

UCRL-JRNL-230192



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A High-Order Finite-Volume Algorithm for Fokker-Planck Collisions in Magnetized Plasmas

Z. Xiong, R. H. Cohen, T. D. Rognlien, X. Q. Xu

April 23, 2007

Journal of Computational Physics

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

A high-order finite-volume algorithm for Fokker-Planck collisions in magnetized plasmas

Z. Xiong, R. H. Cohen, T. D. Rognlien, X. Q. Xu *

*Lawrence Livermore National Laboratory, L-630, P. O. Box 808,
Livermore, California 94550, USA*

Abstract

A high-order finite volume algorithm is developed for the Fokker-Planck Operator (FPO) describing Coulomb collisions in strongly magnetized plasmas. The algorithm is based on a general fourth-order reconstruction scheme for a unstructured grid in the velocity space spanned by parallel velocity and magnetic moment. The method provides density conservation and high-order-accurate evaluation of the FPO independent of the choice of the velocity coordinates. As an example, a linearized FPO in constant-of-motion coordinates, i.e. the total energy and the magnetic moment, is developed using the present algorithm combined with a cut-cell merging procedure. Numerical tests include the Spitzer thermalization problem and the return to isotropy for distributions initialized with velocity space loss cones. Utilization of the method for a nonlinear FPO is straightforward but requires evaluation of the Rosenbluth potentials.

Key words: Fokker-Planck collisions, high order scheme, finite volume, constants-of-motion coordinates

PACS:

1 Introduction

The differential Fokker-Planck Operator (FPO) describes the particle collisions in a fully ionized plasma through electrostatic Coulomb fields. Collisions are ubiquitous, and applications of the FPO range from plasma physics, magnetic and inertial confinement fusion, industrial material processing, and

* Corresponding author. Tel.: +925-423-7578; fax: +925-423-3484. E-mail addresses: xxu@llnl.gov (X.Q.Xu)

astrophysics. Although formally a function of Cartesian velocity (v_x, v_y, v_z) , for highly magnetized plasma, the FPO is traditionally expressed in spherical coordinates (v, θ, ϕ) , where v is the particle speed, θ is the pitch angle and ϕ the gyro-angle. For strong magnetic fields, the characteristic time scale of many physical phenomena can be much longer than the gyro-period, and the characteristic length scale much larger than the gyro-radius; for such phenomena, the distribution function and hence the FPO become independent of gyro-angle ϕ . The θ dependence may be expressed by a series expansion using Legendre polynomials $L_n(\cos \theta)$, and the FPO is obtained by solving a series of one-dimensional equations in v [1] [2]. Based on this approach, Chang and Cooper [3] developed a finite difference scheme which conserves particle number density. It was further extended to conserve particle energy by Epperlein [4]. More recently, a spectral collocation method was also used to solve the v equation by Khabibrakhmanov and Khazanov [5].

Though convenient for the FPO itself, the (v, θ) coordinates typically may not be suitable for simulating spatially inhomogeneous plasmas based on full kinetic equations. In such cases, the choice of velocity coordinates is often dictated not by the FPO but by the spatial advection terms (parallel streaming and perpendicular drifts). Hence the FPO needs to be evaluated in the chosen coordinates and its discretization becomes purely two dimensional. Chacon et al (2000) proposed a tensor formalism for the two dimensional Fokker-Planck operator and studied the finite difference scheme in cylindrical coordinates for improved energy conservation. Other algorithms dealing with non-isotropic, multidimensional FPOs use Cartesian velocity coordinates directly, see, e.g.[7] [8], which are almost never used for magnetized plasma due to its high frequency gyro-motion. Recent attempts [9] [10] to couple the collision operator with the Vlasov equation in different velocity coordinates ignored the spatial dependence of the distribution function in the collision operator.

The so-called constant-of-motion coordinates, e.g. the total energy E and the magnetic moment μ , have been chosen for both particle [11] and continuum [12] gyrokinetic simulations in fusion devices. In such devices, the accurate representation of the particle orbits, both for the passing and trapped particles, is critical. The choice of (μ, E) coordinates is advantageous because (μ, E) remain constant along particle orbits (in the absence of collisions and acceleration). In other words, the velocity coordinates (μ, E) are orthogonal to the spatial coordinates. For instance, the collisionless Vlasov equation in (μ, E) coordinates is particularly simple. To accurately represent collisional effects, the same constant-of-motion coordinates should be used in the FPO. Our experience with the alternative method—interpolating between different velocity coordinates—has been found unsatisfactory, particularly with respect to the conservation properties. In this paper, we present an algorithm for a FPO in constant-of-motion coordinates based on a general high-order finite volume scheme on an unstructured, two dimensional velocity grid.

Although we focus on constant-of-motion coordinates in this paper, our goal is to develop a FPO that is accurate, conservative, yet easily applicable to different coordinates systems. To achieve this, we first choose one convenient but fixed coordinates system, e.g. in this case $(v_{||}, \mu)$, within which we develop a conservative high-order unstructured finite volume scheme for the FPO equations. The finite volume discretization is inherently density conservative, and the unstructured mesh decouples the choice of coordinates and the gridding strategy. Once this is completed, different velocity coordinates can then be mapped *directly* into the chosen $(v_{||}, \mu)$ coordinates, with a regular grid in the former typically becoming an irregular and unstructured grid in the latter. But after the mapping, the solution we obtained maintains high-order accuracy and good conservation. In this sense, the FPO is independent the choice of velocity coordinate, and the constant-of-motion coordinates discussed in this paper is but one such choice. For simplicity, the method is illustrated for the linearized FPO in this paper, where the diffusion coefficients are known, assuming the scattering particles have a Maxwellian velocity distribution. For nonlinear FPO, the diffusion coefficients are obtained by solving Rosenbluth potentials, which is an important but rather independent problem and will be dealt with in a separate paper. Once the nonlinear diffusion coefficients are obtained, the algorithm described here applies directly.

The rest of this paper is organized as follows. The formulation of the FPO in $(v_{||}, \mu)$ coordinates is given in section 2 both for general and linearized cases. The high-order finite volume scheme for general unstructured mesh is presented in section 3. In section 4, we describe the cut-cell method in (μ, E) space, and choices of stencil for finite volume reconstruction. The numerical tests are shown in section 5, and we give the summary and concluding remarks in section 6.

2 Fokker-Planck collision operator

In this section, we present the Fokker-Planck collision operator in $(v_{||}, \mu)$ coordinates and its linearized version about a fixed Maxwellian distribution function describing the background field particles.

