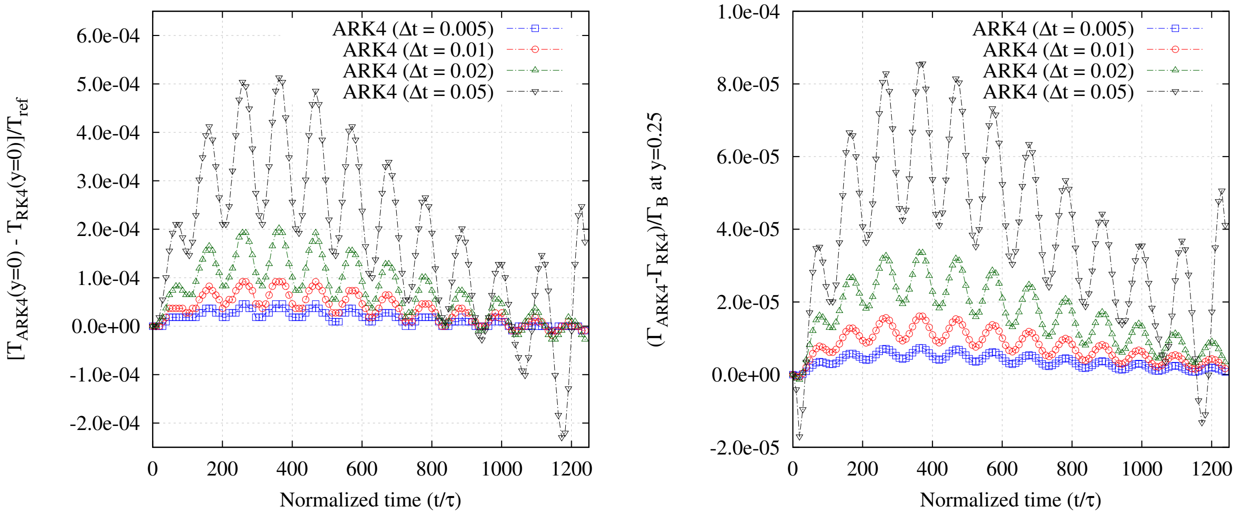


(a) Temperature vs. time

(b) Parallel heat flux vs. time

Figure 4. Temperature and parallel heat flux as a function of time at $y = 0\,\text{m}$ and $y = 0.25\,\text{m}$, respectively: Comparison of the solutions obtained with RK4 and ARK4 methods. A good agreement is observed.

The computational cost of the semi-implicit ARK4 method is compared with that of the explicit RK4 method in Table 1. The wall times and the number of function calls (to the right-hand-side functions $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$) for the ARK4 method with several values of the time step Δt are compared with that of the RK4 method with its largest stable time step. Table 1 shows these metrics for solutions obtained with RK4 and ARK4 at a final simulation time of 10.0 where N is the number of time steps, $\bar{\sigma}_{\hat{\mathcal{V}}}$ and $\bar{\sigma}_{\hat{\mathcal{C}}}$ are the Vlasov and collision



(a) Temperature difference vs. time

(b) Parallel heat flux difference vs. time

Figure 5. The difference in the temperature and parallel heat flux between solutions obtained with the ARK4 method, plotted in Fig. 4, and that obtained with the RK4 method as a function of time: The differences between the explicit and the semi-implicit solutions are relatively small.

CFL numbers defined here as,

$$\bar{\sigma}_{\hat{\mathcal{V}}} = \max\left\{ \frac{\left(\dot{\mathbf{R}}\cdot\hat{r}\right)\Delta t}{\Delta r}, \frac{\left(\dot{\mathbf{R}}\cdot\hat{\theta}\right)\Delta t}{\Delta \theta}, \frac{\dot{v}_{\parallel}\Delta t}{\Delta v_{\parallel}} \right\}, \tag{64}$$

$$\bar{\sigma}_{\hat{\mathcal{C}}} = \max\left\{ \frac{\sigma_{v_\parallel}\Delta t}{\Delta v_\parallel}, \frac{\sigma_{\mu}\Delta t}{\Delta \mu}, \frac{\kappa_{v_\parallel v_\parallel}\Delta t}{\Delta v_\parallel^2}, \frac{\kappa_{\mu\mu}\Delta t}{\Delta \mu^2}, \frac{\kappa_{v_\parallel \mu}\Delta t}{\Delta v_\parallel \Delta \mu} \right\}, \tag{65}$$

η is the wall time, $\eta_{\min,\text{RK4}} = 4.66 \times 10^4$ s is the wall time for the fastest RK4 solution, $N_{\hat{\mathcal{V}}}$ is the number of times the Vlasov operator is called, and $N_{\hat{\mathcal{C}}}$ is the number of times the collision operator is called. The sixth numerical column shows ratio of the wall time of the ARK4 method with a given time step to the wall time of the RK4 method with its largest stable time step; it thus represents the "speedup" achieved by the ARK4 method. The number of function counts for an explicit time integration method is the number of time steps multiplied by the number of stages of the time integration method,

$$N_{\hat{\mathcal{V}}} = N_{\hat{\mathcal{C}}} = N \times s, \tag{66}$$

where s is the number of stages. The ARK methods integrate the Vlasov operator explicitly and the collision operator implicitly, and

$$N_{\hat{\mathcal{V}}} = N \times s; \quad N_{\hat{\mathcal{C}}} = N \times s + N_{\text{Newton}} + N_{\text{GMRES}} \tag{67}$$

because the implicit system is solved using the Jacobian-free Newton–Krylov method as described in Sec. III C (and thus each GMRES iteration involves a function evaluation).

