Discontinuous Galerkin Studies of Collisional Dynamics in Continuum-Kinetic Plasma

John M. Rodman

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Bhuvana Srinivasan, Chair
Colin S. Adams
Wayne A. Scales
Timothy C. Warburton

November 25, 2024
Blacksburg, Virginia

Keywords: Plasma physics, computational modeling, discontinuous Galerkin

Discontinuous Galerkin Studies of Collisional Dynamics in Continuum-Kinetic Plasma

John M. Rodman

(ABSTRACT)

Numerical investigations of collisional physics have historically been impeded by the issue of computational expense. While the continuum-kinetic Vlasov-Maxwell-Fokker-Planck system is well-established in theory and has been used as the basis for many approximate fluid equations, simulations utilizing the distribution function are relatively uncommon, due primarily to the high dimensionality of the problem. However, advances in numerical methods are steadily making these models more accessible. In this work, we utilize the `Gkeyll` framework, which applies a novel, highly efficient discontinuous Galerkin (DG) finite element method to the Vlasov-Maxwell-Fokker-Planck system.

We first investigate the Rayleigh-Taylor (RT) instability in a neutral gas in regimes of finite collisionality which are inaccessible to the fluid codes that are traditionally applied to this instability. Utilizing a spatially constant, finite collision frequency, we demonstrate the ability of the Vlasov-Boltzmann model to approach the fluid result at high collision frequency while also accessing a regime of intermediate collisionality in which the RT instability deviates greatly from classic fluid behavior. We then extend upon this finding by choosing a collision frequency that varies spatially, resulting in new dynamics with asymmetric diffusion affecting the development of the RT instability.

Having demonstrated the utility of collisional kinetic modeling even in the simple case of a neutral gas with a basic collision operator, we transition to development and implementation of a fully-conservative, recovery-based DG algorithm for the full nonlinear Rosenbluth/Fokker-Planck collision operator (FPO). Details of the novel recovery scheme for the cross-derivatives and conservation enforcement are presented, and we show that the scheme converges and exhibits stability criteria as expected. Finally, the FPO is applied to test cases that demonstrate the importance of accurate handling of the velocity-dependent collision frequency as compared to an approximate model.

This work was supported by the DOE SciDAC program under DE-SC0018276 and the NSF CAREER award under grant numbers PHY-1847905 and PHY-2345433. The author acknowledges Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper (http://www.arc.vt.edu). A subset of simulations presented in this article were performed on computational resources managed and supported by Princeton Research Computing which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Research Computing.

Discontinuous Galerkin Studies of Collisional Dynamics in Continuum-Kinetic Plasma

John M. Rodman

(GENERAL AUDIENCE ABSTRACT)

Under the right conditions, the electrons and ions that comprise the particles in a gas separate, or *ionize*, forming a plasma. Plasma is the most common state of matter in the universe, existing at a wide range of scales. Whether concerning a supernova, the solar wind, a plume of material ablated by a laser, or a nuclear fusion reactor, all of these plasmas are governed by the same set of rules, with the main differences being which length and time scales are relevant.

Understanding the dynamics of these collections of ionized particles offers a unique challenge, as particles interact not only by colliding with one another but through longer-range electromagnetic interactions. A number of methods exist for modeling plasmas, and one must choose which of the many scales in the plasma are relevant in order to make the best choice of model. In this work, we apply the continuum-kinetic method, which captures the statistical effect of individual particle motions while avoiding the noise that arises when tracking individual particles directly. Kinetic methods are not applied nearly as often as fluid methods, primarily because of the computational expense involved in resolving the wide range of scales and accounting for quantities that evolve as a function of both position and velocity. However, recent advances in numerical methods have made continuum-kinetic methods much more accessible. This work utilizes the `Gkeyll` code framework, which applies a discontinuous Galerkin method, to simulate plasma with a continuum-kinetic model.

We begin by considering the Rayleigh-Taylor (RT) instability, which occurs when a heavy fluid is balanced atop a lighter fluid and perturbed, resulting in fluid mixing. The RT instability is ubiquitous in nature and is commonly modeled with fluid methods that assume particle collide with one another with effectively infinite frequency. With the continuum-kinetic method, we demonstrate that situations arise where the collision frequency is finite but the RT instability still grows. In these regimes, the instability growth is no longer well-described by fluid methods, and a kinetic model must be applied to accurately predict its evolution.

Following this, we introduce an algorithm that utilizes a novel discontinuous Galerkin (DG) method to model one of the most complex and accurate collision operators for plasmas: the Fokker-Planck operator (FPO). The FPO is notoriously difficult to implement numerically and computationally expensive due to its nonlinear nature, so simulations generally utilize approximate forms rather than the full operator. By applying this DG method, we are able to ensure the numerical FPO implementation maintains many of the desirable properties of the original model while running highly efficiently. We conclude by verifying that the code is stable and highly accurate while reproducing expected results and improvements over simplified collision models.

Acknowledgments

As I make the final edits to this dissertation, I can't help but reflect on the tremendous amount of support I have received throughout this endeavor. This document is the culmination of a five-year effort that would not have gone nearly as smoothly without the help of my amazing friends, family, and collaborators.

To begin, I would like to thank my advisor, Professor Bhuvana Srinivasan. When I had the opportunity to interview with Dr. Srinivasan during my official visit to Virginia Tech, her infections enthusiasm for her research had me ready to get to work, even though I didn't really understand what she was talking about at the time. That energetic environment has remained through my entire time in this program. Regardless of how frustrated or defeated I felt from getting stuck on some problem, even a simple chat with Dr. Srinivasan would invariably set my mind straight and put me on the path towards the solution with renewed optimism. Petr Cagas summed it up nicely as we puzzled over a simulation one day: "well, when in doubt, Bhuvana will save us." I want to thank Dr. Srinivasan for fostering and welcoming me into this incredibly positive, supportive, and productive academic environment that was invaluable through my PhD, and I will strive to bring that same culture with me as I progress into my career.

Dr. Srinivasan is not the only mentor I have had the good fortune of encountering during my time here. Through the first portion of my PhD, I was also supervised by Dr. Petr Cagas. Back when I could only rarely get simulations to run and couldn't comprehend the math involved, Petr was endlessly patient and always willing to help me alongside Dr.

Srinivasan. Petr's philosophies on research and problem-solving have contributed greatly to my own development, and I cannot thank him enough for starting me on the right path so early (even if it takes a lot longer to get my presentations and papers up to my standards now). Later in my time here, I lucked out again and had the opportunity to take over a project that had me working closely with Dr. James Juno. Before working with Jimmy, even I would never have guessed that working with deeply complex numerical methods and debugging physics code could actually be *fun*. Jimmy has been a fount of knowledge, and I'm astounded at how much I have learned from him these past couple years. Beyond that, Jimmy's guidance has been invaluable for my development as a researcher, and I feel much more prepared to begin a career in science after working with him. I would not be in this position today without Petr and Jimmy's guidance, so thank you both again.

I would now like to thank all the other collaborators and group members I've had the pleasure of working with. From Dr. Srinivasan's group, I'd like to mention in particular Dr. Camille Samulski, Dr. Matt Carrier, Dr. Kolter Bradshaw, Dr. Yuzhi Li, Dr. Lujain Almarhabi, Dr. Chirag Skolar, Daniel Alex, and Vignesh Kumar for contributing to the group's great vibes, even when we were all tired from staring at our computers. I thank Dr. Ammar Hakim for his guidance and advice and for his and the rest of the `Gkeyll` team's support as I learned the ropes. I would like to thank my remaining committee members, Dr. Colin Adams, Dr. Wayne Scales, and Dr. Timothy Warburton, for being flexible and providing me with useful insight every step of the way. I also thank the AOE department staff, especially the graduate coordinators during my time, Kelsey Wall and Audriana Cunningham, for being endlessly helpful as I navigated all the administrative complexities of being a grad student.

Of course, I would certainly not be where I am today without the help of my family. I cannot express enough gratitude to my family for their unrelenting support throughout my entire life, not just during my time at Virginia Tech. I am privileged to have parents like Alix and Jeff Rodman and an older brother like Drew, all of whom have had my back and been excited for me through every tiny step down this path I have taken. When the early portion of my PhD was upended along with the rest of the world causing me and Drew to move back home, my family made the best of it, ensuring I had everything I needed to succeed as a grad student. It's difficult to convey how grateful I am to my family for setting me up for success for my entire life, so all I can do here is say thank you.

I would now like to thank my friends for keeping my head on straight through this journey. To Austin, Madalyn, Pat, Sweeney, Red, Cletus, Nicole, Jeff, and other friends from undergrad: we've all come a long way since those early days at Syracuse. I'm grateful to all of you for always managing to keep it fun and for always being down to lose to me at Super Smash Bros. I can't wait to see where we all go next, and I'm looking forward to seeing you all again soon. To Diana, Jamie, Joao, and other friends from Baltimore: thank you for sticking with me all these years, even though we've all moved far apart. To Chris and Sean (and Fiona the cat): I really have no idea how I was lucky enough to end up with you two (three) as roommates when I moved back to Blacksburg. It seems like there was never a dull moment in that house over those years, even when we were all swamped with work. I'm looking forward to playing more bad golf, hearing never-ending slander about the Ravens and Orioles, and getting roasted about how long it took me to beat Malenia for years to come.

Many others have helped me through this journey in some way, and I'll thank some of them here. Thank you to all the friends I made in Blacksburg. You welcomed me in with

open arms even when you all only knew me as Chris and Sean's roommate. Thank you to Dr. David Sprouster for being a fantastic mentor during my summer at Brookhaven National Lab and beyond. Thank you to all of my professors at Syracuse University for setting me up for success, particularly Dr. Ashok Sangani, Dr. Katie Cadwell, Dr. Jay Hubisz, and Dr. Duncan Brown. Thank you to the great teachers I had back at Towson High who started me on this path, namely Mr. Dalsimer, Mr. Biddison, Ms. Bissell-Harris, and Mr. Bradley. Thank you Julian and Corey for making sure I'm able to (and reminded to) exercise and move well after hours spent hunched over my desk. I would probably be permanently curled into an unathletic ball by now if it weren't for your coaching. Thank you to the Chestertown crew for keeping my parents out of trouble and always making fun plans when Drew and I visit. Finally, thank you to the various cafés around Blacksburg and Chestertown for accommodating me as I chugged coffee and frantically typed away.

Thanks again everyone, and I hope you enjoy giving this thesis a read.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Plasma, the ionized gas often referred to as the "fourth state of matter," comprises the vast majority of ordinary matter in our universe. From supernovae and the solar wind to lightning storms and fluorescent light fixtures, plasma exists at a staggering variety of scales. But what exactly is a plasma? Often, plasma is introduced in the context of the three fundamental states of matter. A solid, when heated, eventually undergoes a phase transition to a liquid state, which in turn transitions to a gas. Should the gas be heated further, collisions between gas particles become energetic enough to knock electrons from their nuclei, "ionizing" the gas into a plasma. While this description is tidy, there is no concrete point where all particles simultaneously ionize that would allow us to definitively pronounce that a gas has become a plasma. It is therefore worth defining some set of criteria to differentiate a plasma from a weakly ionized gas. One common definition is given by Chen [25]: "a plasma is a quasineutral gas of charged and neutral particles which exhibits collective behavior." Naturally, we now need two more definitions for this description to make sense. A *quasineutral* gas has approximately the same number density (number of particles per unit volume) of free electrons and ions, that is,

$$n_e \approx Z n_i,$$

where n_e and n_i are the electron and ion number densities, and Z is the charge state of the ions. *Collective behavior* refers to the fact that portions of the plasma affect other regions

via the long-range Coulomb interaction. In a plasma that exhibits collective behavior, the motions of particles in some small region are not simply dependent on the state of that local region, rather, long range electric forces centered far from the region of interest contribute tangibly to the dynamics as well.

The inherently complex interactions between the many particles in a plasma (for example, $\approx 10^{20}$ particles per cubic meter in a tokamak fusion reactor) and the rapidly fluctuating electromagnetic fields may give the impression that attempting to accurately model a plasma is a fruitless effort. However, with careful construction and approximations, clean mathematical models can be developed that allow for accurate prediction of plasma dynamics both analytically and computationally. Computational modeling of plasma physics serves to inform experimental development, so we strive to capture the most accurate behavior in the most computationally efficient manner to further both theoretical understanding and experimental research thrusts.

This work begins with an introduction to plasma kinetic methods in Chapter 1, arriving at the continuum-kinetic model that is our method of choice for this research. We will then discuss the dynamics of particle collisions, an important kinetic phenomenon that is generally approximated or averaged over due to complexity and computational expense. Following the physics background, Chapter 2 is a primer on common numerical methods for solving partial differential equations computationally. The discontinuous Galerkin (DG) method emerges from combining aspects of two of these common methods and is the method utilized in the simulations presented in this work. With the DG method defined, we will work through an example application on a simple equation before showing the continuum-kinetic plasma model as implemented in the plasma simulation framework `Gkeyll`.

Physics studies begin in Chapter 3, where we cover results of fully kinetic simulations of a hydrodynamic instability in `Gkeyll`. First we will see the effects of using a spatially constant,

finite collision frequency on the development of the Rayleigh-Taylor (RT) instability, before allowing the collision frequency to vary in space and time. Results will demonstrate the relevance of collisional dynamics in a regime that is inaccessible to non-kinetic models.

Chapter 4 of this dissertation revisits the complex Fokker-Planck collision operator we will introduce later in this chapter. Using the highly optimized DG framework in `Gkeyll`, we introduce a breakthrough algorithm for the full nonlinear Fokker-Planck collision operator that is conservative in density, momentum, and energy. Results with this new algorithm will then be compared to the simplified methods that have been in use in `Gkeyll`.

With all of this in mind, let's begin by discussing mathematical models of plasma dynamics.

1.1 Plasma Kinetic Models

1.1.1 Klimontovich Equation

In a similar manner to [85], consider a gas composed of N_T total point particles, each with position $\mathbf{X}_i \in \mathbf{x}$ and velocity $\mathbf{V}_i \in \mathbf{v}$. The coordinates $\mathbf{x} = (x, y, z)$ and $\mathbf{v} = (v_x, v_y, v_z)$ define three-dimensional configuration and velocity spaces, and these independent variables can be combined into a single six-dimensional phase space variable, $\mathbf{z} = (\mathbf{x}, \mathbf{v})$. The density of a single particle, i, is

$$N_i(\mathbf{x}, \mathbf{v}, t) = \delta[\mathbf{x} - \mathbf{X}_i]\delta[\mathbf{v} - \mathbf{V}_i], \tag{1.1}$$

where $\delta[x]$ is the Dirac delta function. Velocity is simply the time derivative of position,

$$\frac{d}{dt}\mathbf{X}_i(t) = \mathbf{V}_i(t),$$

and the particle, having charge q_i and mass m_i, is subject to the Lorentz force,

$$\frac{d}{dt}\mathbf{V}_i(t) = \frac{q_i}{m_i}\left[\mathbf{E}^m(\mathbf{X}_i(t),t) + \mathbf{V}_i \times \mathbf{B}^m(\mathbf{X}_i(t),t)\right],$$

where the so-called "microscopic" electric and magnetic fields $\mathbf{E}^m$ and $\mathbf{B}^m$ contain both the large externally-applied fields and the smaller fields produced by other particles in the plasma. The fields are computed using Maxwell's equations,

$$\epsilon_0\mu_0\frac{\partial \mathbf{E}^m(\mathbf{x},t)}{\partial t} - \nabla_{\mathbf{x}} \times \mathbf{B}^m(\mathbf{x},t) = -\mu_0\mathbf{J}^m(\mathbf{x},t), \tag{1.2}$$

$$\frac{\partial \mathbf{B}^m(\mathbf{x},t)}{\partial t} + \nabla_{\mathbf{x}} \times \mathbf{E}^m(\mathbf{x},t) = 0, \tag{1.3}$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{E}^m(\mathbf{x},t) = \frac{\rho^m(\mathbf{x},t)}{\epsilon_0}, \tag{1.4}$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{B}^m(\mathbf{x},t) = 0, \tag{1.5}$$

where the gradients $\nabla_{\mathbf{x}} = (\nabla_x, \nabla_y, \nabla_z)$ and $\nabla_{\mathbf{v}} = (\nabla_{v_x}, \nabla_{v_y}, \nabla_{v_z})$, with microscopic charge density

$$\rho^m(\mathbf{x},t) = \sum_i q_i \int N_i(\mathbf{x},\mathbf{v},t)d\mathbf{v} \tag{1.6}$$

and microscopic current density

$$\mathbf{J}^m(\mathbf{x},t) = \sum_i q_i \int \mathbf{v}N_i(\mathbf{x},\mathbf{v},t)d\mathbf{v}. \tag{1.7}$$

The total density of a single species of particles (e.g. electrons), s, is

$$N_s(\mathbf{x},\mathbf{v},t) = \sum_{i=1}^{N_T} \delta[\mathbf{x} - \mathbf{X}_i]\delta[\mathbf{v} - \mathbf{V}_i].$$

Taking the time derivative of particle density yields the evolution equation,

$$\frac{\partial N_s(\mathbf{x},\mathbf{v},t)}{\partial t} = -\sum_{i=1}^{N_T} \frac{\partial \mathbf{X}}{\partial t} \cdot \nabla_{\mathbf{x}} \delta[\mathbf{x} - \mathbf{X}_i(t)]\delta[\mathbf{v} - \mathbf{V}_i(t)] - \sum_{i=1}^{N_T} \frac{\partial \mathbf{V}}{\partial t} \cdot \nabla_{\mathbf{v}} \delta[\mathbf{x} - \mathbf{X}_i(t)]\delta[\mathbf{v} - \mathbf{V}_i(t)]. \tag{1.8}$$

Inserting the Lorentz force in for the time derivative of $\mathbf{V}_i$ and applying the following property of the delta function,

$$a\delta(a-b) = b\delta(a-b),$$

we can perform some algebraic simplifications to arrive at the Klimontovich equation [68],

$$\frac{\partial N_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} N_s + \frac{q_s}{m_s} (\mathbf{E}^m + \mathbf{v} \times \mathbf{B}^m) \cdot \nabla_{\mathbf{v}} N_s = 0, \tag{1.9}$$

where we have dropped the functional dependence of the quantities for simplicity. While the Klimontovich-Maxwell system of equations does indeed completely describe the evolution of the plasma, this is a rather impractical model for our purposes. In the diffuse interstellar medium, a typical total particle number density is still on the order of $10^6\ m^{-3}$ [41]. Common systems of interest like the ionosphere or fusion plasmas are many orders of magnitude more dense with particles, and evolving and tracking each individual particle in such systems is simply unfeasible. Additionally, there is likely no context where a researcher would need the details of every single particle's position and trajectory; often, we would rather consider averaged flow properties of the plasma as a whole. We instead seek a model that averages out individual particle information while maintaining the effects that arise from discrete particle motion.

1.1.2 Vlasov-Boltzmann Equation

To average out individual particle information, we can borrow a concept from statistical mechanics and perform an ensemble average on the particle density function [e.g. 77, 104],

$$f_s(\mathbf{x}, \mathbf{v}, t) \equiv \langle N_s(\mathbf{x}, \mathbf{v}, t) \rangle. \tag{1.10}$$

The ensemble average operation can be thought of as an average over the infinite number of different microstates of the plasma that share the same bulk macrostate. Perhaps more intuitively, consider two identical plasmas initialized in thermal equilibrium under some constant external electromagnetic field. If a single particle position or velocity is perturbed in one case before the systems are allowed to evolve in time, the exact particle distributions of the plasmas (the microstates) will of course differ as time passes. However, they will still share the same bulk properties (the macrostate). Therefore, we take the ensemble average over all of these possible microstates of the plasma to get a smooth distribution from the "spiky" discrete particle densities, N_s. The resulting quantity, $f_s(\mathbf{x}, \mathbf{v}, t)$, is known as the *distribution function* of species s and quantifies the number of particles per unit phase space volume.

