Numerical study of 1D drift kinetic models with periodic boundary conditions

M. Barnes[1], **F. I. Parra**[1] **and M. R. Hardman**[1]

[1] Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Clarendon Laboratory, Parks Road, Oxford OX1 3PU, UK

E-mail: michael.barnes@physics.ox.ac.uk

1. Introduction

We expect that one of the biggest challenges in numerically solving drift kinetic equations in the plasma edge is treating the motion of electrons along the magnetic field. Because the electrons are light, they move rapidly along the field, placing a severe stability restriction on the step size for explicit time advance schemes. Unfortunately, an implicit treatment is not straightforward due to an implicit dependence of the electrostatic potential on the charged particle distribution functions. One of the main aims of our research is to develop and test a novel analytical model and associated numerical algorithm for relaxing this restriction. As a first step towards this goal, we have developed a new code in the programming language Julia to simulate a simple model for parallel dynamics (described in our Jan 2021 report [1]) without the novel moment-based approach that we intend to ultimately employ. This code will be used to test different numerical approaches and will be built upon to create the code for the moment-based approach. It will also be a useful benchmark against which the moment-based approach can be tested.

In this report we give a brief description of a simple model for parallel plasma dynamics before describing the corresponding code and presenting a comparison of numerical results with the analytical benchmark we have derived.

2. Model equations

A detailed derivation of the model we consider is provided in the previous (Jan 2021) report [1]. Here we provide a brief overview of the model for the Reader's convenience. The model we consider consists of a single ion species of charge e, a single neutral species, and an electron species modelled as having a Boltzmann response, all immersed in a straight, uniform magnetic field in the z direction. We allow for charge exchange collisions between ions and neutrals but do not account for intra-species collisions. Finally, we assume that the plasma is homogeneous in the plane perpendicular to the magnetic field. With these assumptions, our model system of equations is

$$\frac{\partial f_i}{\partial t} + v_\| \frac{\partial f_i}{\partial z} - \frac{e}{m_i}\frac{\partial \phi}{\partial z}\frac{\partial f_i}{\partial v_\|} = -R_{in}\left(n_n f_i - n_i f_n\right), \tag{1}$$

$$\frac{\partial f_n}{\partial t} + v_\| \frac{\partial f_n}{\partial z} = -R_{in}\left(n_i f_n - n_n f_i\right), \tag{2}$$

$$n_s(z,t) = \int_{-\infty}^{\infty} dv_\| f_s(z, v_\|, t), \tag{3}$$

and

$$n_i = N_e \exp\left(\frac{e\phi}{T_e}\right), \tag{4}$$

with $f_s \doteq \int d\vartheta dv_\perp v_\perp F_s$ the marginalized particle distribution function for species s, $v_\|$ and $v_\perp$ the components of the particle velocity parallel and perpendicular to the magnetic field, respectively, ϑ the gyro-angle, m_i the ion mass, t the time, ϕ the electrostatic potential, and R_{in} the charge exchange collision frequency.

For our boundary conditions, we impose periodicity on f_s in both z and $v_\|$, with periods L_z and $L_{v_\|}$, respectively. There is also the option to impose zero boundary conditions on z and $v_\|$ at the upwind boundary of the domain. As f_s should go to zero in $v_\|$ as $v_\| \to \pm\infty$, imposition of zero boundary conditions and periodic boundary conditions should be equivalent as long as $L_{v_\|}$ is sufficiently large. Note that with either choice of boundary conditions, the line-averaged density $\int_0^{L_z} dz\; n_s$ is conserved.

We normalize Eqs. (1)-(4) by defining

$$\tilde{f}_s \doteq f_s \frac{v_{\rm th,i}\sqrt{\pi}}{N_e}, \tag{5}$$

$$\tilde{t} \doteq t\frac{v_{\rm th,i}}{L_z}, \tag{6}$$

$$\tilde{z} \doteq \frac{z}{L_z}, \tag{7}$$

$$\tilde{v}_\| \doteq \frac{v_\|}{v_{\rm th,i}}, \tag{8}$$

$$\tilde{n}_s \doteq \frac{n_s}{N_e}, \tag{9}$$

$$\tilde{\phi} \doteq \frac{e\phi}{T_e}, \tag{10}$$

and

$$\tilde{R}_{\rm in} \doteq \frac{R_{\rm in}}{N_e}\frac{L_z}{v_{\rm th,i}} \tag{11}$$