The maximum stable time step for the RK4 method results in a Vlasov CFL number of $\bar{\sigma}_{\hat{\mathcal{V}}} \sim 0.02$ and a collision CFL number of $\bar{\sigma}_{\hat{\mathcal{C}}} \sim 0.8$, indicating that the largest stable time step is of the order of the collisional time scale and significantly smaller than the advective time scale. The ARK4 method can be used with time steps larger than the largest stable time step for RK4. The wall time and the number of function counts for the collision operator are larger for the ARK4 method at a time step size of 0.005 even though the number of time steps is smaller. Thus, at this time step, the additional cost of solving the nonlinear system outweighs the benefits of the larger time step. At larger time steps, the cost of the ARK4 method reduces, and it is faster than the RK4 method (both in terms of the wall time and number of function counts). At the largest stable time step for the ARK4 methods ($\Delta t = 0.05$), the Vlasov CFL is $\bar{\sigma}_{\hat{\mathcal{V}}} \sim 1.1$. Thus the time step is comparable to the advective time scale, and the wall time is approximately six times smaller than the RK4 method. Figure 4 shows the time evolution of the temperature at $y = 0\,\mathrm{m}$ and the parallel heat flux at $y = 0.25\,\mathrm{m}$ for the solutions compared in Table 1, while Fig. 5 plots the difference between the solutions obtained with the ARK4 method at various time steps and that obtained with the RK4 method at its largest stable time step. Good agreement is observed between the solutions obtained with the ARK4 at the larger time steps and that obtained with the RK4 with the collisional time-scale-constrained time step.

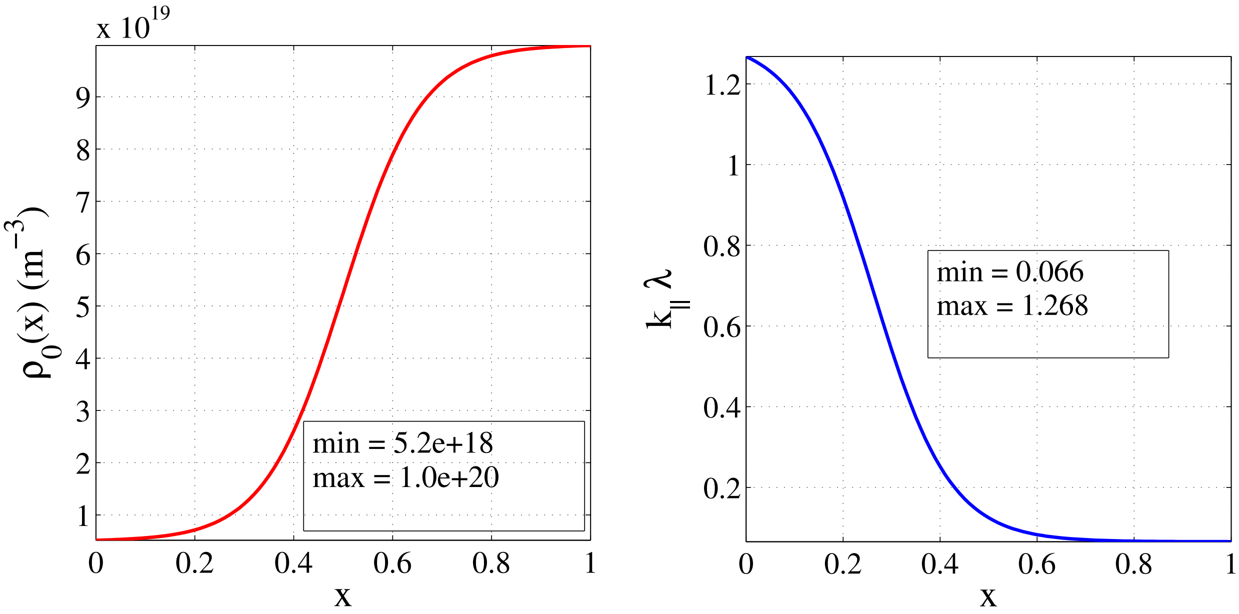
B. Case 2: Varying Collisionality

The second test problem simulates the dynamics over a two-dimensional slab with strongly varying collisionality along one dimension. This is representative of a small patch in the toroidal plane of the edge region of the plasma. The setup is identical to the previous case except that the initial density is specified as

$$\rho\left(x,y,t=0\right) \equiv \rho_0\left(x\right) = \rho_{\rm ref}\left\{1-\delta+\delta\tanh\left(2\pi x-\pi\right)\right\}, \tag{68}$$

where $\delta = 0.475$. Figure 6 shows the density variation along x and the resulting variation of $k_\parallel\lambda$ (ratio of the mean free path to the characteristic length scale). Near $x = 0$, the plasma is weakly collisional with the mean free path comparable to the length scale while the plasma is strongly collisional as $x \rightarrow 1$ with the mean free path much smaller than the length scale. Thus, the collisional times scales are comparable to Vlasov time scales as $x \rightarrow 0$ but are much smaller as $x \rightarrow 1$. In the subsequent discussions, the numerical solutions are obtained on a grid with $32\left(x\right) \times 32\left(y\right) \times 72\left(v_\parallel\right) \times 48\left(\mu\right)$ points, and 1 simulation time unit corresponds to a physical time of $2.3 \times 10^{-5}\,\mathrm{s}$, computed using Eq. (62).