By transitioning from $N_s(\mathbf{x}, \mathbf{v}, t)$ to $f_s(\mathbf{x}, \mathbf{v}, t)$, we are now working with a smooth function rather than individual particles. It is worth noting here that the bulk configuration space properties of the plasma are calculated simply by taking velocity moments of f over the entire velocity space $\mathcal{V}$. The zeroth moment of the distribution function is the particle number density,

$$n_s(\mathbf{x}, t) = \int_{\mathcal{V}} f_s(\mathbf{x}, \mathbf{v}, t)\, d^3v, \tag{1.11}$$

the first velocity moment gives the *particle flux density*,

$$n_s(\mathbf{x},t)\mathbf{u}_s(\mathbf{x},t) = \int_{\mathcal{V}} \mathbf{v} f_s(\mathbf{x},\mathbf{v},t)\, d^3v, \tag{1.12}$$

and the second moment is the *stress tensor* (with trace proportional to the total energy density),

$$\boldsymbol{\mathcal{P}}_s(\mathbf{x},t) = m_s \int_{\mathcal{V}} \mathbf{v}\mathbf{v} f_s(\mathbf{x},\mathbf{v},t)\, d^3v, \tag{1.13}$$

$$\mathcal{E}_s(\mathbf{x},t) = \frac{1}{2}\mathcal{P}_s(\mathbf{x},t) = \frac{1}{2} m_s \int_{\mathcal{V}} v^2 f_s(\mathbf{x},\mathbf{v},t)\, d^3v, \tag{1.14}$$

where $\mathbf{vv}$ is a dyadic tensor. The elements of the stress tensor are defined as

$$\mathcal{P}_{ij} = m_s n_s (u_i u_j + \delta_{ij} v_{th}^2), \tag{1.15}$$

where δ_{ij} is the Kronecker delta. One common third velocity moment is the *particle energy-flux density*,

$$\mathcal{Q}_s(\mathbf{x},t) = \frac{1}{2} m_s \int_{\mathcal{V}} \mathbf{v} v^2 f_s(\mathbf{x},\mathbf{v},t)\, d^3v. \tag{1.16}$$

These velocity moments have been performed in the *laboratory frame*, where the plasma has nonzero bulk velocity, but they can also be taken in the *plasma frame* by multiplying by the relative velocity, $\mathbf{w} = \mathbf{v} - \mathbf{u}$, before integrating. In the plasma's rest frame, the zeroth moment is still n, but the first moment is, of course, 0. The second moment in this frame is the *pressure tensor*,

$$\mathbf{P}_s(\mathbf{x},t) = m_s \int_{\mathcal{V}} \mathbf{w}\mathbf{w} f_s(\mathbf{x},\mathbf{v},t)\, d^3v, \tag{1.17}$$

and its trace gives the scalar pressure, $p = \mathrm{Tr}(\mathbf{P}_s)/3$. Finally, in the plasma rest frame, the

third velocity moment similar to Eq. 1.16 is the *heat flux*,

$$\mathbf{q}_s(\mathbf{x},t) = \frac{1}{2} m_s \int_{\mathcal{V}} \mathbf{w} w^2 f_s(\mathbf{x},\mathbf{v},t)\, d^3v. \tag{1.18}$$

The relationship between laboratory and plasma frame moments can be found by substituting the definition of $\mathbf{w}$ into the integral and expanding, as demonstrated in Section 3.1.2.

From here, the domain $\mathcal{V}$ will be assumed for integrals over d^3v, that is, these integrals are performed over all velocity space. It is commonly known and can be simply shown, [e.g. 18, 25], that when there is no preferred velocity direction and no external forces, the distribution function must be a Maxwellian,

$$f(\mathbf{v}) = \frac{n}{\sqrt{2\pi v_{th}^2}^3} \exp\left(-\frac{(v_x-u_x)^2+(v_y-u_y)^2+(v_z-u_z)^2}{2v_{th}^2}\right), \tag{1.19}$$

with thermal velocity[1], $v_{th} = \sqrt{T/m}$, and no bulk velocity, $u_x = u_y = u_z = 0$. Note that it is conventional in plasma physics to express temperature in units of energy, implicitly multiplying temperature by Boltzmann's constant. Any external forces or inter-particle collisions imply that the velocity directions are no longer independent of one another, invalidating the initial assumption of such derivations. However, as will be demonstrated in Section 1.2.2, collisions serve to thermalize the distribution, driving it to a Maxwellian.

Returning to the kinetic equation, applying the ensemble average operation to the Klimontovich equation yields the evolution equation of the distribution function. First, we re-frame the particle density and microscopic fields discussed earlier as sums of the large-scale quan-

[1] In this form, the thermal velocity is the standard deviation of the Maxwellian distribution. Another common definition of the thermal velocity arises from setting equal the typical kinetic energy of a plasma and its temperature, $mv_{th}^2/2 = T$, yielding a thermal velocity larger than that used in this work by a factor of $\sqrt{2}$.

tities with small-scale perturbations arising from individual particle motion,

$$N_s = f_s + \delta N_s,$$
$$\mathbf{E}^m = \mathbf{E} + \delta\mathbf{E},$$
$$\mathbf{B}^m = \mathbf{B} + \delta\mathbf{B},$$

where the ensemble averages of perturbation quantities individually are 0, i.e. $\langle \delta N_s \rangle = \langle \delta\mathbf{E} \rangle = \langle \delta\mathbf{B} \rangle = 0$, and the macroscopic fields are ensemble averages of their microscopic counterparts, $\langle \mathbf{B}^m \rangle = \mathbf{B}$ and $\langle \mathbf{E}^m \rangle = \mathbf{E}$.

Inserting these forms into Eq. 1.9 and ensemble averaging, we obtain the Vlasov-Boltzmann equation,

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_s = C[f_s], \tag{1.20}$$

where the remaining perturbation terms from the ensemble average are collected into the right-hand-side operator,

$$C[f_s] = -\frac{q_s}{m_s} \langle (\delta\mathbf{E} + \mathbf{v} \times \delta\mathbf{B}) \cdot \nabla_{\mathbf{v}} \delta N_s \rangle. \tag{1.21}$$

The left-hand-side of Eq. 1.20 describes advection of the distribution function in configuration and velocity space. While the distribution function does not include individual particle information because of the ensemble average, the contributions of small-scale, individual particle effects stemming from the perturbation fields are accumulated into the right-hand-side, known as the *collision operator*. If collisional effects are negligible, as is often the case for diffuse, high temperature plasmas [41], we are left with the Vlasov equation,

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = 0. \tag{1.22}$$

However, if we wish to model a regime where collisional physics are relevant, we need a usable form of the collision operator.

1.2 Derivation of the Fokker-Planck Operator

To begin our discussion of collisional modeling in continuum-kinetics, we now derive the Fokker-Planck collision operator. This section follows the process of [54], but many equally valid alternative paths exist, e.g. [41, 77].

Given a particle species with distribution function $f(x, v, t)$, there is a probability, $\mathcal{P}(v, \Delta v)$, that the velocity changes from $v - \Delta v$ to v within time Δt as the result of a collision event. For convenience, the x-dependence of the distribution function will be omitted for the rest of this derivation. If we have a particle with velocity $v - \Delta v$ at time t, then the probability of that same particle having velocity v at time $t + \Delta t$ is $\mathcal{P}(v, \Delta v)$. The phase space density of particles satisfying such a condition is then $f(v - \Delta v, t)\mathcal{P}(v - \Delta v, \Delta v)$. Integrating over all possible Δv gives the total distribution function of particles at time $t + \Delta t$,

$$f(v, t + \Delta t) = \int f(v - \Delta v, t)\mathcal{P}(v - \Delta v, \Delta v) d\Delta v. \tag{1.23}$$

As will be shown, the change in velocity due to a single Coulomb collision is expected to be small, so we can expand the arguments of the integral in $v - \Delta v$,

$$\begin{aligned} f(v - \Delta v, t)\mathcal{P}(v - \Delta v, \Delta v) \approx & f(v, t)\mathcal{P}(v, \Delta v) + (-\Delta v)\frac{\partial}{\partial v}[f(v, t)\mathcal{P}(v, \Delta v)] \\ & + \frac{(\Delta v)^2}{2}\frac{\partial^2}{\partial v^2}[f(v, t)\mathcal{P}(v, \Delta v)] + ... \end{aligned}$$

Because Δv is expected to be small, we can neglect terms of higher order than $(\Delta v)^2$. This

will also be further justified later, where it will be shown that these higher order terms become smaller by a factor of the Coulomb logarithm, which is assumed to be large in plasmas of interest. Therefore, the distribution function of particles at time $t + \Delta t$ becomes approximately,

$$f(v, t+\Delta t) \approx \int \Big(f(v,t)\mathcal{P}(v,\Delta v) - \Delta v \frac{\partial}{\partial v}[f(v,t)\mathcal{P}(v,\Delta v)] + \frac{(\Delta v)^2}{2}\frac{\partial^2}{\partial v^2}[f(v,t)\mathcal{P}(v,\Delta v)] \Big)\, d\Delta v \tag{1.24}$$

Because $\mathcal{P}(v, \Delta v)$ is a probability,

$$\int \mathcal{P}(v,\Delta v) d\Delta v = 1.$$

The expectation values of Δv and $(\Delta v)^2$ (also referred to as the *velocity increments*) are given as

$$\langle \Delta v \rangle = \int \Delta v \mathcal{P}(v,\Delta v)\; d\Delta v,$$
$$\left\langle (\Delta v)^2 \right\rangle \int (\Delta v)^2 \mathcal{P}(v,\Delta v) d\Delta v.$$

Eq. 1.24 can then be rewritten as

$$f(v, t+\Delta t) \approx f(v,t) - \frac{\partial}{\partial v}[f(v,t)\,\langle \Delta v \rangle] + \frac{1}{2}\frac{\partial^2}{\partial v^2}[f(v,t)\,\left\langle (\Delta v)^2 \right\rangle]. \tag{1.25}$$

Because we are searching for the change of the distribution function in time as a result of

collisions, we can now apply the limit definition of the derivative,

$$\begin{aligned}\frac{\partial f(v,t)}{\partial t} &= \lim_{\Delta t \to 0} \frac{f(v,t+\Delta t) - f(v,t)}{\Delta t} \\ &= \frac{1}{\Delta t}\left(-\frac{\partial}{\partial v}(f(v,t)\langle \Delta v \rangle) + \frac{1}{2}\frac{\partial^2}{\partial v^2}(f(v,t)\left\langle (\Delta v)^2 \right\rangle)\right),\end{aligned}$$

to give the final collision operator in 1X1V (one configuration space dimension, one velocity space dimension),

$$C[f(v,t)] = \left(\frac{\partial f}{\partial t}\right)_{\text{coll}} = \underbrace{-\frac{\partial}{\partial v}\left(\frac{\langle \Delta v \rangle}{\Delta t} f(v,t)\right)}_{\text{drag}} + \underbrace{\frac{\partial^2}{\partial v^2}\left(\frac{\langle (\Delta v)^2 \rangle}{2\Delta t} f(v,t)\right)}_{\text{diffusion}}. \tag{1.26}$$

The first term of Eq. 1.26 has the form of an advection equation in velocity space and describes particle slowing due to friction. This drag component serves to group particles around $\mathbf{v} = 0$, corresponding to the distribution function growing a peak close to the origin of velocity space [77]. In contrast, the second term is a diffusion operator, governing diffusive spreading/flattening of f in velocity space. The net effect of collisions on the distribution function is therefore a balance between drag slowing particles towards 0 velocity and velocity space diffusion to higher velocities.

In order to generalize this operator to 3V, the expectation values of the velocity increments are replaced with corresponding vector and tensor quantities,

$$\begin{aligned}\langle \Delta v \rangle &\to \langle \Delta \mathbf{v} \rangle \\ \left\langle (\Delta v)^2 \right\rangle &\to \langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle,\end{aligned}$$

and the derivatives are replaced with divergence operations, yielding

$$C[f(\mathbf{v},t)] = -\nabla_{\mathbf{v}} \cdot \left[\frac{\langle \Delta \mathbf{v} \rangle}{\Delta t} f(\mathbf{v},t) - \nabla_{\mathbf{v}} \cdot \left(\frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle}{2\Delta t} f(\mathbf{v},t) \right) \right], \tag{1.27}$$

or component-wise,

$$C[f(\mathbf{v},t)] = -\frac{\partial}{\partial v_i} \left[\frac{\langle v_i \rangle}{\Delta t} f(\mathbf{v},t) - \frac{\partial}{\partial v_j} \left(\frac{\langle \Delta v_i \Delta v_j \rangle}{2\Delta t} f(\mathbf{v},t) \right) \right]. \tag{1.28}$$

Einstein's summation convention is used in Equation 1.28 and is assumed throughout this work unless otherwise noted. Under this convention, repeated indices in a single term indicates a sum over all values of the indices, e.g.

$$x_i y_i = x_1 y_1 + x_2 y_2 + x_3 y_3,$$

if $x = \{x_1, x_2, x_3\}$ and $y = \{y_1, y_2, y_3\}$. The velocity increments are dependent on the identities of the colliding species, e.g. for collisions of species s with species s', the velocity increments are denoted $\langle \Delta \mathbf{v} \rangle^{ss'}$ and $\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^{ss'}$. Summing over all species with which s collides gives the total collisional contribution to the rate of change of the distribution function of species s.

Now we must calculate the expectation values for the velocity changes. Consider an individual Coulomb collision in 2D between particle A (mass m_s and charge e_s) and particle B (mass $m_{s'}$ and charge $e_{s'}$), as in Figure 1.1. Particle A approaches particle B at distance b with initial velocity $\mathbf{v}$, in a frame where particle B is stationary. Through the Coulomb force, $F = e_s e_{s'}/4\pi\epsilon_0 r(t)^2$, particle A gains momentum in the y-direction and is deflected from its initial path by an angle α.

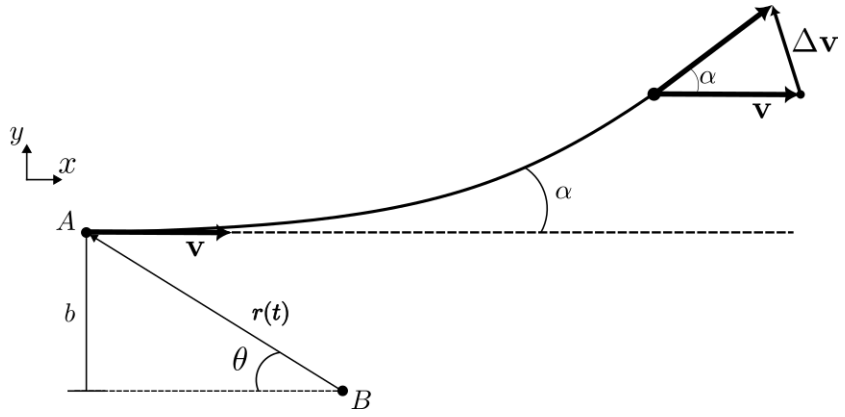

Figure 1.1: Diagram of a Coulomb collision between two particles with like charges. Particle A with velocity $\mathbf{v}$ approaches stationary particle B with impact parameter b. As a result of the collision, the path of A is deflected by angle α.

The momentum gained by particle A can be found by integrating Newton's 2nd Law,

$$\begin{aligned} m_s \Delta v_y &= \int F_y dt \\ &= \int_{-\infty}^{\infty} \frac{e_s e_{s'}}{4\pi\epsilon_0 r(t)^2} \sin\theta dt = \int_{-\infty}^{\infty} \frac{e_s e_{s'}}{4\pi\epsilon_0 r(t)^2} \frac{b}{r(t)} dt \\ &= \frac{e_s e_{s'}}{4\pi\epsilon_0} \int_{-\infty}^{\infty} \frac{b}{((vt)^2 + b^2)^{3/2}} dt \\ &= \frac{e_s e_{s'}}{2\pi\epsilon_0 bv}. \end{aligned}$$

The angle of deflection is then (using a small-angle approximation for α),

$$\sin\alpha \approx \alpha = \frac{\Delta v}{v} = \frac{e_s e_{s'}}{2\pi\epsilon_0 m_s b v^2} = \frac{r_{\min}}{b}.$$

We now introduce the collision cross-section, $\sigma_{ss'} = \pi r_{\min}^2$, where $r_{\min}$ is the velocity-dependent distance of closest approach. The cross-section is related to the collision frequency,

$\nu_{ss'}$, through

$$\nu_{ss'} = n_{s'}\sigma_{ss'}v = \frac{n_{s'}e_s^2 e_{s'}^2}{4\pi\epsilon_0^2 m_s^2 v^3}. \tag{1.29}$$

Clearly, the frequency of small-angle Coulomb collisions varies strongly with velocity, scaling as $\nu \sim v^{-3}$. This property has significant implications on the collisional dynamics of our model, as the fastest particles (in the high-energy tails of f) collide substantially less often than those in the bulk of the plasma that exists within approximately one thermal velocity around the peak of the Maxwellian. The highest-energy electrons, referred to as *superthermals*, have velocities greater than approximately $3v_{th}$ and are responsible for a large portion of the heat flux. These superthermals can escape local temperature gradients without colliding due to their long collision mean-free-paths, $\lambda_{\text{mfp}} = v_{th}/\nu$, depositing their energy elsewhere in the plasma [16, 17]. Accurate treatment of these superthermal electrons is of utmost importance when temperature scale lengths become similar to λ_{mfp}, and, as we will see, the FPO inherently includes a velocity-dependent collision frequency. It is worth mentioning here that a conventional characteristic collision time was defined by Braginskii [15] when considering the friction force between species, given by

$$\tau_{ss'} = \frac{1}{\nu_{ss'}} = \frac{12\pi^{3/2}}{\sqrt{2}} \frac{\epsilon_0^2 m_s^{1/2} T_s^{3/2}}{e_s^2 e_{s'}^2 n_{s'} \log\Lambda_{ss'}}. \tag{1.30}$$

The quantity $\log\Lambda_{ss'}$ is known as the Coulomb logarithm and will be introduced shortly.

