

UKAEA-CCFE-PR(25)345

M. R. Hardman, M. Abazorius, J. Omotani, M. Barnes,
S. L. Newton, J. W. S. Cook, P. E. Farrell, F. I. Parra

A higher-order finite-element implementation of the exact Landau Fokker–Planck collision operator for charged particle collisions in a low density plasma

Enquiries about copyright and reproduction should in the first instance be addressed to the UKAEA Publications Officer, Culham Science Centre, Building K1/O/83 Abingdon, Oxfordshire, OX14 3DB, UK. The United Kingdom Atomic Energy Authority is the copyright holder.

The contents of this document and all other UKAEA Preprints, Reports and Conference Papers are available to view online free at scientific-publications.ukaea.uk/

A higher-order finite-element implementation of the exact Landau Fokker–Planck collision operator for charged particle collisions in a low density plasma

M. R. Hardman, M. Abazorius, J. Omotani, M. Barnes, S. L. Newton, J. W. S. Cook, P. E. Farrell, F. I. Parra

A higher-order finite-element implementation of the nonlinear Fokker–Planck collision operator for charged particle collisions in a low density plasma

M. R. Hardman^{a,b,c}, M. Abazorius^b, J. Omotani^c, M. Barnes^b, S. L. Newton^c, J. W. S. Cook^{c,d}, P. E. Farrell^{e,g}, F. I. Parra^f

^aTokamak Energy Ltd, 173 Brook Drive, Milton Park, Abingdon, OX14 4SD, United Kingdom

^bRudolf Peierls Centre for Theoretical Physics, University of Oxford, Clarendon Laboratory, Parks Road, Oxford, OX1 3PU, United Kingdom

^cUnited Kingdom Atomic Energy Authority, Culham Campus, Abingdon, OX14 3DB, United Kingdom

^dCentre for Fusion, Space and Astrophysics, University of Warwick, Coventry, CV4 7AL, United Kingdom

^eMathematical Institute, University of Oxford, Oxford, OX2 6GG, United Kingdom

^fPrinceton Plasma Physics Laboratory, P.O. Box 451, Princeton, 08540, New Jersey, United States

^gMathematical Institute, Faculty of Mathematics and Physics, Charles University, Prague, 18600, Czechia

Abstract

Collisions between particles in a low density plasma are described by the Fokker–Planck collision operator. In applications, this nonlinear integro-differential operator is often approximated by linearised or ad-hoc model operators due to computational cost and complexity. In this work, we present an implementation of the nonlinear Fokker–Planck collision operator written in terms of Rosenbluth potentials in the Rosenbluth–MacDonald–Judd (RMJ) form. The Rosenbluth potentials may be obtained either by direct integration or by solving partial differential equations (PDEs) similar to Poisson’s equation: we optimise for performance and scalability by using sparse matrices to solve the relevant PDEs. We represent the distribution function using a tensor-product continuous-Galerkin finite-element representation and we derive and describe the implementation of the weak form of the collision operator. We present tests demonstrating a successful implementation using an explicit time integrator and we comment on the speed and accuracy of the operator. Finally, we speculate on the potential for applications in the current and next generation of kinetic plasma models.

Keywords: Fokker–Planck Collision Operator, Finite-Elements, Drift-Kinetics, Performance Optimisation

1. Introduction

A low density plasma is one that can be accurately described by the one-point particle distribution function $F_s(\mathbf{r}, \mathbf{v}, t)$. The distribution function provides us with the number of particles $\delta n(\mathbf{r}, \mathbf{v}, t) = F_s(\mathbf{r}, \mathbf{v}, t) d^3\mathbf{r}d^3\mathbf{v}$ of species s at a time t in the phase space volume around the phase space position $(\mathbf{r}, \mathbf{v})$, with $\mathbf{r}$ the particle position and $\mathbf{v}$ the particle velocity. An equation for the time evolution of the distribution function may be obtained from the BBGKY hierarchy [1], which converts an N -body Hamiltonian system describing a plasma or gas into a statistical description. The resulting equation has the form

$$\begin{aligned} \frac{\partial F_s}{\partial t} + \mathbf{v} \cdot \nabla F_s + \frac{Z_s e}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial F_s}{\partial \mathbf{v}} \\ = \sum_{s'} C_{ss'} [F_s, F_{s'}], \end{aligned} \quad (1)$$

where the left-hand side of the equation is the Vlasov operator, accounting for the acceleration of particles by the large-scale electromagnetic fields. The Boltzmann collision operator on the right-hand side of the equation accounts for the interactions of particles of species s with local small-scale electromagnetic fields generated by interactions between particles of species s' at the same position $\mathbf{r}$. Here, m_s is the species mass, Z_s is the species charge number, e is the unit charge, and $\mathbf{E}$ and $\mathbf{B}$ are the electric and magnetic fields, respectively. Equation (1) is

solved with the boundary condition that $F_s(\mathbf{v}) \rightarrow 0$ as $|\mathbf{v}| \rightarrow \infty$ and an appropriate boundary condition in $\mathbf{r}$.

If the interaction cross section is chosen to be the $1/r$ electrostatic potential, then the collision operator becomes the well-known Fokker–Planck collision operator [2, 3, 4, 5], which we introduce here using the form due to Landau:

$$\begin{aligned} C_{ss'} [F_s, F_{s'}] = \\ \frac{\gamma_{ss'}}{m_s} \frac{\partial}{\partial \mathbf{v}} \cdot \left\{ \int \frac{\partial^2 g}{\partial \mathbf{v} \partial \mathbf{v}'} \cdot \left[\frac{F_{s'}(\mathbf{v}')}{m_{s'}} \frac{\partial F_s}{\partial \mathbf{v}} - \frac{F_s(\mathbf{v})}{m_s} \frac{\partial F_{s'}}{\partial \mathbf{v}'} \right] d^3 \mathbf{v}' \right\}, \end{aligned} \quad (2)$$

where

$$\gamma_{ss'} = \frac{2\pi Z_s^2 Z_{s'}^2 e^4 \ln \Lambda_{ss'}}{(4\pi\epsilon_0)^2}, \quad (3)$$

with $\ln \Lambda_{ss'}$ the Coulomb logarithm [2, 3, 4, 5],

$$g = |\mathbf{v} - \mathbf{v}'|, \quad (4)$$

and $\int d^3 \mathbf{v}'$ denotes a definite integral over all $\mathbf{v}'$. Note that it is common to write $\partial^2 g / \partial \mathbf{v} \partial \mathbf{v}' = (\mathbf{I}g^2 - \mathbf{g}\mathbf{g})/g^3$, with $\mathbf{g} = \mathbf{v} - \mathbf{v}'$ and $\mathbf{I}$ the identity matrix. The operator (2) is widely used in plasma physics and magnetic confinement fusion studies [4, 5]. Despite the complex integro-differential structure of the operator, the Landau form of the Fokker–Planck operator (2) possesses four key properties that we note. First, the collision operator

conserves particle density, i.e.,

$$\int C_{ss'} [F_s, F_{s'}] d^3\mathbf{v} = 0. \quad (5)$$

Second, the collision operator conserves the total momentum in a collision, i.e.,

$$\int (m_s \mathbf{v} C_{ss'} [F_s, F_{s'}] + m_{s'} \mathbf{v} C_{s's} [F_{s'}, F_s]) d^3\mathbf{v} = \mathbf{0}. \quad (6)$$

The same is true for the total energy:

$$\int \left(\frac{1}{2} m_s |\mathbf{v}|^2 C_{ss'} [F_s, F_{s'}] + \frac{1}{2} m_{s'} |\mathbf{v}|^2 C_{s's} [F_{s'}, F_s] \right) d^3\mathbf{v} = 0. \quad (7)$$

Finally, Boltzmann's H-theorem applied to same-species collisions [5] proves that the entropy production

$$\dot{S}_s = - \int \ln F_s C_{ss} [F_s, F_s] d^3\mathbf{v} \geq 0, \quad (8)$$

with equality if and only if F_s is a Maxwellian distribution described by the local density n_s , mean velocity $\mathbf{u}_s$, and temperature T_s , i.e.,

$$F_s = F_{Ms} = \frac{n_s}{\pi^{3/2} v_{th,s}^3} \exp \left[- \left(\frac{\mathbf{v} - \mathbf{u}_s}{v_{th,s}} \right)^2 \right], \quad (9)$$

with $v_{th,s} = \sqrt{2T_s/m_s}$.

Implementing the nonlinear Fokker–Planck collision operator numerically is challenging because of the nonlinear and integro-differential nature of the operator. For a given distribution function F_s , we must carry out a series of difficult integrals to find the coefficients of the operator. Whilst previous authors have implemented the nonlinear Fokker–Planck collision operator, see, e.g., [6, 7, 8, 9, 10, 11, 12, 13], including implementations of the underlying Boltzmann operator [14], it is more typical to either (i) write down an ad-hoc diffusive model operator which may be solved rapidly, yet still has the conservation or H-theorem properties desired for the physics of interest [15, 16, 17, 18]; or (ii) use asymptotic expansions in physics parameters to linearise the kinetic equation (including both the collision operator and the convective left hand side) around a known Maxwellian distribution function for use in a specific application (e.g., transport theory or collisional closures [4, 5, 19, 20]). Such a linearised model that solves for small perturbations to F_s is often referred to as a δF model, in contrast to full- F models that aim to solve for the entire distribution function without linearisations.

In applications relating to hot plasma turbulence on closed magnetic field lines, collisional relaxation timescales are typically long compared to the nonlinear turnover time of the turbulent eddies: energy is injected into the turbulence at velocity scales comparable to the thermal speed, and energy is dissipated at much smaller velocity scales set by velocity diffusion via inter-particle collisions. When the dissipation scale is well separated from the injection scale – near-collisionless plasma

turbulence – details of the velocity-space dissipation mechanism do not affect the large-scale turbulent transport of interest. In these situations, a linearised Fokker–Planck operator or an ad-hoc model operator is an appropriate and relatively inexpensive operator that captures the dissipation of fine velocity-space structure [16, 17].

Systems with closed magnetic field lines are known to be approximately in thermal equilibrium because the system is approximately closed [20, 21], meaning that the distribution function is never far from the Maxwellian around which the collision operator is usually linearised. However, in certain applications, the distribution function of the plasma may be far from Maxwellian: for example, in beam-driven plasmas [22, 23], or in the scrape-off layer of a tokamak plasma [24, 25, 26, 27, 28]. In the latter example, this is due to the presence of the divertor plate or limiter [29], which intercepts the field lines at the edge of the plasma and so makes the system open, preventing local thermal equilibrium. In addition, hot particles may transit rapidly from the hot, closed-magnetic-field-line region of the plasma to the open magnetic field lines at the edge where the plasma is expected to be cooler, potentially resulting in a bimodal distribution of particle energies: the steady state distribution is not known. Therefore, it is not clear whether or not a model or linearised collision operator is adequate for modelling the plasma on open field lines. The only rigorous choice is the nonlinear Fokker–Planck operator.

In this paper we describe the implementation of the nonlinear Fokker–Planck operator appropriate for use in a drift-kinetic [30] model of a magnetised plasma. We will choose a higher-order continuous-Galerkin (C^0) finite-element representation for the distribution function to permit a spectrally-accurate polynomial representation whilst retaining the ability to describe boundary layers in velocity space. This representation does not attempt to guarantee positivity of F_s . The specialisation to drift-kinetics allows us to consider only gyrotropic distribution functions, which, due to rapid gyromotion around the field line, are independent of the gyrophase angle ϑ that measures the position of the particle in the plane perpendicular to the magnetic field. We support $F_s = F_s(v_{\parallel}, v_{\perp})$, with the cylindrical velocity space coordinates $(v_{\parallel}, v_{\perp}, \vartheta)$ defined by

$$v_{\parallel} = \mathbf{v} \cdot \mathbf{b}, \quad v_{\perp} = |\mathbf{v} - v_{\parallel} \mathbf{b}|, \quad \tan \vartheta = - \frac{\mathbf{v} \cdot \mathbf{e}_2}{\mathbf{v} \cdot \mathbf{e}_1}, \quad (10)$$

or equivalently,

$$\mathbf{v} = v_{\parallel} \mathbf{b} + v_{\perp} \mathbf{e}_{\perp}, \quad (11)$$

with $\mathbf{e}_{\perp} = (\cos \vartheta \mathbf{e}_1 - \sin \vartheta \mathbf{e}_2)$. The basis vector $\mathbf{b} = \mathbf{B}/|\mathbf{B}|$ is the unit vector in the direction of the magnetic field. The vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ are orthogonal to $\mathbf{b}$ and satisfy

$$\mathbf{b} \cdot \mathbf{e}_1 \times \mathbf{e}_2 = 1, \quad \mathbf{e}_1 \cdot \mathbf{b} = 0, \quad \mathbf{e}_2 \cdot \mathbf{b} = 0. \quad (12)$$

The numerical implementation described in this paper ensures the near-exact satisfaction of the conservation properties (5)–(7) by achieving high accuracy with the weak formulations and adequate numerical resolution. To avoid carrying out costly numerical integration in $\mathbf{v}'$ in the whole of the velocity

space, as required by the definition (2), we use the Rosenbluth–MacDonald–Judd (RMJ) form of the collision operator [3], given in the next section, where the velocity integrals are rewritten as Rosenbluth potentials: the Rosenbluth potentials may be obtained by solving elliptic PDEs using the higher-order finite-element method, with boundary conditions obtained by direct integration using the formal definitions of the Rosenbluth potentials at the limits of the velocity space. This numerical strategy optimises the scheme for scalability.