Downloaded by Debojyoti Ghosh on June 26, 2017 | http://arc.aiaa.org | DOI: 10.2514/6.2017-4016

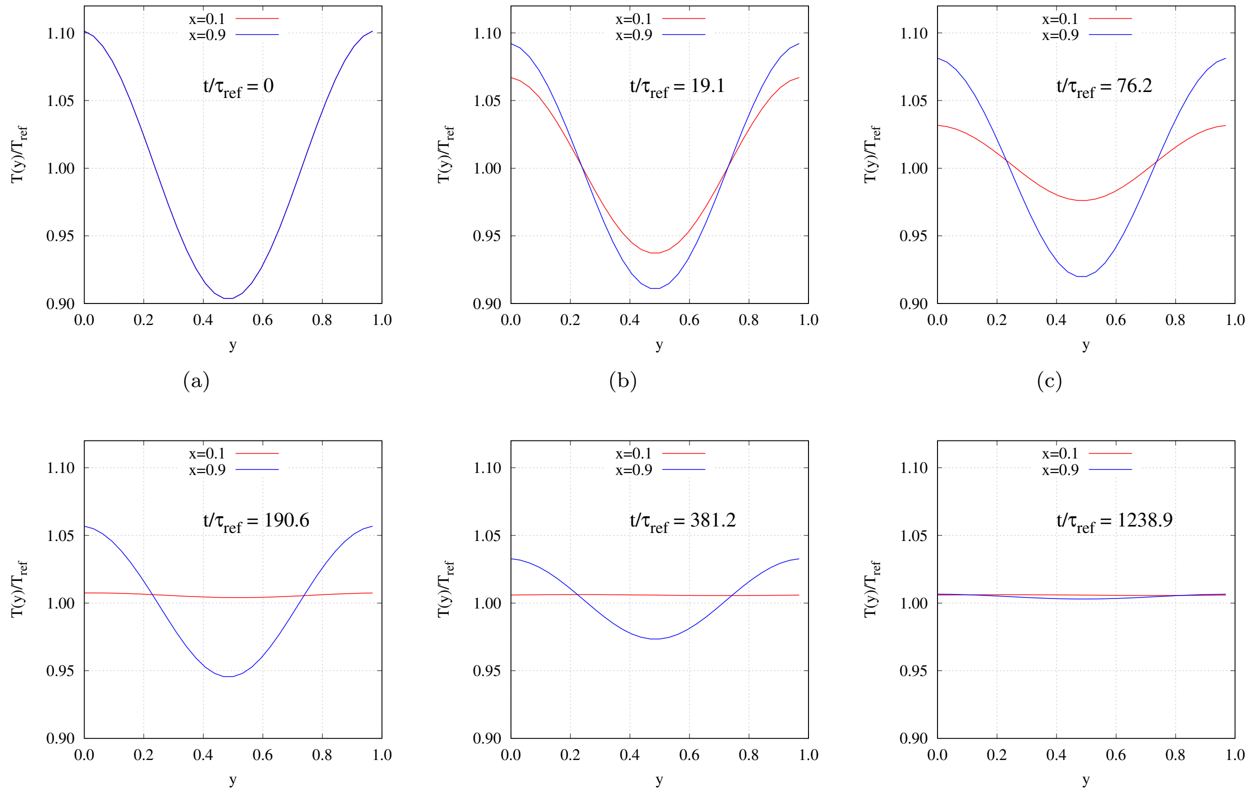


(a) Initial density $\rho_0\left(x\right)$

(b) Ratio of mean free path to length scales

Figure 6. The initial density variation for case 2 and the consequent variation in collisionality along x. The plasma is highly collisional for $x \rightarrow 1$ while it is weakly collisional for $x \rightarrow 0$.

The dynamics of this case are similar to the previous case: There are two time scales. Thermal equilibration of the initially sinusoidal (along y) temperature to a constant occurs on the transport time scale, while collisions drive the parallel heat flux to its "quasi-steady" value on the collisional time scale. The collisions are weak as $x \rightarrow 0$ and the transport time scale is shorter, while strong collisions as $x \rightarrow 1$ results in a longer transport time scale. Figure 7 shows the cross-sectional temperature $T\left(y\right)$ at $x = 0.1, 0.9$ for various times. The solution is obtained with the ARK4 method with a time step of 0.1. The reference collision time $\tau_{\rm ref} = 2.4 \times 10^{-6}\,\mathrm{s}$ is the collision time computed by Eq. (61) at $x = 1$. Figure 8 shows the normalized parallel heat flux Γ/Γ_B as a function of the normalized time $t/\tau_{\rm ref}$. We observe faster equilibration at $x = 0.1$ where the collisions are weak than at $x = 0.9$ where the collisions are strong.



(a) (b) (c)

(d) (e) (f)

Figure 7. Normalized cross-sectional temperature $T(y)/T_{\rm ref}$ at $x = 0.1, 0.9$ at various times. The temperature equilibrates faster at $x = 0.1$ where the plasma is weakly collisional compared with $x = 0.9$ where the plasma is strongly collisional.