For the remainder of this derivation however, we consider the cross-section for a typical particle in the distribution, i.e. one moving with velocity $v \approx v_{th}$. Therefore, the deflection angle is $\alpha = b_{\min}/b$, where

$$b_{\min} = \frac{e_s e_{s'}}{2\pi\epsilon_0 m_s v_{th,s}^2}. \tag{1.31}$$

We have thus calculated the deflection angle of a particle moving at velocity v_{th} undergoing

a Coulomb collision with a stationary particle at a large distance b. In the reference frame where both particles are moving, consider the Lagrangian of the system,

$$L = \frac{m\dot{\mathbf{x}}_s^2}{2} + \frac{m\dot{\mathbf{x}}_{s'}^2}{2} - \frac{e_s e_{s'}}{4\pi\epsilon_0|\mathbf{x_s} - \mathbf{x}|}.$$

Converting into center of mass coordinates with,

$$\mathbf{R} \equiv \frac{m_s\mathbf{x_s} + m_{s'}\mathbf{x}}{m_s + m_{s'}} \quad \mathbf{r} = \mathbf{x_s} - \mathbf{x_{s'}},$$

yields

$$L = \frac{(m_s + m_{s'})\dot{\mathbf{R}}^2}{2} + \frac{1}{2}\frac{m_s m_{s'}\dot{\mathbf{r}}^2}{m_s + m_{s'}} - \frac{e_s e_{s'}}{4\pi\epsilon_0 r}. \tag{1.32}$$

The Euler-Lagrange equation for this system,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{R}}}\right) = \frac{\partial L}{\partial \mathbf{R}},$$

yields

$$(m_s + m_{s'})\mathbf{R} = 0$$
$$\therefore \quad \dot{\mathbf{R}} = \text{const.},$$

so the center of mass of the two particle system moves at a constant velocity. The first term of the Lagrangian, Eq. 1.32, is therefore a constant, and the Lagrangian as a whole resembles that of a particle with mass $m_* = (m_s m_{s'})/(m_s + m_{s'})$ moving at velocity $\mathbf{u} = \dot{\mathbf{r}} = \mathbf{v_s} - \mathbf{v_{s'}}$ in a Coulomb field:

$$L = \text{const.} + \frac{1}{2}\frac{m_s m_{s'}\dot{\mathbf{r}}^2}{m_s + m_{s'}} - \frac{e_s e_{s'}}{4\pi\epsilon_0 r}.$$

As a result, we can treat the system with two moving particles identically to the previous

case, substituting m_* for m_s and $\mathbf{u}$ for $\mathbf{v}$, so the deflection angle in the x-y plane will be the same angle α found previously. If we now allow particle A to also deflect off the x-y plane by an angle ϕ, the changes in velocity of the center of mass system are then,

$$\Delta u_x = \Delta u \sin^2 \frac{\alpha}{2} = u(1 - \cos\alpha)$$
$$\Delta u_y = u \sin\alpha \cos\phi$$
$$\Delta u_z = u \sin\alpha \sin\phi.$$

Returning to the velocity-aligned frame ($\mathbf{u} \to \mathbf{v}$) is a simple change of variables,

$$\mathbf{x_s} = \mathbf{R} + \frac{m_{s'}}{m_s + m_{s'}}\mathbf{r}$$
$$\mathbf{v} = \frac{d}{dt}\left(\mathbf{x}_s\right) = \text{const.} + \frac{m_{s'}}{m_s + m_{s'}}\mathbf{u},$$

thus the velocity changes are

$$\Delta\mathbf{v} = \frac{m_{s'}}{m_s + m_{s'}}\Delta\mathbf{u}$$
$$\Delta v_x = \frac{m_{s'}}{m_s + m_{s'}}u(1 - \cos\alpha)$$
$$\Delta v_y = \frac{m_{s'}}{m_s + m_{s'}}u \sin\alpha \cos\phi$$
$$\Delta v_z = \frac{m_{s'}}{m_s + m_{s'}}u \sin\alpha \sin\phi.$$

These velocity changes can be simplified further by recalling that α is small,

$$\Delta v_x \approx \frac{m_{s'}}{m_s + m_{s'}} \frac{\alpha^2}{2} u = \left(1 + \frac{m_s}{m_{s'}}\right) \left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \frac{1}{2r^2 u^3}$$
$$\Delta v_y \approx \frac{e_s e_{s'}}{2\pi\epsilon_0 m_s} \frac{\cos\phi}{ur}$$
$$\Delta v_z \approx \frac{e_s e_{s'}}{2\pi\epsilon_0 m_s} \frac{\sin\phi}{ur}.$$

We now have the incremental changes to the velocity of particle A as a result of a single collision with particle B, but a real plasma consists of many particles undergoing potentially many such collision events. Therefore, we turn our attention to quantifying the cumulative effect of these collisions on the distribution function. Rather than the previous two-particle system, particle A now travels at velocity $\mathbf{v}$ through a distribution of particles of species b described by distribution function $f_{s'}(\mathbf{x}_{s'}, \mathbf{v}_{s'}, t)$. Consider collisions that occur within an impact parameter between r and $r + dr$ and angle ϕ and $\phi + d\phi$ through some time Δt. The volume described by these conditions multiplied by the density of particles of species b within the differential volume $d^3 v_{s'}$ around $\mathbf{v}$ gives the number of colliding particles,

$$\text{number of colliding particles} = \Delta t (ur d\phi dr)\, d^3 v_{s'} f_{s'}(\mathbf{x}_{s'}, \mathbf{v}_{s'}, t).$$

Integrating over all possible impact parameters r, angles ϕ, and velocities v_b gives the total number of collisions over time Δt (substituting $\mathbf{u} = \mathbf{v} - \mathbf{v}_{s'}$),

$$\text{number of collisions} = \Delta t \iiint |\mathbf{v} - \mathbf{v}_{s'}| f_b(\mathbf{v}_{s'}) r d\phi dr\, d^3 v_{s'},$$

where, again, the position- and time-dependence of the distribution function is omitted for simplicity. Multiplying the number of colliding particles by the velocity changes before integrating over r, ϕ, and v' gives the velocity increments we seek. Beginning with the Δv_x

expectation value,

$$\frac{\langle \Delta v_x \rangle^{ss'}}{\Delta t} = \left(1 + \frac{m_s}{m_{s'}}\right)\left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \iiint \frac{1}{2r^2u^3} r u f_{s'}(\mathbf{v}_{s'}) dr d\phi\, d^3v_{s'},$$

there is a divergent integral in r, so we must introduce some physically-motivated cutoff for the impact parameter integral. The natural choice for the lower bound is the minimum impact parameter derived earlier, $b_{\min}$. While this is a generally sound choice, in the case where temperatures (and therefore velocities) are large, the minimum impact parameter, which is also the classical distance of closest approach, can become smaller than the de Broglie wavelength. In such cases, the de Broglie wavelength should be used instead [41]. The Debye length, λ_D gives a convenient upper bound, as beyond this distance, the Coulomb potential is shielded by collective behavior of the bulk of the plasma. With these cutoffs, the integral becomes,

$$\begin{aligned}\frac{\langle \Delta v_x \rangle^{ss'}}{\Delta t} &= \left(1 + \frac{m_s}{m_{s'}}\right)\left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \int_{b_{\min}}^{\lambda_D} \frac{dr}{r} \int_0^{2\pi} d\phi \int \frac{1}{2u^2} f_{s'}(\mathbf{v}_{s'})\, d^3v_{s'} \\ &= \left(1 + \frac{m_s}{m_{s'}}\right)\left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \pi \ln\left(\frac{\lambda_D}{b_{\min}}\right) \int \frac{1}{u^2} f_{s'}(\mathbf{v}_{s'})\, d^3v_{s'}.\end{aligned}$$

The quantity $\ln(\lambda_D/b_{\min})$ is known as the Coulomb logarithm and is commonly denoted as $\ln\Lambda$. It can be shown that $\ln\Lambda \gg 1$ for most cases of interest (generally $\ln\Lambda$ lies between 10 and 20), and the rest of this analysis makes this assumption [59]. Our velocity increment expectation value then becomes,

$$\frac{\langle \Delta v_x \rangle^{ss'}}{\Delta t} = \Gamma_{ss'}\left(1 + \frac{m_s}{m_{s'}}\right) \int \frac{1}{u^2} f_{s'}(\mathbf{v}_{s'})\, d^3v_{s'}, \tag{1.33}$$

where we have defined,

$$\Gamma_{ss'} \equiv \frac{1}{4\pi}\left(\frac{e_s e_{s'}}{\epsilon_0 m_s}\right)^2 \ln\Lambda. \tag{1.34}$$

Treating $\langle \Delta v_y \rangle / \Delta t$ and $\langle \Delta v_z \rangle / \Delta t$ in the same way, we end up with integrals from 0 to 2π of $\cos\phi$ and $\sin\phi$, respectively, yielding,

$$\frac{\langle \Delta v_y \rangle^{ss'}}{\Delta t} = \frac{\langle \Delta v_z \rangle^{ss'}}{\Delta t} = 0.$$

We now have the elements of the velocity increment vector $\langle \Delta v_i \rangle^{ss'} / \Delta t$, so we next calculate the elements of the tensor $\langle \Delta v_i \Delta v_j \rangle^{ss'} / \Delta t$. To make our job simpler, we can take advantage of the assumption that α is small. Because $\Delta v_x \propto \alpha^2$ and both Δv_y and Δv_z are proportional to α, any products of Δv_x and another increment will be of at least order α^3, which we take as negligible. Additionally, the ϕ-integral for $\langle \Delta v_y \Delta v_z \rangle / \Delta t$ and $\langle \Delta v_z \Delta v_y \rangle / \Delta t$ are over $\sin\phi\cos\phi$, so those terms will be 0. Therefore, we only need to calculate $\langle (\Delta v_y)^2 \rangle^{ss'} / \Delta t$ and $\langle (\Delta v_z)^2 \rangle^{ss'} / \Delta t$. Beginning with $\langle (\Delta v_y)^2 \rangle^{ss'} / \Delta t$,

$$\begin{aligned}
\frac{\langle (\Delta v_y)^2 \rangle^{ss'}}{\Delta t} &= \left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \int_{\min}^{\lambda_D} \int_0^{2\pi} \int \frac{\cos^2\phi}{ur} f_{s'}(\mathbf{v}_{s'})\, d^3 v_{s'} d\phi dr \\
&= \left(\frac{e_s e_{s'}}{2\pi\epsilon_0 m_s}\right)^2 \pi \ln\Lambda \int \frac{1}{u} f_{s'}(\mathbf{v}_{s'})\, d^3 v_{s'} \\
&= \Gamma_{ss'} \int \frac{1}{u} f_{s'}(\mathbf{v}_{s'})\, d^3 v_{s'}.
\end{aligned}$$

Similarly, we determine,

$$\frac{\langle (\Delta v_z)^2 \rangle^{ss'}}{\Delta t} = \Gamma_{ss'} \int \frac{1}{u} f_{s'}(\mathbf{v}_{s'})\, d^3 v_{s'}.$$

Now that we have calculated the velocity increments, we have to transform the coordinate

system back from velocity-aligned coordinates, $\{\widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \widehat{\mathbf{z}}\}$, to a stationary lab frame, which can be any arbitrary coordinate system defined by basis vectors $\{\widehat{\mathbf{e}}_i, \widehat{\mathbf{e}}_j, \widehat{\mathbf{e}}_k\}$. Taking advantage of the fact that we know only $\langle \Delta v_x \rangle^{ss'}$, $\langle (\Delta v_y)^2 \rangle^{ss'}$, and $\langle (\Delta v_z)^2 \rangle^{ss'}$ are non-zero,

$$\frac{\langle \Delta v_i \rangle^{ss'}}{\Delta t} = \frac{\langle \widehat{\mathbf{e}}_i \cdot \widehat{\mathbf{x}} \Delta v_x \rangle^{ss'}}{\Delta t} = \frac{\langle (u_i/u) \Delta v_x \rangle^{ss'}}{\Delta t},$$

and

$$\begin{aligned}
\frac{\langle \Delta v_i \Delta v_j \rangle^{ss'}}{\Delta t} &= \frac{\langle (\widehat{\mathbf{e}}_i \cdot (\widehat{\mathbf{y}} \Delta v_y + \widehat{\mathbf{z}} \Delta v_z))(\widehat{\mathbf{e}}_j \cdot (\widehat{\mathbf{y}} \Delta v_y + \widehat{\mathbf{z}} \Delta v_z)) \rangle^{ss'}}{\Delta t} \\
&= \frac{\langle (\widehat{\mathbf{e}}_i \cdot \widehat{\mathbf{y}})(\widehat{\mathbf{e}}_j \cdot \widehat{\mathbf{y}})(\Delta v_y)^2 + (\widehat{\mathbf{e}}_i \cdot \widehat{\mathbf{z}})(\widehat{\mathbf{e}}_j \cdot \widehat{\mathbf{z}})(\Delta v_z)^2 \rangle^{ss'}}{\Delta t} \\
&= \frac{\langle (\widehat{\mathbf{e}}_\mathbf{i} \cdot \widehat{\mathbf{e}}_\mathbf{j} - (\widehat{\mathbf{e}}_\mathbf{i} \cdot \widehat{\mathbf{x}})(\widehat{\mathbf{e}}_\mathbf{j} \cdot \widehat{\mathbf{x}}))(\Delta v_y)^2 \rangle^{ss'}}{\Delta t} \\
&= \frac{\langle (\delta_{ij} - u_i u_j / u^2)(\Delta v_y)^2 \rangle^{ss'}}{\Delta t}.
\end{aligned}$$

Calculating the integrals for the expectation values yields,

$$\frac{\langle \Delta v_i \rangle^{ss'}}{\Delta t} = \Gamma_{ss'} \left(1 + \frac{m_s}{m_{s'}}\right) \int \frac{u_i}{u^3} f_{s'}(\mathbf{v}')\, d^3 v', \tag{1.35}$$

$$\frac{\langle \Delta v_i \Delta v_j \rangle}{\Delta t} = \Gamma_{ss'} \int \frac{u^2 \delta_{ij} - u_i u_j}{u^3} f_{s'}(\mathbf{v}')\, d^3 v', \tag{1.36}$$

where we replace the dummy integration variable $v_{s'}$ with v' for notational consistency with other works. As in [101], we now define the so-called Rosenbluth potentials,

$$H_{s'}(\mathbf{v}) \equiv \int \frac{1}{u} f_{s'}(\mathbf{v}')\, d^3 v' = \int \frac{1}{|\mathbf{v} - \mathbf{v}'|} f_{s'}(\mathbf{v}')\, d^3 v' \tag{1.37}$$

$$G_{s'}(\mathbf{v}) \equiv \int u f_{s'}(\mathbf{v}')\, d^3 v' = \int |\mathbf{v} - \mathbf{v}'| f_{s'}(\mathbf{v}')\, d^3 v', \tag{1.38}$$

which have the convenient properties

$$\frac{\partial H_{s'}(\mathbf{v})}{\partial v_i} = \int \frac{u_i}{u^3} f_{s'}(\mathbf{v}')\, d^3v',$$
$$\frac{\partial^2 G_{s'}(\mathbf{v})}{\partial v_i \partial v_j} = \int \frac{u^2\delta_{ij} - u_i u_j}{u^3} f_{s'}(\mathbf{v}')\, d^3v' = \int w_{ij} f_{s'}(\mathbf{v}')\, d^3v',$$

with the tensor

$$\mathbf{w} = \nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} u = \frac{u^2\mathbf{I} - \mathbf{uu}}{u^3},$$

where $\otimes$ indicates a tensor product and $\mathbf{uu}$ is a dyadic tensor.

The quantities $H_{s'}$ and $G_{s'}$ can also be written as the solutions to the following Poisson equations,

$$\nabla^2_{\mathbf{v}} H_{s'}(\mathbf{v}) = -4\pi f_{s'}(\mathbf{v}) \tag{1.39}$$
$$\nabla^2_{\mathbf{v}} G_{s'}(\mathbf{v}) = 2H_{s'}(\mathbf{v}), \tag{1.40}$$

hence Rosenbluth *potentials*. It is worthwhile to note that the Rosenbluth potentials can be determined analytically when considering collisions with a Maxwellian, yielding [41]

$$H_{s',M} = \frac{n_{s'}}{v} \mathrm{erf}\left(\frac{v}{\sqrt{2}v_{th,s'}}\right), \tag{1.41}$$

$$G_{s',M} = n_{s'}\sqrt{2}v_{th,s'}\left[\frac{1}{\sqrt{\pi}}\exp\left(-\frac{v^2}{2v^2_{th,s'}}\right) + \mathrm{erf}\left(\frac{v}{\sqrt{2}v_{th,s'}}\right)\left(\frac{\sqrt{2}v_{th,s'}}{v} + \frac{v}{\sqrt{2}v_{th,s'}}\right)\right], \tag{1.42}$$

where the speed, v, is instead the relative speed when there is nonzero bulk velocity, $v \to |\mathbf{v} - \mathbf{u}|$.

Defining,

$$h_{ss'}(\mathbf{v}) \equiv \Gamma_{ss'}\left(1+\frac{m_s}{m_{s'}}\right)H_{s'}(\mathbf{v})$$
$$g_{ss'}(\mathbf{v}) \equiv \Gamma_{ss'}G_{s'}(\mathbf{v}),$$

yields new forms of the velocity increments

$$\frac{\langle\Delta\mathbf{v}\rangle^{ss'}}{\Delta t} = \Gamma_{ss'}\left(1+\frac{m_s}{m_{s'}}\right)\nabla_{\mathbf{v}}H_{s'}(\mathbf{v}) = \nabla_{\mathbf{v}}h_{ss'}(\mathbf{v}) \tag{1.43}$$
$$\frac{\langle\Delta\mathbf{v}\Delta\mathbf{v}\rangle^{ss'}}{\Delta t} = \Gamma_{ss'}\nabla_{\mathbf{v}}\otimes\nabla_{\mathbf{v}}G_{s'}(\mathbf{v}) = \nabla_{\mathbf{v}}\otimes\nabla_{\mathbf{v}}g_{ss'}(\mathbf{v}). \tag{1.44}$$

With these forms of the velocity increments, the component-wise Fokker-Planck operator, Equation 1.28, accounting for collisions between species s and s' can now be written,

$$C_{ss'}[f_s,f_{s'}] = -\frac{\partial}{\partial v_i}\left[\frac{\partial h_{ss'}(\mathbf{v})}{\partial v_i}f_s(\mathbf{v}) - \frac{1}{2}\frac{\partial}{\partial v_j}\left(\frac{\partial^2 g_{ss'}(\mathbf{v})}{\partial v_j\partial v_k}f_s(\mathbf{v})\right)\right]. \tag{1.45}$$

By inserting the integral forms of the potentials into Equations 1.43 and 1.44 and performing some nontrivial algebra and simplifications, the Landau form [74] of the FPO is obtained,

$$C_{ss'}[f_s,f_{s'}] = \\ -\frac{m_s\Gamma_{ss'}}{2}\frac{\partial}{\partial v_i}\int w_{ij}\left[\frac{f_s(\mathbf{v})}{m_{s'}}\frac{\partial f_{s'}(\mathbf{v}')}{\partial v_j'} - \frac{f_{s'}(\mathbf{v}')}{m_s}\frac{\partial f_s(\mathbf{v})}{\partial v_j}\right]d^3v', \tag{1.46}$$

or, in vector notation,

$$C_{ss'}[f_s,f_{s'}] = \\ -\frac{m_s\Gamma_{ss'}}{2}\nabla_{\mathbf{v}}\cdot\int\mathbf{w}\cdot\left[\frac{f_s(\mathbf{v})}{m_{s'}}\nabla_{\mathbf{v}'}f_{s'}(\mathbf{v}') - \frac{f_{s'}(\mathbf{v}')}{m_s}\nabla_{\mathbf{v}}f_s(\mathbf{v})\right]d^3v'. \tag{1.47}$$

Now, consider the collision operator for species s undergoing only same-species collisions (simplified from Equation 1.27),

$$C_{ss}[f_s] = -\nabla_{\mathbf{v}} \cdot \left[\frac{\langle \Delta \mathbf{v} \rangle^s}{\Delta t} f_s(\mathbf{v}) - \nabla_{\mathbf{v}} \cdot \left(\frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^{ss}}{2\Delta t} f_s(\mathbf{v}) \right) \right].$$

This form of the collision operator is inconvenient to implement numerically, as the second term is actually a mixed advection-diffusion term. Utilizing the following relations,

$$\begin{aligned} \mathrm{Tr}\left(\frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^{ss}}{\Delta t} \right) &= \mathrm{Tr}(\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} g_{ss}) = 2\Gamma_{ss} H_s, \\ \nabla_{\mathbf{v}} \cdot \frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^{ss}}{\Delta t} &= \nabla_{\mathbf{v}} (\nabla_{\mathbf{v}}^2 g_{ss}) = 2\Gamma_{ss} \nabla_{\mathbf{v}} H_s, \end{aligned}$$

we can rewrite the collision operator,

$$\begin{aligned} C_{ss}[f_s] &= -\nabla_{\mathbf{v}} \cdot \left[\frac{\langle \Delta \mathbf{v} \rangle^s}{\Delta t} f_s - f_s \nabla_{\mathbf{v}} \cdot \frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^s}{2\Delta t} - \frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^s}{2\Delta t} \cdot \nabla_{\mathbf{v}} f_s \right] \\ &= -\nabla_{\mathbf{v}} \cdot \left[(\nabla_{\mathbf{v}} h_{ss}) f_s - f_s \Gamma_{ss} \nabla_{\mathbf{v}} H_s - \frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^s}{2\Delta t} \cdot \nabla_{\mathbf{v}} f_s \right] \\ &= -\nabla_{\mathbf{v}} \cdot \left[2 f_s \Gamma_{ss} \nabla_{\mathbf{v}} H_s - f_s \Gamma_{ss} \nabla_{\mathbf{v}} H_s - \frac{\langle \Delta \mathbf{v} \Delta \mathbf{v} \rangle^s}{2\Delta t} \cdot \nabla_{\mathbf{v}} f_s \right] \\ &= -\frac{1}{2} \nabla_{\mathbf{v}} \cdot \left[2 f_s \Gamma_{ss} \nabla_{\mathbf{v}} H_s - \Gamma_{ss} (\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} G_s) \cdot \nabla_{\mathbf{v}} f_s \right], \end{aligned}$$

where we have used the fact that, for a single species colliding with itself, $h_{ss} = 2\Gamma_{ss} H_s$. Defining the drag coefficient, $\mathbf{a} = 2\Gamma_{ss} \nabla_{\mathbf{v}} H_s$, and diffusion tensor, $\mathbf{D} = \Gamma_{ss} \nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} G_s$, yields the advection-diffusion form of the FPO for single-species collisions,

$$C_{ss}[f_s] = -\frac{1}{2} \nabla_{\mathbf{v}} \cdot (\mathbf{a} f_s - \mathbf{D} \cdot \nabla_{\mathbf{v}} f_s). \tag{1.48}$$

1.2.1 Conservation Properties of the FPO

In this section, we will verify that the FPO conserves density, momentum, and energy as one would expect from a robust collision operator. Proofs of conservation for the entire Vlasov-Maxwell-Fokker-Planck system can be found in [64]. While all the presented forms of the FPO are equivalent, we will use the Landau form for these proofs. For sanity and brevity, we will use this more compact form of Eq. 1.47,

$$C_{ss'}[f_s] = \gamma_{ss'} \nabla_{\mathbf{v}} \cdot \int \mathbf{K}_{ss'} \, d^3v', \tag{1.49}$$

with

$$\mathbf{K}_{ss'} = \mathbf{w} \cdot \left(\frac{f_s(\mathbf{v})}{m_{s'}} \nabla_{\mathbf{v}'} f_{s'}(\mathbf{v}') - \frac{f_{s'}(\mathbf{v}')}{m_s} \nabla_{\mathbf{v}} f_s(\mathbf{v}) \right),$$
$$\gamma_{ss'} = -\frac{m_s \Gamma_{ss'}}{2}.$$

Conservation of density. For the FPO to conserve particles, the following must be true [88]:

$$\int C_{ss'}[f_s] \, d^3v = 0.$$

Inserting Eq. 1.49 into this integral and applying the divergence theorem in v,

$$\int C_{ss'}[f_s] \, d^3v = \gamma_{ss'} \iint \nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'} \, d^3v \, d^3v'$$
$$= \gamma_{ss'} \oint \left(\int \mathbf{K}_{ss'} \cdot \widehat{\mathbf{n}} \, d^3v' \right) d\Omega,$$

where Ω is the boundary of a closed surface, we can take advantage of the fact that f and

$\partial f_s/\partial v$ vanish as velocity goes to infinity (and thus so does $\mathbf{K}_{ss'}$), giving

$$\int C_{ss'}[f_s]\, d^3v = 0$$

and proving conservation of density.