with $v_{\rm th,i} \doteq \sqrt{2T_e/m_i}$. In terms of these normalised quantities, Eqs (1)-(4) become

$$\frac{\partial \tilde{f}_i}{\partial \tilde{t}} + \tilde{v}_\| \frac{\partial \tilde{f}_i}{\partial \tilde{z}} - \frac{1}{2}\frac{\partial \tilde{\phi}}{\partial \tilde{z}}\frac{\partial \tilde{f}_i}{\partial \tilde{v}_\|} = -\tilde{R}_{\rm in}\left(\tilde{n}_n \tilde{f}_i - \tilde{n}_i \tilde{f}_n\right), \tag{12}$$

$$\frac{\partial \tilde{f}_n}{\partial \tilde{t}} + \tilde{v}_\parallel \frac{\partial \tilde{f}_n}{\partial \tilde{z}} = -\tilde{R}_{\text{in}} \left(\tilde{n}_i \tilde{f}_n - \tilde{n}_n \tilde{f}_i \right), \tag{13}$$

$$e^{\tilde{\phi}} = \tilde{n}_i = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} d\tilde{v}_\parallel \tilde{f}_i, \tag{14}$$

and

$$\tilde{n}_n = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} d\tilde{v}_\parallel \tilde{f}_n. \tag{15}$$

3. Numerical implementation

The algorithms described in this Section have been implemented in the code, written in the Julia programming language, currently available on GitHub at `https://github.com/mabarnes/moment_kinetics`.

3.1. Time advance

We evolve Eqs. (12)-(15) using a time-marching scheme (as opposed to an eigensolver) due to its efficiency and to the nonlinear nature of the system of partial differential equations. In particular, we employ a member of the family of Strong Stability Preserving (SSP) Runge-Kutta (RK) schemes; see, e.g., [2, 3, 4]. Current SSPRK options implemented in the code are SSPRK1 (forward Euler), SSPRK2 (Heun's method) and SSPRK3 (Shu-Osher method). The user can also specify the use of 'flip-flop' Lie operator splitting, described in Appendix A. Operator splitting limits the time advance scheme to second order accuracy in step size, but could be useful for separately treating different pieces of physics. Here we describe the current default option, which is the Shu-Osher method (SSPRK3) without operator splitting.

For convenience of notation, we express the normalised drift kinetic equations for the ions and neutrals in the vector form

$$\frac{\partial \mathbf{f}}{\partial t} = G[\mathbf{f}], \tag{16}$$

with $\mathbf{f} = (\tilde{f}_i, \tilde{f}_n)^T$ and G the drift kinetic operator account for parallel streaming, parallel acceleration (for the ions) and charge exchange collisions. The Shu-Osher method for advancing this system of equations is a 3-stage, SSPRK scheme that is 3rd order accurate in time step size Δt. It is given by

$$\begin{aligned}
\mathbf{f}^{(1)} &= \mathbf{f}^n + \Delta t G\left[\mathbf{f}^n\right], \\
\mathbf{f}^{(2)} &= \frac{3}{4}\mathbf{f}^n + \frac{1}{4}\left(\mathbf{f}^{(1)} + \Delta t G\left[\mathbf{f}^{(1)}\right]\right), \\
\mathbf{f}^{n+1} &= \frac{1}{3}\mathbf{f}^n + \frac{2}{3}\left(\mathbf{f}^{(2)} + \Delta t G\left[\mathbf{f}^{(2)}\right]\right),
\end{aligned} \tag{17}$$

where the superscript n denotes the time level.

3.2. Spatial discretisation

There are two discretisation schemes implemented in the code: finite differences and Chebyshev (pseudo)spectral elements. The user can choose at run-time which scheme to use for each of the z and $v_\|$ coordinates.