The computational costs of the ARK4 and RK4 methods are compared in Table 2. The columns are the same as Table 1 and are reported for solutions at a simulation time of 4.0. The largest stable time step for the RK4 method corresponds to a Vlasov and collisional CFL numbers of $\bar{\sigma}_{\hat{\mathcal{V}}} \sim 0.01, \bar{\sigma}_{\hat{\mathcal{C}}} \sim 0.8$, respectively and shows that the collisional time scale severely constrains the explicit time integrator. The ratio of the wall time for the fastest RK4 solution to the wall time for the ARK4 solutions $(\eta_{\rm min,RK4})/\eta$ (seventh column) is the speedup. The largest stable time step for the ARK4 method and corresponds to a Vlasov CFL of ~ 1.1. The wall time is ten times smaller than that of the RK4 method with its largest time step. Figure 9 shows the evolution of the temperature at $y = 0$ and $x = 0.1, 0.9$ for the cases in Table 2, while Fig. 10 plots the difference between the solutions obtained with the ARK4 method at various time steps and that obtained with the RK4 method at its largest stable time step. A good agreement is observed between the solution obtained by the RK4 with the collisional-time-scale constrained time step and the ARK4 methods with time steps comparable to the Vlasov time scale.

V. Conclusions

The gyrokinetic VFP equations describe the dynamics of tokamak edge plasma, which is characterized by disparate temporal scales. Near the hot core, the mean free paths are much smaller than the characteristic length scales, defined as the temperature and density gradient length scales. In the intermediate region near the separatrix, the mean free paths are comparable to the characteristic length scales, while at the cold edge, the mean free paths are much smaller. The collisional time scale varies from being much larger than the Vlasov time scale near the core to being significantly smaller near the edge, thus rendering the governing equations stiff. This paper presents a semi-implicit algorithm for the VFP equations where the collision term is integrated implicitly in time while the Vlasov term is integrated explicitly. The algorithm is developed

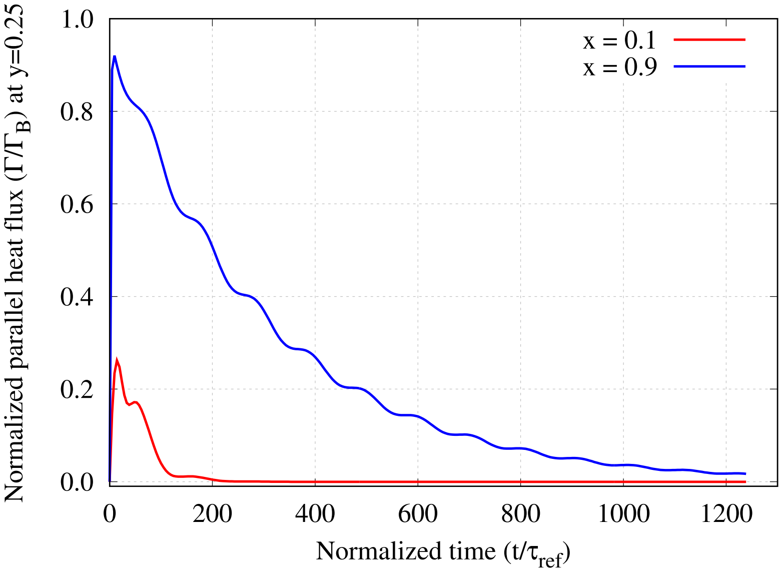


Figure 8. Parallel heat flux as a function of time at $x = 0.1, 0.9$ and $y = 0.25\,\mathrm{m}$. The heat flux decays to zero faster at $x = 0.1$ where the plasma is weakly collisional.

Table 2. Wall times (in seconds) and number of function calls to the right-hand-side functions $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ for the RK4 and ARK4 methods for a final simulation time of 4.0.

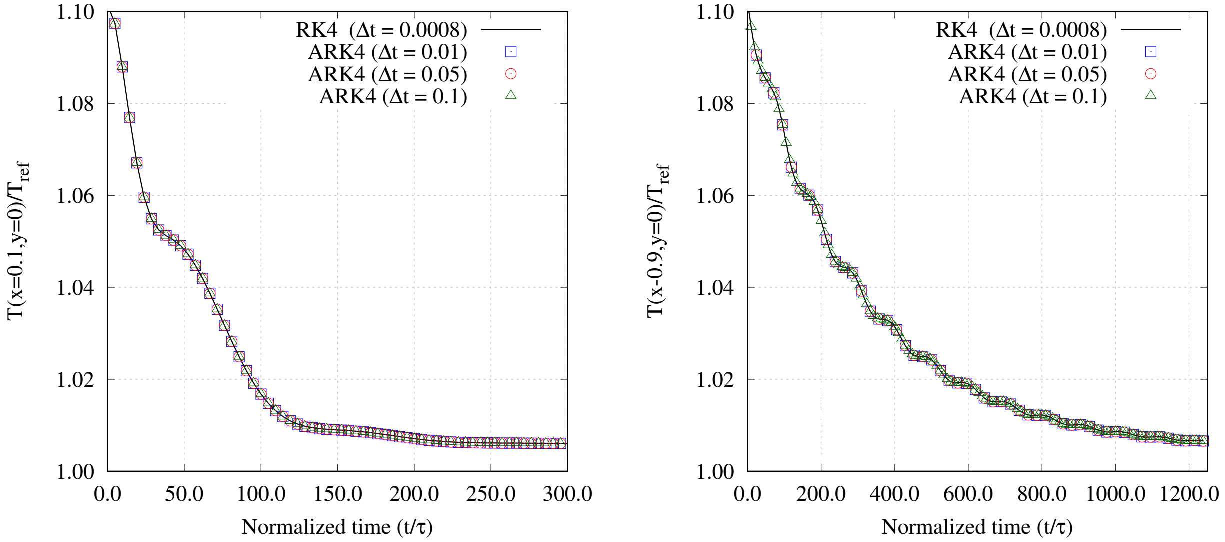
Name	Δt	N	$\bar{\sigma}_{\hat{\mathcal{V}}}$	$\bar{\sigma}_{\hat{\mathcal{C}}}$	η (seconds)	$\eta_{\min,\mathrm{RK4}}/\eta$	$N_{\hat{\mathcal{V}}}$	$N_{\hat{\mathcal{C}}}$
RK4	0.0008	5000	0.01	0.8	3.78×10^4	1	20000	20000
ARK4	0.01	400	0.1	9.6	2.96×10^4	0.78	2400	22905
	0.05	80	0.6	48	7.42×10^3	0.20	480	5744
	0.10	40	1.1	96	3.72×10^3	0.10	240	2950