Conservation of momentum. In momentum conserving collisions between species s and s', the momentum lost by one species must be gained by the other species,

$$\mathbf{F}_{ss'} + \mathbf{F}_{s's} = \int m_s \mathbf{v} C_{ss'}[f_s]\, d^3v + \int m_{s'} \mathbf{v} C_{s's} f_{s'}\, d^3v = 0. \tag{1.50}$$

Beginning with the force on s,

$$\mathbf{F}_{ss'} = \gamma_{ss'} m_s \iint \mathbf{v}\, (\nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'})\, d^3v\, d^3v',$$

we integrate by parts,

$$\begin{aligned}\gamma_{ss'} m_s \iint \mathbf{v}\, (\nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'})\, d^3v\, d^3v' &= \gamma_{ss'} m_s \cancelto{0}{\oint \int \mathbf{v} \mathbf{K}_{ss'} \cdot \widehat{\mathbf{n}}\, d\Omega\, d^3v'} - \gamma_{ss'} m_s \iint \nabla_{\mathbf{v}} \mathbf{v} \cdot \mathbf{K}_{ss'}\, d^3v\, d^3v' \\ &= -\gamma_{ss'} m_s \iint \mathbf{K}_{ss'}\, d^3v\, d^3v'.\end{aligned}$$

Substituting back in for $\mathbf{K}$,

$$\mathbf{F}_{ss'} = -\gamma_{ss'} m_s \iint \mathbf{w} \cdot \left(\frac{f_s(\mathbf{v})}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v}')}{\partial \mathbf{v}'} - \frac{f'_s(\mathbf{v}')}{m_s} \frac{\partial f_s(\mathbf{v})}{\partial \mathbf{v}} \right) d^3v\, d^3v',$$

we can now exchange the species s and s' (note $\gamma_{ss'} = m_{s'}\gamma_{s's}/m_s$) then swap the dummy

variables v and v' under the integral,

$$\begin{aligned}\mathbf{F}_{s's} &= -\gamma_{s's} m_{s'} \iint \mathbf{w} \cdot \left(\frac{f_{s'}(\mathbf{v})}{m_s} \frac{\partial f_s(\mathbf{v}')}{\partial \mathbf{v}'} - \frac{f_s(\mathbf{v}')}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v})}{\partial \mathbf{v}} \right) d^3v \, d^3v', \\ &= -\gamma_{ss'} m_s \iint \mathbf{w} \cdot \left(\frac{f_{s'}(\mathbf{v}')}{m_s} \frac{\partial f_s(\mathbf{v})}{\partial \mathbf{v}} - \frac{f_s(\mathbf{v})}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v}')}{\partial \mathbf{v}'} \right) d^3v' \, d^3v, \\ &= -\mathbf{F}_{ss'},\end{aligned}$$

demonstrating that $\mathbf{F}_{ss'} = -\mathbf{F}_{s's}$, satisfying Eq. 1.50 and proving that the FPO conserves momentum.

Conservation of energy. Similarly to momentum conservation, the rate of work done on or by species s through collisions with species s' must be balanced by the rate of energy transfer from collisions of species s' with species s. Mathematically, the following must hold:

$$\mathcal{E}_{ss'} + \mathcal{E}_{s's} = \frac{1}{2} m_s \int v^2 C_{ss'}[f_s] \, d^3v + \frac{1}{2} m_{s'} \int v^2 C_{s's}[f_{s'}] \, d^3v = 0. \tag{1.51}$$

Integrating $\mathcal{E}_{ss'}$ by parts,

$$\begin{aligned}\mathcal{E}_{ss'} &= \frac{1}{2} \gamma_{ss'} m_s \iint v^2 \nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'} \, d^3v \, d^3v' \\ &= -\frac{1}{2} \gamma_{ss'} m_s \iint \nabla_{\mathbf{v}} v^2 \cdot \mathbf{K} \, d^3v \, d^3v' \\ &= -\gamma_{ss'} m_s \iint \mathbf{v} \cdot \mathbf{K} \, d^3v \, d^3v' \\ &= -\gamma_{ss'} m_s \iint \mathbf{v} \cdot \mathbf{w} \cdot \left(\frac{f_s(\mathbf{v})}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v}')}{\partial \mathbf{v}'} - \frac{f_{s'}(\mathbf{v}')}{m_s} \frac{\partial f_s(\mathbf{v})}{\partial \mathbf{v}} \right) d^3v \, d^3v',\end{aligned}$$

we again exchange the species and swap v and v',

$$\mathcal{E}_{s's} = -\gamma_{ss'} m_s \iint \mathbf{v}' \cdot \mathbf{w} \cdot \left(\frac{f_{s'}(\mathbf{v}')}{m_s} \frac{\partial f_s(\mathbf{v})}{\partial \mathbf{v}} - \frac{f_s(\mathbf{v})}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v}')}{\partial \mathbf{v}'} \right) d^3v' \, d^3v.$$

Therefore,

$$\mathcal{E}_{ss'} + \mathcal{E}_{s's} = -\gamma_{ss'} m_s \iint \mathbf{u} \cdot \mathbf{w} \cdot \mathbf{K}\, d^3v\, d^3v',$$

and

$$\begin{aligned}\mathbf{u} \cdot \mathbf{w} &= \mathbf{u} \cdot (\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} u) = \mathbf{u} \cdot \frac{u^2\mathbf{I} - \mathbf{uu}}{u^3} \\ &= \frac{u^2\mathbf{u}}{u^3} - \frac{\mathbf{u} \cdot \mathbf{u}}{u^3}\mathbf{u} \\ &= \frac{\mathbf{u}}{u} - \frac{\mathbf{u}}{u} = 0,\end{aligned}$$

demonstrating that the FPO satisfies Eq. 1.51 and conserves energy.

1.2.2 Boltzmann's H-Theorem for the FPO

In order to determine the long-term steady-state solution of the FPO, we turn to Boltzmann's H-theorem [13, 24, 54]. We first define the quantity H,

$$H_s(\mathbf{x}, t) = \int f_s(\mathbf{x}, \mathbf{v}, t) \ln f_s(\mathbf{x}, \mathbf{v}, t)\, d^3v, \tag{1.52}$$

which can be interpreted as an entropy density and has the following time derivative,

$$\frac{\partial H_s}{\partial t} = \int (1 + \ln f_s) \frac{\partial f_s}{\partial t}\, d^3v. \tag{1.53}$$

Considering a spatially uniform plasma under no external forces and only affected by collisions, the $\partial f_s/\partial t$ term can be replaced by the collision operator to quantify the rate of

entropy production from collisions between s and s',

$$\begin{aligned}\frac{\partial H_{ss'}}{\partial t} &= \int (1+\ln f_s) C_{ss'}[f_s]\, d^3v \\ &= \gamma_{ss'} \iint (1+\ln f_s) \nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'}\, d^3v\, d^3v'.\end{aligned}$$

Integrating by parts in v, we find

$$\begin{aligned}\frac{\partial H_{ss'}}{\partial t} &= \gamma_{ss'} \iint (1+\ln f_s) \nabla_{\mathbf{v}} \cdot \mathbf{K}_{ss'}\, d^3v\, d^3v' = -\gamma_{ss'} \iint \mathbf{K}_{ss'} \cdot \nabla_{\mathbf{v}} \ln f_s\, d^3v\, d^3v' \\ &= -\gamma_{ss'} \iint \mathbf{w} \cdot \left(\frac{f_s(\mathbf{v})}{m_{s'}} \frac{\partial f_{s'}(\mathbf{v}')}{\partial \mathbf{v}'} - \frac{f_{s'}(\mathbf{v}')}{m_s} \frac{\partial f_s(\mathbf{v})}{\partial \mathbf{v}} \right) \cdot \nabla_{\mathbf{v}} \ln f_s\, d^3v\, d^3v' \\ &= -\gamma_{ss'} \iint \mathbf{w} \cdot \left(\frac{f_s(\mathbf{v})}{m_{s'}} \nabla_{\mathbf{v}'} f_{s'}(\mathbf{v}') - \frac{f_{s'}(\mathbf{v})}{m_s} \nabla_{\mathbf{v}} f_s(\mathbf{v}) \right) \cdot \nabla_{\mathbf{v}} \ln f_s\, d^3v\, d^3v'.\end{aligned}$$

As in the conservation proofs, we swap the species and dummy variables to get $H_{s's}$,

$$\frac{\partial H_{s's}}{\partial t} = -\frac{m_s}{m_{s'}} \gamma_{ss'} \iint \mathbf{w} \cdot \left(\frac{f_{s'}(\mathbf{v}')}{m_s} \nabla_{\mathbf{v}} f_s(\mathbf{v}) - \frac{f_s(\mathbf{v})}{m_{s'}} \nabla_{\mathbf{v}'} f_{s'}(\mathbf{v}') \right) \cdot \nabla_{\mathbf{v}'} \ln f_{s'}\, d^3v'\, d^3v.$$

Boltzmann's H-theorem for the collision operator states that [88]

$$\frac{\partial H_{ss'}}{\partial t} + \frac{\partial H_{s's}}{\partial t} \geq 0. \tag{1.54}$$

Substituting in for the time derivatives, we find

$$\frac{\partial H_{ss'}}{\partial t} + \frac{\partial H_{s's}}{\partial t} = -\frac{\gamma_{ss'}}{m_s} \iint (\mathbf{w} \cdot \mathbf{K}) \cdot \left(\frac{1}{m_s} \nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) - \frac{1}{m_{s'}} \nabla_{\mathbf{v}'} \ln f_{s'}(\mathbf{v}') \right) d^3v\, d^3v',$$

and noting that

$$f_s(\mathbf{v}) f_{s'}(\mathbf{v}') \left(\frac{1}{m_{s'}} \nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) - \frac{1}{m_s} \nabla_{\mathbf{v}'} \ln f_{s'}(\mathbf{v}') \right) = \mathbf{K},$$

we can rewrite this result as

$$\frac{\partial H_{ss'}}{\partial t} + \frac{\partial H_{s's}}{\partial t} = -\frac{\gamma_{ss'}}{m_s} \iint \mathbf{a} \cdot \mathbf{w} \cdot f_s(\mathbf{v}) f_{s'}(\mathbf{v}') \mathbf{a} \, d^3v \, d^3v', \tag{1.55}$$

where we have defined

$$\mathbf{a} = \frac{1}{m_s} \nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) - \frac{1}{m_{s'}} \nabla_{\mathbf{v}'} \ln f_{s'}(\mathbf{v}').$$

To verify that Eq. 1.55 is always greater than or equal to zero, first note that f is a probability and is therefore always positive, as is the product $f_s f_{s'}$ in Eq. 1.55. The rest of the vector/tensor product in the integrand can be expressed as

$$\begin{aligned} \mathbf{a} \cdot \mathbf{w} \cdot \mathbf{a} = \mathbf{a} \cdot (\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} u) \cdot \mathbf{a} &= \mathbf{a} \cdot \frac{u^2 \mathbf{I} - \mathbf{u}\mathbf{u}}{u^3} \cdot \mathbf{a} \\ &= \frac{u^2 \mathbf{a} - (\mathbf{a} \cdot \mathbf{u})\mathbf{u}}{u^3} \cdot \mathbf{a} \\ &= \frac{a^2 u^2 - (\mathbf{a} \cdot \mathbf{u})^2}{u^3} \geq 0. \end{aligned}$$

Therefore, the integral is always positive, and the negative sign in front is canceled by the negative sign in the definition of $\gamma_{ss'}$, verifying the H-theorem, Eq. 1.54, and proving that entropy production from collisions is always positive or zero.

The steady-state solution of the collision operator is determined by considering the case where entropy production is 0,

$$\frac{\partial H_{ss'}}{\partial t} + \frac{\partial H_{s's}}{\partial t} = 0.$$

In order for the entropy production to be identically 0, the following must hold:

$$\frac{a^2 u^2 - (\mathbf{a} \cdot \mathbf{u})^2}{u^3} = 0,$$

which is the case only if $\mathbf{a}$ and $\mathbf{u}$ are parallel. As in [88], the definition of $\mathbf{a}$ can thus be written as

$$\frac{1}{m_s}\nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) - \frac{1}{m_{s'}}\nabla_{\mathbf{v}'} \ln f_{s'}(\mathbf{v}') = G(\mathbf{v}, \mathbf{v}')\mathbf{u}, \tag{1.56}$$

where $G(\mathbf{v}, \mathbf{v}')$ is some unknown function. For this equality to be true for all possible combinations of $\mathbf{v}$ and $\mathbf{v}'$, $G(\mathbf{v}, \mathbf{v}')$ must be a constant, G_0. Setting $\mathbf{v}' = 0$, we have

$$\begin{aligned}\frac{1}{m_s}\nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) &= G_0\mathbf{v} + \frac{1}{m_{s'}}\nabla_{\mathbf{v}'} \ln f_{s'}(0) \\ &= G_0\mathbf{v} + G_1,\end{aligned} \tag{1.57}$$

where the $\ln f_{s'}$ term on the right-hand-side is some new constant G_1, or, with $\mathbf{v} = 0$,

$$\begin{aligned}\frac{1}{m_{s'}}\nabla_{\mathbf{v}} \ln f_{s'}(\mathbf{v}') &= G_0\mathbf{v}' + \frac{1}{m_s}\nabla_{\mathbf{v}} \ln f_s(0) \\ &= G_0\mathbf{v}' + G_1,\end{aligned}$$

Rearranging Eq. 1.57 and solving for f_s, we arrive at

$$\nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) = m_s G_0\mathbf{v} + G_1 = m_s G_2(\mathbf{v} + \mathbf{G_3})$$

Because f should approach 0 as $v \to \infty$, the constant G_2 must be negative,

$$\nabla_{\mathbf{v}} \ln f_s(\mathbf{v}) = m_s G_0\mathbf{v} + G_1 = -m_s G_2(\mathbf{v} + \mathbf{G_3}),$$

so we find f_s must have the following functional form,

$$f_s(\mathbf{v}) = G_4 \exp\left(-m_s G_2|\mathbf{v} + G_3|^2\right). \tag{1.58}$$

The constants in this expression can be determined such that the bulk plasma properties are recovered when taking velocity moments of the distribution function, e.g. [18], eventually leading us to the Maxwellian distribution as the long-time equilibrium solution to the collision operator for both colliding species,

$$f_s(\mathbf{x},\mathbf{v},t) = \frac{n_s}{\sqrt{2\pi v_{th}^2}^3}\exp\left(-\frac{(v_x-u_x)^2+(v_y-u_y)^2+(v_z-u_z)^2}{2v_{th,s}^2}\right),$$
$$f_{s'}(\mathbf{x},\mathbf{v},t) = \frac{n_{s'}}{\sqrt{2\pi v_{th}^2}^3}\exp\left(-\frac{(v_x-u_x)^2+(v_y-u_y)^2+(v_z-u_z)^2}{2v_{th,s}^2}\right).$$

While the number densities may differ, the two Maxwellians have the same temperature and bulk velocity. As mentioned in Section 1.1.2, the steady state distribution function is Eq. 1.19 when there is no preferred velocity direction, but this discussion demonstrates that the collision operator serves to force a perturbed distribution function back to Maxwellian, even under the influence of external forces that introduce anisotropy.

1.3 Approximate Collision Operators

1.3.1 Dougherty-Lenard-Bernstein Operator

The Fokker-Planck collision operator, while an accurate model for collision dynamics, is a complex nonlinear integro-differential equation requiring calculation of the Rosenbluth potentials for the drag and diffusion coefficients. We seek a simplified collision model that maintains similar structure, i.e. is still a Fokker-Planck operator including the effects of drag and diffusion but does not require full calculation of the velocity increments via Rosenbluth potentials. Therefore, consider the following substitution of moments of the distribution

function for the drag and diffusion coefficients proposed by [32, 52, 75],

$$C_{ss}[f_s] = \nu_{ss}\nabla_{\mathbf{v}} \cdot \left[(\mathbf{v} - \mathbf{u}_s) f_s + \frac{T_s}{m_s} \nabla_{\mathbf{v}} f_s \right], \tag{1.59}$$

where $\nu_{ss}(\mathbf{x}, t)$ is an average collision frequency, and $T_s(\mathbf{x}, t)$ and $\mathbf{u}_s(\mathbf{x}, t)$ are calculated from velocity moments of f_s. This form can also be found by making the following substitutions for the Rosenbluth potentials,

$$H_s^{\text{LBO}} = -\frac{\nu_{ss}}{\Gamma_{ss}} \frac{(\mathbf{v} - \mathbf{u}_s)^2}{2}, \tag{1.60}$$

$$G_s^{\text{LBO}} = \frac{\nu_{ss}}{\Gamma_{ss}} \frac{T_s}{m_s} \mathbf{v}^2, \tag{1.61}$$

and inserting into the full FPO. These simplifications are significant in that the coefficients are now functions of configuration space only, rather than all phase space like $\mathbf{a}$ and $\mathbf{D}$. Additionally, the diffusion tensor has been reduced to a scalar factor. However, this model, known as the Dougherty-Lenard-Bernstein operator (LBO), retains important aspects of the full FPO. First, we clearly have not made any changes to the structure of the equation, so this collision model still includes the effects of drag and diffusion on the distribution function. The LBO also maintains the conservation properties present in the full FPO, and the steady-state solution is a Maxwellian [see 52, 64, for details]. One significant shortcoming of this model however, is the use of the mean collision frequency, ν_{ss}. The effective collision frequency of the full FPO is a function of particle velocity, proportional to v^{-3}. As a result, the LBO will overestimate the effect of collisions in the high-energy tails of the distribution function. While this is generally not an issue when considering dynamics of the bulk plasma, applications of the LBO to regimes where the motion of the fastest particles is relevant, such as fusion energy, will not correctly capture the important motions of those highest-energy particles. Regardless, the LBO model is still an efficient simplification of the FPO that

maintains some important properties. However, what if this model is still more complex than necessary for a given study?

1.3.2 Bhatnagar-Gross-Krook Operator

The FPO and LBO relax a non-equilibrium distribution function to Maxwellian via the cumulative effects of drag and diffusion. We can instead choose to skip the drag and diffusion contributions and directly relax the distribution function to a Maxwellian,

$$C[f_{ss}] = \nu_{ss}(f_{M,s} - f_s), \tag{1.62}$$

where $f_{M,s}$ is a Maxwellian distribution calculated from moments of f_s. This model is known as the Bhatnagar-Gross-Krook (BGK) operator [9]. Clearly, the form of the BGK operator is a significant departure from the FPO and LBO, as we have chosen to directly force the distribution function towards Maxwellian at a rate proportional to the collision frequency. This simple collision operator is well-suited to large-angle binary collisions between neutral particles, rather than the small-angle Coulomb collisions discussed to this point. The lack of velocity-dependence in the collision frequency results in, similarly to the LBO, a significant overestimation of the effects of collisions in the high-energy-tails of the distribution function. However, two important aspects are maintained: relaxation to steady-state Maxwellian and conservation of particles, momentum, and energy. Conservation is rather simply proved by recalling that f_M is constructed using moments of f, so taking the first three moments of the BGK operator will result in the right-hand-side vanishing identically. The BGK operator is therefore best suited to studies of neutral particle interactions (neutral-neutral, ion-neutral, or electron-neutral) where the gas/plasma is sufficiently collisional such that any departures from Maxwellian are small and ideally occur near the peak of the distribution function.