3.2.1. Finite difference discretisation. For the finite difference discretisation, the corresponding coordinate grid is uniform on the domain $[-L/2, L/2]$, with L the coordinate box length. The default method employed for derivatives is 3rd order upwind differences, though 1st and 2nd order schemes are also available as options. For an overview of upwind differences and a discussion of the merits of the different upwind schemes, see, e.g. [5]. The associated integration weights used for field-line averages in z and/or for the $v_\|$ integration required for obtaining fields/moments are obtained using the composite Simpson's rule (sometimes referred to as composite Simpson's 1/3 rule:

$$\int_0^L dx\ f(x) \approx \frac{h}{3} \sum_{j=1}^{(N-1)/2} \left(f(x_{2j-1}) + 4f(x_{2j}) + f(x_{2j+1}) \right), \tag{18}$$

where N is the number of grid points in the coordinate x, and $h = L/(N-1)$ is the uniform grid spacing. The composite rule (18) is only applicable for N odd, so it is supplemented at the boundary by Simpson's 3/8 rule when N is even.

3.2.2. Chebyshev spectral elements. When using Chebyshev spectral elements (see, e.g., [6]), the corresponding coordinate grid is the Gauss-Chebyshev-Lobatto grid on each element. For a description of Chebyshev-Gauss quadrature, see, e.g. [7]. Inclusion of the endpoints within each element facilitates enforcement of continuity at element boundaries, and the use of Chebyshev polynomials as a basis enables the use of Fast Fourier Transforms. In our code, these transforms are done using the widely-used FFTW library [8]. The associated integration weights used for field-line averages in z and/or for the $v_\|$ integration required for obtaining fields/moments are obtained using Clenshaw-Curtis quadrature rules [9]. Clenshaw-Curtis quadrature is convenient, as it allows for the use of endpoints in the integration domain (which is dictated by the use of a Gauss-Chebyshev-Lobatto grid) while still exactly integrating polynomials up to degree $N-1$, with N the number of points within the element.

4. Numerical results

To benchmark our numerical implementation of Eqs. (12)-(15), we compare our simulation results with the analytical benchmarks developed in [1]. In particular, we have initialised the distribution functions for the ions and neutrals to be of the form

$$\tilde{f}_s = \frac{n_s}{N_e} \left(\frac{T_e}{\overline{\overline{T}}_s} \right)^{1/2} \exp\left(-\tilde{v}_\|^2 \frac{T_e}{\overline{\overline{T}}_s} \right), \tag{19}$$

with $n_s = \overline{n}_s + \delta n_s$, and an overline denoting a field line average. The piece of the density that varies along z, δn_s, is chosen to be small compared to $\overline{n}_s$ ($\delta n_s/\overline{n}_s = 0.001$) so that the system of equations can be linearised to a good approximation. This facilitates comparisons with the linear analytical theory presented in [1]. For all cases shown here, $\overline{n}_i = \overline{n}_n = N_e/2$, $\overline{T}_i = \overline{T}_n$ and $m_i = m_n$. Both the electron-ion temperature ratio and the charge exchange collision frequency are varied, and damping rates and frequencies are extracted by considering the time evolution of the spatially-varying component of the electrostatic potential, $\delta\phi$. In particular, a least-squares fit for $\delta\phi(t)/\delta\phi(t_0)$ is done for each simulation to a function of the form $\exp(\gamma(t-t_0))\cos(\omega t - \varphi)/\cos(\omega t_0 - \varphi)$ to obtain the damping rate $-\gamma$, frequency ω and phase φ. The results are given in Fig. 1. There is excellent agreement across a wide range of temperature ratios and charge exchange collision frequencies, both for the damping of finite frequency modes (corresponding to the solid lines) and to a zero frequency mode that appears at larger collisionalities (dashed-dotted lines).