within COGENT, a high-order finite-volume code for tokamak edge simulations that solve the gyrokinetic VFP equations on mapped multiblock grids. High-order, multistage ARK methods are implemented for the semi-implicit time integration of the governing equations, and their performance is compared with that of the explicit RK methods. The test problems simulate parallel ion heat transport and are representative of the tokamak edge region. The ARK methods are observed to achieve their theoretical orders of convergence. Wall times of the ARK4 method with time steps comparable to the Vlasov time scale are compared with those of the RK4 method with its largest stable time step, and a significant speedup is observed. The solutions obtained with the semi-implicit approach agree well with those obtained with explicit time integration.

The results presented in this paper do not use any preconditioning while solving for the implicit stages in the ARK time integrators. The implementation of an efficient preconditioner for the Fokker–Planck collision term is an area of current research. Future work will aim to incorporate other numerically stiff terms in to the IMEX formulation, such as electrostatic plasma waves, ion acoustic modes, and parallel electron and ion transport.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program.



(a) $x = 0.1$

(b) $x = 0.9$

Figure 9. Comparison of the temperature $T(x,y)/T_{\rm ref}$ evolution at $x = \{0.1, 0.9\}, y = 0$ for cases reported in Table 2. A good agreement is observed between the explicit (RK4) and the semi-implicit (ARK4) solutions.

References

[1] Cohen, R. H. and Xu, X. Q., "Progress in Kinetic Simulation of Edge Plasmas," *Contributions to Plasma Physics*, Vol. 48, No. 1-3, 2008, pp. 212–223.

[2] Dorf, M. A., Cohen, R. H., Dorr, M., Rognlien, T., Hittinger, J., Compton, J., Colella, P., Martin, D., and McCorquodale, P., "Simulation of neoclassical transport with the continuum gyrokinetic code COGENT," *Physics of Plasmas*, Vol. 20, No. 1, 2013.

[3] Dorf, M. A., Cohen, R. H., Dorr, M., Hittinger, J., and Rognlien, T. D., "Progress with the COGENT Edge Kinetic Code: Implementing the Fokker-Planck Collision Operator," *Contributions to Plasma Physics*, Vol. 54, No. 4-6, 2014, pp. 517–523.

[4] Thomas, A., Tzoufras, M., Robinson, A., Kingham, R., Ridgers, C., Sherlock, M., and Bell, A., "A review of Vlasov–Fokker–Planck numerical modeling of inertial confinement fusion plasma," *Journal of Computational Physics*, Vol. 231, No. 3, 2012, pp. 1051–1079, Special Issue: Computational Plasma Physics.

[5] Lee, W. W., "Gyrokinetic approach in particle simulation," *Physics of Fluids*, Vol. 26, No. 2, 1983, pp. 556–562.

[6] Chang, C. S. and Ku, S., "Spontaneous rotation sources in a quiescent tokamak edge plasma," *Physics of Plasmas*, Vol. 15, No. 6, 2008.

[7] Heikkinen, J., Kiviniemi, T., Kurki-Suonio, T., Peeters, A., and Sipilä, S., "Particle Simulation of the Neoclassical Plasmas," *Journal of Computational Physics*, Vol. 173, No. 2, 2001, pp. 527—548.

[8] Heikkinen, J. A., Henriksson, S., Janhunen, S., Kiviniemi, T. P., and 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*, Vol. 46, No. 7-9, 2006, pp. 490–495.

[9] Chen, G., Chacón, L., and Barnes, D., "An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm," *Journal of Computational Physics*, Vol. 230, No. 18, 2011, pp. 7018–7036.

[10] Chen, G. and Chacón, L., "An energy- and charge-conserving, nonlinearly implicit, electromagnetic 1D-3V Vlasov–Darwin particle-in-cell algorithm," *Computer Physics Communications*, Vol. 185, No. 10, 2014, pp. 2391–2402.