We emphasise that the novel contribution of this work is the demonstration that higher-order finite-element methods such as those used by mature computational fluid dynamics frameworks, e.g., [31], can achieve a scalable, conservative implementation of the nonlinear Fokker–Planck collision operator. Higher-order finite-element methods which can achieve accuracy by refinement of both element size and polynomial order, h - p refinement, should be considered as numerical methods for models within plasma physics [32]. The remainder of this paper is structured as follows. In the next section, we write the collision operator in the RMJ form. In section 3, we obtain the weak-form representation of the problem that we will implement numerically. In section 4 we prescribe numerical-error-correcting terms to ensure that conservative properties (5)–(7) are satisfied to machine precision in the time advance. In section 5 we provide results from numerical tests of our implementation. In section 6, we discuss the outlook for the use of the operator in a production code. Appendix A, Appendix B and Appendix C contain useful results pertaining to the calculation of the Rosenbluth potentials.

2. RMJ form of the collision operator

The operator in the RMJ form [3] in $(v_{\parallel}, v_{\perp})$ coordinates is most usefully written in terms of collisional fluxes:

$$C_{ss'}[F_s, F_{s'}] = \frac{\partial \Gamma_{\parallel}}{\partial v_{\parallel}} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} (v_{\perp} \Gamma_{\perp}), \quad (13)$$

where the fluxes are defined by

$$\Gamma_{\parallel} = \frac{\gamma_{ss'}}{m_s^2} \left(\frac{\partial F_s}{\partial v_{\parallel}} \frac{\partial^2 G_{s'}}{\partial v_{\parallel}^2} + \frac{\partial F_s}{\partial v_{\perp}} \frac{\partial^2 G_{s'}}{\partial v_{\perp} \partial v_{\parallel}} - 2 \frac{m_s}{m_{s'}} F_s \frac{\partial H_{s'}}{\partial v_{\parallel}} \right), \quad (14)$$

and

$$\Gamma_{\perp} = \frac{\gamma_{ss'}}{m_s^2} \left(\frac{\partial F_s}{\partial v_{\parallel}} \frac{\partial^2 G_{s'}}{\partial v_{\parallel} \partial v_{\perp}} + \frac{\partial F_s}{\partial v_{\perp}} \frac{\partial^2 G_{s'}}{\partial v_{\perp}^2} - 2 \frac{m_s}{m_{s'}} F_s \frac{\partial H_{s'}}{\partial v_{\perp}} \right), \quad (15)$$

and the Rosenbluth potentials are

$$G_{s'}(\mathbf{v}) = \int F_{s'}(\mathbf{v}') g \, d^3 \mathbf{v}' \quad (16)$$

and

$$H_{s'}(\mathbf{v}) = \int \frac{F_{s'}(\mathbf{v}')}{g} \, d^3 \mathbf{v}'. \quad (17)$$

In the drift-kinetic limit the largest piece of the distribution functions are independent of gyroangle [4], i.e., $F_s = F_s(v_{\parallel}, v_{\perp})$

and $F_{s'} = F_{s'}(v_{\parallel}, v_{\perp})$. In terms of $(v_{\parallel}, v_{\perp})$ coordinates, for gyrotropic distributions the Rosenbluth potentials simplify to

$$G_{s'} = \int_{-\infty}^{\infty} \int_0^{\infty} 4U^{1/2} E(m) F_{s'}(v'_{\parallel}, v'_{\perp}) v'_{\perp} \, dv'_{\perp} dv'_{\parallel}, \quad (18)$$

and

$$H_{s'} = \int_{-\infty}^{\infty} \int_0^{\infty} 4U^{-1/2} K(m) F_{s'}(v'_{\parallel}, v'_{\perp}) v'_{\perp} \, dv'_{\perp} dv'_{\parallel}, \quad (19)$$

where

$$U = U(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) = (v_{\parallel} - v'_{\parallel})^2 + (v_{\perp} + v'_{\perp})^2, \quad (20)$$

and

$$m = m(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) = \frac{4v_{\perp} v'_{\perp}}{U(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp})}, \quad (21)$$

and we have used the definitions of the complete elliptic integral of the first and second kinds,

$$K(m) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - m \sin^2 \theta}} \, d\theta \quad (22)$$

and

$$E(m) = \int_0^{\pi/2} \sqrt{1 - m \sin^2 \theta} \, d\theta, \quad (23)$$

respectively.

2.1. Finding elliptic problems for the Rosenbluth potentials

As noted in the original derivation by Rosenbluth, MacDonald, and Judd [3], the potentials defined by equations (16) and (17) may also be defined as the solutions of the elliptic problems

$$\frac{\partial^2 G}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial G}{\partial v_{\perp}} \right) = 2H, \quad (24)$$

and

$$\frac{\partial^2 H}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial H}{\partial v_{\perp}} \right) = -4\pi F. \quad (25)$$

Given a known F , we recognise the Poisson's equation for the Rosenbluth potential H , and the biharmonic equation for G . Obtaining the Rosenbluth potentials through an elliptic solve is potentially numerically advantageous compared to evaluating the formal definitions directly, see, e.g., (16) and (17). This is because several algorithms with $O(N^2)$ or $O(N^2 \ln N^2)$ complexity for solving Poisson equations are known [33, 34, 35, 36], where N is of order the number of points in the array that represents one of the velocity dimensions, whereas a direct evaluation of the formal definition leads to a scaling of $O(N^4)$ due to the need to evaluate an integral with a kernel function that is a function of $(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp})$.

In practice, for a finite simulation domain, boundary conditions for the elliptic solve must be supplied on the bounding surfaces where $v_{\parallel} = \pm V_{\parallel}$ and $v_{\perp} = V_{\perp}$, with $V_{\parallel}$ the maximum $|v_{\parallel}|$ on grid, and $V_{\perp}$ the maximum $v_{\perp}$ on grid. This necessitates $O(N^3)$ operations to obtain the boundary data through direct integration using the formal definition of the appropriate

Rosenbluth potential. Parallelisation over many processes may be able to alleviate the cost of obtaining boundary data, as this part of the calculation is embarrassingly parallel.

Having motivated the use of sparse elliptic solvers to find the Rosenbluth potentials, it remains to formulate the appropriate PDEs for the derivatives of the Rosenbluth potentials appearing in the fluxes (14) and (15). We choose to solve for the derivatives of the Rosenbluth potentials directly: this avoids derivatives in the scheme higher than second order, which improves numerical accuracy. Henceforth, we will also refer to the derivatives of the Rosenbluth potentials as Rosenbluth potentials. First, we introduce the shorthand notation $G_{20} = \partial^2 G / \partial v_{\parallel}^2$, $G_{02} = \partial^2 G / \partial v_{\perp}^2$, $G_{11} = \partial^2 G / \partial v_{\parallel} \partial v_{\perp}$, $H_{10} = \partial H / \partial v_{\parallel}$, and $H_{01} = \partial H / \partial v_{\perp}$. Direct differentiation of equations (24) and (25) yields the required differential definitions of the Rosenbluth potentials. Taking care to write the results in forms that are easy to integrate by parts through the identity

$$\frac{\partial^2 G}{\partial v_{\perp}^2} = \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial G}{\partial v_{\perp}} \right) - \frac{1}{v_{\perp}} \frac{\partial G}{\partial v_{\perp}}, \quad (26)$$

and making liberal use of equations (24) and (25), we find that useful equations for the Rosenbluth potentials derived from G are

$$\frac{\partial^2 G_{20}}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial G_{20}}{\partial v_{\perp}} \right) = 2 \frac{\partial^2 H}{\partial v_{\parallel}^2}, \quad (27)$$

$$\frac{\partial^2 G_{11}}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial G_{11}}{\partial v_{\perp}} \right) - \frac{G_{11}}{v_{\perp}^2} = 2 \frac{\partial^2 H}{\partial v_{\parallel} \partial v_{\perp}}, \quad (28)$$

and

$$\begin{aligned} \frac{\partial^2 G_{02}}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial G_{02}}{\partial v_{\perp}} \right) - \frac{4G_{02}}{v_{\perp}^2} \\ = \frac{2}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial H}{\partial v_{\perp}} \right) - \frac{2}{v_{\perp}} \frac{\partial H}{\partial v_{\perp}} - \frac{4H}{v_{\perp}^2} + \frac{2G_{20}}{v_{\perp}^2}. \end{aligned} \quad (29)$$

Note that we must obtain H via (25) to make use of equations (27) and (28). Equation (29) also requires the solution to (27). For the derivatives of H , we similarly obtain the PDEs

$$\frac{\partial^2 H_{10}}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial H_{10}}{\partial v_{\perp}} \right) = -4\pi \frac{\partial F}{\partial v_{\parallel}}, \quad (30)$$

and

$$\frac{\partial^2 H_{01}}{\partial v_{\parallel}^2} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} \left(v_{\perp} \frac{\partial H_{01}}{\partial v_{\perp}} \right) - \frac{H_{01}}{v_{\perp}^2} = -4\pi \frac{\partial F}{\partial v_{\perp}}. \quad (31)$$

Note that we have written the elliptic equations (27), (28), (29), (30), and (31), in a form that will be amenable to integration by parts in the test-function analysis required for a weak-form implementation. The numerical implementation of these equations first uses equations (25), (30) and (31) to find H and its derivatives from F . Then, equations (27), (28), and (29) may be solved for the derivatives of G .

2.2. Evaluating the Rosenbluth potential boundary data

To obtain the boundary data required to solve the elliptic problems (27), (28), (29), (30), and (31), we must obtain formal definitions for each of the required Rosenbluth potentials by differentiating the formal definitions (16) and (17). We use integration by parts to obtain formal definitions where the integration kernels are very similar to those appearing in the integrals (16) and (17), which can be carried out with known methods once they are expressed in the forms (18) and (19).

We start by computing

$$\frac{\partial G_{s'}}{\partial \mathbf{v}} = \int F_{s'}(\mathbf{v}') \frac{\partial g}{\partial \mathbf{v}} d^3 \mathbf{v}' = - \int F_{s'}(\mathbf{v}') \frac{\partial g}{\partial \mathbf{v}'} d^3 \mathbf{v}', \quad (32)$$

where we have used that $\partial g / \partial \mathbf{v} = -\partial g / \partial \mathbf{v}'$. Using integration by parts, and that $F_{s'}(\mathbf{v}') \rightarrow 0$ as $|\mathbf{v}'| \rightarrow \infty$, we find that

$$\frac{\partial G_{s'}}{\partial \mathbf{v}} = \int \frac{\partial F_{s'}}{\partial \mathbf{v}'} g d^3 \mathbf{v}'. \quad (33)$$

We can use the same method to find that

$$\frac{\partial H_{s'}}{\partial \mathbf{v}} = \int \frac{\partial F_{s'}}{\partial \mathbf{v}'} \frac{1}{g} d^3 \mathbf{v}'. \quad (34)$$

Direct differentiation of equation 33 shows that

$$\frac{\partial^2 G_{s'}}{\partial \mathbf{v} \partial \mathbf{v}} = \int \frac{\partial F_{s'}}{\partial \mathbf{v}'} \frac{\mathbf{v} - \mathbf{v}'}{g} d^3 \mathbf{v}'. \quad (35)$$

Equations (33) and (34) are vector equations and (35) is a tensor equation. We extract the required results by taking dot products with the unit vectors $\mathbf{b}$ and $\mathbf{e}_{\perp}$, noting that

$$\frac{\partial}{\partial \mathbf{v}} = \mathbf{b} \frac{\partial}{\partial v_{\parallel}} + \mathbf{e}_{\perp} \frac{\partial}{\partial v_{\perp}} + \frac{\mathbf{e}_{\perp} \times \mathbf{b}}{v_{\perp}} \frac{\partial}{\partial \theta}. \quad (36)$$