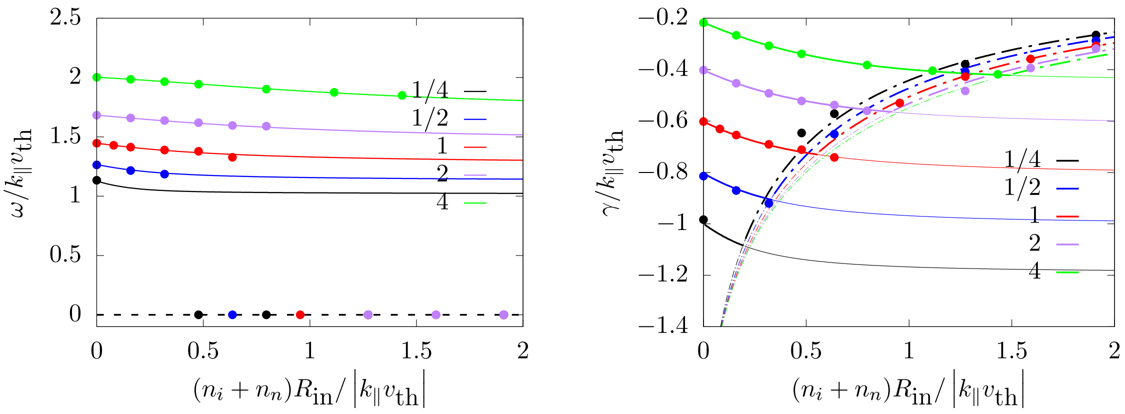

Figure 1. Normalized growth rate and real frequency as a function of the ion-electron temperature ratio. Note that the normalising $v_{\rm th}$ employed here differs from the $v_{\rm th,i}$ employed in the text: It is chosen to be $v_{\rm th} = \sqrt{2T_i/m_i}$ to facilitate comparison with the analytical results obtained in [1].

The minor discrepancies between the analytical and numerical damping rates that are apparent for a handful of the cases are due to the simultaneous presence of both modes with similar damping rates. This necessitates in some cases resolving the damping of both modes over many orders of magnitude before the least damped mode dominates the numerical solution – a challenging task given the ever-increasing filamentation of the velocity space due to phase mixing. This should be possible to eliminate by carefully initialising the simulation so that only the least damped mode is present, though we have not yet attempted this. An example of a case in which both modes are present, as well as a case in which they are not, is given in Fig. 2.

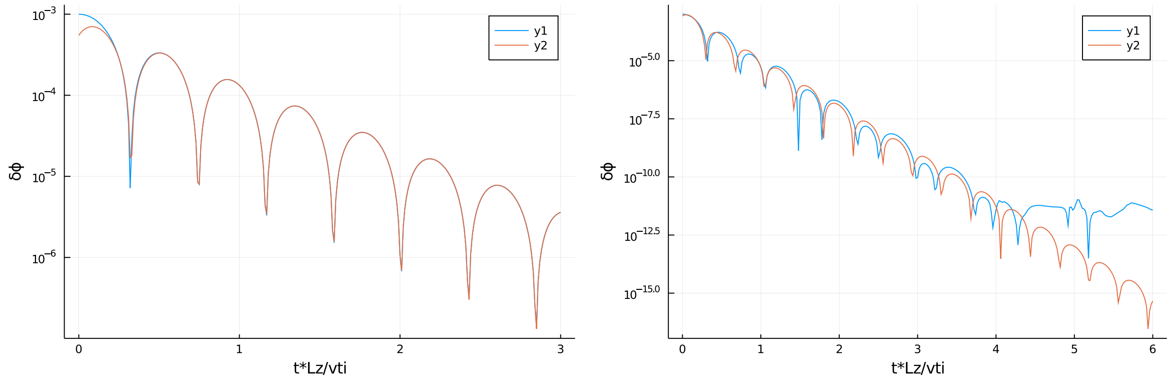

Figure 2. Time evolutions of the absolute value of the spatially-varying electrostatic potential $\delta\phi$ (blue) and the result of a least-squares fit (orange) to obtain the damping rate, frequency and phase. The left plot corresponds to $T_e/T_i = 2$ and $\tilde{R}_{\text{in}} = 0$, and the right plot corresponds to $T_e/T_i = 1$ and $\tilde{R}_{\text{in}} = 4$ (≈ 0.7 in terms of the normalised frequency used in Fig. 1).

5. Future plans

With the core 1+1D code developed and successfully benchmarked against analytical theory, we will now turn our attention to including kinetic electron dynamics and to trialing a version of the moment-based approach in which the density moment is split off from the kinetic equation. In parallel, we plan to explore the idea of using an interpolation-free semi-Lagrange scheme [10] for the solution of the kinetic equation as a novel approach to ameliorating the time step restriction posed by the kinetic electron dynamics.