2.1 General form

Following Trubnikov [13], the Fokker-Planck collision operator C can be written in a divergence form as,

$$C(f_\alpha) = \sum_{\beta} C^{\alpha/\beta}(f_\alpha) = - \sum_{\beta} \frac{\partial S_i^{\alpha/\beta}}{\partial v_i} \quad (1)$$

where $S^{\alpha/\beta}$ is the flux of the ‘test’ particles of species α due to collision with ‘field’ particles of species β (including $\beta = \alpha$). The flux $S^{\alpha/\beta}$ consists of a friction term and a diffusion term:

$$S_i^{\alpha/\beta} = \frac{F_i^{\alpha/\beta}}{m_\alpha} f_\alpha - D_{ik}^{\alpha/\beta} \frac{\partial f_\alpha}{\partial v_k}. \quad (2)$$

Using the so-called Trubnikov-Rosenbluth potentials, we can write [1] [13]

$$\frac{F_i^{\alpha/\beta}}{m_\alpha} = -\frac{m_\alpha}{m_\beta} n_\beta \Lambda^{\alpha/\beta} \frac{\partial h^\beta}{\partial v_i}, \quad D_{ik}^{\alpha/\beta} = -n_\beta \Lambda^{\alpha/\beta} \frac{\partial^2 g^\beta}{\partial v_i \partial v_j} \quad (3)$$

where $\Lambda^{\alpha/\beta}$ is defined as

$$\Lambda^{\alpha/\beta} = \Lambda_c \left(\frac{4\pi Z_\alpha Z_\beta e^2}{m_\alpha} \right)^2 \quad (4)$$

with Coulomb logarithm $\Lambda_c = \log(8\pi(n_e \lambda_D^3))$, λ_D is the Debye length. m is the particle mass, and n_β is the number density of the field particle.

The Trubnikov-Rosenbluth potentials are defined as

$$h^\beta = -\frac{1}{4\pi} \int \frac{f_\beta(\vec{v}')}{|\vec{v} - \vec{v}'|} d\vec{v}', \quad g^\beta = -\frac{1}{8\pi} \int |\vec{v} - \vec{v}'| f_\beta(\vec{v}') d\vec{v}', \quad (5)$$

where the distribution function is normalized to unity. With the following identities,

$$\nabla_v^2 |\vec{v} - \vec{v}'| = \frac{2}{|\vec{v} - \vec{v}'|}, \quad \nabla_v^2 \frac{1}{|\vec{v} - \vec{v}'|} = -4\pi \delta(\vec{v} - \vec{v}'), \quad (6)$$

it is easily established that

$$\nabla_v^2 g^\beta = h^\beta, \quad \nabla_v^2 h^\beta = f_\beta. \quad (7)$$

2.2 $v_{\parallel} - \mu$ coordinates

For magnetized plasmas, the parallel velocity v_{\parallel} and magnetic moment μ are defined as

$$v_{\parallel} = v_z, \quad \mu = \frac{m_{\alpha}}{2B}(v_x^2 + v_y^2), \quad (8)$$

where B is the local magnetic field strength and is assumed to be in the z direction. Note that v_{\parallel} and μ are an appealing choice for velocity space coordinates for magnetized plasmas after gyro-averaging, as μ is a constant of motion, while the use of v_{\parallel} makes f a single valued function in velocity space, avoiding the complicated multi-sheet problems which occur in reconstruction in (μ, E) space. Furthermore, the Jacobian of the (v_{\parallel}, μ) coordinates is a constant. Finally, a commonly used conservative form of the Vlasov equation is expressed in (v_{\parallel}, μ) space.

Hence, by ignoring the gyro-phase dependence, we transform the resulting axisymmetric collision operator from Cartesian (v_x, v_y, v_z) coordinates to (v_{\parallel}, μ) coordinates and obtain

$$C(f_{\alpha}) = -\frac{\partial S_x^{\alpha/\beta}}{\partial v_x} - \frac{\partial S_y^{\alpha/\beta}}{\partial v_y} - \frac{\partial S_z^{\alpha/\beta}}{\partial v_z} = \frac{\partial \Gamma_{v_{\parallel}}}{\partial v_{\parallel}} + \frac{\partial \Gamma_{\mu}}{\partial \mu} \quad (9)$$

where the fluxes $\Gamma_{v_{\parallel}}$ and Γ_{μ} are defined as

$$\Gamma_{v_{\parallel}} = D_v f_{\alpha} + D_{vv} \frac{\partial f_{\alpha}}{\partial v_{\parallel}} + D_{v\mu} \frac{\partial f_{\alpha}}{\partial \mu} \quad (10)$$

$$\Gamma_{\mu} = D_{\mu} f_{\alpha} + D_{\mu v} \frac{\partial f_{\alpha}}{\partial v_{\parallel}} + D_{\mu\mu} \frac{\partial f_{\alpha}}{\partial \mu}, \quad (11)$$

and the diffusion coefficients are given by

$$D_v = n_{\beta} \Lambda^{\alpha/\beta} \frac{m_{\alpha}}{m_{\beta}} \frac{\partial h^{\beta}}{\partial v_{\parallel}} \quad (12)$$

$$D_{vv} = -n_{\beta} \Lambda^{\alpha/\beta} \frac{\partial^2 g^{\beta}}{\partial v_{\parallel}^2} \quad (13)$$

$$D_{v\mu} = -2 n_{\beta} \Lambda^{\alpha/\beta} \frac{m_{\alpha}}{B} \mu \frac{\partial^2 g^{\beta}}{\partial v_{\parallel} \partial \mu} \quad (14)$$

$$D_{\mu} = 2 n_{\beta} \Lambda^{\alpha/\beta} \frac{m_{\alpha}^2}{B m_{\beta}} \mu \frac{\partial h^{\beta}}{\partial \mu} \quad (15)$$

$$D_{\mu v} = -2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha}{B} \mu \frac{\partial^2 g^\beta}{\partial v_{\parallel} \partial \mu} \quad (16)$$

$$D_{\mu\mu} = -2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha^2}{B^2} \mu \left(2\mu \frac{\partial^2 g^\beta}{\partial \mu^2} + \frac{\partial g^\beta}{\partial \mu} \right) \quad (17)$$

The Trubnikov-Rosenbluth potentials in (v_{\parallel}, μ) coordinates satisfy

$$\frac{\partial^2 g^\beta}{\partial v_{\parallel}^2} + \frac{m_\alpha}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial g^\beta}{\partial \mu} \right) = h^\beta \quad (18)$$

$$\frac{\partial^2 h^\beta}{\partial v_{\parallel}^2} + \frac{m_\alpha}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial h^\beta}{\partial \mu} \right) = f_\beta. \quad (19)$$

Thus, the equations (9)-(19) constitute the fully nonlinear Fokker-Planck collision operator in (v_{\parallel}, μ) coordinates.

2.3 Linearization

The fully nonlinear FPO given above can be simplified or linearized if the collisions between the test particles and the background plasma is more important than the collisions among the test particle themselves. Moreover, if the background plasma has a large heat capacity, it will remain close to a Maxwellian, i.e.

$$f_\beta(v) = F_M^\beta(v) = (\sqrt{\pi} v_{t\beta})^{-3} \exp(-v^2/v_{t\beta}^2) \quad (20)$$

where $v_{t\beta} = (2T_\beta/m_\beta)^{\frac{1}{2}}$ is the thermal velocity of field particles, and T_β is the its temperature.