Assuming that $F_{s'} = F_{s'}(v'_{\parallel}, v'_{\perp})$, which implies that $G_{s'} = G_{s'}(v_{\parallel}, v_{\perp})$, we find that

$$\frac{\partial^2 G_{s'}}{\partial v_{\parallel} \partial v_{\perp}} = 2\pi \int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial^2 F_{s'}}{\partial v'_{\parallel} \partial v'_{\perp}} I_{G1} v'_{\perp} dv'_{\perp} dv'_{\parallel}, \quad (37)$$

$$\frac{\partial H_{s'}}{\partial v_{\parallel}} = 2\pi \int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial F_{s'}}{\partial v'_{\parallel}} I_{H0} v'_{\perp} dv'_{\perp} dv'_{\parallel}, \quad (38)$$

$$\frac{\partial H_{s'}}{\partial v_{\perp}} = 2\pi \int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial F_{s'}}{\partial v'_{\perp}} I_{H1} v'_{\perp} dv'_{\perp} dv'_{\parallel}, \quad (39)$$

$$\frac{\partial^2 G_{s'}}{\partial v_{\parallel}^2} = 2\pi \int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial F_{s'}}{\partial v'_{\parallel}} (v_{\parallel} - v'_{\parallel}) I_{H0} v'_{\perp} dv'_{\perp} dv'_{\parallel}, \quad (40)$$

and

$$\frac{\partial^2 G_{s'}}{\partial v_{\perp}^2} = 2\pi \int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial F_{s'}}{\partial v'_{\perp}} (v_{\perp} I_{H1} - v'_{\perp} I_{H2}) v'_{\perp} dv'_{\perp} dv'_{\parallel}, \quad (41)$$

where

$$I_{G1} = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(\mathbf{e}_{\perp} \cdot \mathbf{e}'_{\perp}) d\theta', \quad (42)$$

$$I_{H0} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{g} d\theta', \quad (43)$$

$$I_{H1} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\mathbf{e}_{\perp} \cdot \mathbf{e}'_{\perp}}{g} d\vartheta', \quad (44)$$

and

$$I_{H2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{(\mathbf{e}_{\perp} \cdot \mathbf{e}'_{\perp})^2}{g} d\vartheta'. \quad (45)$$

The main advantage of this formulation is that the integrands have rather simple numerators and denominators. Note that inspection of the integral (19) reveals that the integrand there diverges logarithmically as $m \rightarrow 1$ since $K \rightarrow \ln(4/\sqrt{1-m})$ as $m \rightarrow 1$. The kernels (43)-(45) also diverge logarithmically where $v'_{\parallel} = v_{\parallel}$ and $v'_{\perp} = v_{\perp}$: this kind of divergence can be handled numerically by a change of variables in the affected elements [37]. The functions I_{G1} , I_{H0} , I_{H1} , and I_{H2} are evaluated explicitly in Appendix A.

3. Obtaining the weak formulation of the problem

We consider the collisional relaxation problem

$$\frac{\partial F}{\partial t} = \frac{\partial \Gamma_{\parallel}}{\partial v_{\parallel}} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} (v_{\perp} \Gamma_{\perp}). \quad (46)$$

in $v_{\parallel} \in [-V_{\parallel}, V_{\parallel}]$, $v_{\perp} \in [0, V_{\perp}]$ and t , where $V_{\parallel}$ and $V_{\perp}$ are the maximum values of $v_{\parallel}$ and $v_{\perp}$ on the grid, respectively. The solution $F = F(v_{\parallel}, v_{\perp}, t)$, and the fluxes $\Gamma_{\parallel} = \Gamma_{\parallel}(v_{\parallel}, v_{\perp}, t) = \Gamma_{\parallel}[F(v_{\parallel}, v_{\perp}, t)]$ and $\Gamma_{\perp} = \Gamma_{\perp}(v_{\parallel}, v_{\perp}, t) = \Gamma_{\perp}[F(v_{\parallel}, v_{\perp}, t)]$ are functionals of F . We note that the fluxes in velocity space are defined explicitly by equations (14) and (15). As we only consider the self-collision operator in what follows, we neglect the species index s in the following analysis where it is convenient to do so.

We divide the domain into a rectangular grid of $N_{2D} = N_{\parallel}N_{\perp}$ elements. We use $N_{\parallel}$ 1D elements in the $v_{\parallel}$ direction and $N_{\perp}$ 1D elements in the $v_{\perp}$ direction. Each 2D element is a tensor product of two 1D elements. On each 1D element we represent the function with Lagrange polynomials of order N_p using the $N_{GR} = N_p + 1$ (normalised) grid points within the elements

$$x_j \in \{x_0, x_1, \dots, x_{N_p-1}, x_{N_p}\}, \quad (47)$$

with $x_0 = -1$ and $x_{N_p} = 1$ (Gauss–Legendre–Lobatto quadrature points [38]) on elements that do not include $v_{\perp} = 0$. On the element including $v_{\perp} = 0$, we take $x_{N_p} = 1$ but we use Gauss–Legendre–Radau quadrature points to ensure that $x_0 > -1$ [38].

The transformation between $(v_{\parallel}, v_{\perp})$ and the local coordinate $x^{(r)}$ in the r^{th} 1D element is

$$v_{\parallel} = s_{\parallel}^{(r)} x^{(r)} + c_{\parallel}^{(r)}, \quad v_{\perp} = s_{\perp}^{(r)} x^{(r)} + c_{\perp}^{(r)} \quad (48)$$

where $s_{\parallel}^{(r)}$, $c_{\parallel}^{(r)}$, $s_{\perp}^{(r)}$ and $c_{\perp}^{(r)}$ are constants in each element (labelled here by r) which may vary between elements, and $x^{(r)} \in [-1, 1]$ for all r , except in the element that includes the origin of $v_{\perp}$, which has $x^{(r)} \in (-1, 1]$.

3.1. The basis functions

We introduce tensor-product 2D basis functions

$$\Phi_{jk}^{(rp)}(v_{\parallel}, v_{\perp}) = \varphi_j^{(r)}(v_{\parallel}) \varphi_k^{(p)}(v_{\perp}), \quad (49)$$

where the 1D basis functions are

$$\varphi_j^{(r)}(v) = l_j(x^{(r)}(v)) \Theta(v - v(x_0^{(r)})) \Theta(v(x_{N_p}^{(r)}) - v), \quad (50)$$

with l_j the j^{th} Lagrange polynomial on the element, v a placeholder for either $v_{\parallel}$ or $v_{\perp}$, and $\Theta(v)$ the Heaviside function. Expanding the solution in these basis functions, we write

$$\begin{aligned} F(v_{\parallel}, v_{\perp}) &= \sum_{r,p} \sum_{j,k} F_{jk}^{rp} \Phi_{jk}^{(rp)}(v_{\parallel}, v_{\perp}) \\ &= \sum_{r,p} \sum_{j,k} F_{jk}^{rp} \varphi_j^{(r)}(v_{\parallel}) \varphi_k^{(p)}(v_{\perp}), \end{aligned} \quad (51)$$

with

$$F_{jk}^{rp} = F(v_{\parallel}(x_j^{(r)}), v_{\perp}(x_k^{(p)})). \quad (52)$$

Note that the basis functions have the cardinality property

$$\varphi_j^{(r)}(v(x_k^{(p)})) = \delta_{jk} \delta_{rp}, \quad (53)$$

with $\delta_{jk} = 1$ if $j = k$ and 0 if $j \neq k$. This choice of basis functions does not guarantee positivity of the distribution function F – this scheme relies on spectral accuracy implied by the higher-order polynomial representation to obtain convergence to physical solutions. In particular, note that the interpolant for F may be negative even if F_{jk}^{rp} are positive, due to oscillations in the Lagrange polynomial basis functions.

3.2. The projection onto the polynomial basis

To project equation (46) onto the basis functions $\Phi_{jk}^{(rp)}(v_{\parallel}, v_{\perp})$, we multiply by the basis function $\Phi_{mn}^{(qs)}(v_{\parallel}, v_{\perp})$, and integrate over velocity space corresponding to a single 2D element. The limits of this element are $v_{\parallel U}^{(q)} = v_{\parallel}(x_{N_p}^{(q)})$, $v_{\parallel L}^{(q)} = v_{\parallel}(x_0^{(q)})$, $v_{\perp U}^{(s)} = v_{\perp}(x_{N_p}^{(s)})$, and $v_{\perp L}^{(s)} = v_{\perp}(x_0^{(s)})$, respectively. The strong-form equation (46) is then replaced by the weak-form equation

$$\begin{aligned} \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \Phi_{mn}^{(qs)} \frac{\partial F}{\partial t} v_{\perp} dv_{\perp} dv_{\parallel} &= \\ \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \Phi_{mn}^{(qs)} \left(\frac{\partial \Gamma_{\parallel}}{\partial v_{\parallel}} + \frac{1}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} (v_{\perp} \Gamma_{\perp}) \right) v_{\perp} dv_{\perp} dv_{\parallel}, \end{aligned} \quad (54)$$

valid for each of the 2D basis functions retained in the scheme.

3.3. The mass matrix

The left hand side of equation (54) takes the form

$$\int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \Phi_{mn}^{(qs)} \frac{\partial F}{\partial t} v_{\perp} dv_{\perp} dv_{\parallel} = \sum_{j,k} M_{mj}^{(q)} M_{nk}^{(s)} \frac{\partial F_{jk}^{qs}}{\partial t}, \quad (55)$$

where we have defined the 1D parallel and perpendicular mass matrices

$$\begin{aligned} M_{mj}^{(q)} &= \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \varphi_m^{(q)}(v_{\parallel}) \varphi_j^{(q)}(v_{\parallel}) dv_{\parallel} \\ &= s_{\parallel}^{(q)} \int_{-1}^1 l_m(x) l_j(x) dx, \end{aligned} \quad (56)$$

and

$$\begin{aligned} M_{\perp nk}^{(s)} &= \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \varphi_n^{(s)}(v_{\perp}) \varphi_k^{(s)}(v_{\perp}) v_{\perp} dv_{\perp} \\ &= s_{\perp}^{(s)} \int_{-1}^1 l_n(x) l_k(x) (s_{\perp}^{(s)} x + c_{\perp}^{(s)}) dx, \end{aligned} \quad (57)$$

respectively. In equations (56) and (57) we have indicated for completeness how to evaluate these matrix elements in terms of Lagrange polynomials.

3.4. The nonlinear stiffness matrices for the collision operator

The form of the right hand side of equation (54) and the forms of the fluxes, given by equations (14) and (15), respectively, suggest that we should integrate by parts to bring all derivatives down to first order. Carrying out this step, we find that for the parallel flux term

$$\begin{aligned} &\int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \Phi_{mn}^{(qs)} \frac{\partial \Gamma_{\parallel}}{\partial v_{\parallel}} v_{\perp} dv_{\perp} dv_{\parallel} = \\ &\delta_{mN_P} \int_{v_{\perp L}^{(q)}}^{v_{\perp U}^{(q)}} \Phi_{mn}^{(qs)}(v_{\parallel U}^{(s)}, v_{\perp}) \Gamma_{\parallel}(v_{\parallel U}^{(s)}, v_{\perp}) v_{\perp} dv_{\perp} \\ &- \delta_{m0} \int_{v_{\perp L}^{(q)}}^{v_{\perp U}^{(q)}} \Phi_{mn}^{(qs)}(v_{\parallel L}^{(s)}, v_{\perp}) \Gamma_{\parallel}(v_{\parallel L}^{(s)}, v_{\perp}) v_{\perp} dv_{\perp} \\ &- \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \frac{\partial \Phi_{mn}^{(qs)}}{\partial v_{\parallel}} \Gamma_{\parallel} v_{\perp} dv_{\perp} dv_{\parallel}. \end{aligned} \quad (58)$$

Similarly, for the perpendicular flux term, we have that

$$\begin{aligned} &\int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \frac{\Phi_{mn}^{(qs)}}{v_{\perp}} \frac{\partial}{\partial v_{\perp}} (v_{\perp} \Gamma_{\perp}) v_{\perp} dv_{\perp} dv_{\parallel} = \\ &= \delta_{nN_P} \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \Phi_{mn}^{(qs)}(v_{\parallel}, v_{\perp U}^{(s)}) v_{\perp U}^{(s)} \Gamma_{\perp}(v_{\parallel}, v_{\perp U}^{(s)}) dv_{\parallel} \\ &- \delta_{n0} \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \Phi_{mn}^{(qs)}(v_{\parallel}, v_{\perp L}^{(s)}) v_{\perp L}^{(s)} \Gamma_{\perp}(v_{\parallel}, v_{\perp L}^{(s)}) dv_{\parallel} \\ &- \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \frac{\partial \Phi_{mn}^{(qs)}}{\partial v_{\perp}} \Gamma_{\perp} v_{\perp} dv_{\perp} dv_{\parallel}. \end{aligned} \quad (59)$$

The boundary flux terms in equations (58) and (59) will cancel identically at the assembly stage, vanish at $v_{\perp} = 0$, or vanish at $v_{\perp} = V_{\perp}$, $v_{\parallel} = -V_{\parallel}$, and $v_{\parallel} = V_{\parallel}$ by the boundary conditions that $F \rightarrow 0$ as $|\mathbf{v}| \rightarrow \infty$.