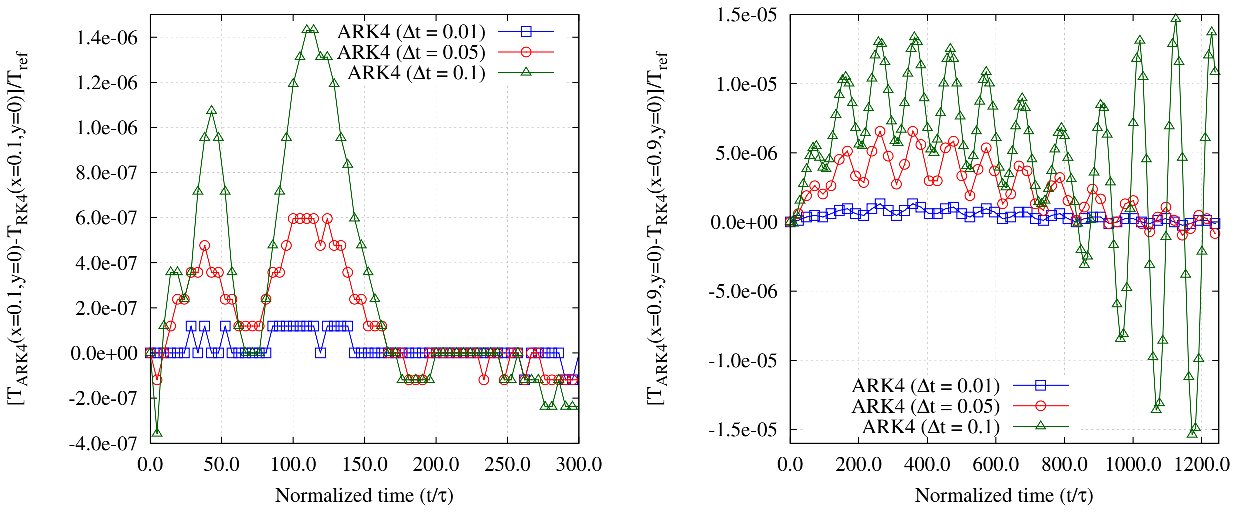
[11] Chen, G. and Chacón, L., "A multi-dimensional, energy- and charge-conserving, nonlinearly implicit, electromagnetic Vlasov–Darwin particle-in-cell algorithm," *Computer Physics Communications*, Vol. 197, 2015, pp. 73–87.

[12] Grandgirard, V., Brunetti, M., Bertrand, P., Besse, N., Garbet, X., Ghendrih, P., Manfredi, G., Sarazin, Y., Sauter, O., Sonnendrücker, E., Vaclavik, J., and Villard, L., "A drift-kinetic semi-Lagrangian 4D code for ion turbulence simulation," *Journal of Computational Physics*, Vol. 217, No. 2, 2006, pp. 395–423.

[13] Idomura, Y., Ida, M., and Tokuda, S., "Conservative gyrokinetic Vlasov simulation," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 13, No. 1, 2008, pp. 227–233, Vlasovia 2006: The Second International Workshop on the Theory and Applications of the Vlasov Equation.

[14] Scott, B., "Gyrokinetic study of the edge shear layer," *Plasma Physics and Controlled Fusion*, Vol. 48, No. 5A, 2006, pp. A387.

[15] 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., and Umansky, M., "Edge gyrokinetic theory and continuum simulations," *Nuclear Fusion*, Vol. 47, No. 8, 2007, pp. 809.



(a) $x = 0.1$

(b) $x = 0.9$

Figure 10. The difference in the cross-sectional temperature at $x = \{0.1, 0.9\}$, $y = 0$ between solutions obtained with ARK4, plotted in Fig. 9, and that obtained with RK4 as a function of time: The differences between the explicit and the semi-implicit solutions are relatively small.

[16] Xu, X. Q., Xiong, Z., Gao, Z., Nevins, W. M., and McKee, G. R., "TEMPEST Simulations of Collisionless Damping of the Geodesic-Acoustic Mode in Edge-Plasma Pedestals," *Phys. Rev. Lett.*, Vol. 100, May 2008, pp. 215001.

[17] Xu, X., Bodi, K., Cohen, R., Krasheninnikov, S., and Rognlien, T., "TEMPEST simulations of the plasma transport in a single-null tokamak geometry," *Nuclear Fusion*, Vol. 50, No. 6, 2010, pp. 064003.

[18] Filbet, F., Sonnendrücker, E., and Bertrand, P., "Conservative Numerical Schemes for the Vlasov Equation," *Journal of Computational Physics*, Vol. 172, No. 1, 2001, pp. 166–187.

[19] Hahm, T. S., "Nonlinear gyrokinetic equations for turbulence in core transport barriers," *Physics of Plasmas*, Vol. 3, No. 12, 1996, pp. 4658–4664.

[20] Rosenbluth, M. N., MacDonald, W. M., and Judd, D. L., "Fokker-Planck Equation for an Inverse-Square Force," *The Physical Review*, Vol. 107, Jul 1957, pp. 1–6.

[21] Taitano, W., Chacón, L., Simakov, A., and 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*, Vol. 297, 2015, pp. 357–380.

[22] Landau, L. D., "The kinetic equation in the case of Coulomb interaction," *Zh. Eksper. i Teoret. Fiz.*, Vol. 7, No. 2, 1937, pp. 203–209.

[23] Chang, J. and Cooper, G., "A practical difference scheme for Fokker-Planck equations," *Journal of Computational Physics*, Vol. 6, No. 1, 1970, pp. 1–16.

[24] Kho, T. H., "Relaxation of a system of charged particles," *Physics Review A*, Vol. 32, Jul 1985, pp. 666–669.