1.4 Summary of Chapter 1

In this chapter, we introduced the definitions of a plasma and some ways the motion of individual particles can be modeled. While simple to derive, the Klimontovich equation, Eq. 1.9, contains substantially more information than we could ever use. Taking the ensemble average of the Klimontovich equation results in the Vlasov-Boltzmann equation, Eq. 1.22, that describes the evolution of the phase space distribution of particles. An important piece of the Boltzmann equation is the collision operator, which, when derived from the Klimontovich equation, is expressed in the form of small-scale electromagnetic fluctuations due to single-particle dynamics. To derive at a more usable form of the collision operator, we began by considering the effect of collisions on the distribution function as a probabilistic phenomenon, resulting in the Fokker-Planck collision operator, Eq. 1.28. The velocity increments in the FPO were then formulated by considering small-angle Coulomb collisions, eventually leading us to three equivalent forms of the FPO: the Rosenbluth form (Eq. 1.45), the Landau form (Eq. 1.46), and the advection-diffusion form (Eq. 1.48). Conservation properties of the FPO were then demonstrated, and Boltzmann's H-theorem was utilized to show that the steady-state solution to the FPO is a Maxwellian. While these collision models are accurate, they are deeply complex nonlinear equations of f. We therefore sought simplifications to the FPO, first by approximating the velocity increments to arrive at the LBO, Eq. 1.59. An even simpler model was then constructed by considering the cumulative effect of the FPO and LBO: relaxation to a Maxwellian distribution. This simpler BGK model, Eq. 1.62, forgoes the finer-scale features of the LBO and FPO but is conceptually and mathematically simpler and is actually well-suited to modeling neutral species collisions.

With this physical background in mind, the question remains of how to solve these types of equations computationally. In the next chapter, we will introduce some common numerical

methods used to solve partial differential equations computationally, and we will discuss the sophisticated discontinuous Galerkin method that is used in the plasma simulation framework `Gkeyll` that powers all simulations in this work.

Chapter 2

The Discontinuous Galerkin Finite Element Method

2.1 Background: Common Numerical Methods

Numerical solution of partial differential equations is not often a straightforward process. The first step we must consider when applying computational techniques to such problems is how to represent a continuous function or set of functions and their derivatives in a manner usable by computers. This process generally involves "discretizing" the continuous function and defining a computational space with finite extents in independent variables that is then decomposed into cells or nodes that "contain" the discrete representation. A variety of techniques exist for discretizing functions in this manner, each with a distinct form of the discrete representation. The three most commonly utilized techniques are known as finite difference, finite volume, and finite element methods, and we will go over the basic ideas of each in this section. Each of these methods has a long history of use in a variety of fields, and a more in-depth introduction to each can be found in many numerical methods textbooks, e.g. [56, 58]. To help clarify our discussion of discretization techniques, we will apply each method to the one-dimensional advection equation for scalar quantity $u(x,t)$ with flux $F(u(x,t))$,

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0. \tag{2.1}$$

2.1.1 Finite Difference Method

We will begin with the finite difference method, as it is the conceptually simplest technique of the three. The finite difference technique emerges from the Taylor expansion of some function, $F(x)$, at point $x_0 + h$,

$$F(x_0 + h) = F(x_0) + F'(x_0)h + F''(x_0)\frac{h^2}{2!} + \cdots,$$

where the remaining terms are proportional to higher powers of h. Accumulating all terms higher than second order in h into a single remainder term, $R(x)$, and rearranging to solve for the first x derivative of F yields

$$F'(x_0) = \frac{F(x_0 + h) - F(x_0)}{h} - \frac{R(x)}{h}, \tag{2.2}$$

known as a forward difference formula, which bears remarkable similarity to the limit definition of the derivative if the remainder is sufficiently small and we take the limit as $h \to 0$. We can quantify the magnitude of the residual (the *truncation error*) by noting the first truncated term, proportional to h^2, would retain one factor of h after the division, so this scheme is said to be *first-order*. This approximation of $F'(x_0)$ appears sound, but we can gain accuracy if we assume the function is smooth and include information from the expansion in the other direction away from x_0,

$$F(x_0 - h) = F(x_0) - F'(x_0)h + F''(x_0)\frac{h^2}{2} + \cdots,$$

to achieve the backward difference formula,

$$F'(x_0) = \frac{F(x_0) - F(x_0 - h)}{h} - \frac{R(x)}{h}. \tag{2.3}$$

Adding these two formulas, accumulating the residuals, and solving for the derivative again yields the central difference formula,

$$F'(x_0) = \frac{F(x_0 + h) - F(x_0 - h)}{2h} - \frac{R(x)}{h}. \tag{2.4}$$

Because the terms in the residual proportional to h^2 have opposite signs in the two Taylor expansions, they cancel in the addition, resulting in a leading term in the residual proportional to h^3 and a second-order approximation.

With this approximation of the derivative in mind, the finite difference method involves laying out a grid of points in the computational domain $x_k, k \in N$ with grid spacing $h_k = x_{k+1} - x_k$, where N is the total number of points. We can now insert a finite difference like the central difference formula, Eq. 2.4, for spatial derivative of the advection equation, Eq. 2.1,

$$\frac{\partial u_h(x_k, t)}{\partial t} + \frac{F_h(x_{k+1}, t) - F_h(x_{k-1}, t)}{h_k + h_{k-1}} = R_h(x), \tag{2.5}$$

where we have updated the notation slightly, denoting our discrete solution and flux as u_h and F_h, and redefining $x_{k\pm1} \equiv x_k \pm h$ and $h_k \equiv x_{k+1} - x_k$. Finite difference methods inherently assume u_h and F_h are well approximated by polynomials around x_k because with the Taylor expansions, we are essentially interpolating the continuous quantities between these grid points using Taylor polynomials. Any deviations from this approximation accumulate in the residual, $R_h(x)$, quantifying the discrepancy between the approximate solution and the exact solution of Eq. 2.1. We can then specify the manner in which we want the discrete solution to satisfy the original equation. Because our solution is defined at the grid locations x_k, we can require the residual to vanish at each grid point, $R(x_k) = 0$. Enforcing this constraint at each grid point results in a linear system of N equations that must be solved to compute the solution $u_h(x_k, t)$ at each grid point. It is worth mentioning that the order of accuracy of a

finite difference formula can be arbitrarily increased by simply including Taylor expansions of more neighboring points. This idea can be applied to construct high-order one-sided schemes, e.g. a second-order scheme using information from the lower side of point k that uses data from points k, $k-1$, and $k-2$.

The finite difference method is applied here to the spatial derivative of Eq. 2.1, but the time derivative has not yet been addressed. A proper introduction to time integration techniques requires a rigorous discussion of consistency, stability, and convergence that is beyond the scope of this work [see 58, 87], but we now briefly digress to a high-level exploration of these topics for completeness. A numerical scheme is *stable* if numerical errors remain bounded as the simulation advances in time to infinity, and it is *consistent* if the difference between the continuous differential equation (e.g. Eq. 2.1) and the discrete scheme (e.g. Eq. 2.5) vanishes as grid spacing and timestep approach 0. *Convergence* refers to the fact that the discrete solution should be an accurate representation of the exact numerical solution to the continuous differential equation. As grid spacing and timestep are decreased, the difference between the discrete and exact solutions (the truncation error) tends to 0 in a convergent scheme. The rate at which the truncation error decreases as the computational mesh is refined is the *observed* order of accuracy (as opposed to the *formal* order of accuracy found when deriving the scheme above) [87]. While a time derivative could intuitively be handled in exactly the same manner as the spatial derivatives above, care must be taken as different combinations of temporal and spatial discretizations can have drastically different effects on the stability of the scheme.

Consider replacing the time derivative in Eq. 2.5 with a forward difference in time with interval Δt. Setting the residual to be 0 at point x_k again, we arrive at

$$\frac{u_h(x_k, t+\Delta t) - u_h(x_k, t)}{\Delta t} = -\frac{F_h(x_{k+1}, t) - F_h(x_{k-1}, t)}{h_k + h_{k-1}}, \tag{2.6}$$

giving us a simple method to calculate the solution at the next timestep. This is an *explicit* scheme known as the (forward) Euler method. In an explicit scheme, we are solving for a single unknown value at time $t + \Delta t$ using known quantities at time t. Naturally, such a conveniently simple technique has a drawback: the timestep must be very small for the overall scheme to be numerically stable, as can be demonstrated by von Neumann stability analysis [58]. In contrast, the *implicit* backward Euler method would arise if we instead evaluate our equation at $t + \Delta t$ and take a backward difference in time,

$$\frac{u_h(x_k, t + \Delta t) - u_h(x_k, t)}{\Delta t} + \frac{F_h(x_{k+1}, t + \Delta t) - F_h(x_{k-1}, t + \Delta t)}{h_k + h_{k-1}} = 0. \tag{2.7}$$

Three unknown quantities appear in this form, so computing the solution $u_h(x_k, t + \Delta t)$ with an implicit method generally involves iterative techniques or large linear solves. In contrast to the explicit method, the implicit method is unconditionally stable, so increasing the timestep only reduces accuracy rather than introducing instability. However, it is worth noting that both of these schemes are first-order in time due to the use of a single-sided finite difference approximation of the derivative. Similarly to the spatial finite difference, higher order central difference schemes can be constructed such as the leapfrog method, Lax-Friedrichs method, or the Crank-Nicolson method. More information on time integration methods and deeper discussion of the discretization methods introduced in this section can be found in any standard numerical methods textbook, such as [58]. The remaining discussion in this section is focused exclusively on spatial discretization, as the techniques that comprise the bulk of this dissertation use a *semi-discrete* formulation as part of a *method of lines* approach. Semi-discrete schemes first discretize only in space to reduce a partial differential equation in space and time to an ordinary differential equation in time which can then be solved with standard time integration methods.

Boundary conditions are generally straightforward to implement in finite difference codes,

as Dirichlet (fixed value) and Neumann (fixed derivative) conditions result in simple mathematical forms for updating boundary grid points. Extending the finite difference method to higher dimensions involves simply laying down more grid points and solving a (substantially) larger linear system, greatly increasing computational expense. Additionally, introducing nonuniformity into the grid spacing must be performed carefully to avoid a notable decrease in the order of accuracy [58]. The finite difference method is then generally best suited for uniform geometries and smooth functions that are well approximated as local polynomials of order equal to the scheme's order of accuracy (e.g. the second order central scheme in Eq. 2.5 is exact for quadratic functions). While those restrictions are not an issue for many problems, few theoretical leaps are required to arrive at finite volume methods, which add flexibility in terms of geometry and information locality.

2.1.2 Finite Volume Method

In essence, for the finite difference method, we have laid out a grid of points within a computational domain and defined our solution at each of those points. The finite volume method emerges when we instead consider the computational grid not as a grid of points but as a grid of *cells* with the approximate solution defined as some constant value within each cell. The average value in cell V_k with bounds $[x_{k-1/2}, x_{k+1/2}]$ is notated with a bar, e.g. $\bar{u}_h^k(t)$, and defined as

$$\bar{u}_h^k(t) = \frac{1}{h} \int_{x_{k-1/2}}^{x_{k+1/2}} u(x,t)\, dx,$$

where $h = x_{k+1/2} - x_{k-1/2}$. Replacing the continuous solution with the cell average in Eq. 2.1 yields a new form of the equation that holds in each cell individually,

$$\frac{\partial \bar{u}_h^k(t)}{\partial t} + \frac{\partial F(\bar{u}_h^k)}{\partial x} = R^k(x),$$

where $R^k(x)$ is the residual for cell V_k, and $x \in [x_{k-1/2}, x_{k+1/2}]$. Whereas the remainder is 0 at each point in the finite difference method, here we choose to require that R^k vanishes on average in each cell. Taking the cell average of this equation yields

$$\frac{1}{h}\int_{x_{k-1/2}}^{x_{k+1/2}} \frac{\partial \bar{u}_h^k(t)}{\partial t}\,dx + \frac{1}{h}\int_{x_{k-1/2}}^{x_{k+1/2}} \frac{\partial F(\bar{u}_h^k)}{\partial x}\,dx = \frac{\partial \bar{u}_h^k(t)}{\partial t} + \frac{1}{h}\oint_{\delta V_k} F(\bar{u}_h^k)\,dx$$

$$= \frac{\partial \bar{u}_h^k(t)}{\partial t} + \frac{1}{h}(F_*^{k+1/2} - F_*^{k-1/2}) = 0, \tag{2.8}$$

where we have applied the divergence theorem to convert the flux integral into two *surface terms* with *numerical fluxes*, $F^{k\pm1/2}$, that are evaluated at the boundaries of cell V_k. Equation 2.8 states that the cell average $\bar{u}_h$ is conserved within cell V_k: changes in $\bar{u}_h^k$ are due solely to the fluxes into and out of cell V_k. It is worth noting that this conservative form arises because the advection equation is a statement of conservation; discretizing a non-conservative equation would lead to source terms. Because the finite volume method guarantees the discrete solution is conserved cell-wise, it is a natural choice when one seeks to study physical systems described by hyperbolic conservation laws like the Euler equations. A diagram of a typical cell and its neighbors in a finite volume scheme is presented in Figure 2.1.

Cell average quantities are, in general, not continuous across cell boundaries, so an important choice must be made on evaluation the fluxes for the surface terms while ensuring continuity of flow between cells. Some simple options are to use the average of the fluxes calculated from the solutions on both sides of the interface (known as a central flux),

$$F^{k+1/2} = \frac{1}{2}(F(\bar{u}_h^k) + F(\bar{u}_h^{k+1})),$$

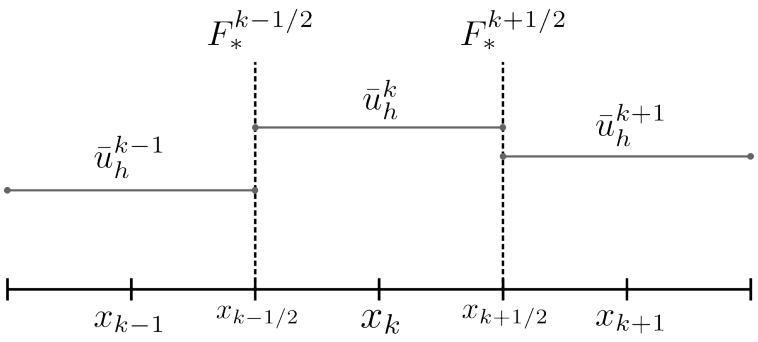

Figure 2.1: Diagram of a typical cell in a finite volume scheme. Note the cell average solutions, $\bar{u}_h^k$, are discontinuous and multiply defined at the cell interfaces, $x_{k\pm1/2}$, where the numerical fluxes, $F_*^{k\pm1/2}$, are evaluated.

a flux calculated using the average solution at the interface,

$$F^{k+1/2} = F\left(\frac{u_h^k + u_h^{k+1/2}}{2}\right),$$

or a one-sided flux based on the direction of information flow. However, as noted by Godunov [46], a Riemann problem arises at each cell interface,

$$\bar{u}_h = \begin{cases} \bar{u}_h^k, & x < x_{k+1/2} \\ \bar{u}_h^{k+1}, & x > x_{k+1/2}. \end{cases}$$

The Riemann problem can be solved exactly for a given differential equation in certain cases, and this exact solution is used to compute the numerical flux at each interface. Otherwise, an expensive nonlinear equation must be solved at each cell interface every timestep to compute the exact solution [57]. Alternatively, an approximate Riemann solver such as the Roe solver [99] or HLL solver [53] can be utilized to reduce computational expense with some reduced accuracy, which is generally acceptable as we are already working with cell average quantities

in the finite volume method.

While at first glance the introduction of numerical flux only serves to add complexity to the finite volume method relative to the finite difference method, the conservative form of the discrete equation offers an immediate advantage over the finite difference method: flexibility in geometry. As Eq. 2.8 requires only information from cell V_k and its nearest neighbors, as long as the numerical flux functions are defined such that they are symmetric across cell interfaces, there will be global conservation of u over the computational domain. Therefore, the finite volume method can handle any type of mesh, regardless of if said mesh is rectangular Cartesian, triangular, or completely unstructured, as long as the entire computational domain is tiled completely and each cell boundary touches two cells. Additionally, the finite volume formalism allows for boundary conditions to be included quite simply, either directly as the flux contribution to a cell on the domain boundary side or by way of ghost cells outside the domain that are used to compute fluxes on the domain boundary side of the last cell.

The semi-discrete finite volume scheme above, Eq. 2.8, is only first-order in space. In order to increase the accuracy of this method, we can choose to use the cell averages of neighboring cells to reconstruct a higher order representation. For example, for a linear representation,

$$p_k(x) = a_k + b_k x,$$

two pieces of information are necessary to determine the coefficients a_k and b_k, and we can constrain the reconstructed solution to solve for these coefficients by requiring the cell

average of $p_k(x)$ to match the solution cell average in each respective cell,

$$\frac{1}{h}\int_{x_{k-1/2}}^{x_{k+1/2}} p_k(x)\,dx = \bar{u}_h^k$$

$$\frac{1}{h}\int_{x_{k+1/2}}^{x_{k+3/2}} p_k(x)\,dx = \bar{u}_h^{k+1}.$$

Should we desire an even higher order of accuracy, a polynomial reconstruction of order N will require $N+1$ pieces of information, so the reconstruction stencil must be expanded to include even more cells. Such a process requires the grid to be structured in some way, removing that major strength of the finite volume method. To summarize, if we seek a higher order of accuracy in our finite volume scheme, more pieces of information are required to constrain a higher order polynomial representation, and because each cell in the finite volume method only contains a single average value, we must include more cells in the reconstruction operation. Rather than using a large stencil to achieve a higher order polynomial representation of the solution, we can sidestep the problem by introducing more degrees of freedom within each cell from the start, which leads us to the finite element method.

2.1.3 Finite Element Method

Finite element methods emerge naturally from this problem of seeking to increase the order of accuracy of a finite volume scheme to an arbitrary degree while maintaining geometric flexibility. To increase the available degrees of freedom for our representation, we introduce a set of N_p total nodes within a given element K_j with bounds $[x^j, x^{j+1}]$. Note that the bounds of the element are now at integer indices, rather than the half indices used in the finite volume method. We have also switched from working with cells to *elements*, but the concept is effectively the same.

From this point, we follow the approach outlined in [56]. We seek a representation of our solution as a linear combination of basis functions $\psi_n(x)$ with scalar coefficients,

$$u_h^j(x,t) = \sum_{i=1}^{N_p} a_n(t)\psi_n(x).$$

We are largely free to choose the basis functions as we see fit, but some choices are better than others, as will be seen later. As noted at the end of the finite volume method discussion, a linear polynomial representation requires two degrees of freedom, which will be captured here in two locations called *nodes*. If we define these two nodes in element K_j to be located at the boundaries of the element, x^j and x^{j+1}, we can define our basis functions as Lagrange interpolation polynomials relative to each node,

$$\psi_n(x) = l_n^k(x) = \frac{x - x_{k+1-n}}{x_{k+n} - x_{k+1-n}}.$$

The coefficients are then simply $u(x)$ evaluated at each corresponding node, resulting in the following polynomial representation,

$$u_h^j(x,t) = u^j(t)\frac{x - x_{j+1}}{x_j - x_{j+1}} + u^{j+1}(t)\frac{x - x_j}{x_{j+1} - x_j},$$

where $u^j \equiv u(x^j)$. Recovering the global representation of our solution involves simply summing over all N_T nodes,

$$u_h(x,t) = \sum_{n=1}^{N_T} u^n(t)N^n(x),$$

where $N^n(x)$ is the piecewise linear shape function,

$$N^n(x) = \begin{cases} \dfrac{x - x_{n-1}}{x_n - x_{n-1}}, & x_{n-1} \leq x < x_n \\ 1 - \dfrac{x - x_n}{x_{n+1} - x_n}, & x_n \leq x < x_{n+1} \\ 0 & \text{otherwise} \end{cases}$$

When evaluated at node n, the shape function is simply the Kronecker delta, $N^n(x^k) = \delta_{nk}$.