We are now in a position to write down the matrix row for an arbitrary polynomial. We use the expansion (51) for both

the distribution function F and the Rosenbluth potentials. With these choices, recalling the definitions of the fluxes $\Gamma_{\parallel}$ and $\Gamma_{\perp}$, equations (14) and (15), respectively, the result is

$$\begin{aligned} \sum_{j,k} M_{mj}^{(q)} M_{\perp nk}^{(s)} \frac{\partial F_{jk}^{qs}}{\partial t} = & - \frac{\gamma_{ss'}}{m_s^2} \sum_{jklr} F_{jk}^{qs} \left[\left[\frac{\partial^2 G_{s'}}{\partial v_{\parallel}^2} \right]_{lr}^{(qs)} Y_{\parallel 2mjl}^{(q)} Y_{\perp 0nkr}^{(s)} \right. \\ & + \left[\frac{\partial^2 G_{s'}}{\partial v_{\perp} \partial v_{\parallel}} \right]_{lr}^{(qs)} Y_{\parallel 1mjl}^{(q)} Y_{\perp 3nkr}^{(s)} - 2 \frac{m_s}{m_{s'}} \left[\frac{\partial H_{s'}}{\partial v_{\parallel}} \right]_{lr}^{(qs)} Y_{\parallel 1mlj}^{(q)} Y_{\perp 0nkr}^{(s)} \\ & + \left[\frac{\partial^2 G_{s'}}{\partial v_{\perp} \partial v_{\parallel}} \right]_{lr}^{(qs)} Y_{\parallel 3mjl}^{(q)} Y_{\perp 1nkr}^{(s)} + \left[\frac{\partial^2 G_{s'}}{\partial v_{\perp}^2} \right]_{lr}^{(qs)} Y_{\parallel 0mjl}^{(q)} Y_{\perp 2nkr}^{(s)} \\ & \left. - 2 \frac{m_s}{m_{s'}} \left[\frac{\partial H_{s'}}{\partial v_{\perp}} \right]_{lr}^{(qs)} Y_{\parallel 0mjl}^{(q)} Y_{\perp 1nkr}^{(s)} \right], \end{aligned} \quad (60)$$

where we have neglected the interior boundary terms as vanishing at the assembly stage, and we have neglected the extreme boundary terms to impose the natural boundary condition on F that $\Gamma_{\parallel}(v_{\parallel} = \pm V_{\parallel}, v_{\perp}) = 0$ and $\Gamma_{\perp}(v_{\parallel}, v_{\perp} = V_{\perp}) = 0$. This is equivalent to imposing $F = 0$ as $V_{\parallel}, V_{\perp} \rightarrow \infty$. In equation (60), the stiffness matrices with three indices are

$$\begin{aligned} Y_{\parallel 0mjl}^{(q)} &= \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \varphi_m^{(q)} \varphi_j^{(q)} \varphi_l^{(q)} dv_{\parallel}, \\ Y_{\parallel 1mjl}^{(q)} &= \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \frac{\partial \varphi_m^{(q)}}{\partial v_{\parallel}} \varphi_j^{(q)} \varphi_l^{(q)} dv_{\parallel}, \\ Y_{\parallel 2mjl}^{(q)} &= \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \frac{\partial \varphi_m^{(q)}}{\partial v_{\parallel}} \frac{\partial \varphi_j^{(q)}}{\partial v_{\parallel}} \varphi_l^{(q)} dv_{\parallel}, \\ Y_{\parallel 3mjl}^{(q)} &= \int_{v_{\parallel L}^{(q)}}^{v_{\parallel U}^{(q)}} \varphi_m^{(q)} \frac{\partial \varphi_j^{(q)}}{\partial v_{\parallel}} \varphi_l^{(q)} dv_{\parallel}, \end{aligned} \quad (61)$$

and

$$\begin{aligned} Y_{\perp 0nkr}^{(s)} &= \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \varphi_n^{(s)} \varphi_k^{(s)} \varphi_r^{(s)} v_{\perp} dv_{\perp}, \\ Y_{\perp 1nkr}^{(s)} &= \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \frac{\partial \varphi_n^{(s)}}{\partial v_{\perp}} \varphi_k^{(s)} \varphi_r^{(s)} v_{\perp} dv_{\perp}, \\ Y_{\perp 2nkr}^{(s)} &= \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \frac{\partial \varphi_n^{(s)}}{\partial v_{\perp}} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} \varphi_r^{(s)} v_{\perp} dv_{\perp}, \\ Y_{\perp 3nkr}^{(s)} &= \int_{v_{\perp L}^{(s)}}^{v_{\perp U}^{(s)}} \varphi_n^{(s)} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} \varphi_r^{(s)} v_{\perp} dv_{\perp}. \end{aligned} \quad (62)$$

Note that the stiffness matrices in (60) are all 1D integrals of 1D basis functions, as a result of the choice to use the representation (51) where the 2D basis function $\Phi_{mn}^{(qs)}(v_{\parallel}, v_{\perp})$ is a product of two 1D Lagrange polynomials – one for the $v_{\parallel}$ dimension, and one for the $v_{\perp}$ dimension.

The assembly step is carried out using the usual element-wise finite-element assembly algorithm, by defining a com-

pound index that indexes over the nonzero entries in the assembled matrix equation. We use continuity of F to demand that $F_{jN_p}^{q,s} = F_{j0}^{q,s+1}$, $F_{N_p k}^{q,s} = F_{0k}^{q+1,s}$, and remove the duplicated points at interior element boundaries by summing the matrix rows there.

3.5. The weak form of the equations for the Rosenbluth potentials

We need to determine the coefficients derived from the Rosenbluth potentials. We start by considering the solution of Poisson's equation, equation (25). Multiplying by the 2D basis function $\Phi_{jk}^{(rp)} = \varphi_j^{(r)}(v_{\parallel}) \varphi_k^{(p)}(v_{\perp})$ and integrating over velocity space, integrating by parts on the left-hand side and neglecting the boundary terms we have that

$$\begin{aligned} & - \int_{v_{\parallel}^{(r)}} \int_{v_{\perp}^{(p)}} \left(\frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\parallel}} \frac{\partial H}{\partial v_{\parallel}} + \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\perp}} \frac{\partial H}{\partial v_{\perp}} \right) v_{\perp} dv_{\perp} dv_{\parallel} \\ & = -4\pi \int_{v_{\parallel}^{(r)}} \int_{v_{\perp}^{(p)}} \Phi_{jk}^{(rp)} F v_{\perp} dv_{\perp} dv_{\parallel}. \end{aligned} \quad (63)$$

We can neglect boundary terms because the assembly step will cancel all terms due to interior boundaries, whereas exterior boundary terms only appear in rows that will be replaced by a Dirichlet boundary condition. Defining the matrices

$$K_{\perp nk}^{(s)} = - \int_{v_{\perp}^{(s)}} \frac{\partial \varphi_n^{(s)}}{\partial v_{\perp}} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} v_{\perp} dv_{\perp} \quad (64)$$

and

$$K_{\parallel nk}^{(s)} = - \int_{v_{\parallel}^{(s)}} \frac{\partial \varphi_n^{(s)}}{\partial v_{\parallel}} \frac{\partial \varphi_k^{(s)}}{\partial v_{\parallel}} dv_{\parallel} \quad (65)$$

and expanding

$$F = \sum_{rp} \sum_{jk} \Phi_{jk}^{(rp)} F_{jk}^{rp}, \quad H = \sum_{rp} \sum_{jk} \Phi_{jk}^{(rp)} H_{jk}^{rp}, \quad (66)$$

we find that the row of the unassembled matrix is

$$\sum_{mn} \left(K_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} + K_{\perp kn}^{(p)} M_{\parallel jm}^{(r)} \right) H_{mn}^{rp} = -4\pi \sum_{mn} M_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} F_{mn}^{rp}. \quad (67)$$

We impose Dirichlet boundary conditions on the assembled matrices using the values of the required functions computed numerically by direct integration, as outlined in section 2.2. Once the coefficients H_{jk}^{rp} are known then the same matrices can be used in an identical fashion to solve for G_{jk}^{rp} . A similar matrix equation can be written down to solve for G_{20}^{rp} and H_{10}^{rp} , the components of $G_{20} = \partial^2 G / \partial v_{\parallel}^2$ and $H_{10} = \partial H / \partial v_{\parallel}$, respectively, with the only difference being in the source terms on the right-hand side. Explicitly, these results are

$$\sum_{mn} \left(K_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} + K_{\perp kn}^{(p)} M_{\parallel jm}^{(r)} \right) G_{20mn}^{rp} = 2 \sum_{mn} K_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} H_{mn}^{rp}, \quad (68)$$

and

$$\sum_{mn} \left(K_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} + K_{\perp kn}^{(p)} M_{\parallel jm}^{(r)} \right) H_{10mn}^{rp} = -4\pi \sum_{mn} P_{\parallel jm}^{(r)} M_{\perp kn}^{(p)} F_{mn}^{rp}. \quad (69)$$

where we have defined

$$P_{\parallel mj}^{(r)} = \int_{v_{\perp}^{(r)}} \varphi_m^{(r)} \frac{\partial \varphi_j^{(r)}}{\partial v_{\parallel}} dv_{\parallel}. \quad (70)$$

To improve numerical accuracy, we choose to find $\partial H / \partial v_{\parallel}$ by a separate Poisson solve rather than by differentiating H .

To find the equations for the other Rosenbluth potentials in the fluxes, we must repeat the exercise above. The PDE defining $G_{11} = \partial^2 G / \partial v_{\parallel} \partial v_{\perp}$ is distinct from Poisson's equation and will require different matrix elements. We follow identical steps as used to derive the matrix row equation for H_{jk}^{rp} . First, we integrate over velocity space to obtain the weak form equation

$$\begin{aligned} & - \int_{v_{\parallel}^{(r)}} \int_{v_{\perp}^{(p)}} \left(v_{\perp}^2 \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\parallel}} \frac{\partial G_{11}}{\partial v_{\parallel}} + v_{\perp}^2 \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\perp}} \frac{\partial G_{11}}{\partial v_{\perp}} \right. \\ & \quad \left. + v_{\perp} \Phi_{jk}^{(rp)} \frac{\partial G_{11}}{\partial v_{\perp}} + \Phi_{jk}^{(rp)} G_{11} \right) dv_{\perp} dv_{\parallel} \\ & = 2 \int_{v_{\parallel}^{(r)}} \int_{v_{\perp}^{(p)}} v_{\perp}^2 \Phi_{jk}^{(rp)} \frac{\partial^2 H}{\partial v_{\parallel} \partial v_{\perp}} dv_{\perp} dv_{\parallel}, \end{aligned} \quad (71)$$

and we use this to define the unassembled matrix row equation:

$$\begin{aligned} & \sum_{mn} \left(K_{\parallel jm}^{(r)} R_{\perp kn}^{(p)} + M_{\parallel jm}^{(r)} J_{\perp kn}^{(p)} - M_{\parallel jm}^{(r)} P_{\perp kn}^{(p)} - M_{\parallel jm}^{(r)} N_{\perp kn}^{(p)} \right) G_{11mn}^{rp} \\ & = 2 \sum_{mn} P_{\parallel jm}^{(r)} U_{\perp kn}^{(p)} H_{mn}^{rp}, \end{aligned} \quad (72)$$

where we have defined the matrix elements

$$\begin{aligned} R_{\perp nk}^{(s)} &= \int_{v_{\perp}^{(s)}} \varphi_n^{(s)}(v_{\perp}) \varphi_k^{(s)}(v_{\perp}) v_{\perp}^2 dv_{\perp}, \\ N_{\perp nk}^{(s)} &= \int_{v_{\perp}^{(s)}} \varphi_n^{(s)}(v_{\perp}) \varphi_k^{(s)}(v_{\perp}) dv_{\perp}, \\ J_{\perp nk}^{(s)} &= - \int_{v_{\perp}^{(s)}} \frac{\partial \varphi_n^{(s)}}{\partial v_{\perp}} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} v_{\perp}^2 dv_{\perp}, \\ P_{\perp nk}^{(s)} &= \int_{v_{\perp}^{(s)}} \varphi_n^{(s)} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} v_{\perp} dv_{\perp}, \\ U_{\perp nk}^{(s)} &= \int_{v_{\perp}^{(s)}} \varphi_n^{(s)} \frac{\partial \varphi_k^{(s)}}{\partial v_{\perp}} v_{\perp}^2 dv_{\perp}. \end{aligned} \quad (73)$$

Similarly, the PDE for $H_{01} = \partial H / \partial v_{\perp}$, equation (31), has the weak form

$$\begin{aligned} & \sum_{mn} \left(K_{\parallel jm}^{(r)} R_{\perp kn}^{(p)} + M_{\parallel jm}^{(r)} J_{\perp kn}^{(p)} - M_{\parallel jm}^{(r)} P_{\perp kn}^{(p)} - M_{\parallel jm}^{(r)} N_{\perp kn}^{(p)} \right) H_{01mn}^{rp} \\ & = -4\pi \sum_{mn} M_{\parallel jm}^{(r)} U_{\perp kn}^{(p)} F_{mn}^{rp}. \end{aligned} \quad (74)$$