In this case, the Trubnikov-Rosenbluth potentials depends only on the total velocity v , i.e. $h^\beta = h^\beta(v)$ and $g^\beta = g^\beta(v)$, where v is defined as

$$v = \sqrt{v_{\parallel}^2 + 2\mu B/m_\alpha}. \quad (21)$$

The derivatives of h and g can be calculated explicitly as

$$\frac{dh^\beta}{dv} = \frac{1}{4\pi v^2} H\left(\frac{v}{v_{t\beta}}\right) \quad (22)$$

$$\frac{dg^\beta}{dv} = \frac{T_\beta}{8\pi m_\beta v^2} H\left(\frac{v}{v_{t\beta}}\right) - \frac{1}{8\pi} G\left(\frac{v}{v_{t\beta}}\right) \quad (23)$$

$$\frac{d^2 g^\beta}{dv^2} = -\frac{T_\beta}{4\pi m_\beta v^3} H\left(\frac{v}{v_{t\beta}}\right) \quad (24)$$

where the error function $G(x)$, and function $H(x)$ are defined as

$$G(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy, \quad H(x) = \frac{4}{\sqrt{\pi}} \int_0^x y^2 e^{-y^2} dy = G(x) - xG'(x). \quad (25)$$

Consequently, the diffusion coefficients in (10) and (11) are given by

$$\tilde{D}_v = n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha v_{\parallel}}{m_\beta v} \frac{dh^\beta}{dv} \quad (26)$$

$$\tilde{D}_{vv} = -n_\beta \Lambda^{\alpha/\beta} \left(\frac{2\mu B}{m_\alpha v^3} \frac{dg^\beta}{dv} + \frac{v_{\parallel}^2}{v^2} \frac{d^2 g^\beta}{dv^2} \right) \quad (27)$$

$$\tilde{D}_{v\mu} = 2n_\beta \Lambda^{\alpha/\beta} \left(\frac{\mu v_{\parallel}}{v^3} \frac{dg^\beta}{dv} - \frac{\mu v_{\parallel}}{v^2} \frac{d^2 g^\beta}{dv^2} \right) \quad (28)$$

$$\tilde{D}_\mu = 2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha \mu}{m_\beta v} \frac{dh^\beta}{dv} \quad (29)$$

$$\tilde{D}_{\mu v} = 2n_\beta \Lambda^{\alpha/\beta} \left(\frac{\mu v_{\parallel}}{v^3} \frac{dg^\beta}{dv} - \frac{\mu v_{\parallel}}{v^2} \frac{d^2 g^\beta}{dv^2} \right) \quad (30)$$

$$\tilde{D}_{\mu\mu} = 2n_\beta \Lambda^{\alpha/\beta} \left[\left(\frac{2\mu^2}{v^3} - \frac{m_\alpha \mu}{vB} \right) \frac{dg^\beta}{dv} - \frac{2\mu^2}{v^2} \frac{d^2 g^\beta}{dv^2} \right]. \quad (31)$$

This constitutes the linearized Fokker-Planck collision operator in (v_{\parallel}, μ) coordinates.

3 Finite volume discretization

3.1 General formula

To solve the FPO numerically, consider a fixed partition of (v_{\parallel}, μ) plane by a set of non-overlapping polygonal sub-domains Ω_i , $i = 1 \dots N$. We may call each sub-domain a cell. On any cell Ω_i , the following Fokker-Planck equation

$$\frac{\partial f}{\partial t} = \nabla \cdot \Gamma = \frac{\partial \Gamma_{v_{\parallel}}}{\partial v_{\parallel}} + \frac{\partial \Gamma_{\mu}}{\partial \mu} \quad (32)$$

can be integrated on the cell volume. By Gauss theorem, we obtain the basic form of the finite-volume scheme

$$\frac{\partial \bar{f}_i}{\partial t} = \frac{1}{V_i} \sum_{j \in \partial \Omega_i} \int (\Gamma \cdot n_j) dS_j \quad \text{where} \quad \bar{f}_i = \frac{1}{V_i} \int_{\Omega_i} f dv. \quad (33)$$

Here \bar{f}_i is the cell averaged value of distribution function f on cell i , with V_i and $\partial \Omega_i$ being its volume and boundary. The \bar{f}_i , rather than a point-wise value, is the fundamental variable solved in the finite volume method, and its rate of change is determined by the fluxes across the cell edges. The fluxes $\Gamma_{v_{||}}$ and Γ_{μ} are given in (10) and (11), but to evaluate them, we must first obtain the point wise values of f and its derivatives on the cell edges. Hence a central piece of the finite-volume algorithm is the so-called reconstruction, i.e. finding point-wise values of $f(v_{||}, \mu)$ from the available cell-averaged values \bar{f}_i .

The basic idea of reconstruction on a particular cell is to use the cell-averaged values for this cell and the neighboring cells to form a local approximation function to $f(v_{||}, \mu)$, such that when this approximate function is averaged over any of the cells involved, the known cell-averaged value is recovered. Hence let us assume f to be a generic function of $v_{||}$ and μ and approximate f , in the neighborhood of cell p , by a polynomial of $v_{||}$ and μ , i.e.,

$$f_p(v_{||}, \mu) = \sum_{i=0}^n \sum_{j=0}^{n-i} a_{ij} (v_{||} - v_{||o})^i (\mu - \mu_o)^j. \quad (34)$$

Here n is the desired order of the approximating polynomial, and a_{ij} are the unknown coefficients. For a given order n , the total number of a_{ij} is $(n+1)(n+2)/2$. The $(v_{||o}, \mu_o)$ is an appropriately chosen reference point. Here we observe that if f_p is a unique approximation based on $(v_{||o}, \mu_o)$, i.e., given all the a_{ij} determined uniquely, then it can be equivalently written for another reference point $(v'_{||o}, \mu'_o)$. The new coefficients will become a'_{ij} and in general $a'_{ij} = a'_{ij}(a_{mn}, v_{||o}, v'_{||o}, \mu_o, \mu'_o)$, but the f_p itself remains the same. In other words, the choice of $(v_{||o}, \mu_o)$ is not essential to the approximation. Thus, we can choose a global reference point and use it for the all reconstructions. The origin $v_{||o} = \mu_o = 0$ is clearly a convenient choice and is used throughout the paper. We will see its advantage in the next section.