Similarly to the previous methods, this solution representation is not exact and brings a residual into the advection equation,

$$\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial t} + R(x) = 0.$$

In the finite element method, we require that the residual is orthogonal to all test functions belonging to some set $\mathcal{W}$. That is, the advection equation multiplied by any basis function $w(x) \in \mathcal{W}$ and integrated over the computational domain, Ω, must vanish

$$\int_\Omega \left(\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial t} \right) w(x)\, dx = 0.$$

For a more detailed motivation of this choice of requirement for the residual as a consequence of L2 error norm minimization, see [64]. With the discrete representation of the solution and condition for the error in hand, the next step is to define the set of basis functions. A common tactic known as the Galerkin method emerges if we choose weight functions that span the same space as the basis functions used in the representation of the solution. Applying this to the example above, the basis functions would be $N^n(x)$, yielding

$$\int_\Omega \left(\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial t} \right) N^n(x)\, dx = 0, \tag{2.9}$$

for each node $n \in [1...N_T]$. Recalling that u_h and F_h are expressed as expansions in the basis set, Eq. 2.9 can be rewritten as

$$\int_\Omega N^n(x)\frac{\partial}{\partial t}\left(\sum_i^{N_T} u^i(t)N^i(x)\right) + N^n(x)\frac{\partial}{\partial x}\left(\sum_i^{N_T} F^i(t)N^i(x)\right) dx = 0.$$

It is worth emphasizing that expanding the solution in the basis set effectively separates the spatial and temporal dependence of the solution. The spatial dependence of the discrete representation is contained in the basis functions, $\psi(x)$, while the coefficients of those basis functions (the *expansion coefficients*), $u^j(t)$, are what vary in time. Defining the mass matrix, $\mathcal{M}$, and stiffness matrix, $\mathcal{S}$, as

$$\begin{aligned} \mathcal{M}_{ij} &= \int_\Omega N^i(x)N^j(x)\,dx \\ \mathcal{S}_{ij} &= \int_\Omega N^i(x)\frac{dN^j(x)}{dx}\,dx, \end{aligned} \tag{2.10}$$

we can rewrite our scheme in the following form,

$$\mathcal{M}\frac{\partial \mathbf{u}^n}{\partial t} + \mathcal{S}\mathbf{F}^n = 0.$$

We have introduced the vectors $\mathbf{u}^n = [u^1(t)), ..., u^{N_T}(t)]$ and $\mathbf{F}^n = [F^1(t), ..., F^{N_T}(t)]$, containing the expansion coefficients of our solution and fluxes evaluated at each node. While these quantities are defined over the entire domain, no assumptions are made regarding geometry nor the order of the representations in the individual elements. As long as the elements maintain shared nodes at their boundaries, any number of nodes can be added to a single element to increase the degrees of freedom.

Galerkin finite element methods offer the ability to achieve arbitrarily high-order representations within any single element without adjusting other elements, and there are no global

assumptions on element size or shape. As discussed in greater depth in [56], some disadvantages of the Galerkin finite element scheme shown here are that we must invert the mass matrix to solve for the unknown expansion coefficients every timestep and are subsequently forced into an implicit time integration technique. Ideally, we would be able to utilize a numerical method that maintains the highlighted strengths of the finite element method while including advantages from other methods, specifically the conservative form and cell-wise locality of information present in the finite volume method. Thankfully, those that came before have already arrived at the solution: the discontinuous Galerkin method.

2.2 Introduction to the Discontinuous Galerkin Method

The discontinuous Galerkin (DG) method is generally attributed to Reed and Hill [94], where it was used to numerically solve the neutron transport equation. Following decades of advancements, detailed in [31], the DG method with Runge-Kutta time integration was applied to multidimensional, hyperbolic conservation laws by [29, 30]. Since that accomplishment, DG has continued to gain popularity in studying phenomena described by hyperbolic equations, from acoustics to shallow water equations to, of course, plasma physics. Substantially more historical context and areas of application, including elliptic problems like the Poisson equation, are given in [56] and [31].

DG methods have demanded such attention from these fields for several reasons. As a subset of finite element methods, the spatial order of accuracy of a DG scheme can be made arbitrarily high via choice of basis set. Similarly to finite volume methods, the formalism involves surface flux terms that keep information local to the cell of interest and its neighbors, and representations are discontinuous across cell boundaries, allowing for accurate capturing of discontinuities such as shocks. The flux functions required for transfer of information

between cells can be tailored for application to specific physics problems, e.g. if flow is known to primarily favor one direction.

In short, DG is a remarkably powerful tool that is actively applied to plasma physics problems like those we seek to study in this work, such as in [50], [116], and [79].

All DG schemes described in this thesis are implemented in the plasma simulation framework `Gkeyll`, (`gkyl.readthedocs.io`). `Gkeyll` is an open-source framework that includes solvers for the Vlasov-Maxwell [50, 63], full-f gyrokinetics [81, 108], and multi-fluid [137] systems.

2.2.1 Weak Equality and Choice of Basis Set

Our discussion of DG begins similarly to the finite element method, where we divide our computational domain into elements, K_j (cells and elements are used interchangeably here). We make no assumptions about the geometry of these elements beyond that they completely tile the domain without overlap. Within these elements, we again choose a set of basis functions, ψ_i, this time explicitly choosing basis functions that are piecewise polynomials of order p, i.e. are elements of the vector space $\mathcal{S}^p \equiv \{\psi : \psi \in \mathcal{P}^p\}$ for piecewise polynomials $\mathcal{P}^p$ defined over a single cell. From here, we diverge from the traditional finite element derivation by defining the interval $I = [-1, 1]$ as the bounds of each element in *logical* or *computational space*. Where previously integrals were performed over the entire computational grid, instead they are now taken over I, as the solutions are now discontinuous and localized to individual elements. For example, the inner product of two functions over an element is

$$\langle f, w \rangle = \int_{-1}^{1} f(x) w(x)\, dx.$$

We again choose to represent the approximate solution, f_h, as an expansion in the basis set with N_p basis functions,

$$f_h^j(x,t) = \sum_{n=0}^{N_p-1} \widehat{f}_n^j(t)\psi_n(x), \tag{2.11}$$

where the *expansion coefficients*, $\widehat{f}_n^j$, are functions only of time, and the basis functions, $\psi_i(x)$, are only defined over the bounds of the element,

$$\psi_n(\xi) = \begin{cases} \psi_n(\xi), & \xi \in [-1,1] \\ 0 & \text{otherwise,} \end{cases}$$

for the transformed computational space coordinate, ξ. In writing the representation this way, we have implicitly chosen the *modal* expression of f_h^j. This is contrasted with the *nodal* expression we saw before in the finite element method discussion, which looks like

$$f_h^j(x,t) = \sum_{n=0}^{N_p-1} f_n(x_n^j,t)l_n^j(x)$$

in the DG method. While these representations may seem quite distinct, they are simply two methods of storing and expressing the same information. As seen before, a discretized polynomial representation of order N_p-1 is uniquely defined by the values of the N_p polynomial coefficients. In DG, those pieces of information can be stored either as weights for the N_p basis functions in the modal expansion or as values at each of the N_p nodes defined within the element for the nodal description. Both expressions are equally valid for DG purposes, and transformation between them is simple with the Vandermonde matrix,

$$\mathcal{V}_{mn}^j \widehat{f}_n^j(t) = f_n(x_n^j,t)$$

with

$$\mathcal{V}_{mn}^j = \psi_n(x_m^j).$$

However, the DG schemes that emerge from these two expressions requires different treatment and algorithms [18, 64]. Either way, the complete discrete representation is a direct sum of the local polynomial solutions,

$$f_h(x,t) = \bigoplus_{j=0}^{N_p-1} f_h^j(x,t).$$

As `Gkeyll` utilizes the modal description, that will be the focus of the remainder of this work, but nodal methods are explored in great depth in [56].

An example case of discretizing a function with a modal DG expression with polynomial order $p = 1$ and $p = 2$ basis sets is shown in Figure 2.2. The function $f(x) = \exp(-(x+0.1)^2)$ is discretized over three cells with intervals $[-3,-1]$, $[-1,1]$, and $[1,3]$. Within each cell, each representation is weakly equal to the original function over its respective basis set,

$$\int_{I_j} (f(x) - f_h(x))\psi_i(x)\,dx = 0.$$

Attempting to resolve the important parts of a function (e.g. the peak of the bell curve here) with so few cells with wide spacing relative to the function is generally inadvisable. However, in this case, while the $p = 1$ representation is a poor approximation of the original function, the $p = 2$ representation matches relatively well even with such poor resolution. This is an example of the utility of being able to increase the polynomial order of the discrete representations arbitrarily; with suitably high order polynomials, discretizations on coarse resolutions can be accurate.

The next step after choosing the discrete representation is determining the semi-discrete form of the equation of interest. For simplicity and consistency with the previous section, we will again utilize the advection equation, Eq. 2.1. Recall the form of the advection equation from

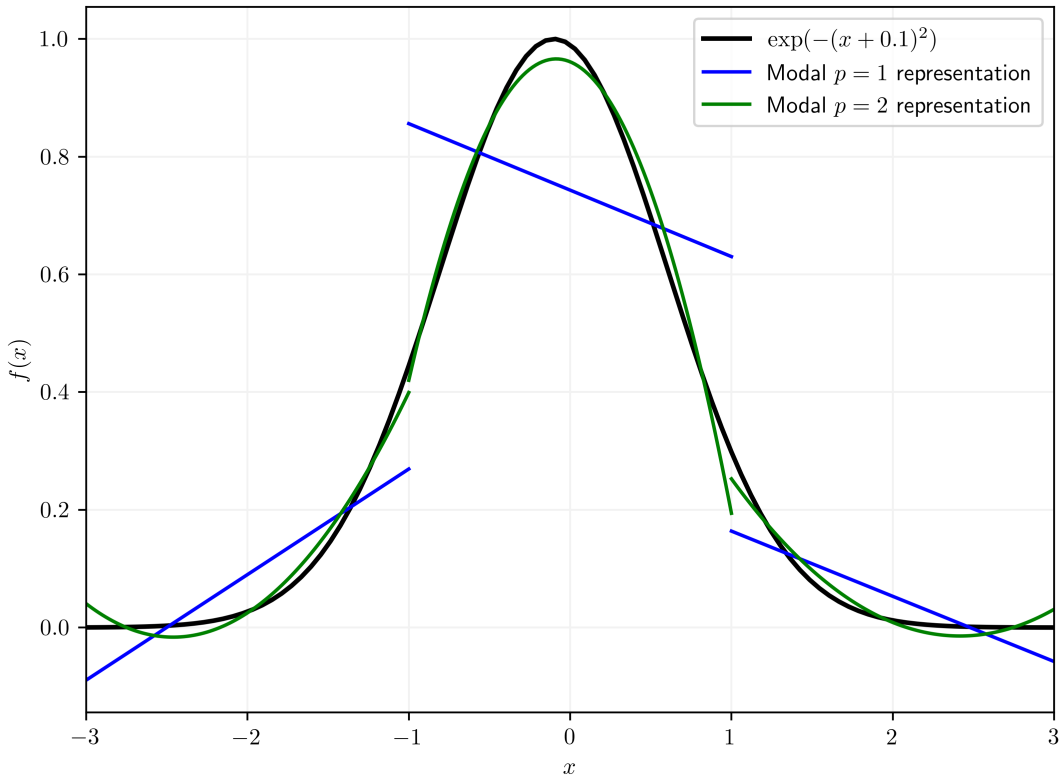

Figure 2.2: Example of a DG discretization using the modal description. The function $\exp\left(-(x+0.1)^2\right)$ is discretized using a polynomial order $p=1$ basis (blue) and $p=2$ basis (green) over three cells with intervals $[-3,-1]$, $[-1,1]$, and $[1,3]$. Both representations are weakly equal to the original function within their respective cells.

the finite element method,

$$\int_\Omega \left(\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial t} \right) w(x)\, dx = 0. \tag{2.12}$$

where an inner product is taken between the differential equation and a test function, and we required the residual to be orthogonal to all test functions in the computational domain. DG methods apply a very similar idea through the concept of *weak equality*. The functions

f and g are weakly equal if the following is true,

$$\int_I (f-g)\psi_i \, dx = 0, \quad \forall i \in [0, \cdots, N_p - 1],$$

or, equivalently,

$$f \overset{\circ}{=} g,$$

that is, the difference between the two functions is orthogonal to some test function. In DG, point-wise equality is not enforced. We only require weak equality, where the projections of quantities onto a set of test functions are equal. This means that the behavior of individual functions in each cell is largely unimportant; as long as their projections onto the basis set are identical, they are treated as equal. The application of weak equality can be viewed as either reducing the interval over which the integral is taken in the finite element method, Eq. 2.12, from the entire computational domain down to a single element or as substitution of the approximate forms of the solutions and replacing the strict equality in Eq. 2.1 with weak equality,

$$\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial x} \overset{\circ}{=} 0.$$

Weak equality is of fundamental importance to the DG method, so we will take a moment to summarize before moving on. Solution quantities are discretized via projection onto our chosen basis set of piecewise polynomials of order p, $\psi_i \in \mathcal{P}^p$, yielding approximate solutions of the form in Eq. 2.11 that are defined within each cell of the computational grid. Discretizing a function onto a computational grid in this manner results in some degree of error, which is accumulated into the residual term, $R(x)$, as in the approximate form of the advection equation,

$$\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial x} + R(x) \overset{\circ}{=} 0.$$

Similarly to the Galerkin finite element method, we require the residual to be orthogonal to

all basis functions. Because the basis functions are defined within individual elements, the inner product to enforce orthogonality is performed over the bounds of an element, j, given by the interval I_j. Writing out the weak equality statement as an inner product yields

$$\int_{I_j} \left(\frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial x} + R(x) \right) \psi_i(x)\, dx = 0, \quad \forall i \in [0, \cdots, N_p - 1].$$

Dropping the residual term due to the orthogonality constraint and rearranging the equation gives

$$\int_{I_j} \frac{\partial u_h}{\partial t} \psi_i(x)\, dx = -\int_{I_j} \frac{\partial F_h}{\partial x} \psi_i(x)\, dx.$$

We then integrate the flux integral by parts to turn the single flux integral into a volume contribution and a surface contribution as in the finite volume method,

$$\begin{aligned} \int_{I_j} \frac{\partial u_h(x,t)}{\partial t} \psi_i(x) dx &= -F_h(x_j, t)\psi_i^-(x_j) + F_h(x_{j+1}, t)\psi_i^-(x_{j+1}) + \int_{I_j} F_h(x,t) \frac{\partial \psi_i(x)}{\partial x}\, dx, \\ &= -[F_h(x,t)\psi_i^-(x)] + \int_{I_j} F_h(x,t) \frac{\partial \psi_i(x)}{\partial x}\, dx \end{aligned} \tag{2.13}$$

where the brackets around the surface term indicate that the quantity is evaluated at the upper and lower cell boundaries and ψ_i^- indicates the basis function is evaluated just inside the cell boundary (ψ_i^+ would be evaluated just outside the boundary). Equation 2.13 is known as the *weak form* of the advection equation, because we are only requiring solutions to satisfy the original differential equation in the weak sense. While it may seem excessive to do so in one dimension, for generality we can rewrite the surface term as a surface integral over the boundary of the element, ∂I_j,

$$\int_{I_j} \frac{\partial u_h(x,t)}{\partial t} \psi_i(x) dx = -\oint_{\partial I_j} \psi_i^-(x)\, \widehat{\mathbf{n}} \cdot \widehat{\mathbf{G}}\, dS + \int_{I_j} F_h(x,t) \frac{\partial \psi_i(x)}{\partial x}\, dx.$$

We have replaced the product $F_h \psi_i$ in the surface term with generic numerical flux $\widehat{\mathbf{G}}$ be-

cause values of discontinuous quantities are multiply defined at cell interfaces. As in the finite volume method, choice of numerical flux function is of utmost importance and varies depending on the equation system. If we take the advective flux to be a simple linear flux, $F(u) = au$, the weak form becomes

$$\int_{I_j} \frac{\partial u_h(x,t)}{\partial t}\psi_i(x)dx = -\oint_{\partial I_j} \psi_i^-(x)\,\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}}\,dS + \int_{I_j} au_h(x,t)\frac{\partial \psi_i(x)}{\partial x}\,dx \tag{2.14}$$

where $\widehat{\mathbf{G}}$ should have some functional dependence on the product au_h evaluated from the cells on either side of the boundary. We want to ensure the scheme guarantees conservation, so the fluxes should be of Godunov-type: balanced on both sides of the interface

$$\oint_{\partial I_j} \psi_i^-\,\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}}dS = -\oint_{\partial I_j} \psi_i^+\,\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}}dS.$$

Examples of these matching fluxes include central flux,

$$\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}} = \frac{1}{2}\widehat{\mathbf{n}}\cdot a(u_h^+ + u_h^-),$$

or, more appropriately for an advection problem, an upwind flux,

$$\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}} = \begin{cases} au_h^-, & if \mathrm{a} > 0 \\ au_h^+, & if \mathrm{a} < 0 \end{cases}$$

At this point, it is time to confront the choice of basis functions used in our DG discretization. Through our derivation of the weak form of the advection equation, we have asserted the following about the basis set:

1. Basis functions are polynomials of order p, i.e. $\psi_i \in \mathcal{P}^p$.

2. Basis functions are defined piecewise on a reference element rather than over the whole computational domain.

With those two conditions in mind, there is still some freedom in the actual form of the basis functions and the question arises of what constitutes a "good" basis set. An extensive discussion on the choice of basis functions in `Gkeyll` is provided in [18] and [64], but we will summarize some of the key points here. One way of quantifying the effectiveness of a basis set is through the condition number of the mass matrix, Eq. 2.10, which characterizes the relative change in the solution of the linear system due to small changes in the inputs. The logarithm of the condition number is an estimate of the worst case number of digits of precision lost every time the linear system is solved. For the simple case of a pure monomial basis functions, $\psi_p(x) = x^p$, the logarithms of the condition numbers of order $p = 2$, 3, and 4 are approximately 1.3, 2.0, and 2.7, respectively, and continue to increase as polynomial order increases. As this scheme will be applied to real computational simulations, one must consider that numerical data types have some degree of precision already (7 digits for single point floats, 15-16 for doubles [1]). Increasing the order of accuracy further would approach numerical precision where this poor conditioning would affect simulations.

If we instead construct an orthonormal basis via the Gram-Schmidt procedure, the condition number of the mass matrix is instead guaranteed to be 1, regardless of polynomial order. Additionally, the orthonormal basis constructed in [18] is shown to be more linearly independent than the monomial basis, so the orthonormal basis is better equipped to approximate a wide variety of functional forms. The basis set can be extended to higher dimensions by taking the tensor product of one dimensional basis sets, greatly increasing the number of basis functions as the number of dimensions, d, increases. Commonly referred to as the "curse of dimensionality," the number of basis functions increases grows exponentially according to $(p+1)^d$. As we aim to perform simulations in up to six-dimensional phase space,

`Gkeyll` implements a reduced basis set known as the Serendipity polynomial space [4]. The Serendipity basis set is constructed utilizing the concept of super-linear order, which is the sum of all powers greater than 1 in a product. Beginning with a basis set of a tensor product of monomials, a Serendipity set of order p is constructed by removing terms of super-linear order greater than p. For example, if super-linear order $p = 2$ for a space with variables (x, v_x, v_y, v_z), the term $x^2 v_x^2 v_y v_z$, with super-linear order $2 + 2 = 4$, would be removed when constructing the Serendipity set.