To complete the set of Rosenbluth potentials, we solve the elliptic problem (29) for $G_{02} = \partial^2 G / \partial v_{\perp}^2$. Following steps

similar to those above, we obtain the weak-form equation

$$\begin{aligned}
& - \int_{v_{\parallel L}^{(r)}}^{v_{\parallel U}^{(r)}} \int_{v_{\perp L}^{(p)}}^{v_{\perp U}^{(p)}} \left(v_{\perp}^2 \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\parallel}} \frac{\partial G_{02}}{\partial v_{\parallel}} + v_{\perp}^2 \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\perp}} \frac{\partial G_{02}}{\partial v_{\perp}} \right. \\
& \quad \left. - v_{\perp} \Phi_{jk}^{(rp)} \frac{\partial G_{02}}{\partial v_{\perp}} - 4 \Phi_{jk}^{(rp)} G_{02} \right) dv_{\perp} dv_{\parallel} \\
& = -2 \int_{v_{\parallel L}^{(r)}}^{v_{\parallel U}^{(r)}} \int_{v_{\perp L}^{(p)}}^{v_{\perp U}^{(p)}} \left(v_{\perp}^2 \frac{\partial \Phi_{jk}^{(rp)}}{\partial v_{\perp}} \frac{\partial H}{\partial v_{\perp}} + 2 v_{\perp} \Phi_{jk}^{(rp)} \frac{\partial H}{\partial v_{\perp}} \right. \\
& \quad \left. + 2 \Phi_{jk}^{(rp)} H - \Phi_{jk}^{(rp)} G_{20} \right) dv_{\perp} dv_{\parallel}.
\end{aligned} \tag{75}$$

The corresponding row of the unassembled matrix therefore becomes

$$\begin{aligned}
& \sum_{mn} \left(K_{\parallel jm}^{(r)} R_{\perp kn}^{(p)} + M_{\parallel jm}^{(r)} (J_{\perp kn}^{(p)} - P_{\perp kn}^{(p)} - 4 N_{\perp kn}^{(p)}) \right) G_{02mn}^{rp} \\
& = 2 \sum_{mn} M_{\parallel jm}^{(r)} (J_{\perp kn}^{(p)} - 2 P_{\perp kn}^{(p)} - 2 N_{\perp kn}^{(p)}) H_{mn}^{rp} \\
& \quad + 2 \sum_{mn} M_{\parallel jm}^{(r)} N_{\perp kn}^{(p)} G_{20mn}^{rp}.
\end{aligned} \tag{76}$$

3.6. Velocity space integration in the spectral element scheme

To compute the boundary data for the elliptic problems obtained in the last section, we need to integrate a function $F = F(v_{\parallel}, v_{\perp})$ multiplied by a kernel function $\mathcal{G} = \mathcal{G}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp})$: we wish to compute

$$I = \int_{-\infty}^{\infty} \int_0^{\infty} \mathcal{G}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) F(v'_{\parallel}, v'_{\perp}) v'_{\perp} dv'_{\perp} dv'_{\parallel}. \tag{77}$$

We expand F in the Lagrange polynomial basis functions using equation (51) and thus obtain that

$$I = \sum_{rp} \sum_{jk} F_{jk}^{rp} I_{jk}^{(rp)} \tag{78}$$

with the integration over local elements

$$I_{jk}^{(rp)} = \int_{v_{\parallel L}^{(r)}}^{v_{\parallel U}^{(r)}} \int_{v_{\perp L}^{(p)}}^{v_{\perp U}^{(p)}} \mathcal{G}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) \varphi_j^{(r)}(v'_{\parallel}) \varphi_k^{(p)}(v'_{\perp}) v'_{\perp} dv'_{\perp} dv'_{\parallel}. \tag{79}$$

When assembling the integral over elements, one must recall that the nodal value of F at element boundaries has an interpolating polynomial that contributes to the quadrature in two elements.

4. Numerical-error-correcting terms

The numerical scheme for evolving the distribution function due to the Fokker–Planck collision operator is chosen here for performance and scalability, rather than for exact machine-precision satisfaction of the conservative properties (5)–(7). To ensure that the numerical scheme for the collision operator can also preserve the density, parallel velocity and total energy at each time step to machine precision, for time-evolving simulations we introduce ad-hoc conserving terms which make a

correction which is at most of order of the discretisation error. Noting the definitions of the plasma density n_s and parallel flow $u_{\parallel, s}$,

$$n_s = \int \int F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{80}$$

and

$$n_s u_{\parallel, s} = \int \int v_{\parallel} F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{81}$$

respectively, we define

$$\begin{aligned}
C_{ss} [F_s, F_s] &= C_{ss}^* [F_s, F_s] \\
&\quad - \left(z_0 + z_1 (v_{\parallel} - u_{\parallel, s}) + z_2 ((v_{\parallel} - u_{\parallel, s})^2 + v_{\perp}^2) \right) F_s,
\end{aligned} \tag{82}$$

where $C_{ss}^* [F_s, F_s]$ denotes the numerically calculated finite-element collision operator given by $\partial F / \partial t$ in equation (60), and the coefficients z_0, z_1 and z_2 are determined by the requirements that (5)–(7) are exactly satisfied. Instead of formulating the equations for $\{z_j\}$ in terms of total energy

$$\varepsilon_s = \frac{m_s}{2} \int \int (v_{\parallel}^2 + v_{\perp}^2) F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{83}$$

we write them in terms of pressure p_s , using the definition $\varepsilon_s = 3p_s/2 + m_s n_s u_{\parallel, s}^2/2$. Then, the conservation laws (5)–(7) lead to the matrix equation

$$\begin{pmatrix} m_s n_s & 0 & 3p_s \\ 0 & p_{\parallel, s} & q_{\parallel, s} \\ 3p_s & q_{\parallel, s} & \bar{R}_s \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} m_s \Delta n_s \\ m_s (n_s \Delta u_{\parallel, s} - u_{\parallel, s} \Delta n_s) \\ 3 \Delta p_s \end{pmatrix}, \tag{84}$$

where the vector components on the right hand side are the moments of $C_{ss}^* [F_s, F_s]$ that should vanish up to discretisation error, i.e.,

$$\begin{aligned}
\Delta n_s &= \int \int C_{ss}^* [F_s, F_s] 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \\
\Delta u_{\parallel, s} &= \frac{1}{n_s} \int \int v_{\parallel} C_{ss}^* [F_s, F_s] 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \\
\Delta p_s &= \frac{m_s}{3} \int \int ((v_{\parallel} - u_{\parallel, s})^2 + v_{\perp}^2) C_{ss}^* [F_s, F_s] 2\pi v_{\perp} dv_{\perp} dv_{\parallel},
\end{aligned} \tag{85}$$

and the components of the matrix on the left-hand side are given by the moments of F_s . We have that the total pressure $p_s = (2p_{\perp, s} + p_{\parallel, s})/3$ with the parallel and perpendicular pressures given by

$$p_{\parallel, s} = m_s \int \int (v_{\parallel} - u_{\parallel, s})^2 F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{86}$$

and

$$p_{\perp, s} = \frac{m_s}{2} \int \int v_{\perp}^2 F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{87}$$

respectively. The parallel heat flux is given by

$$q_{\parallel, s} = m_s \int \int (v_{\parallel} - u_{\parallel, s}) ((v_{\parallel} - u_{\parallel, s})^2 + v_{\perp}^2) F_s 2\pi v_{\perp} dv_{\perp} dv_{\parallel}, \tag{88}$$

and the higher-order moment

$$\tilde{R}_s = m_s \iint ((v_{\parallel} - u_{\parallel,s})^2 + v_{\perp}^2)^2 F_s \, 2\pi v_{\perp} dv_{\perp} dv_{\parallel}. \quad (89)$$

We will demonstrate in the next section that Δn_s , $\Delta u_{\parallel,s}$, and Δp_s defined in equation (85) are indeed bounded by the discretisation error.

We note the similarity of these error-correcting terms to those employed for similar reasons where the density, parallel flow, and pressure are required to be conserved exactly [13, 39].

5. Numerical implementation and results

We have implemented an explicit form of the weak-form collision operator in Julia [40], in the drift-kinetic version of the pre-sheath plasma code “moment.kinetics” [41, 42, 43], using shared-memory parallelism implemented with MPI using features introduced in MPI-3 [44], in particular `MPI.Win_allocate_shared` [45]. Specifically, we have implemented the assembled weak-form problems defined by equations (60), (67), (68), (69), (72), (74), (76), using sparse matrices [46], and with appropriate calculations of the Rosenbluth potential boundary data using equations (19) and (37)-(41) with integration weights defined by (79). The basic quadrature used to calculate these weights is a tensor product of two 1D Gauss–Legendre quadratures, except near logarithmic divergences, where Gauss–Laguerre quadratures are employed with changes of variables. We use Gauss–Legendre polynomials to define the Lobatto and Radau collocation grid points. We have implemented the scheme for arbitrary positive order of polynomials.

The source code for the implementation is available and documented [47]. The test and simulation results that are presented in the remainder of this section are supported by publicly available data [48].

5.1. Evaluation tests

We wish to test the three properties of the collision operator (5)-(7). To facilitate this test we define three quantities which measure the change in the moments of the distribution function due to the collision operator, given by equation (85). We test in a grid resolution scan whether or not the collision operator vanishes on a prescribed Maxwellian distribution up to expected discretisation error, i.e., whether or not the operator well satisfies

$$C_{ss} [F_{Ms}, F_{Ms}] = 0. \quad (90)$$

We now proceed to describe the details of the test. In figure 1 we carry out the resolution test for varying $N_{\text{EL}} = N_{\perp} = N_{\parallel}/2$ at fixed $N_{\text{GR}} = 5$, corresponding to 4th order polynomials, although we support any $N_{\text{GR}} \geq 2$. Here N_{EL} is the number of elements in the $v_{\perp}$ dimension and half the number of elements in the $v_{\parallel}$ dimension. The quantity N_{GR} is the number of points per 1D element, in both the parallel and perpendicular dimensions. We take the maximum velocity to be $V_{\parallel} = V_{\perp} = 6c_{\text{ref}}$, with $c_{\text{ref}} = \sqrt{2T_{\text{ref}}/m_{\text{ref}}}$, where T_{ref} and m_{ref} are the reference

Quantity	Normalisation
n_s	n_{ref}
$u_{\parallel,s}$	c_{ref}
p_s	$m_{\text{ref}} c_{\text{ref}}^2$
$v_{\parallel}$	c_{ref}
$v_{\perp}$	c_{ref}
F_s	$n_{\text{ref}}/\pi^{3/2} c_{\text{ref}}^3$
G_s	$n_{\text{ref}} c_{\text{ref}}$
H_s	$n_{\text{ref}}/c_{\text{ref}}$

Table 1: List of normalisations used in the numerical implementation of the self-collision operator.

temperature and reference mass, respectively. Note that reducing $V_{\parallel}$ and $V_{\perp}$ for a fixed integrand reduces the accuracy of the numerical integration because the true velocity integrals should extend to infinite velocities. We choose to carry out the test for a species of mass $m_s/m_{\text{ref}} = 1$ with a Maxwellian with a normalised density $n_s/n_{\text{ref}} = 1$, a normalised parallel flow $u_{\parallel,s}/c_{\text{ref}} = 1$, and a normalised temperature $T_s/T_{\text{ref}} = 1$. In figure 1, and in the remainder of the paper, we plot normalised quantities, with the normalisations given in table 1.

In figure 1, we plot both the infinity norm of the error ϵ_{∞} and the L_2 norm of the error ϵ_2 of calculating the collision operator with respect to the expected value (which is zero). Here, the infinity norm of the error in a normalised distribution F is defined by

$$\epsilon_{\infty}(F) = \max_v |F - F_{\text{EXACT}}|, \quad (91)$$

with F_{EXACT} the normalised, exact, analytically computed value. The L_2 norm of the error is defined by

$$\epsilon_2(F) = \sqrt{\frac{\int (F - F_{\text{EXACT}})^2 v_{\perp} dv_{\perp} dv_{\parallel}}{\int v_{\perp} dv_{\perp} dv_{\parallel}}}, \quad (92)$$

where the integration is carried out numerically over the finite range of velocities on the grid. We see that the infinity norm gives a larger value than the L_2 norm in all cases by a factor of an order of magnitude. This is due to numerical oscillations near $v_{\perp} = 0$ where the differential equations become singular. The error in computing the collision operator decreases according to expected scaling for error in differentiation

$$\left(\frac{1}{N_{\text{EL}}}\right)^{N_{\text{GR}}-1}. \quad (93)$$

The quantities Δn_s , $\Delta u_{\parallel,s}$, and Δp_s approach zero rapidly at (or better than) the expected scaling for numerical integration errors

$$\left(\frac{1}{N_{\text{EL}}}\right)^{N_{\text{GR}}+1}. \quad (94)$$

We have carried out this test using multiple other values of N_{GR} and demonstrated the same results. To demonstrate the attained performance of the explicit collision operator, in figure 2, we plot the timing data (in milliseconds) for completing the initialisation and evaluation of the collision operator on 2 cores.