Appendix A. Operator splitting

We evolve Eqs. (12)-(15) using Heun's method with 'flip-flop' Lie operator splitting. To facilitate this it is convenient to define the vector $\mathbf{f}$ whose components are the distribution functions $\tilde{f}_i$ and $\tilde{f}_n$; i.e., $\mathbf{f} = (\tilde{f}_i, \tilde{f}_n)^T$. In terms of $\mathbf{f}$ the system of equations is

$$\frac{\partial \mathbf{f}}{\partial t} + A[\mathbf{f}] + B[\mathbf{f}] = C[\mathbf{f}], \tag{A.1}$$

where

$$A[\mathbf{f}] \doteq \tilde{v}_{\|} \begin{pmatrix} 1 & 0 \\ 0 & v_{\text{th},n}/v_{\text{th,i}} \end{pmatrix} \frac{\partial \mathbf{f}}{\partial z}, \tag{A.2}$$

$$B[\mathbf{f}] \doteq -\frac{1}{2} \begin{pmatrix} \partial\tilde{\phi}/\partial\tilde{z} & 0 \\ 0 & 0 \end{pmatrix} \frac{\partial \mathbf{f}}{\partial \tilde{v}_{\|}}, \tag{A.3}$$

and

$$C[\mathbf{f}] \doteq -\tilde{R}_{\text{in}} \begin{pmatrix} \tilde{n}_n & -\tilde{n}_i \\ -\tilde{n}_n & \tilde{n}_i \end{pmatrix} \mathbf{f}. \tag{A.4}$$

Splitting the operators and employing Heun's method results in the following time advance scheme:

$$\begin{aligned}
\mathbf{f}_A^{(1),n} &= \mathbf{f}^n - \Delta t A\left[\mathbf{f}^n\right] \\
\mathbf{f}_A^{(2),n} &= \mathbf{f}^n - \Delta t A\left[\mathbf{f}_A^{(1),n}\right] \\
\mathbf{f}_A^{n} &= \frac{1}{2}\left(\mathbf{f}_A^{(1),n} + \mathbf{f}_A^{(2),n}\right),
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
\mathbf{f}_B^{(1),n} &= \mathbf{f}_A^n - \Delta t B\left[\mathbf{f}_A^n\right] \\
\mathbf{f}_B^{(2),n} &= \mathbf{f}_A^n - \Delta t B\left[\mathbf{f}_A^{(1),n}\right] \\
\mathbf{f}_B^{n} &= \frac{1}{2}\left(\mathbf{f}_B^{(1),n} + \mathbf{f}_B^{(2),n}\right),
\end{aligned} \tag{A.6}$$

and

$$\begin{aligned}
\mathbf{f}_C^{(1),n} &= \mathbf{f}_B^n - \Delta t C\left[\mathbf{f}_B^n\right] \\
\mathbf{f}_C^{(2),n} &= \mathbf{f}_B^n - \Delta t C\left[\mathbf{f}_B^{(1),n}\right] \\
\mathbf{f}^{n+1} &= \frac{1}{2}\left(\mathbf{f}_C^{(1),n} + \mathbf{f}_C(2), n\right),
\end{aligned} \tag{A.7}$$

with the order of operations (A, followed by B, followed by C) reversed every time step to obtain second order acuracy in Δt.

[1] F. I. Parra, M. Barnes, and M. R. Hardman. 1d drift kinetic models with periodic boundary conditions. *Excalibur/Neptune Report*, 1:2047357–TN–01–02 M1.1, 2021.

[2] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillator shock-capturing schemes. *J. Comp. Phys.*, 77:439–471, 1988.

[3] S. Gottlieb and C.-W. Shu. Total variation diminishing runge-kutta methods. *Mathematics of Computation*, 67:73–85, 1998.

[4] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Rev.*, 43:89, 2001.

[5] D. R. Durran. *Numerical methods for fluid dynamics.* Springer, 2010.

[6] J. P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover, New York, 2000.

[7] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, New York, 1972.

[8] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[9] C. W. Clenshaw and A. R Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2:197, 1960.

[10] H. Ritchie. Eliminating the interpolation associated with the semi-lagrange scheme. *Monthly Weather Review*, 114:135–146, 1986.