[25] Larsen, E., Levermore, C., Pomraning, G., and Sanderson, J., "Discretization methods for one-dimensional Fokker-Planck operators," *Journal of Computational Physics*, Vol. 61, No. 3, 1985, pp. 359 – 390.

[26] Epperlein, E., "Implicit and Conservative Difference Scheme for the Fokker-Planck Equation," *Journal of Computational Physics*, Vol. 112, No. 2, 1994, pp. 291 – 297.

[27] Lemou, M. and Mieussens, L., "Implicit Schemes for the Fokker–Planck–Landau Equation," *SIAM Journal on Scientific Computing*, Vol. 27, No. 3, 2005, pp. 809–830.

[28] Buet, C. and Cordier, S., "Conservative and Entropy Decaying Numerical Scheme for the Isotropic Fokker–Planck–Landau Equation," *Journal of Computational Physics*, Vol. 145, No. 1, 1998, pp. 228–245.

[29] Berezin, Y., Khudick, V., and Pekker, M., "Conservative finite-difference schemes for the Fokker-Planck equation not violating the law of an increasing entropy," *Journal of Computational Physics*, Vol. 69, No. 1, 1987, pp. 163–174.

[30] Filbet, F. and Pareschi, L., "A Numerical Method for the Accurate Solution of the Fokker–Planck–Landau Equation in the Nonhomogeneous Case," *Journal of Computational Physics*, Vol. 179, No. 1, 2002, pp. 1–26.

[31] Buet, C., Cordier, S., Degond, P., and Lemou, M., "Fast Algorithms for Numerical, Conservative, and Entropy Approximations of the Fokker–Planck–Landau Equation," *Journal of Computational Physics*, Vol. 133, No. 2, 1997, pp. 310–322.

[32] Banks, J. W., Brunner, S., Berger, R. L., and Tran, T. M., "Vlasov simulations of electron-ion collision effects on damping of electron plasma waves," *Physics of Plasmas*, Vol. 23, No. 3, 2016.

[33] Brunner, S., Tran, T., and 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, October 2010.

[34] James, R., "The solution of Poisson's equation for isolated source distributions," *Journal of Computational Physics*, Vol. 25, No. 2, 1977, pp. 71–93.

[35] Belli, E. A. and Candy, J., "Full linearized Fokker–Planck collisions in neoclassical transport simulations," *Plasma Physics and Controlled Fusion*, Vol. 54, No. 1, 2012, pp. 015015.

[36] McCoy, M., Mirin, A., and Killeen, J., "FPPAC: A two-dimensional multispecies nonlinear Fokker-Planck package," *Computer Physics Communications*, Vol. 24, No. 1, 1981, pp. 37–61.

[37] Landreman, M. and Ernst, D. R., "New velocity-space discretization for continuum kinetic calculations and Fokker–Planck collisions," *Journal of Computational Physics*, Vol. 243, 2013, pp. 130–150.

[38] Pataki, A. and Greengard, L., "Fast elliptic solvers in cylindrical coordinates and the Coulomb collision operator," *Journal of Computational Physics*, Vol. 230, No. 21, 2011, pp. 7840–7852.

[39] Larroche, O., "Kinetic simulation of a plasma collision experiment," *Physics of Fluids B*, Vol. 5, No. 8, 1993, pp. 2816–2840.

[40] Chacón, L., Barnes, D. C., Knoll, D. A., and Miley, G. H., "An Implicit Energy-Conservative 2D Fokker–Planck Algorithm: II. Jacobian-Free Newton–Krylov Solver," *Journal of Computational Physics*, Vol. 157, No. 2, 2000, pp. 654–682.

[41] Chacón, L., Barnes, D. C., Knoll, D. A., and Miley, G. H., "An Implicit Energy-Conservative 2D Fokker–Planck Algorithm: I. Difference Scheme," *Journal of Computational Physics*, Vol. 157, No. 2, 2000, pp. 618–653.

[42] Lemou, M., "Multipole expansions for the Fokker-Planck-Landau operator," *Numerische Mathematik*, Vol. 78, No. 4, 1998, pp. 597–618.

[43] Larroche, O., "Kinetic simulations of fuel ion transport in ICF target implosions," *The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics*, Vol. 27, No. 2, 2003, pp. 131–146.

[44] Lemou, M. and Mieussens, L., "Fast implicit schemes for the Fokker–Planck–Landau equation," *Comptes Rendus Mathematique*, Vol. 338, No. 10, 2004, pp. 809–814.

[45] Casanova, M., Larroche, O., and Matte, J.-P., "Kinetic simulation of a collisional shock wave in a plasma," *Phys. Rev. Lett.*, Vol. 67, Oct 1991, pp. 2143–2146.

[46] Taitano, W., Chacón, L., and Simakov, A., "An adaptive, conservative 0D–2V multispecies Rosenbluth–Fokker–Planck solver for arbitrarily disparate mass and temperature regimes," *Journal of Computational Physics*, Vol. 318, 2016, pp. 391–420.

[47] Epperlein, E. M., Rickard, G. J., and Bell, A. R., "Two-Dimensional Nonlocal Electron Transport in Laser-Produced Plasmas," *Physical Review Letters*, Vol. 61, Nov 1988, pp. 2453–2456.

[48] Mousseau, V. and Knoll, D., "Fully Implicit Kinetic Solution of Collisional Plasmas," *Journal of Computational Physics*, Vol. 136, No. 2, 1997, pp. 308–323.