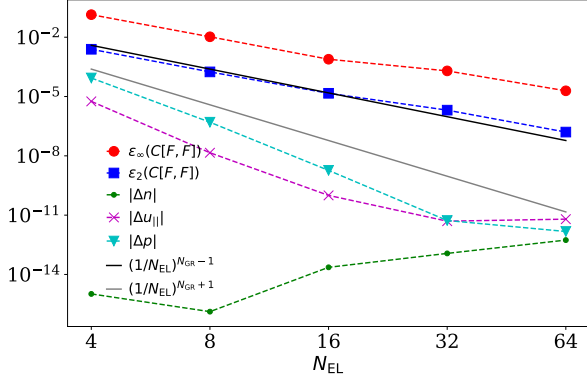


Figure 1: The numerical error for a test carried out with $N_{\text{GR}} = 5$ points per element. The infinity norm of the error (ϵ_∞ , see (91)) and L_2 norm of the error (ϵ_2 , see (92)) of evaluating the collision operator are shown and compared to the expected scalings for differentiation and integration (93) and (94), respectively. This test does not use the numerical-error-correcting terms that are introduced in section 4, in order to show the numerical errors in the conserved moments native to the finite-element scheme.

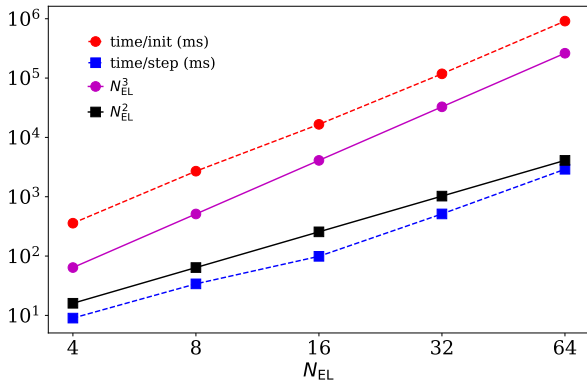


Figure 2: For the test shown in figure 1 carried out on 2 cores, timing data for the initialisation (init) and a single evaluation of the collision operator (step) is given in milliseconds.

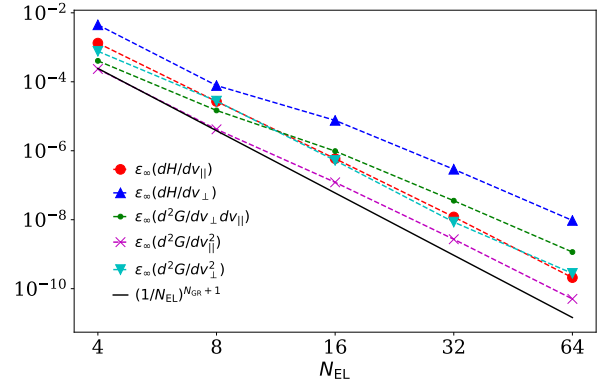


Figure 3: The infinity norm of the error ϵ_∞ in computing the Rosenbluth potentials for a shifted Maxwellian distribution, for $N_{\text{GR}} = 5$.

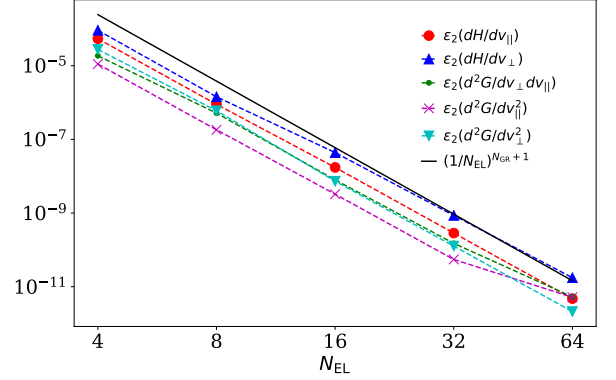


Figure 4: The L_2 norm of the error ϵ_2 in computing the Rosenbluth potentials for a shifted Maxwellian distribution, for $N_{\text{GR}} = 5$.

The expected scaling for the initialisation is N_{EL}^3 , by virtue of the calculation of the integration weights for the boundary data. The expected scaling for the evaluation of the collision operator depends on which operation dominates the calculation. If it is the computation of the boundary data (using equations (37)–(41) with integration weights defined by (79)) it is N_{EL}^3 , whereas if it is the elliptic solve or the assembly of the right hand side of equation (60) then the scaling would be expected to be N_{EL}^2 due to the sparse nature of these operations. We see that a scaling close to N_{EL}^2 is achieved for the evaluation step.

To understand the dominant source of the numerical error in the collision operator, we find it useful to plot the infinity and L_2 norm error measures of the numerically calculated Rosenbluth potentials $\partial H/\partial v_{\parallel}$, $\partial H/\partial v_{\perp}$, $\partial^2 G/\partial v_{\parallel}^2$, $\partial G/\partial v_{\perp}$, $\partial^2 G/\partial v_{\parallel} \partial v_{\perp}$, and $\partial^2 G/\partial v_{\perp}^2$. The exact values are known for shifted Maxwellian distributions, see, e.g., Appendix B. We plot the error in our numerical calculation of the potentials data for $N_{\text{GR}} = 5$ in figures 3 and 4. The L_2 norm error is smaller by one or two orders of magnitude than the infinity norm error. This is due to numerical oscillations near $v_{\perp} = 0$. However, in both cases the errors decay to zero approximately at the rate given by (94). Note that our numerical calculation of the boundary data does involve a numerical differentiation of F , see equa-

Simulation	#1	#2	#3
$V_{\parallel}/c_{\text{ref}} = V_{\perp}/c_{\text{ref}}$	2.25	3	4
$N_{\text{EL}} = N_{\perp} = N_{\parallel}/2$	4	8	16
$\nu_{ss}\Delta t$	10^{-3}	10^{-3}	0.25×10^{-3}
time/step (ms)	7	21	89
# cores	4	16	64

Table 2: List of resolution parameters used in the numerical simulation of the self-collision operator, time per step in the simulation in milliseconds, and the number of cores used in the simulation.

tions (37)-(41), which may explain why the rate of convergence is slower for some Rosenbluth potentials. For completeness, in Appendix C we show the numerical error resulting from computing the potentials from direct integration.

We take the results in figures 3 and 4 to indicate that the dominant source of numerical error in evaluating the collision operator comes from numerical differentiation, rather than from the errors in obtaining the Rosenbluth potentials. Indeed, comparable levels of error to that seen in computing the collision operator may be obtained by simply using the weak method to differentiate F to find the second derivatives in $v_{\parallel}$ and $v_{\perp}$.

5.2. Relaxation to a Maxwellian distribution

It is important to test whether or not the numerical self-collision operator can provide a stable, steady-state numerical solution which is close to a Maxwellian distribution, with positive-definite entropy production, equation (8). In this section we show tests where we integrate in time the collisional relaxation problem (46), in weak form (60), both with and without the numerical-error-correcting terms (82). We show that the numerical-error-correcting terms are only necessary to ensure that we can obtain a stable solution for low numerical resolution. We impose only natural boundary conditions on F by neglecting the boundary terms introduced by integration by parts in the projection onto the weak basis. For explicit expressions for these terms, see equations (58) and (59).

For the time evolving tests, we initialise the normalised distribution function F to

$$F = n_0 \exp \left[-\frac{v_{\parallel}^2 + (v_{\perp} - c_{\text{ref}})^2}{c_{\text{ref}}^2/4} \right], \quad (95)$$

with n_0 a dimensionless factor calculated to ensure that the density of F satisfies $n = 1$ (see equation (80)). The form in equation (95) was chosen to provide a substantially non-Maxwellian initial distribution function. The time integration scheme is the Runge–Kutta strong-stability-preserving 4-stage method [49, 50, 51]. We use a collision frequency $\nu_{ss} = \gamma_{ss} n_{\text{ref}} / m_s^2 c_{\text{ref}}^3$, and we run for a time of $200/\nu_{ss}$. We use the resolutions detailed in table 2 to generate three different simulations with increasing resolution and maximum velocity on grid, keeping $N_{\text{GR}} = 5$ and varying $V_{\parallel} = V_{\perp}$ and N_{EL} .

5.2.1. Collisional evolution without numerical-error-correcting terms

In figure 5, we show time traces of the change in the density, parallel flow and pressure over the course of a simulation where the numerical-error-correcting terms are not employed. The pressure is not well conserved at long times for low resolution. This is to be anticipated from the results in figure 1 where the change in the pressure was not as small as the change in the density. The velocity remains well conserved here because the simulation is symmetric in $v_{\parallel}$. Despite the lack

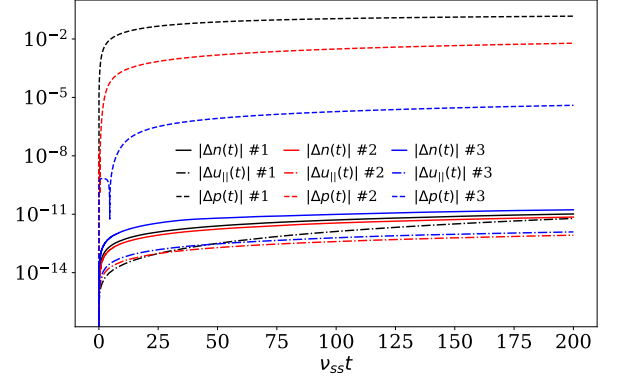


Figure 5: The changes in the first three moments of the distribution function n_s , $u_{\parallel,s}$, and p_s as a result of time evolution with the Fokker–Planck collision operator defined by equation (60), i.e., without the numerical-error-correcting terms. The pressure moment is not well conserved at low resolution, as would be expected from figure 1. The resolutions for simulations #1, #2, and #3 are provided in table 2

of exact numerical conservation of the moments, the collision operator without numerical-error-correcting terms still pushes the distribution towards a Maxwellian distribution, as shown in figure 6, where we show the infinity norm of $F - F_M(t)$, $L_{\infty}(F - F_M(t)) = \max_v |F - F_M(t)|$, where the Maxwellian distribution is constructed with the time-evolving values of the moments. The maximum in the infinity norm is taken over the set of nodal values of F , F_{jk}^{tp} . Simulations with increasing numerical resolution have a smaller $L_{\infty}(F - F_M)$, suggesting a strong convergence. These results show that numerical-error-correcting terms are thus only required to fix the solution to a Maxwellian distribution with the exact same density, mean velocity and pressure as the initial condition.

5.2.2. Collisional evolution with numerical-error-correcting terms

We now show the result of evolving equation (60) with the numerical-error-correcting terms (82). In figure 7, we show time traces of the change in the density, parallel flow and pressure over the course of the simulation. The small errors in the moments increase with increasing resolution, suggesting conservation accurate to machine precision, with the error coming from round-off errors.

In figure 8 we show the infinity norm $L_{\infty}(F - F_M(t))$, demonstrating a strong convergence, with very similar values for the infinity norm as in figure 6. This suggests that the

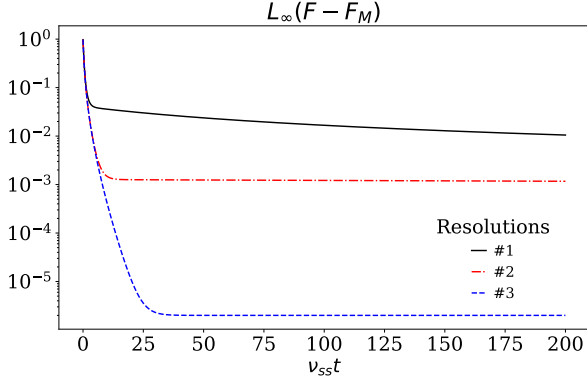


Figure 6: The infinity norm of $F - F_M(t)$, $L_\infty(F - F_M(t)) = \max_v |F - F_M(t)|$, where the Maxwellian is computed at each timestep with the updated values of the pressure, velocity and density moments. Despite the loss of pressure conservation shown in figure 5, this figure indicates that F_s becomes increasingly close to a Maxwellian distribution. The resolutions for simulations #1, #2, and #3 are provided in table 2.

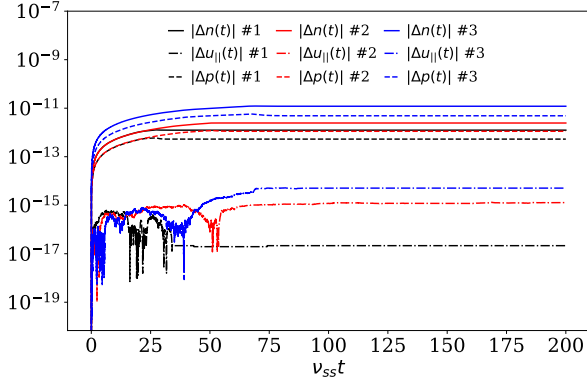


Figure 7: The changes in the first three moments of the distribution function n_s , $u_{||,s}$, and p_s as a result of time evolution with the Fokker–Planck collision operator defined by equations (60) and (82) (i.e., with the numerical conserving terms). The moments are well conserved, despite the low resolution used. The resolutions for simulations #1, #2, and #3 are provided in table 2

numerical-error-correcting terms act to conserve the moments to machine-precision without a detrimental effect on the shape of the distribution function.