We now have the semi-discrete form of the advection equation, Eq. 2.14, with approximate solutions projected onto the Serendipity basis set, and a numerical flux that makes sense with the physical interpretation of the equation. The remaining detail is to determine how to advance these discretized solutions in time. Considering only explicit time discretization techniques, recall the forward Euler method introduced early in this section, Eq. 2.6, which can be written more generally as

$$\mathcal{F}(f,t) = f + \Delta t \mathcal{L}(f,t),$$

where $\mathcal{L}(f,t)$ is the spatial discretization portion of the differential equation. A desirable property of potential combinations of spatial and temporal discretizations is that they are *total variation diminishing* (TVD) [47], which indicates a scheme is able to accurately capture shocks and discontinuities without introducing unphysical oscillations. The conditions for this property can be met when utilizing the forward Euler method as the time integrator with a given spatial discretization, assuming a timestep sufficiently small to meet the Courant-Friedrichs-Levy (CFL) constraint. However, a major strength of DG is the ability to easily increase the order of accuracy of the spatial discretization to an arbitrarily high degree. Applying a first-order time discretization like the forward Euler method limits the overall order of the scheme, and, as briefly mentioned before, the forward Euler method is

very restrictive on the timestep to achieve a stable scheme (conditional stability). Strong-stability preserving Runge-Kutta (SSP-RK) methods were devised [110, 111] in the context of hyperbolic conservation laws to solve ordinary differential equations of the form

$$\frac{\partial f}{\partial t} = \mathcal{L}(f, t),$$

while maintaining the TVD property and allowing for higher-order temporal discretizations and potentially less restrictive timesteps [35, 47]. SSP-RK timesteppers utilize forward Euler operations through multiple stages to arrive at the updated solution. Several SSP-RK methods are implemented in `Gkeyll`, such as the following four stage, third order method,

$$f^{(1)} = \frac{1}{2} f^n + \frac{1}{2}\mathcal{F}[f^n, t^n], \tag{2.15}$$

$$f^{(2)} = \frac{1}{2} f^{(1)} + \frac{1}{2}\mathcal{F}[f^{(1)}, t^n + \Delta t/2], \tag{2.16}$$

$$f^{(3)} = \frac{2}{3} f^n + \frac{1}{6} f^{(2)} + \frac{1}{6}\mathcal{F}[f^{(2)}, t^n + \Delta t], \tag{2.17}$$

$$f^{n+1} = \frac{1}{2} f^{(3)} + \frac{1}{2}\mathcal{F}[f^{(3)}, t^n + \Delta t/2], \tag{2.18}$$

which allows for twice as large of a timestep as the forward Euler method.

Returning to the example semi-discrete form of the advection equation, Eq. 2.14 can be rewritten explicitly in the form required for the SSP-RK timesteppers by substituting in the approximate solutions with their modal expressions,

$$\int_{I_j} \frac{\partial(\widehat{u}_n^j(t)\psi_n(x))}{\partial t}\psi_i(x)dx = \\ -\oint_{\partial I_j} \widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}}\psi_i^-(x)dS + \int_{I_j} a\widehat{u}_m^j(t)\psi_m(x)\frac{\partial\psi_i(x)}{\partial x}\,dx,$$

where we utilize Einstein's summation convention of treating repeated indices in a product

as a sum over those indices,

$$u_n \psi_n = \sum_{n=0}^{N-1} f_n \psi_n.$$

Pulling the expansion coefficients out of the spatial integrals allows the semi-discrete equation to be reformulated as

$$\frac{\partial \widehat{u}_n^j}{\partial t} = \left(\mathcal{M}_{ni}^j\right)^{-1} \left[- \oint_{\partial I_j} \widehat{\mathbf{n}} \cdot \widehat{\mathbf{G}} \psi_i^-(x) dS + \widehat{u}_m^j(t) \int_{I_j} a \psi_m(x) \frac{\partial \psi_i(x)}{\partial x} \, dx \right]. \tag{2.19}$$

In this form, it appears as though we will be required to invert the mass matrix every timestep (or multiple times per step if using a multi-stage Runge-Kutta method), but our choice of orthonormal basis set means the mass matrix is simply the identity matrix. With the timestepping method chosen, we have completed our DG treatment of the advection equation in one dimension.

2.2.2 Application of Weak Equality: Recovery

When applying the DG method to cases beyond simple linear advection, situations arise where the discontinuous polynomial representations become an issue, such as in diffusion equations with second order spatial derivatives. While the numerical flux for an advection term can handle these discontinuities rather easily, diffusion terms introduce additional complexity due to the spatial derivatives that appear in the resulting surface terms. Applying weak equality to a second derivative $g(x,t) = \partial^2 f(x,t)/\partial x^2$, and integrating the volume

integral by parts twice,

$$\begin{aligned}\int_{I_j} g_h^j \psi_i \, dx &= \int_{I_j} \frac{\partial^2 f_h^j}{\partial x} \psi_i \, dx, \\ &= \left[\frac{\partial f_h^j}{\partial x} \psi_i \right] - \int_{I_j} \frac{\partial f_h^j}{\partial x} \frac{d\psi_i}{dx} \, dx, \\ &= \left[\frac{\partial f_h^j}{\partial x} \psi_i - f_h^j \frac{d\psi_i}{dx} \right] - \int_{I_j} f_h^j \frac{d^2 \psi_i}{dx^2} \, dx,\end{aligned}$$

we are left with aforementioned surface terms that require both the value and derivative of f at each cell boundary to properly handle the discontinuity between cells [44]. One such solution to this problem is known as the recovery process (and the resulting schemes are generally known as recovery discontinuous Galerkin schemes) [61, 134], which has proven well-equipped to handle diffusive problems [51, 135]. In short, the recovery process utilizes weak equality along with information from two neighboring cells to construct a polynomial of order $2p+1$ that is continuous across the cell boundary and allows us to take a derivative across that boundary without losing an order of accuracy. Recovery is similar in spirit to the reconstruction process outlined in Section 2.1.2, which has also been applied to DG methods [118]. However, the goal of recovery is not just to achieve a higher order representation but to leverage the new, continuous solution for evaluation of fluxes.

Mathematically, consider a one-dimensional grid with two cells, I_L and I_R, and approximate solutions of order p defined in each cell, $q_{h,L}(x)$ and $q_{h,R}(x)$. We seek a polynomial, $\widetilde{q}(x)$, that is higher order than the representations in the left and right cells and is continuous across the cell boundary. The trick is to require that $\widetilde{q}$ is weakly equal to both approximations in

their respective cells, so we can form the linear system,

$$\int_{I_L} (\widetilde{q}(x) - q_{h,L}(x))\psi_{Li}(x)\,dx = 0$$
$$\int_{I_R} (\widetilde{q}(x) - q_{h,R}(x))\psi_{Ri}(x)\,dx = 0.$$

For example, if the representations $q_{h,L}(x)$ and $q_{h_R}(x)$ are expanded in the basis set of $p = 1$ polynomials, the basis functions are

$$\psi_0(x) = \frac{1}{\sqrt{2}}, \qquad \psi_1(x) = \frac{\sqrt{3}x}{\sqrt{2}},$$

which can be shifted into the left and right cells. The linear system is then comprised of 4 equations:

$$\int_{I_L} (\widetilde{q}(x) - q_{h,L}(x))\psi_{L0}(x)\,dx = 0,$$
$$\int_{I_L} (\widetilde{q}(x) - q_{h,L}(x))\psi_{L1}(x)\,dx = 0,$$
$$\int_{I_R} (\widetilde{q}(x) - q_{h,R}(x))\psi_{R0}(x)\,dx = 0,$$
$$\int_{I_R} (\widetilde{q}(x) - q_{h,R}(x))\psi_{R1}(x)\,dx = 0,$$

and we therefore have 4 pieces of information available for our recovered polynomial. With these four equations and four unknowns, $\widetilde{q}$ can be a third order polynomial,

$$\widetilde{q}(x) = \widetilde{q}_0 + \widetilde{q}_1 x + \widetilde{q}_2 x^2 + \widetilde{q}_3 x^3.$$

In general, there will be $2p+2$ degrees of freedom in this recovery procedure, so the recovery polynomial will be of at most order $2p + 1$. Because the order of the recovered polynomial

is much greater than the order of the basis set, taking derivatives of $\widetilde{q}$ for use in updates will not lower the order of our scheme when the resulting derivatives are projected back onto the original basis set. An example of recovery is presented in Figure 2.3, where the operation is performed between the left and center cells. Note that while the $p = 1$ linear polynomial representations do not capture the original function very well, the third order recovered polynomial is a much more accurate fit. The recovered value at the cell interface, $x = -1$, can be used to calculate numerical fluxes as necessary.

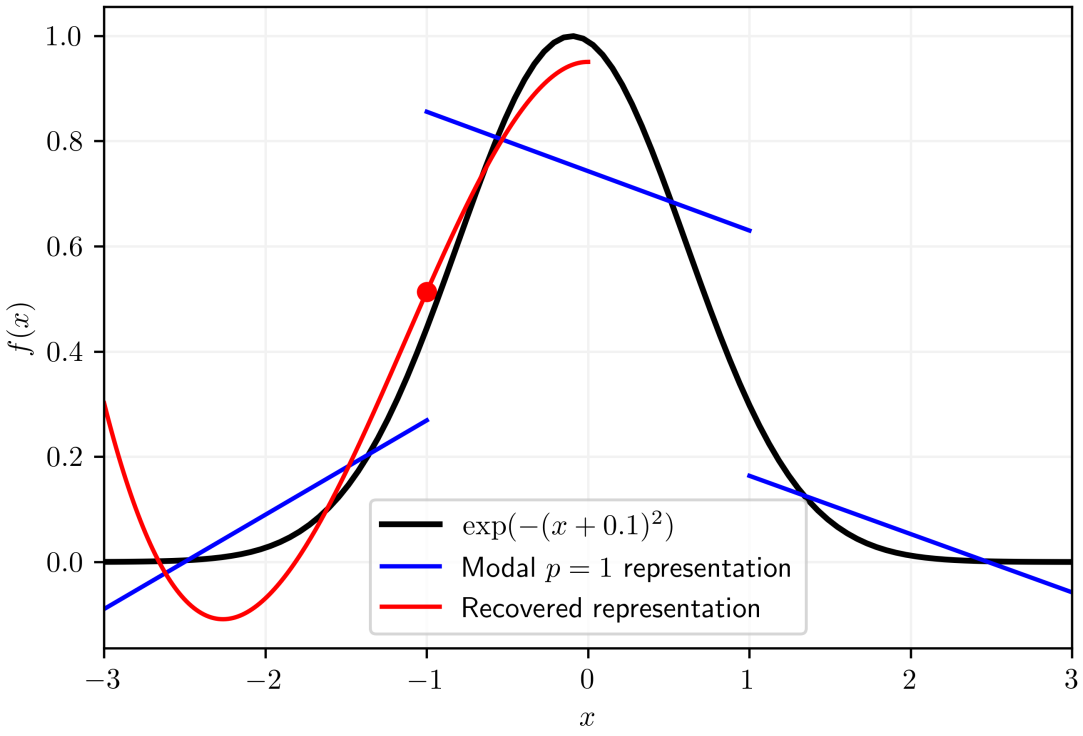

Figure 2.3: Example application of recovery, where a third order polynomial is constructed using information from the left and right cells that are defined over $[-3, -1]$ and $[-1.1]$. The higher order representation is a better approximation of the original function than the $p = 1$ linear representations, and the value of the recovered polynomial at the cell interface can be used to compute numerical fluxes easily.

2.2.3 Leveraging the Discontinuous Galerkin Method: Pre-computation for an Alias-Free Scheme

Before we transition to the application of the DG method to the Vlasov-Maxwell/Boltzmann system of equations, we must cover one of the unique aspects of `Gkeyll` that allows us to run accurate, high-dimensional kinetic simulations that have previously been computationally unfeasible. This advancement, discussed in detail in [50, 64], is the application of computer algebra software to calculate integrals symbolically to machine precision prior to runtime, generating computational kernels that serve to eliminate aliasing errors and avoid matrix operations and quadrature. Aliasing errors are commonly known to generate runaway numerical instabilities for kinetic equations, and these errors can emerge as uncontrolled and unpredictable sources of energy. While nodal DG methods have utilized over-integration (or polynomial de-aliasing) and split form expressions that average between conservative and non-conservative forms of the governing equation, these techniques only control and limit the growth of energy, rather than eliminating the problem at its source [139]. Additionally, if one is integrating via quadrature and chooses to add more quadrature nodes to achieve a higher-order accurate solution, the computational expense quickly balloons as a tensor product as dimensionality increases, and again, the kinetic simulations we aim to conduct can be up to six-dimensional. The modal scheme used in `Gkeyll` uses an orthonormal basis set, sparsifying matrices that form from products of basis functions. Because these matrices are sparse and the modal expression allows us to pull the expansion coefficients out of spatial integrals, these integrals can be symbolically precomputed. For example, consider the volume term of Eq. 2.19,

$$a\widehat{u}_m^j(t)\int_{I_j}\psi_m(x)\frac{\partial\psi_i(x)}{\partial x}\,dx, \tag{2.20}$$

where we have assumed the advection velocity is a constant scalar and noted that the mass matrix becomes the identity with an orthonormal basis. The integrand is now simply a product of basis functions (if a was an expansion as well, there would be another factor of $\psi(x)$), which can be calculated in software such as Maxima CAS, [83] the `Gkeyll` team's computer algebra software of choice. Performing the integral and summing over indices m yields the $i \in [0, \cdots, N_p]$ components of the volume contribution to the time derivative in 2.19. The resulting components are expanded out and simplified within Maxima CAS to generate computational kernels with of the minimum necessary operations, consisting of only addition and multiplication operations. By specifying the significant figures used in this analytic computation, the integrals, and therefore the kernels, are exact to machine precision. Examples of these kernels can be seen in the GitHub page for the C backend of `Gkeyll`, https://github.com/ammarhakim/gkylzero.

2.3 Discontinuous Galerkin Discretization of the Vlasov-Maxwell/Fokker-Planck System

With this background in DG methods and the subtle yet vital differences between them and the more commonly applied methods, we can now apply the DG method to discretize the Vlasov-Maxwell/Fokker-Planck system. This demonstration covers the basic important concepts that must be considered when applying the modal DG method to our equation system, but we will avoid the majority of the deeper details and subtleties that must be considered when developing a robust, accurate numerical scheme. An interested reader is encouraged to refer to previous work of the `Gkeyll`team for a complete presentation of the motivations and details of a modal DG implementation of the Vlasov-Maxwell/Fokker-Planck system, e.g. [18, 52, 63, 64]. The full system, with species indices dropped for notational

clarity, consists of the Vlasov-Fokker-Planck equation,

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{v} f) + \nabla_{\mathbf{v}} \cdot \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) f = \left(\frac{\partial f}{\partial t} \right)_C , \tag{2.21}$$

where the collision operator is left generic but can be, for example, any of the three discussed in Chapter 1, and Maxwell's equations,

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla_{\mathbf{x}} \times \mathbf{E} = 0 \tag{2.22}$$

$$\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} - \frac{1}{\mu_0} \nabla_{\mathbf{x}} \times \mathbf{B} = -\mathbf{J} \tag{2.23}$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{B} = 0 \tag{2.24}$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}. \tag{2.25}$$

As a reminder, the distribution function is a function of all phase space, $f(\mathbf{x}, \mathbf{v}, t)$, and the remaining quantities are functions of configuration space only, $\mathbf{E}(\mathbf{x}, t)$, $\mathbf{B}(\mathbf{x}, t)$, $\mathbf{J}(\mathbf{x}, t)$, and $\rho(\mathbf{x}, t)$. For the purposes of this exercise, we will choose the Doughtery-Lenard-Bernstein operator (LBO) [52] as our collision operator,

$$\left(\frac{\partial f}{\partial t} \right)_C = \nu \nabla_{\mathbf{v}} \cdot \left[(\mathbf{v} - \mathbf{u}) f + \frac{T}{m} \nabla_{\mathbf{v}} f \right]. \tag{2.26}$$

To construct the DG discretization of the Vlasov-Maxwell-Fokker-Planck system, we can handle each equation independently. Beginning with the Vlasov portion of Eq. 2.21, and setting aside the collision operator for the moment. We first rewrite the equation in the following form for simplicity,

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{z}} \cdot (\boldsymbol{\alpha} f) = 0, \tag{2.27}$$

where we define the phase space variable $\mathbf{z} = (\mathbf{x}, \mathbf{v})$, phase space gradient, $\nabla_{\mathbf{z}} = (\nabla_{\mathbf{x}}, \nabla_{\mathbf{v}})$ and the phase space flux $\boldsymbol{\alpha} = (\mathbf{v}, \mathbf{F}/m)$, with the Lorentz force, $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$. We can

now apply the same method we used for the advection equation in the previous section, by replacing the quantities of interest with their modal DG approximations and enforcing weak equality,

$$\frac{\partial f_h}{\partial t} + \nabla_{\mathbf{z}} \cdot (\boldsymbol{\alpha}_h f_h) \stackrel{\circ}{=} 0,$$

Again, rewriting the weak equality condition in terms of an inner product with a basis function over a cell K_j,

$$\int_{K_j} \left[\frac{\partial f_h^j}{\partial t} + \nabla_{\mathbf{z}} \cdot (\boldsymbol{\alpha}_h^j f_h^j) \right] \psi_i(\mathbf{z}) d\mathbf{z} = 0,$$

we can then integrate the flux term by parts and isolate the time derivative to arrive at the semi-discrete form of the Vlasov equation,

$$\int_{K_j} \frac{\partial f_h^j}{\partial t} \psi_i(\mathbf{z}) d\mathbf{z} = - \oint_{\partial K_j} \psi_i(\mathbf{z}) \widehat{\mathbf{n}} \cdot \widehat{\mathbf{G}} dS + \int_{K_j} \boldsymbol{\alpha}_h^j f_h^j \cdot \nabla_{\mathbf{z}} \psi_i(\mathbf{z}) d\mathbf{z}, \tag{2.28}$$

where the numerical flux is a function of the flux on both sides of the cell interfaces, $\widehat{\mathbf{G}} = \widehat{\mathbf{G}}(\boldsymbol{\alpha}_h^{j-} f_h^{j-}, \boldsymbol{\alpha}_h^{j+} f_h^{j+})$. As before, the fluxes should be continuous across cell boundaries for conservation. The choice of flux function for this application in `Gkeyll` is a Lax-Friedrichs flux,

$$\widehat{\mathbf{n}} \cdot \widehat{\mathbf{G}} = \frac{1}{2} \widehat{\mathbf{n}} \cdot (\boldsymbol{\alpha}_h^+ f_h^+ \boldsymbol{\alpha}_h^- f_h^-) - \frac{\max |\widehat{\mathbf{n}} \cdot \boldsymbol{\alpha}_h|}{2} (f_h^+ - f_h^-), \tag{2.29}$$

which includes a penalty term that limits the size of jumps by introducing numerical diffusion [64].

Discretization of Maxwell's equations requires a concept we have not yet encountered: contraction of the grid and set of basis functions [18]. Our set of basis functions $\psi_i(\mathbf{z})$ are constructed for the set of independent phase space variables $\mathbf{z} = (\mathbf{x}, \mathbf{v})$. This is correct for quantities that are functions of all phase space variables, but the electric and magnetic fields

are functions of only configuration space, $\mathbf{x}$. Therefore, we adjust our method to use configuration space cells Ω_j and basis functions that are instead functions of the configuration space variable $\mathbf{x}$ only, $\phi(\mathbf{x}) \in \mathcal{P}^p$. The weak forms of Ampère's and Faraday's laws are then

$$\int_{\Omega_j} \left[\frac{\partial \mathbf{B}_h^j}{\partial t} + \nabla_{\mathbf{x}} \times \mathbf{E}_h^j \right] \phi_i(\mathbf{x}) d\mathbf{x} = 0,$$
$$\int_{\Omega_j} \left[\epsilon_0 \frac{\partial \mathbf{E}_h^j}{\partial t} - \frac{1}{\mu 0} \nabla_{\mathbf{x}} \times \mathbf{B}_h^j \right] \phi_i(\mathbf{x}) d\mathbf{x} = -\int_{\Omega_j} \mathbf{J}_h^j \phi_i(\mathbf{x}) d\mathbf{x},$$

and we note that we are not enforcing the divergence constraints of Gauss' laws for electricity and magnetism. Therefore, the divergences of $\mathbf{E}_h$ and $\mathbf{B}_h$ may become unphysical during simulations [64]. By applying identities of the curl operator and the divergence theorem, we arrive at the semi-discrete form of Maxwell's equations,

$$\int_{\Omega_j} \frac{\partial \mathbf{B}_h^j}{\partial t} d\mathbf{x} + \oint_{\partial\Omega} d\mathbf{s} \times (\phi^- \widehat{\mathbf{F}}_E^*) - \int_{\Omega_j} \nabla_{\mathbf{x}} \phi \times \mathbf{E}_h d\mathbf{x} = 0 \tag{2.30}$$
$$\epsilon_0 \int_{\Omega_j} \frac{\partial \mathbf{E}_h^j}{\partial t} d\mathbf{x} - \frac{1}{\mu_0} \oint_{\partial\Omega} d\mathbf{s} \times (\phi^- \widehat{\mathbf{F}}_B^*) - \frac{1}{\mu_0} \int_{\Omega_j} \nabla_{\mathbf{x}} \phi \times \mathbf{B}_h d\mathbf{x} = -\mu_0 \int_{\Omega_j} \phi \mathbf{J}_h d\mathbf{x}. \tag{2.31}$$

Just like the other cases, we must carefully choose the forms of the flux functions $\widehat{\mathbf{F}}_E^*$ and $\widehat{\mathbf{F}}_B^*$. One may be tempted to select central fluxes in order to enforce conservation of electromagnetic energy, but this is not generally the best choice in DG. In traditional finite element discretizations of Maxwell's equations, spurious modes appear in the fields that are difficult, if not impossible, to separate from the physical modes [113]. These unphysical solutions also arise in discontinuous Galerkin discretizations in three dimensions due to poor representation of the curl operator [55]. Utilization of central fluxes in a three-dimensional discretization of Maxwell's equations preserves these spurious modes, resulting in inaccurate solutions. A simple and more robust choice of numerical flux is an upwind flux, which introduces dissipation on the unphysical modes and separates them from the physical modes [64].