3.2 High-order reconstruction

To obtain an accurate representation of FPO with modest grid resolution, we choose a fourth order approximation $n = 4$. Therefore, 15 coefficients a_{ij} in (34) need to be determined. For convenience, we reorder the a_{ij} s by a subscript

k , where $k = 0, \dots, 14$, and for each k , the corresponding exponents of $v_{||}$ and μ are denoted as i_k and j_k , where $i_k + j_k \leq n$. So each a_k is the coefficient of a base polynomial of the order $i_k + j_k$. To find these a_k on a particular cell, we first construct a 15-cell stencil consisting of the cell itself and 14 neighboring cells, chosen as described below in section 4.2. Then by integrating (34) over each of the cells within the stencil, we obtain a system of linear equations for the a_k :

$$B_{mk} a_k = \bar{f}_m, \quad m, k = 0, 1, \dots, 14. \quad (35)$$

where the matrix $B_{mk} = [b_{mk}]$, and b_{mk} is given by

$$b_{mk} = \frac{1}{V_m} \int_{\Omega_m} v_{||}^{i_k} \mu^{j_k} dv_{||} d\mu, \quad \text{where} \quad V_m = \int_{\Omega_m} dv_{||} d\mu, \quad (36)$$

that is, b_{mk} is the cell averaged value of the k th base polynomial on the m th cell. Note that the b_{mk} depends on the metrics of the m th cell only, but not on \bar{f}_m . More importantly, b_{mk} does not depend on stencil, i.e., the value of b_{mk} is the same regardless which cell's stencil it is on. This is because the polynomials in b_{mk} are based on a fixed global origin (see discussion in section 3). It suffices to compute b_{mk} only once.

To evaluate b_{mk} for each cell, we notice that the simplex in the 2D plane is a triangle, and any polygonal cell can be divided into one or several triangles (in 3D it would be tetrahedron). It suffices to integrate the base polynomials over a simplex. Let Δ be such a simplex, and $\Delta = \{(v_{||0}, \mu_0), (v_{||1}, \mu_1), (v_{||2}, \mu_2)\}$; for fourth order accuracy, we choose 16-point Gaussian quadrature

$$\int_{\Delta} f(v_{||}, \mu) dv_{||} d\mu = \sum_{i=0}^{15} w_i f(v_{||i}, \mu_i) = J \sum_{i=0}^{15} w_i f(\tilde{v}_{||i}, \tilde{\mu}_i) \quad (37)$$

where the $(\tilde{v}_{||}, \tilde{\mu})$ are the Gaussian abscissa in the standard triangle at $(0,0), (0,1)$ and $(1,0)$. The transformation between $(v_{||i}, \mu_i)$ and $(\tilde{v}_{||i}, \tilde{\mu}_i)$ is given by

$$v_{||i} = v_{||0} + (v_{||1} - v_{||0}) \tilde{v}_{||i} + (v_{||2} - v_{||0}) \tilde{\mu}_i \quad (38)$$

$$\mu_i = \mu_0 + (\mu_1 - \mu_0) \tilde{v}_{||i} + (\mu_2 - \mu_0) \tilde{\mu}_i. \quad (39)$$

Here $J = |(v_{||1} - v_{||0})(\mu_2 - \mu_0) - (v_{||2} - v_{||0})(\mu_1 - \mu_0)|$ is the Jacobian. The numerical values of the Gaussian weights w_i and abscissa $(\tilde{v}_{||i}, \tilde{\mu}_i)$ are given in the Appendix.

Once the b_{mk} are known, the coefficients a_k can be solved directly by inverting B_{mk} in (35), e.g. by the Gaussian elimination method. Substituting the

coefficients into (34), we obtain the the pointwise reconstruction formula for $f(v_{||}, \mu)$. Moreover, the pointwise derivatives $f_{v_{||}}$ and f_{μ} , needed for flux evaluation, can also be obtained as

$$f_{v_{||}} = \sum_{i_k=1}^n \sum_{j_k=0}^{n-i_k} i_k a_k v_{||}^{i_k-1} \mu^{j_k}, \quad f_{\mu} = \sum_{i_k=0}^n \sum_{j_k=1}^{n-i_k} j_k a_k v_{||}^{i_k} \mu^{j_k-1} \quad (40)$$

To integrate the fluxes across a cell edge, again we use fourth-order Gaussian quadrature. For example, given the fluxes $\Gamma_p = \Gamma \cdot n$ at three Gaussian points on a edge, the edge flux is computed by

$$\int_l \Gamma dl = \sum_{p=0}^2 c_p \Gamma_p = \frac{l}{2} \sum_{p=0}^2 c_p \Gamma(\tilde{x}_p). \quad (41)$$

Here l is the length of the edge. The $c_p = \{5/9, 8/9, 5/9\}$ are the weights and $\tilde{x}_p = \{-\sqrt{15}/5, 0, \sqrt{15}/5\}$ are the Gaussian abscissa on $[-1, 1]$.

Since each interior cell edge belongs to two cells, the flux across it can be computed from either of the cells. To have a conservative scheme, a unique definition of the flux must be used. Here we use an upwind approach, that is, assuming edge pq is shared by cell p and q , the flux F is chosen according to

$$\Gamma = \begin{cases} \int \Gamma_{pq} \cdot n_{pq} dl & \text{if } \int \Gamma_{pq} \cdot n_{pq} dl > 0 \\ \int \Gamma_{qp} \cdot n_{qp} dl & \text{otherwise} \end{cases} \quad (42)$$

Here Γ_{pq} and Γ_{qp} are the computed point-wise fluxes from cell p to cell q and vice versa. Also, n_{pq} is the unit normal of edge pq pointing from p to q .

We note a few features of this reconstruction method. First, unlike the conventional finite-volume method, the current reconstruction does not require one to specify the location of \bar{f}_i ; it suffices for \bar{f}_i to be associated with a cell. Second, the metric information of each cell is included automatically in the base polynomial average b_{mk} . So the reconstruction procedure applies equally to structured or unstructured meshes with arbitrary cell shapes. Furthermore, since for each cell the b_{mk} need to be computed only initially, and then, to construct the matrix B_{mk} , we only need to choose the b_{mk} from the cells in the stencil.

A properly chosen stencil is the key to the reconstruction algorithm. The minimum requirement is that the resulting matrix from the stencil should be invertible and well-conditioned. In practice, the cells in a stencil should form a simply-connected domain in the neighborhood of the current cell and

have sufficient support in both directions to avoid a singularity in the resulting matrix. Since the FPO is a convection-diffusion equation in velocity space, the construction of the stencils should take into account the direction of the edge fluxes. In next section, we discuss this in the context of constant-of-motion coordinates.

4 Constant-of-motion coordinates

As mentioned earlier, the general finite volume algorithm developed here does not require the mesh to be regular or even structured. The advantage of this method is that it can use a (v_{\parallel}, μ) mesh that may be mapped directly from another mesh in a different velocity coordinate system, and yet the same algorithm still applies.

One choice of velocity space coordinates for gyrokinetic simulation, as noted in section 1, is the total energy E and magnetic moment μ . In the absence of collision and acceleration, the E and μ are conserved along a particle orbit. So for the Vlasov equation, the numerical approximation of the spatial derivative is effectively decoupled from the velocity space operation. This prevents the spatial approximation from introducing numerical diffusion into the velocity space and is a particularly good choice for accurately calculating particle orbits. An example of a regular (μ, E) mesh is shown in figure 1a), where the boundary $v_{\parallel} = 0$ is a straight line cutting through the background grid and separates the physical (above, $v_{\parallel}^2 > 0$) and non-physical (below, $v_{\parallel}^2 < 0$) zone.