Finally, in figure 9 we show the entropy production for the simulation using numerical-error-correcting terms, calculated using the definition (8) and using the following approximation for the logarithm of the distribution function

$$\ln F = \sum_{rp} \sum_{ij} \ln(|F_{jk}^{rp}| + \epsilon) \Phi_{jk}^{(rp)}, \quad (96)$$

where $\epsilon = 10^{-15}$. The approximation (96) is adequate if the solution is converging with increasing resolution in a strong sense. Figure 9 shows positive-definite entropy production, although there is no guarantee that $\dot{S}$ should be a positive-definite quantity in our numerical scheme.

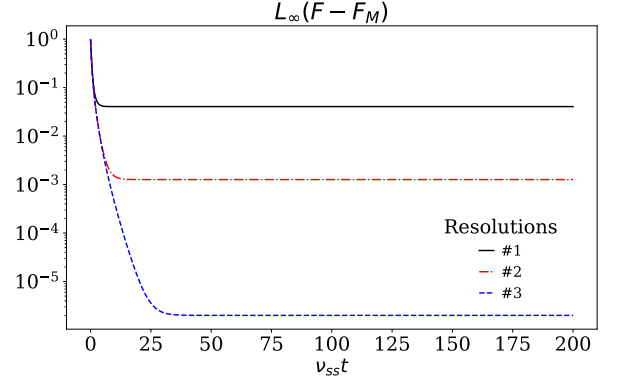


Figure 8: The infinity norm of $F - F_M$, $L_\infty(F - F_M) = \max_v |F - F_M|$, for the test including numerical-error-correcting terms. This figure indicates that F becomes increasingly close to F_M before converging on a steady-state numerical Maxwellian distribution. The resolutions for simulations #1, #2, and #3 are provided in table 2.

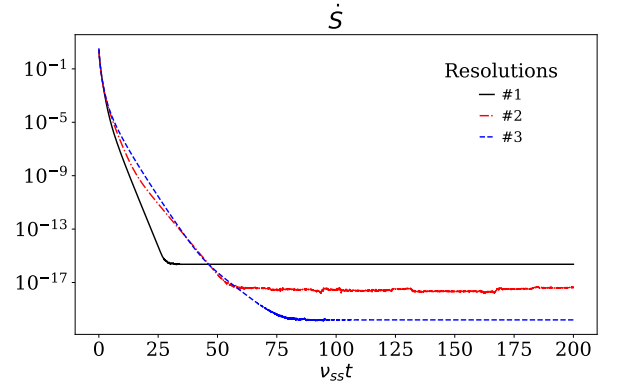


Figure 9: The entropy production $\dot{S}$, for the test including numerical-error-correcting terms. The entropy production $\dot{S}$ is defined in equation (8), where $\ln F$ is calculated using an approximation, equation (96). Note that $\dot{S}$ remains positive and tends to 0^+ . The resolutions for simulations #1, #2, and #3 are provided in table 2.

6. Discussion

In this paper we have investigated a particular finite-element, continuous-Galerkin, weak-form representation of the explicit nonlinear Fokker–Planck collision operator. We choose to use the Rosenbluth–MacDonald–Judd form of the Fokker–Planck operator to permit the use of sparse elliptic solves for determining the Rosenbluth potentials of the nonlinear operator. We have demonstrated that this choice can lead to an optimal scaling of the cost of evaluating the operator for a single time step $\propto N_{\text{EL}}^2$, with N_{EL} the number of elements in a single velocity space dimension, $v_{||}$ or $v_{\perp}$. We have implemented and tested the method for arbitrary number of points per element $N_{\text{GR}} \geq 2$, but we only present results for $N_{\text{GR}} = 5$ for brevity. We also demonstrated a successful time-evolving simulation with low resolution, demonstrating that the self-collision operator can successfully relax the distribution function to a stable steady state that is close to a Maxwellian distribution.

Table 2 indicates that the time per step of the collision opera-

tor is potentially small enough to be of the correct order of magnitude to permit the use of the collision operator in drift-kinetic turbulence simulations where the spatial domain is fully distributed in memory. In future work, the time evolving “moment kinetics” framework will be leveraged to include the Fokker–Planck collision operator in 1, 2, and 3 dimensional simulations of plasma on open field lines. The extension of the operator here to include cross-species collisions is readily achievable by adding further contributions to the Rosenbluth potentials.

To help permit the collision operator to be routinely used alongside other physics features we could further optimise the implementation for speed to accommodate larger problem sizes. This might be achieved with an extension from shared-memory-only parallelism to allow for distributed-memory parallelisation across nodes. The use of distributed memory to parallelise the collision operator calculation is motivated by the observation that the dominant costs which contribute to the time taken to evaluate the operator are the calculation of the boundary data and the assembly of the right-hand side of equation (60). Both of these steps are embarrassingly parallel.

Alternatively, we might consider potential optimisations of the numerical method. First, the boundary data for the elliptic solves may be determined using a multipole expansion of the formal definition of G and H , equations (16) and (17), respectively. This method may permit the evaluation of the boundary data using only an order unity number of velocity integrals, providing the maximum value of $v_{\parallel}$ and $v_{\perp}$ on the grid, $V_{\parallel}$ and $V_{\perp}$, respectively, are sufficiently large. Second, the boundary data may be evaluated at fewer locations and a larger-scale interpolation of the Rosenbluth potentials on the boundary might be constructed. This might save computation time without sacrificing significant accuracy, if $V_{\parallel}$ and $V_{\perp}$ are large enough for the Rosenbluth potentials to have a simple functional form on the boundary. Third, one could choose to use a different interpolation scheme defining the right-hand side of equation (60). One could consider using the quadrature rules commonly employed in spectral element codes that yield diagonal mass matrices [52] to reduce the number of operations due to the nonlinear stiffness matrices defined by equations (61) and (62). Fourth, to enable stable solutions with timestep sizes not limited by diffusion in $v_{\parallel}$ and $v_{\perp}$, one could consider replacing the explicit time-integration method presented here with an implicit time integrator, see, e.g., [53].

Finally, it is useful to consider the benefits and limitations of the finite-element method presented here for Fokker–Planck collisions, to assess the suitability of the method for integration with a solver that treats the Vlasov part of the kinetic equation and the large-scale electric and magnetic fields. The primary benefit of the higher-order finite element method is the potential for spectrally accurate solutions in the presence of boundary layers in velocity space. However, the method does not guarantee positivity of the solution. The structure of the finite-element mass and stiffness matrices means that sparse matrices can be used. To store the matrices needed for the 2D Poisson solvers for the Rosenbluth potentials, we only require $O(N_{\text{EL}}^2 N_{\text{GR}}^4)$ matrix entries, and for each spatial point we will require to store an order unity number of arrays of the size of the 2D distribution

function F , i.e., $O(N_{\text{EL}}^2 N_{\text{GR}}^2)$ values. Had we chosen to use velocity coordinates with explicit spatial variation (e.g., magnetic moment $\mu = v_{\perp}^2/2|B|$), then the coefficients in the equations for the Rosenbluth potentials would depend on spatial position and we would need to store a matrix of size $O(N_{\text{EL}}^2 N_{\text{GR}}^4)$ for each location in the spatial grid. Compared to a linear advection operator, the nonlinear collision operator in two velocity coordinates is more expensive to evaluate. As a result, a model which combines both nonlinear collisions and linear advection will likely be limited by the cost of evaluating the Fokker–Planck operator, unless care is taken to design methods where the Fokker–Planck operator is evaluated infrequently.

CRediT authorship contribution statement

M. R. Hardman: Writing - Original Draft, Writing - Review & Editing, Investigation, Formal analysis, Visualization, Conceptualization, Software. **M. Abazorius:** Formal analysis, Conceptualization. **J. Omotani:** Writing - Review & Editing, Formal analysis, Conceptualization, Software, Resources. **M. Barnes:** Writing - Review & Editing, Resources, Project administration, Funding acquisition, Supervision. **S. L. Newton:** Writing - Review & Editing. **J. W. S. Cook:** Writing - Review & Editing. **P. E. Farrell:** Writing - Review & Editing, Formal analysis. **F. I. Parra:** Conceptualization.

Data availability

Data supporting the results presented in this study are publicly available [48].

Acknowledgements

The authors are grateful for valuable discussions with W. Arter and E. Threlfall.

Funding sources

M. R. Hardman was employed by Tokamak Energy Ltd. **M. R. Hardman** and **M. Barnes** were supported by the United Kingdom Atomic Energy Authority ExCALIBUR programme grant. The ExCALIBUR programme (<https://excalibur.ac.uk/>) is supported by the UKRI Strategic Priorities Fund. The programme is co-delivered by the Met Office and EPSRC in partnership with the Public Sector Research Establishment, the United Kingdom Atomic Energy Authority and UKRI research councils, including NERC, MRC and STFC.

P. E. Farrell was supported by EPSRC grants EP/R029423/1 and EP/W026163/1, the Donatio Universitatis Carolinae Chair “Mathematical modelling of multicomponent systems”, and the UKRI Digital Research Infrastructure Programme through the Science and Technology Facilities Council’s Computational Science Centre for Research Communities (CoSeC).

This work has been part funded by the EPSRC Energy Programme [grant number EP/W006839/1].

This work was supported by the U.S. Department of Energy

under contract number DE-AC02-09CH11466. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Appendix A. Evaluating the gyroaveraged functions

To see how to evaluate the required gyroaveraged functions I_{G1} , I_{H0} , I_{H1} , and I_{H2} , defined in equations (42)-(45), consider

$$I_{G0} = \frac{1}{2\pi} \int_{-\pi}^{\pi} g \, d\vartheta'. \quad (\text{A.1})$$

Expanding g , we have that

$$I_{G0}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left((v_{\parallel} - v'_{\parallel})^2 + v_{\perp}^2 + v'_{\perp}{}^2 - 2v_{\perp}v'_{\perp} \cos(\vartheta' - \vartheta) \right)^{1/2} d\vartheta'. \quad (\text{A.2})$$

Here we can recognise an elliptic integral. Suitable rearrangement and relabeling gives us

$$I_{G0}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) = \frac{2}{\pi} U^{1/2} E(m) \quad (\text{A.3})$$

with U and m functions of $(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp})$ defined by equations (20) and (21), respectively, and the elliptic integral of the second kind is defined by equation (23). Noting that $\mathbf{e}_{\perp} \cdot \mathbf{e}'_{\perp} = \cos(\vartheta' - \vartheta)$, the integral for I_{G1} evaluates to

$$I_{G1}(v_{\parallel}, v_{\perp}, v'_{\parallel}, v'_{\perp}) = -\frac{2}{\pi} U^{1/2} \left(\frac{2-m}{3m} E(m) - \frac{2}{3} \frac{(1-m)}{m} K(m) \right), \quad (\text{A.4})$$

with the elliptic integral of the first kind defined by equation (22), and we have used the identity

$$\int_0^{\pi/2} (1 - 2 \sin^2 \theta) \sqrt{1 - m \sin^2 \theta} \, d\theta = \frac{2-m}{3m} E(m) - \frac{2}{3} \frac{(1-m)}{m} K(m). \quad (\text{A.5})$$

The remaining integrals are

$$I_{H0} = \frac{2}{\pi} U^{-1/2} K(m), \quad (\text{A.6})$$

$$I_{H1} = -\frac{2}{\pi} U^{-1/2} \left(\frac{m-2}{m} K(m) + \frac{2}{m} E(m) \right) \quad (\text{A.7})$$

and

$$I_{H2} = \frac{2}{\pi} U^{-1/2} \left(\frac{3m^2 - 8m + 8}{3m^2} K(m) + \frac{4m - 8}{3m^2} E(m) \right). \quad (\text{A.8})$$

Here we have used that

$$\int_0^{\pi/2} (1 - 2 \sin^2 \theta) (1 - m \sin^2 \theta)^{-1/2} d\theta = \frac{m-2}{m} K(m) + \frac{2}{m} E(m), \quad (\text{A.9})$$

and

$$\int_0^{\pi/2} (1 - 2 \sin^2 \theta)^2 (1 - m \sin^2 \theta)^{-1/2} d\theta = \frac{3m^2 - 8m + 8}{3m^2} K(m) + \frac{4m - 8}{3m^2} E(m). \quad (\text{A.10})$$

Appendix B. Rosenbluth potentials for shifted-Maxwellian distributions

For the shifted Maxwellian distribution of species s , given by equation (9), the Rosenbluth potential $G_s(\mathbf{v})$ is given by [4] (Chpt. 5, Sec. 5.2, eqn. (5.49))

$$G_s(\mathbf{v}) = \frac{n_s v_{\text{th},s}}{2\eta} \left(\frac{2\eta}{\sqrt{\pi}} \exp[-\eta^2] + (1 + 2\eta^2) \text{erf}(\eta) \right), \quad (\text{B.1})$$

and

$$H_s(\mathbf{v}) = \frac{n_s}{v_{\text{th},s}} \frac{\text{erf}(\eta)}{\eta}, \quad (\text{B.2})$$

with $\eta = |\mathbf{v} - \mathbf{u}_s|/v_{\text{th},s}$. The results for derivatives of G_s and H_s may be obtained analytically by direct differentiation. In the drift-kinetic limit $\mathbf{u}_s = u_{\parallel,s} \mathbf{b}$.