[49] Kingham, R. and Bell, A., "An implicit Vlasov–Fokker–Planck code to model non-local electron transport in 2–D with magnetic fields," *Journal of Computational Physics*, Vol. 194, No. 1, 2004, pp. 1–34.

[50] Thomas, A. G. R., Kingham, R. J., and Ridgers, C. P., "Rapid self-magnetization of laser speckles in plasmas by nonlinear anisotropic instability," *New Journal of Physics*, Vol. 11, No. 3, 2009, pp. 033001.

[51] Ascher, U. M., Ruuth, S. J., and Spiteri, R. J., "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations," *Applied Numerical Mathematics*, Vol. 25, No. 2-3, 1997, pp. 151–167.

[52] Pareschi, L. and Russo, G., "Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation," *Journal of Scientific Computing*, Vol. 25, No. 1-2, 2005, pp. 129–155.

[53] Kennedy, C. A. and Carpenter, M. H., "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Applied Numerical Mathematics*, Vol. 44, No. 1-2, 2003, pp. 139–181.

[54] Durran, D. R. and Blossey, P. N., "Implicit–Explicit Multistep Methods for Fast-Wave–Slow-Wave Problems," *Monthly Weather Review*, Vol. 140, No. 4, April 2012, pp. 1307–1325.

[55] Giraldo, F. X., Restelli, M., and Läuter, M., "Semi-implicit formulations of the Navier-Stokes equations: Application to nonhydrostatic atmospheric modeling," *SIAM Journal on Scientific Computing*, Vol. 32, No. 6, 2010, pp. 3394–3425.

[56] Giraldo, F. X., Kelly, J. F., and Constantinescu, E., "Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA)," *SIAM Journal on Scientific Computing*, Vol. 35, No. 5, 2013, pp. B1162–B1194.

[57] Ghosh, D. and Constantinescu, E. M., "Semi-implicit Time Integration of Atmospheric Flows with Characteristic-Based Flux Partitioning," *SIAM Journal on Scientific Computing*, Vol. 38, No. 3, 2016, pp. A1848–A1875.

[58] Cheng, C. and Knorr, G., "The integration of the Vlasov equation in configuration space," *Journal of Computational Physics*, Vol. 22, No. 3, 1976, pp. 330–351.

[59] Candy, J. and Waltz, R. E., "Anomalous Transport Scaling in the DIII–D Tokamak Matched by Supercomputer Simulation," *Phys. Rev. Lett.*, Vol. 91, Jul 2003, pp. 045001.

[60] Idomura, Y., Ida, M., Kano, T., Aiba, N., and Tokuda, S., "Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation," *Computer Physics Communications*, Vol. 179, No. 6, 2008, pp. 391–403.

[61] Maeyama, S., Ishizawa, A., Watanabe, T.-H., Nakajima, N., Tsuji-Iio, S., and Tsutsui, H., "A hybrid method of semi-Lagrangian and additive semi-implicit Runge–Kutta schemes for gyrokinetic Vlasov simulations," *Computer Physics Communications*, Vol. 183, No. 9, 2012, pp. 1986–1992.

[62] Dorf, M. A., Cohen, R. H., Compton, J. C., Dorr, M., Rognlien, T. D., Angus, J., Krasheninnikov, S., Colella, P., Martin, D., and McCorquodale, P., "Progress with the COGENT Edge Kinetic Code: Collision Operator Options," *Contributions to Plasma Physics*, Vol. 52, No. 5-6, 2012, pp. 518–522.

[63] Knoll, D. and Keyes, D., "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397.

[64] Cohen, R. H., Dorf, M., and Dorr, M., "Reduced Electron Models for Edge Simulation," *Contributions to Plasma Physics*, Vol. 52, No. 5-6, 2012, pp. 529–533.

[65] Colella, P., Dorr, M., Hittinger, J., and Martin, D., "High–order, finite–volume methods in mapped coordinates," *Journal of Computational Physics*, Vol. 230, No. 8, 2011, pp. 2952–2976.

[66] McCorquodale, P., Dorr, M., Hittinger, J., and Colella, P., "High–order finite–volume methods for hyperbolic conservation laws on mapped multiblock grids," *Journal of Computational Physics*, Vol. 288, 2015, pp. 181–195.

[67] Jiang, G.-S. and Shu, C.-W., “Efficient implementation of weighted ENO schemes,” *Journal of Computational Physics*, Vol. 126, No. 1, 1996, pp. 202–228.

[68] Jackson, J. D., *Classical electrodynamics*, Wiley, New York, NY, 3rd ed., 1999.

[69] Falgout, R. D. and Yang, U. M., *hypre: A Library of High Performance Preconditioners*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 632–641.

[70] Butcher, J., *Numerical Methods for Ordinary Differential Equations*, Wiley, 2003.

[71] Dennis, J. and Schnabel, R., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial and Applied Mathematics, 1996.

[72] Saad, Y., *Iterative Methods for Sparse Linear Systems: Second Edition*, Society for Industrial and Applied Mathematics, 2003.

[73] Saad, Y. and Schultz, M. H., “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[74] Pernice, M. and Walker, H. F., “NITSOL: A Newton Iterative Solver for Nonlinear Systems,” *SIAM Journal on Scientific Computing*, Vol. 19, No. 1, 1998, pp. 302–318.

[75] Braginskii, S. I., “Transport Processes in a Plasma,” *Reviews of Plasma Physics*, Vol. 1, 1965, pp. 205.