However, existing FPOs are typically not written in constant-of-motion coordinates. So to evaluate the FPO term, we adopt the following strategy. First, we employ a cell-merging technique (described below) to treat the cells being cut by the boundary. This is to avoid cut-cells which can be arbitrarily small and thus dominated by a Courant-Fredrich-Lewy time step constraint. Then, the resulting (μ, E) grid is mapped into (v_{\parallel}, μ) space, and we apply the finite volume algorithm describe in section 3. The resulting cut-cell may have various shapes and different connectivity patterns to its neighbors. For instance, figure 1b) shows the corresponding (v_{\parallel}, μ) mesh mapped directly from the (μ, E) mesh in figure 1a). Note that the $v_{\parallel} = 0$ boundary now becomes the vertical axis about which the mesh is symmetric. The left and right half planes correspond to particles with different signs of the parallel streaming velocity.

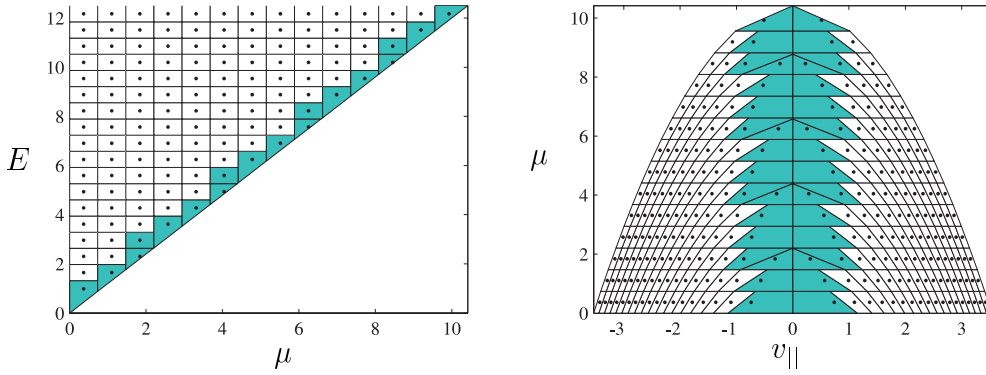


Fig. 1. Velocity space expressed in (μ, E) and the corresponding (v_{\parallel}, μ) coordinates.

4.1 Cell merging and mapping

The cut-cell merging procedure starts by computing the cell center of the regular background (μ, E) grid. If the center of a cell is above the $v_{\parallel} = 0$ boundary (physical domain), this cell will be retained. Otherwise, it will be discarded. If the cell that is discarded has a part in the physical zone, then this part will be merged into its neighboring cells. We follow Ye *et al* [14] and choose to merge cells in the vertical E direction. Since the $v_{\parallel} = 0$ boundary is a straight line, after the cell merging, there are totally four different types of cut-cells, shown in figure 2, with a regular cell for comparison. Note that depending on the slope of $v_{\parallel} = 0$ boundary, i.e., the local B field, there could be no neighboring cells available in the E direction at the upper-right corner. In such cases, we merge the cut-cells in μ direction. So the last one in figure 2 can appear only at the upper right corner with highest values of E and μ .

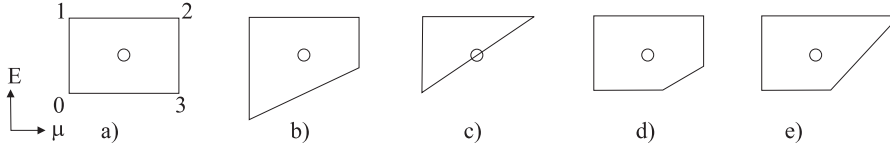


Fig. 2. Computational cells in (μ, E) coordinates. a) is a regular cell and b-e) are cut cells at $v_{\parallel} = 0$ boundary. b) and e) have been merged with neighboring cells. e) only appears at the corner of highest E and μ .

As a result of the merging, the cut-cells may share some of their edges with more than one neighboring cell (see figure 1a). When we assemble the edge fluxes, contributions from different neighboring cells must be accounted appropriately.

The mapping between (μ, E) into a (v_{\parallel}, μ) coordinates is through the following formula expressing energy conservation

$$v_{\parallel} = \pm \sqrt{E - \mu B - q\Phi}, \quad (43)$$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Δi_k	-2	-1	-1	0	0	0	0	0	1	1	1	2	2	-1
Δj_k	0	1	0	2	1	-1	-1	-2	1	0	-1	0	1	-2

Table 1

Index shifts relative to cell (i, j) in a 15-cell stencil.

where B and Φ are the magnetic field and the electric potential. μ is unchanged, but becomes the second coordinate. Through this mapping, each (μ, E) cell, (cut cell or regular cell) will be mapped into two symmetric (v_{\parallel}, μ) cells with opposite sign for v_{\parallel} . Because the mapping is analytic, particle conservation holds in both coordinates. In this mapping, a cell boundary of constant E is not mapped strictly to a straight cell boundary in (v_{\parallel}, μ) . But this poses no problem, as we only need to map the vertices of a cell. The initialization of \bar{f} , as well as the ensuing computations, is carried out entirely in (v_{\parallel}, μ) space. It is equivalent to change the cell boundary in (μ, E) to a slightly curved line such that when mapped to (v_{\parallel}, μ) , it becomes straight line. On the other hand, one might think using finite volume directly in (μ, E) coordinates, but the problem is that when getting to the turning point boundary $v_{\parallel} = 0$, it is difficult for the reconstruction to include cells on the other sheet. Whereas in the (v_{\parallel}, μ) coordinates, there is no multi-sheet problem; the reconstruction at the turning point boundary is essentially the same as anywhere else.

4.2 Reconstruction stencil

As mentioned earlier, the 4th order reconstruction scheme requires a 15-cell stencil. Since the FPO represents convection and diffusion in velocity space, and the direction of the edge fluxes is not usually known *a priori*, we choose the stencil somewhat centered on the cell on which the reconstruction is being sought.

Since each cell in (μ, E) directly corresponds to two cells in (v_{\parallel}, μ) , to make the notation clear, we may describe the stencil using (μ, E) cells which are Cartesian except near the $v_{\parallel} = 0$ boundary.

By the cell cutting procedure, each cell, regular or cut, has one and only one cell center. They are the cell centers of the background grid, and for the cut-cells, typically not the geometric center of the cell. Nevertheless, the cell center index can be used as a label to uniquely identify each cell. Let us assume the cell where the reconstruction is sought has a cell center located at index (i, j) , then corresponding to figure 3, besides cell (i, j) , the base stencil consists of 14 cells with the index shift shown in Table 1.