Appendix C. Computing the Rosenbluth potentials by direct integration

A more direct, but less efficient, method for computing the Rosenbluth potentials is to use the integral expressions (37)-(41) for all $(v_{\parallel}, v_{\perp})$ rather than just the boundary values. Here we show the results of such a calculation to demonstrate the correct implementation of (37)-(41) and the results in Appendix A.

In figure C.10 we plot the infinity-norm errors on the calculation by direct integration of the derivatives of the Rosenbluth potentials for a Maxwellian input, for which the results are known analytically [4] (see, e.g., Appendix B). We see that the integration error becomes small for increasing resolution, indicating that the definitions of the integrands are correct. However, the errors eventually deviate from the expected scaling. This is due to problems carrying out the integral accurately in the region on the integrand where $\mathbf{v}'$ is such that $F_s(\mathbf{v}') \sim O(1)$. This problem might be addressed with an improved integration quadrature, or by using higher than double precision to compute the integrand. Note that this difficulty does not affect the integration of the potentials in the far-field region at the velocity space boundary – meaning that near-machine-precision accuracy can be achieved in the numerical method presented in the main text. This is evident from figures 3 and 4.

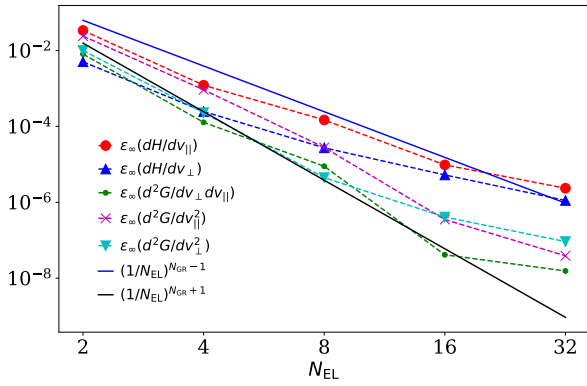


Figure C.10: The infinity norm of the errors ϵ of the potentials $\partial H/\partial v_{||}$, $\partial H/\partial v_{\perp}$, $\partial G/\partial v_{\perp}$, $\partial^2 G/\partial v_{\perp}^2$, $\partial^2 G/\partial v_{||}^2$, $\partial^2 G/\partial v_{\perp}\partial v_{||}$ for a Maxwellian input distribution, compared to the expected scalings for differentiation and integration, equations (93) and (94), respectively.

References

- [1] M. Kardar, Statistical Physics of Particles, Cambridge University Press, 2007.
- [2] L. D. Landau, Die kinetische gleichung für den fall coulombscher wechselwirkung, Phys. Z. Sowjet. 10 (1936) 154.
- [3] M. N. Rosenbluth, W. M. MacDonald, D. L. Judd, Fokker-Planck equation for an inverse-square force, Phys. Rev. 107 (1957) 1–6.
- [4] R. D. Hazeltine, J. D. Meiss, Plasma Confinement, Dover, New York, 2003.
- [5] P. Helander, D. J. Sigmar, Collisional Transport in Magnetized Plasmas, Cambridge University Press, Cambridge, UK, 2002.
- [6] L. Chacón, D. Barnes, D. Knoll, G. Miley, An Implicit Energy-Conservative 2D Fokker-Planck Algorithm: I. Difference Scheme, J. Comput. Phys. 157 (2000) 618–653. doi:https://doi.org/10.1006/jcph.1999.6394.
- [7] F. Alouani-Bibi, M. Shoucri, J.-P. Matte, Different Fokker-Planck approaches to simulate electron transport in plasmas, Comput. Phys. Commun. 164 (2004) 60–66.
- [8] A. Pataki, L. Greengard, Fast elliptic solvers in cylindrical coordinates and the Coulomb collision operator, J. Comput. Phys. 230 (2011) 7840–7852.
- [9] A. Chankin, D. Coster, G. Meisl, Development and benchmarking of a new kinetic code for plasma periphery (KIPP), Contributions to Plasma Physics 52 (2012) 500–504. doi:https://doi.org/10.1002/ctpp.201210039.
- [10] W. Taitano, L. Chacón, A. Simakov, K. Molvig, A mass, momentum, and energy conserving, fully implicit, scalable algorithm for the multi-dimensional, multi-species rosenbluth-fokker-planck equation, J. Comput. Phys. 297 (2015) 357–380. doi:https://doi.org/10.1016/j.jcp.2015.05.025.
- [11] E. Hirvijoki, M. F. Adams, Conservative discretization of the Landau collision integral, Phys. Plasmas 24 (2017) 032121.
- [12] W. Taitano, L. Chacón, A. Simakov, An adaptive, implicit, conservative, 1D-2V multi-species Vlasov-Fokker-Planck multi-scale solver in planar geometry, J. Comput. Phys. 365 (2018) 173–205. doi:https://doi.org/10.1016/j.jcp.2018.03.007.
- [13] M. Abazorius, F. I. Parra, F. Militello, Kinetic analysis of the collisional layer, APS Division of Plasma Physics Meeting Abstracts (2021) CP11–003.
- [14] G. J. Wilkie, T. Keßler, S. Rjasanow, Multidisciplinary benchmarks of a conservative spectral solver for the nonlinear Boltzmann equation, Comput. Phys. Commun. 291 (2023) 108812.
- [15] P. J. Catto, K. T. Tsang, Linearized gyro-kinetic equation with collisions, Phys. Fluids 20 (1977) 396–401.
- [16] I. G. Abel, M. Barnes, S. C. Cowley, W. Dorland, A. A. Schekochihin, Linearized model Fokker-Planck collision operators for gyrokinetic simulations. I. Theory, Phys. Plasmas 15 (2008) 122509.
- [17] M. Barnes, I. Abel, W. Dorland, D. Ernst, G. Hammett, P. Ricci, B. Rogers, A. Schekochihin, T. Tatsuno, Linearized model Fokker-Planck collision operators for gyrokinetic simulations. ii. numerical implementation and tests, Phys. Plasmas 16 (2009).
- [18] H. Sugama, T.-H. Watanabe, M. Nunami, Linearized model collision operators for multiple ion species plasmas and gyrokinetic entropy balance equations, Phys. Plasmas 16 (2009) 112503.
- [19] P. J. Catto, A. N. Simakov, Drift ordered short mean free path description for magnetized plasma allowing strong spatial anisotropy, Phys. Plasmas 11 (2004) 90.
- [20] F. L. Hinton, R. D. Hazeltine, Theory of plasma transport in toroidal confinement systems, Rev. Mod. Phys. 48 (1976) 239–308.
- [21] I. G. Abel, G. G. Plunk, E. Wang, M. Barnes, S. C. Cowley, W. Dorland, A. A. Schekochihin, Multiscale gyrokinetics for rotating tokamak plasmas: fluctuations, transport and energy flows, Rep. Prog. Phys. 76 (2013) 116201.
- [22] S. P. Hirshman, Classical collisional theory of beam-driven plasma currents, Phys. Fluids 23 (1980) 1238–1243. doi:10.1063/1.863103.
- [23] W. W. Heidbrink, Basic physics of Alfvén instabilities driven by energetic particles in toroidally confined plasmas, Phys. Plasmas 15 (2008) 055501. doi:10.1063/1.2838239.
- [24] D. Tskhakaya, F. Subba, X. Bonnin, D. P. Coster, W. Fundamenski, R. A. Pitts, JET EFDA Contributors, On kinetic effects during parallel transport in the SOL, Contributions to Plasma Physics 48 (2008) 89–93. doi:https://doi.org/10.1002/ctpp.200810015.
- [25] M. Francisquez, T. N. Bernard, B. Zhu, A. Hakim, B. N. Rogers, G. W. Hammett, Fluid and gyrokinetic turbulence in open field-line, helical plasmas, Phys. Plasmas 27 (2020) 082301. doi:10.1063/1.5005333.
- [26] P. Cagas, A. Hakim, B. Srinivasan, Plasma-material boundary conditions for discontinuous galerkin continuum-kinetic simulations, with a focus on secondary electron emission, J. Comput. Phys. 406 (2020) 109215. doi:https://doi.org/10.1016/j.jcp.2019.109215.
- [27] A. V. Chankin, G. Corrigan, JET Contributors, Coupled KIPP-EDGE2D modeling of parallel transport in the SOL and divertor plasma for the ITER baseline scenario, Plasma Phys. Control. Fusion 65 (2023) 085003. doi:10.1088/1361-6587/acd86e.
- [28] G. Wilkie, F. Laggner, R. Hager, A. Rosenthal, S.-H. Ku, R. Churchill, L. Horvath, C. Chang, A. Bortolon, Reconstruction and interpretation of ionization asymmetry in magnetic confinement via synthetic diagnostics, Nucl. Fusion 64 (2024) 086028. doi:10.1088/1741-4326/ad556b.
- [29] P. C. Stangeby, A tutorial on some basic aspects of divertor physics, Plasma Phys. Control. Fusion 42 (2000) B271. doi:10.1088/0741-3335/42/12B/321.
- [30] R. Hazeltine, Recursive derivation of drift-kinetic equation, Plasma Phys. 15 (1973) 77–80.
- [31] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. Kirby, S. Sherwin, Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192 (2015) 205–219. doi:https://doi.org/10.1016/j.cpc.2015.02.008.
- [32] E. J. Threlfall, R. J. Akers, W. Arter, M. Barnes, M. Barton, C. Cantwell, P. Challenor, J. W. S. Cook, P. V. Coveney, T. Dodwell, B. D.udson, P. E. Farrell, T. Goffrey, M. Green, S. Guillas, M. Hardman, P. A. Hill, L. Kimpton, C. MacMackin, B. McMillan, D. Moxey, G. R. Mudalige, J. T. Omotani, J. T. Parker, F. P. Diaz, O. Parry, T. Rees, C. P. Ridgers, W. Saunders, S. J. Sherwin, S. Thorne, J. Williams, S. A. Wright, Y. Yang, Software for fusion reactor design: Excalibur project neptune: towards exascale plasma edge simulations, 29th IAEA Fusion Energy Conference, proceedings (2023).
- [33] P. N. Swarztrauber, A direct method for the discrete solution of separable elliptic equations, SIAM Journal on Numerical Analysis 11 (1974) 1136–1150. doi:10.1137/0711086.
- [34] U. Schumann, R. A. Sweet, A direct method for the solution of Poisson’s equation with neumann boundary conditions on a staggered grid of arbitrary size, J. Comput. Phys. 20 (1976) 171–182. doi:https://doi.org/10.1016/0021-9991(76)90062-0.
- [35] A. Toselli, O. B. Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer-Verlag Berlin, Heidelberg, 2005. doi:https://doi.org/10.1007/b137868.
- [36] W. Hackbusch, Multi-Grid Methods and Applications, Springer

- Berlin, Heidelberg, 2010. doi:<https://doi.org/10.1007/978-3-662-02427-0>.
- [37] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, 2001.
 - [38] F. B. Hildebrand, *Introduction to numerical analysis*, Courier Corporation, 1987.
 - [39] M. Barnes, F. I. Parra, M. R. Hardman, J. Omotani, Numerical study of 1+1d drift kinetic models for parallel dynamics in the plasma edge, *Excalibur/Neptune Report* (2021) 2047357–TN–D2.2+M2.5.
 - [40] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM Rev.* 59 (2017) 65–98. doi:10.1137/141000671.
 - [41] J. Omotani, S. L. Newton, F. I. Parra, M. Barnes, M. R. Hardman, Demonstration of moment-kinetics approach for edge modelling, *European Fusion Theory Conference* 2023.
 - [42] M. R. Hardman, F. I. Parra, J. Omotani, M. Barnes, S. L. Newton, ExB drift physics on open field lines in a drift-kinetic model, *European Fusion Theory Conference* 2023.
 - [43] https://github.com/mabarnes/moment_kinetics.
 - [44] The MPI standard, version 3.0.
URL <https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf>
 - [45] MPI_Win_allocate_shared.
URL <https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report/node263.htm>
 - [46] <https://docs.julialang.org/en/v1/stdlib/sparsearrays/>.
 - [47] https://github.com/mabarnes/moment_kinetics, commit 8bff6a5.
 - [48] M. Hardman, et al., Dataset: tests of a finite-element implementation of the nonlinear fokker-planck collision operator (2024). doi:<http://dx.doi.org/10.5287/ora-g7mrageej>.
 - [49] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillator shock-capturing schemes, *J. Comp. Phys.* (1988) 77:439–471.
 - [50] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta methods, *Mathematics of Computation* (1998) 67:73–85.
 - [51] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.* (2001) 43:89.
 - [52] S. A. Teukolsky, Short note on the mass matrix for Gauss–Lobatto grid points, *J. Comput. Phys.* 283 (2015) 408–413.
 - [53] L. Chacón, D. Barnes, D. Knoll, G. Miley, An Implicit Energy-Conservative 2D Fokker–Planck Algorithm: II. Jacobian-Free Newton–Krylov Solver, *J. Comput. Phys.* 157 (2000) 654–682. doi:<https://doi.org/10.1006/jcph.1999.6395>.