In figure 3, we show such a stencil in both coordinates where all the cells in

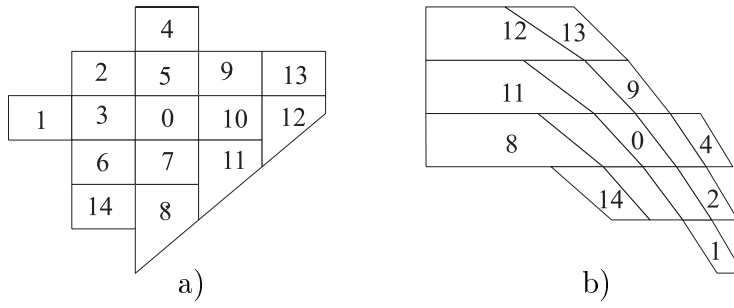


Fig. 3. A 15-cell stencil in the a) (μ, E) and b) corresponding $(v_{||}, \mu)$ coordinates. This stencil does not cross cut-cell boundary, and only part of the corresponding cells in $(v_{||}, \mu)$ coordinates are marked.

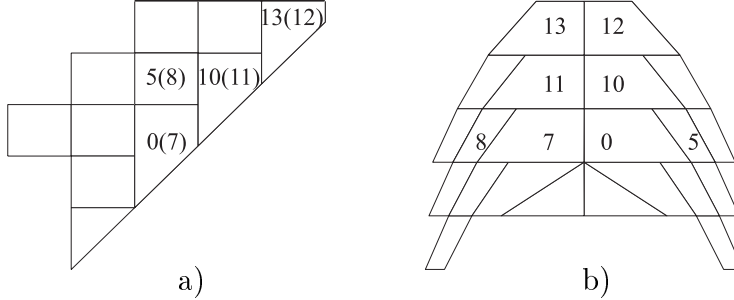


Fig. 4. A 15-cell stencil in the a) (μ, E) and b) corresponding $(v_{||}, \mu)$ coordinates. This stencil crosses cut-cell boundary, and the overlapping cells are marked in both coordinates

this stencil belong to the same sheet in (μ, E) . The numbers marked on the cells in the $(v_{||}, \mu)$ coordinates correspond the cells marked with same number in the (μ, E) coordinates. In figure 4, a stencil with its center cell close to the turning point boundary is shown. Because of the turning point boundary, some cells in the stencil are not available since the index shift place them into nonphysical zone. But this simply means that the cells are on the other sheet, i.e. the other half plane across the vertical axis in $(v_{||}, \mu)$ space. The numbers in the parentheses designate cells on the other sheet with the same (μ, E) indices that need to be included in the stencil.

In general, a stencil need not be static; its composition can change with time according to the direction of the local flux. Examples in a one dimensional system include upwind, TVD and ENO/WENO schemes. Although for a multi-dimensional system a complete theory is still lacking, in practice there are still ways to incorporate the basic idea of dynamic stencil. For instance, we can make a slight variation of the above static stencil to form a stencil with a certain degree of upwinding. The idea is to keep the maximum extension of the stencil in (μ, E) directions the same, but choose the last two cells, 13 and 14, based on the direction of the edge fluxes of cell 0. Hence, the index shift

of the cell 13 and 14 can be determined by

$$\Delta i_{13} = \begin{cases} -2 & \text{if } \Gamma_E > 0 \\ 2 & \text{otherwise,} \end{cases} \quad \text{and} \quad \Delta j_{14} = \begin{cases} -2 & \text{if } \Gamma_\mu > 0 \\ 2 & \text{otherwise} \end{cases} \quad (44)$$

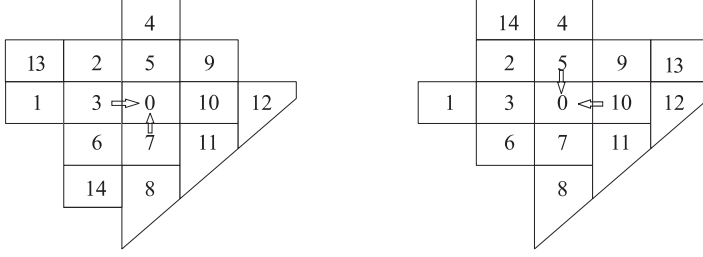


Fig. 5. Dynamic upwind stencil consisting of 15 cells in (μ, E) coordinates. The arrows indicate the direction of the net edge fluxes of cell 0.

Here Γ_E and Γ_μ are the net fluxes through the edge of the cell 0 which are obtained from the $v_{||}$ and μ space at the previous time step. In figure 5, two examples are shown for the location of cells 13 and 14 based on the sign of Γ_E and Γ_μ . The dynamic stencil can adjust itself to bias towards the upwind direction of the edge fluxes, so it offers better numerical stability and robustness compared to a static stencil.

For the same stencil to be used near a boundary, ghost cells must be provided. At the top boundary $E = E_{max}$, we add two more rows of ghost cells and specify their values by assuming the distribution function decays exponentially in the E direction. Similarly, at $\mu = 0$ boundary, we add two more columns of cells, and assign their cell averaged values by linear extrapolation based on three interior neighboring cells.

After the cut-cell merging and mapping, and the construction of the stencil, the FPO in constant-of-motion coordinates can now be evaluated directly in the $(v_{||}, \mu)$ coordinates.

5 Numerical Tests

5.1 Reconstruction accuracy

We first test the accuracy of the reconstruction scheme using an analytic drifting Maxwellian. The normalized density and mean temperature are $n_\alpha = 1, T_\alpha = 1$, and the normalized drift velocity is $v_m = 0.1$. The boundaries of the Cartesian mesh are set at $E_{max} = \mu_{max} = 16$. Starting from cell-averaged

$N_\mu \times N_E$	$\frac{1}{N} \sum_{i=0}^N f_i - f_M $	$ U - U_M $	$ T - T_M $
20×20	1.3929×10^{-3}	2.6042×10^{-3}	2.1463×10^{-2}
40×40	2.1665×10^{-4}	5.6951×10^{-4}	1.7826×10^{-3}
80×80	2.9407×10^{-5}	9.1703×10^{-5}	1.3963×10^{-4}
160×160	5.9352×10^{-6}	1.4444×10^{-5}	5.4600×10^{-6}

Table 2

Errors of reconstruction for a normalized drift Maxwellian. The columns are resolution, point-wise values the cell corners, the mean drift velocity U , and the mean temperature T . N is total number of cell corners. The velocity space boundaries are at $E_{max} = \mu_{max} = 16$ and the magnetic field $B = 1.2$

initial values, we use the reconstruct scheme to compute the point-wise value of f at the cell centers, the mean drift velocity and mean temperature. The last two correspond to the first and second moments of the distribution function, respectively. The errors of the reconstruction with different resolutions are summarized in Table 2. The results confirm the fourth order accuracy of the reconstruction scheme.

5.2 Spitzer thermalization

We then test the linear FP collision operator on the Spitzer thermalization problem [13], which describes the thermal relaxation rate of species α on a background fixed Maxwellian species β . Assuming the distribution of α particles also remains Maxwellian during the relaxation, the theoretical rate of temperature change for species α is given by

$$\frac{dT_\alpha}{dt} = -\frac{8m_\alpha}{3m_\beta\sqrt{\pi}} \frac{T_\alpha - T_\beta}{\tau^{\alpha/\beta} \{T_\alpha + \frac{m_\alpha}{m_\beta} T_\beta\}} \quad (45)$$

Here $\tau^{\alpha/\beta}$ is the basic relaxation time defined by

$$\tau^{\alpha/\beta}(\epsilon) = \frac{\sqrt{m_\alpha}}{\pi\sqrt{2}e_\alpha^2 e_\beta^2} \frac{\epsilon^{3/2}}{\Lambda_c n_\beta} \quad (46)$$

We solve the time dependent FPO equation in (μ, E) space, and use a standard 4th-order Runge-Kutta scheme for the time integration. We consider both cases with the background temperature higher (heating) and lower (cooling) than the initial test particle temperature. The parameters are as follows. $m_\alpha = m_\beta = 2m_p$, $e_\alpha = e_\beta = q$, where m_p is proton mass, and q the charge of electron. The density of the background particles is $n_\beta = 10^{14}/cm^3$, and we choose the

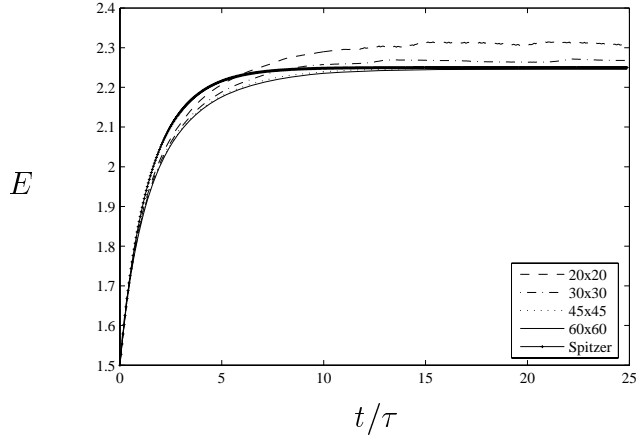


Fig. 6. Mean energy evolution of Spitzer thermalization. Field particle temperature is 1.5keV and initial test particle temperature is at 1keV.

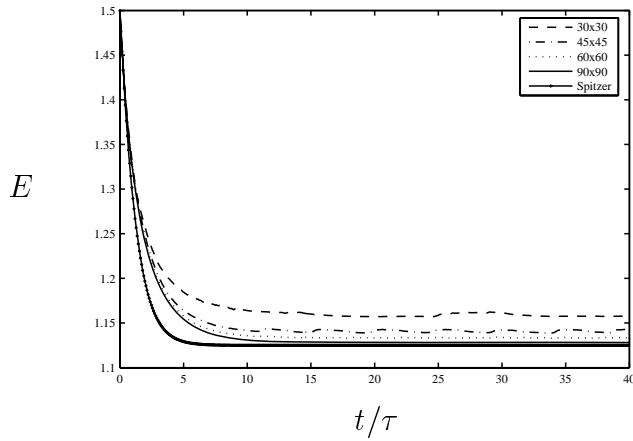


Fig. 7. Mean energy evolution of Spitzer thermalization. Field particle temperature is 0.75 keV and initial test particle temperature is at 1 keV

Coulomb logarithm $\Lambda_c = 16$. The initial temperature of the test particles is $T_\alpha = 1\text{keV}$, the background temperature is set $T_\beta = 1.5\text{keV}$ for heating and $T_\beta = 0.75\text{keV}$ for cooling.

Figure 6 and 7 show the time history of the mean energy of the test species during heating and cooling with different grid resolutions. The density is conserved and the momentum remains zero during the process. The mean energy evolution at different resolutions shows the convergence to the field particle energy. It can be seen the initial development follows the analytic curve very closely, because the test particles is initialized with a Maxwellian distribution. Then the relaxation starts to deviate from the analytic curve, as the distribution of the α particles changes from its initial Maxwellian as the collisional evolution progresses. However, when thermal equilibrium is approached, the curve follow the theory again as now the distribution function returns to being Maxwellian, but at the same temperature as the background particles. The

time history for the cooling case follows a similar process.

5.3 Return to isotropy from loss cone distribution

In this section we show the relaxation to a Maxwellian for the test particles starting from an initial condition with a velocity space loss cone. Loss cone situations are common in plasma confinement experiments, particularly for the plasmas at the edge of the fusion device. In such cases, particles in part of the velocity space escape from the magnetic confinement, leaving behind a loss cone region with no particles inside. However, because of the collisional effects, particles from other parts of the velocity space will be scattered into the loss cone region, and the loss cone will gradually get filled up. So for the test particles, it is a process of a return-to-isotropy, and the its distribution function will eventually recover to a Maxwellian.

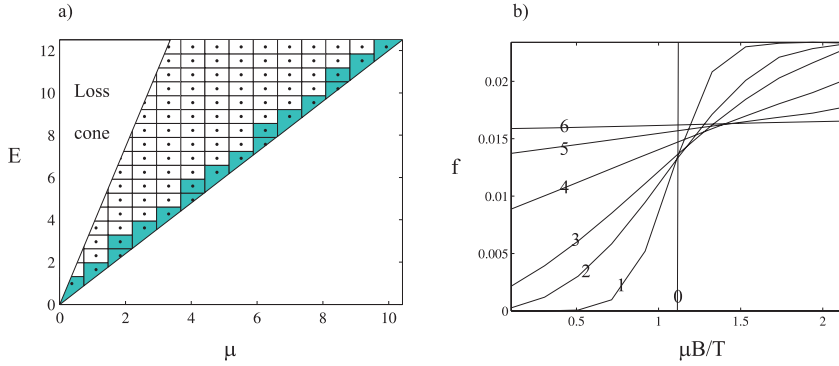


Fig. 8. a) Loss cone in (μ, E) velocity space, b) distribution function f as a function of $\mu B/T$ for at a fixed energy $E/T = 2.387$ during the filling of the loss cone. The normalized times (by the basic relaxation time τ) for the curves 0-6 are $t/\tau = 0, 0.027, 0.135, 0.27, 0.675, 1.35, 2.7$.

Figure 8a shows a typical loss cone in velocity space for purely magnetic mirror trapping. In this case the loss cone boundary is a straight line whose slope is set by the maximum of the B field. The following parameters are chosen to compute the loss cone problem using the present linear FPO in (μ, E) space. The slopes of the loss cone and the $v_{\parallel} = 0$ boundaries are $B_{max} = 2.0$ and $B_{min} = 1.2$. The initial condition for the distribution function f is a Maxwellian at $T_{\alpha} = 1$ but with the loss cone region zeroed out. The initial normalized density is thus 0.6991 instead of unity. Both the test and background particles have twice the proton mass and unit charge. The background temperature $T_{\beta} = 1$. The grid resolution used is $N_E = 45$ and $N_{\mu} = 50$. Let $T_0 = 1keV$ be the reference temperature, the magnetic moment is normalized by T_0/B_{min} and the normalized energy is E/T_0 . Figure 8b shows f as a function of μ at a fixed $E/T_0 = 2.387$ at different times of the loss-cone filling.

Time is normalized as before by the initial basic relaxation time τ . Starting from a step function at time $t = 0$, the loss cone (left) is gradually filled by particles from the trapped region (right), so the increase of f in the loss cone is accompanied by the decrease of f in the trapped region. Eventually, the loss cone is completely filled up, and f is relaxed to a Maxwellian. In this case, f becomes independent to μ as shown by the curve 6 in 8b.

6 Summary and discussion

In this paper, a new fourth-order finite-volume algorithm is developed for the Fokker-Planck collision (FPO) operator for highly magnetized plasmas. The underlying velocity-space coordinates are chosen to be the parallel velocity v_{\parallel} and magnetic moment μ , and the FPO and its linearized form are derived in these coordinates. The finite-volume reconstruction scheme is based on two dimensional Gaussian quadrature and can be applied to arbitrary unstructured meshes. As a result, the underlying mesh in (v_{\parallel}, μ) can be generated by directly mapping a grid from a different, user-chosen coordinate set, and the resulting FPO still conserves the number of particles to roundoff error, and the momentum and energy up to fourth order truncation error.

As an application, we compute the linearized collision operator in constant-of-motion coordinates (μ, E) . A cell-merging method at the turning point boundary is employed and the resulting mesh is mapped into (v_{\parallel}, μ) coordinates analytically. An 15-cell stencil is devised for the reconstruction and the fluxes are evaluated by the fourth order Gaussian quadrature. The accuracy of the reconstruction algorithm is validated using a drifting Maxwellian. The method is applied to Spitzer thermalization problem with both heating and cooling cases, and good agreement with the theoretical results is obtained. A return to isotropy was obtained from the present FPO operator for the test particles starting from a distribution function with a velocity space loss cones.

We would like to make a few final remarks here. First, the present reconstruction scheme is not specific to velocity space; it can be applied to configuration space as well. Second, though we focused on the linear FPO here, the method is directly applicable to a fully nonlinear collision operator once the Trubnikov-Rosenbluth potentials are known. Lastly, the present finite-volume scheme can be generalized to three dimensionals in a straightforward way.

Appendix: Gaussian quadrature

The 4th-order Gaussian quadrature on the standard triangle $\Delta = \{(0, 0), (0, 1), (1, 0)\}$ is given by

$$\int_{\Delta} f(\tilde{x}, \tilde{y}) d\tilde{x}d\tilde{y} = \sum_{i=0}^{15} w_i f(\tilde{x}_i, \tilde{y}_i), \quad (47)$$

where the standard abscissa and weights are given in the following table.

i	w_i	\tilde{x}_i	\tilde{y}_i
0	0.144315607677787	0.333333333333333	0.333333333333333
1	0.095091634267285	0.081414823414554	0.459292588292723
2	0.095091634267285	0.459292588292723	0.081414823414554
3	0.095091634267285	0.459292588292723	0.459292588292723
4	0.103217370534718	0.658861384496480	0.170569307751760
5	0.103217370534718	0.170569307751760	0.658861384496480
6	0.103217370534718	0.170569307751760	0.658861384496480
7	0.032458497623198	0.898905543365938	0.050547228317031
8	0.032458497623198	0.050547228317031	0.898905543365938
9	0.032458497623198	0.050547228317031	0.050547228317031
10	0.027230314174435	0.008394777409958	0.263112829634638
11	0.027230314174435	0.008394777409958	0.728492392955404
12	0.027230314174435	0.263112829634638	0.008394777409958
13	0.027230314174435	0.263112829634638	0.728492392955404
14	0.027230314174435	0.728492392955404	0.008394777409958
15	0.027230314174435	0.728492392955404	0.263112829634638

Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under Contract W-7405-ENG-48. This research was supported by the LLNL LDRD program as Project 04-SI-003 and DOE Edge Simulation Laboratory project.

The authors are grateful to the stimulating discussions with Drs. B. Cohen, M. Dorr, J. Hittinger, G. Kerbel and W. Nevins.

References

- [1] M. N. Rosenbluth, W. M. MacDonald, D. L. Judd, Fokker-Planck equation for an inverse-square force, *Phys. Rev.*, 107 (1957) 1-6.
- [2] J. Killeen, G. D. Kerbel, M. G. McCoy, A. A. Mirin, *Computational methods for kinetic models of magnetically confined plasmas*, Springer-Verlag, (1986)
- [3] J. S. Chang, G. Cooper, A practical difference scheme for Fokker-Planck equations, *J. Comput. Phys.* 6(1) (1970) 1-16.
- [4] E. M. Epperlein, Implicit and conservative difference scheme for the Fokker-Planck equation, *J. Comput. Phys.* 112(2) (1994) 291-297.
- [5] I. K. Khabibrakhmanov, G. V. Khazanov, The spectral collocation method for the kinetic equation with the nonlinear two-dimensional coulomb collisional operator, *J. Comput. Phys.* 161(2) (2000) 558-575.
- [6] L. Chacon, D. C. Barnes, D. A. Knoll, G. H. Miley, An implicit energy-conservative 2D Fokker-Planck algorithm: I. difference scheme, *J. Comput. Phys.* 157(2) (2000) 618-653.
- [7] C. Bueta, S. Cordier, P. Degond, M. Lemou, Fast algorithms for numerical, conservative, and entropy approximations of the Fokker-Planck-Landau equation, *J. Comput. Phys.* 133(2) (1997) 310-322.
- [8] L. Pareschi, G. Russo, G. Toscani, Fast spectral methods for the Fokker-Planck-Landau collision operator, *J. Comput. Phys.* 165(1) (2000) 216-236.
- [9] F. Filbet, L. Pareschi, A numerical method for the accurate solution of the Fokker-Planck-Landau equation in the nonhomogeneous case, *J. Comput. Phys.* 179(1) (2002) 1-26.
- [10] N. Crouseilles, F. Filbet, Numerical approximation of collisional plasmas by high order methods, *J. Comput. Phys.* 201(2) (2004) 564-572.
- [11] Y. Idomura, S. Tokuda, Y. Kishimoto, Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas using a canonical Maxwellian distribution, *Nuclear Fusion* 43, (2003) 234-243.
- [12] X. Xu, Z. Xiong, *et al*, Tempest Continuum Simulations and Edge Gyrokinetic Theory, *Nuclear Fusion*, submitted, 2006.
- [13] B. A. Trubnikov, Particle interactions in a fully ionized plasma, *Review of Plasma Physics* 1, (1965) 105-204.
- [14] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (2) (1999) 209-240.