Another option proposed by Hesthaven and Warburton is a stabilized central flux [55], which yields a similarly physically accurate frequency spectrum with minimal added computational expense through addition of damping of the spurious modes [22].

We have now discretized the Vlasov-Maxwell system in DG, so all that remains to complete the Vlasov-Maxwell/Fokker-Planck system is the discrete form of the collision operator. Again, we choose the LBO as our collision operator for this exercise,

$$\left(\frac{\partial f_h^j}{\partial t}\right)_C \stackrel{\circ}{=} \nu \nabla_{\mathbf{v}} \cdot \left[(\mathbf{v} - \mathbf{u}_h^j) f_h^j + \frac{T_h^j}{m} \nabla_{\mathbf{v}} f_h^j\right].$$

Following exactly the same process as for the Vlasov equation, we integrate by parts once to obtain

$$\frac{1}{\nu} \int_{K_j} \frac{\partial f_h^j}{\partial t} \psi_i(\mathbf{z}) = \oint_{\partial K_j} \psi_i(\mathbf{z}) \widehat{\mathbf{n}} \cdot \widehat{\mathbf{G}} dS - \int_{K_j} \nabla_{\mathbf{v}} \psi_i(\mathbf{z}) \cdot \left[(\mathbf{v} - \mathbf{u}_h^j) f_h^j + \frac{T_h^j}{m} \nabla_{\mathbf{v}} f_h^j\right] d\mathbf{z}, \quad (2.32)$$

where the numerical flux $\widehat{\mathbf{G}}$ contains surface contributions from both the drag and diffusion terms,

$$\widehat{\mathbf{G}} = \frac{1}{2}(\mathbf{v} - \mathbf{u})(f_h^{j+} + f_h^{j-}) + \frac{\max|\mathbf{v} - \mathbf{u}|}{2}(f_h^{j+} - f_h^{j-}) + \frac{T_h^j}{m} \nabla_{\mathbf{v}} \widetilde{f}_h^j.$$

Note the distribution function in last term of the numerical flux, arising from the diffusion surface term, is a recovered quantity because we are taking a derivative across a boundary. While this weak form of the LBO is consistent with previous methods, [52] have demonstrated that this formalism does not conserve momentum or energy. Verifying conservation of momentum and energy involves setting the test function ψ_i as either $m\mathbf{v}$ or $v^2/2$ for momentum or energy, respectively, and summing over all cells in the domain. Because we ensured the fluxes are matched at cell boundaries, surface terms cancel out at every internal surface within the computational domain and leave only contributions at the domain

boundaries. Our boundary condition in velocity space is the zero-flux condition,

$$\widehat{\mathbf{G}}(\mathbf{x}, \mathbf{v}_{max}) \cdot \widehat{\mathbf{n}} = \widehat{\mathbf{G}}(\mathbf{x}, \mathbf{v}_{min}) \cdot \widehat{\mathbf{n}} = 0,$$

so the summed surface terms at domain boundaries also vanish. However, as [52] show in more detail, the sum over all cells of the volume term in Eq. 2.32 does not cancel out due to the gradient of f. To solve this, we move the gradient off of the volume term by integrating it by parts again. A single integration by parts for the drag term and two integrations by parts for the diffusion term yields the following scheme,

$$\begin{aligned}\frac{1}{\nu}\int_{K_j}\frac{\partial f_h^j}{\partial t}\psi_i(\mathbf{z}) = \oint_{\partial K_j}\psi_i(\mathbf{z})\widehat{\mathbf{n}}\cdot\widehat{\mathbf{G}}dS - \oint_{\partial K_j}\frac{T_h^j}{m}\widetilde{f}_h^j\nabla_{\mathbf{v}}\psi_i(\mathbf{z})\cdot\widehat{\mathbf{n}}dS \\ - \int_{K_j}\nabla_{\mathbf{v}}\psi_i(\mathbf{z})\cdot(\mathbf{v}-\mathbf{u}_h^j)f_h^j - \nabla_{\mathbf{v}}^2\psi_i(\mathbf{z})\frac{T_h^j}{m}f_h^j d\mathbf{z},\end{aligned} \tag{2.33}$$

where we note that, unlike the Vlasov equation or Maxwell's equations, the two surface terms that emerge from integrating the diffusion term by parts twice utilize the recovery process on the distribution function. When verifying conservation of energy and momentum in the same way as before, a system of equations emerges after the summation over all cells that serve to constrain the values of T_h and $\mathbf{u}_h$ such that the scheme is conservative in mass, momentum, and energy. It has been shown that our discrete Vlasov-Maxwell system conserves particles and energy, though the discontinuous electric and magnetic fields result in the scheme not conserving momentum [63]. However, the errors in momentum still demonstrate high-order convergence with increasing configuration space resolution [64]. Together, equations 2.28, 2.30, 2.31, and 2.33 comprise our semi-discrete formalism of the Vlasov-Maxwell-Fokker-Planck system.

As mentioned in Section 2.2.1, the time discretization methods of choice in `Gkeyll` are

SSP-RK timesteppers. Explicit timestepping schemes introduce an upper bound on the size of timestep our scheme can use, given by the CFL condition. Broadly speaking, the CFL condition exists to prevent information from traveling across multiple cells within a single timestep by limiting the size of the timestep such that the largest characteristic velocity in the scheme does not have enough time to travel the length of the grid spacing, Δx. In other words,

$$\Delta t < \sigma \frac{\Delta x}{a},$$

where a is that largest characteristic velocity, which is often the speed of light for electromagnetic problems. An additional factor of safety, σ, is also often included in the timestep calculation, and in the case of modal DG, we take that factor to be $\sigma = 1/(2p+1)$. In the case of our semi-discrete Vlasov-Maxwell/Fokker-Planck system that is defined in phase space, the CFL constraint instead uses a CFL frequency [63],

$$\omega_i = \frac{\alpha_i}{\Delta z_i}, \qquad i = 1, \cdots, n_d,$$

where α_i is the largest "speed" in each of the n_d total dimensions. Practically for the Vlasov/Fokker-Planck contribution, for each configuration space dimension, α_i will be the velocity value at the domain edge in each direction, and in each velocity space dimension, α_i is the largest acceleration due to the Lorentz force in the domain in each direction. Collisional contributions to the timestep are treated similarly, where, in the LBO for example, there is a maximum advection speed associated with the drag term and a characteristic speed from the diffusion term. With the CFL frequency calculated in each direction, the CFL condition used to determine the maximum timestep is then

$$\Delta t \leq \frac{CFL}{2p+1} \sum_{i=1}^{n_d} \omega_i, \tag{2.34}$$

where *CFL* is the so-called CFL number, which is often simply 1.0 for explicit timesteppers. For the four stage, third order SSP-RK method introduced earlier, Eqs. 2.15-2.18, $CFL = 2.0$, allowing for twice as large of a timestep relative to other methods. We now have the spatial and temporal discretizations for the Vlasov-Maxwell/Fokker-Planck equations, so after summarizing the introductory information in this chapter, we will move onto some physics studies using this model.

2.4 Summary of Chapter 2

In this chapter, we moved from pure plasma physics into the realm of computational methods. To begin, we discussed some of the most common numerical methods used for solving partial differential equations computationally: the finite difference method, finite volume method, and finite element method. Each has their own advantages and disadvantages, and considering all of them served to help us consider capabilities we would like to have for kinetic plasma simulations. Thankfully, we did not have to lock ourselves into one of the three methods and miss out entirely on the others' advantages. The discontinuous Galerkin (DG) method offers the locality of information and discontinuous solutions of the finite volume method while allowing us to increase the order of accuracy of our scheme arbitrarily in each cell like finite element methods. We then applied the DG method to the simple linear advection equation to study weak equality, orthonormal basis functions, and explicit timestepping. While some of the finer points of DG were glossed over in this introduction, the fundamental process of DG discretization and weak equality will be used substantially in Chapter 4 of this dissertation, so this background will be relevant as we progress. For more rigorous treatments of DG, readers are referred to the text by Hesthaven and Warburton [56] and publications of other members of the `Gkeyll` team for application of DG to the

Vlasov-Maxwell [18, 63], gyrokinetics [82], and multifluid systems [136] in `Gkeyll`, respectively. We concluded our discussion of DG with an introduction to recovery, the process by which we construct higher-order continuous representations of DG discretized quantities using information from multiple cells.

Finally, we were prepared to apply the DG method to the Vlasov-Maxwell/Fokker-Planck system of equations. Following the same procedure as the advection equation yielded us the semi-discrete Vlasov equation, Eq. 2.28. Discretizing Maxwell's equations required us to contract the grid and basis sets from phase space to configuration space, but the process was identical otherwise. We then applied weak equality to arrive at the discrete form of the Dougherty-Lenard-Bernstein collision operator, where we were required to handle the drag and diffusion terms differently, an idea which we will encounter again in greater depth in Chapter 4. As we saw, a single integration by parts of the LBO was not the correct technique, for the gradient left in the volume term from the diffusion term caused the scheme to not conserve momentum or energy. A second integration by parts of the diffusion volume term solved the problem, and we put recovery into practice by including recovered quantities in the two diffusion surface terms. With the semi-discrete form of the Vlasov-Maxwell/Fokker-Planck system attained, we ended this discussion by introducing the CFL constraint on the timestep and discussed the calculation of the maximum allowable timestep.

From here, we have what we need to begin studying physics problems with this model. In the next chapter, the DG discretization of the collisional Vlasov system is applied to the Rayleigh-Taylor instability.

Chapter 3

Continuum-Kinetic Simulations of the Rayleigh-Taylor Instability in `Gkeyll`

In this chapter, results are presented of two published studies of the Rayleigh-Taylor (RT) instability using the DG implementation of the Vlasov-BGK model in `Gkeyll` [97, 98]. As will be discussed, the RT instability is one that is historically treated almost exclusively with fluid techniques. Fluid models take velocity space moments of the kinetic equations to average out individual particle information, reducing the dimensionality of the problem from phase space to configuration-space and inherently assuming distribution functions are always Maxwellian (i.e. the limit of infinite collisionality). This approximation is useful in many regimes where the dynamics of interest occur on time scales much larger than those of individual particle motions. However, the RT instability is present in a staggering range of scales, implying the existence of regimes in which kinetic physics is relevant. The question is, how does the evolution of the RT instability differ from the fluid result when collisionality is finite?

We consider the simple case of an uncharged species in hydrodynamic equilibrium under the effect of gravity. The collision model of choice for these neutral species applications is the BGK operator, Eq. 1.62. In the first section, published in [97], the collision frequency is a constant value everywhere in the domain. Following that, results are presented of simulations where the collision frequency is allowed to vary in space and time for more realistic behavior,

published in [98].

3.1 Constant Collision Frequency Simulations of the Rayleigh-Taylor Instability

Rayleigh-Taylor (RT) instabilities occur when a dense fluid is accelerated into a lighter fluid, for example under the influence of a gravitational field [93, 128]. While this instability is traditionally studied in a strictly fluid regime [70, 106], applying a fully-kinetic treatment allows for study of a range of collisionality, from collisionless and intermediate, where fluid models are not applicable, to highly collisional regimes approaching the fluid limit [42, 92, 102, 138].

This work explores a fully-kinetic treatment of the classical RT instability for a single neutral particle species for varying collisionality, with a future goal of extending into a collisional two-species plasma with evolving electromagnetic fields. A body of literature exists studying magnetohydrodynamic and extended-magnetohydrodynamic modeling of the RT instability [120, 122, 123], the role of viscosity, resistivity, and thermal conduction in RT and magneto-RT instability growth [7, 117, 121], and the role of incorporating some kinetic effects on the magneto-RT instability through use of higher-fidelity fluid models [60, 119].

Kinetic effects can emerge when mean-free-paths are long relative to a relevant characteristic length scale. Shock-driven implosion experiments at the OMEGA Laser facility [12] have shown evidence of kinetic phenomena in high-energy-density regimes, such as non-hydrodynamic mixing, thermal decoupling, and species separation [95, 96, 100]. Emergence of kinetic effects within a shock may imply the presence of kinetic effects for the RT instability when mean-free-paths are long relative to the fluid interface. Other implosion experiments

at OMEGA have studied the physics relevant to RT instability growth in core-collapse supernovae but focused on a purely hydrodynamic interpretation of the results [33, 71, 72]. As there is evidence of a transition from a hydrodynamic to a kinetic regime within OMEGA high-energy-density experiments, fully-kinetic simulations to accompany RT experiments may offer a novel explanation of disparities between experiment and hydrodynamic simulation.

For these studies, the continuum-kinetic capabilities of the plasma simulation framework `Gkeyll` [45] are used to evolve particle distribution functions, f. `Gkeyll` uses a discontinuous Galerkin method [29, 30, 94] to discretize and evolve the Boltzmann equation [50, 63],

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} f + \mathbf{a}\cdot\nabla_{\mathbf{v}} f = \left(\frac{\partial f}{\partial t}\right)_C, \tag{3.1}$$

where $\mathbf{x}$ and $\mathbf{v}$ are the two independent particle position and velocity, respectively. Acceleration vector, $\mathbf{a}$, is simply gravity, $\mathbf{g}$, for this work, as only neutral particles will be considered. The right-hand term accounts for particle collisions and is approximated here by the Bhatnagar-Gross-Krook (BGK) operator [9, 18],

$$\left(\frac{\partial f}{\partial t}\right)_C = \nu(f_M - f), \tag{3.2}$$

where f_M is an ideal Maxwellian distribution function calculated from moments of f, and ν is the collision frequency. The BGK operator is necessarily conservative in number density, momentum, and energy when ν is constant with respect to particle velocity as it is in this work. This approximation is appropriate for neutral species, as considered here. For a plasma, ν is generally known to scale with particle velocity as v^{-4}. Assuming constant ν for a plasma would overestimate energy-fluxes in the high-energy tails of the distribution.

`Gkeyll` discretizes f on a phase space grid of up to six dimensions by decomposing f using a set of piecewise polynomials with superlinear order up to p [4]. Distribution functions are then evolved in time using a strong-stability-preserving Runge-Kutta method.

3.1.1 Problem Description

Distribution functions in this work are 5-dimensional, with two spatial and three velocity space dimensions, (x, y, v_x, v_y, v_z), and have initial conditions derived from hydrostatic equilibrium with

$$\nabla p = -nm\mathbf{g}. \tag{3.3}$$

All units are normalized using a particle species of mass $m = 1.0$, upper bound density $n_1 = 1.0$, and gravity $g = 1.0$. Initial number density and pressure profiles are as follows

$$n(y) = \frac{n_0}{2}\tanh\left(\frac{\alpha y}{L_y}\right) + \frac{3}{2}n_0, \tag{3.4}$$

$$p(y) = \frac{mgn_0}{2}\left[\ln\left(\cosh\left(\frac{\alpha y}{L_y}\right)\right) + 3y\right] + \frac{3}{2}n_0 T_0, \tag{3.5}$$

where $n_0 = 0.5$ is density at the center of the interface, $L_y = 1.0$ is half the length of the simulation domain in y, and T_0 is an arbitrary constant chosen to ensure the minimum pressure in the domain is positive. With the density and pressure profiles above, the interface between the high- and low-density regions is continuous and has width defined by α. Simulations are initialized with $\alpha = 25$ to ensure the width of the interface is small relative to the domain size. This initial density profile corresponds to an Atwood number, $A_t = (n_1 - n_2)/(n_1 + n_2)$, of 1/3. Boundary conditions are periodic in x and static reservoir in y, where the edge ghost layers of cells are a continuation of the initial conditions and do not evolve in time. Distribution functions are initially Maxwellian in velocity space, according

to,

$$f(\mathbf{v}) = \frac{n}{(2\pi v_{th}^2)^{3/2}} \exp\left(-\frac{(\mathbf{v}-\mathbf{u})^2}{2v_{th}^2}\right), \tag{3.6}$$

for initial bulk velocity $\mathbf{u}$, where $v_{th} = \sqrt{T/m}$ is thermal velocity. The pressure profile given by Eq. 3.5 is used to calculate a temperature $T = p/n$, which is then used to initialize the Maxwellian distribution.

While these initial conditions are hydrostatic, they are not a true Boltzmann equilibrium for the case of finite collision frequency, as any deviations from Maxwellian are not immediately damped out by collisions. Additionally, if the collision frequency is not sufficiently high, the interface diffuses and the fluid layers mix before the instability grows.

To generate the RT instability, a single-mode sinusoidal perturbation of wavenumber k is applied to the y-direction bulk velocity, u_y, according to

$$u_y = -0.1 v_{th,c} \cos(kx) \exp\left(-\frac{y^2}{2y_r^2}\right), \tag{3.7}$$

where $k = \pi/(2L_x)$, $v_{th,c}$ is initial thermal velocity at the center of the domain, $L_x = 0.75$ is half the simulation domain length in x, and $y_r = L_y/10$ is a characteristic decay length for the perturbation. Initial conditions of n, v_{th}^2, and u_y are shown in Figure 3.1.

3.1.2 Results

Collision frequencies are calculated from a chosen Knudsen number, $Kn = \lambda_m/L_x$, i.e., the ratio between particle mean-free-path λ_m and scale length L_x. Collision frequencies are assumed to be constant spatially and temporally, according to $\nu = v_{th,c}/\lambda_m$. However, collision frequency is generally known to scale with density and temperature [15], and RT instability simulations with spatially-varying collisionality will be explored in future work.

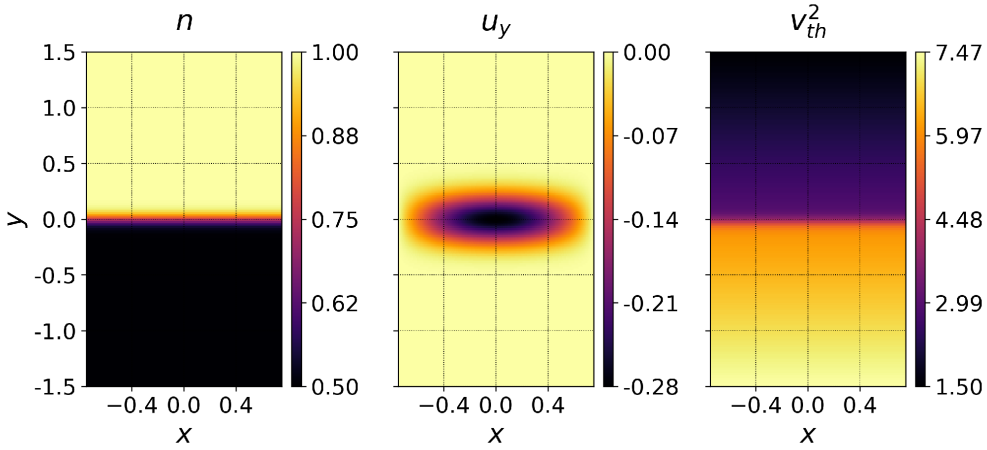

Figure 3.1: Initial conditions in number density (left), bulk velocity (center), and square of thermal velocity (right).

In this work, values of Kn are chosen as 0.1, 0.01, and 0.001. Simulations are run to an end time of 3 classical RT instability growth periods, $\tau_{RT} = 1/\sqrt{kgA_t}$. Time evolution of number density and temperature for each case is shown in Figure 3.2 and Figure 3.3 respectively, for times 0, $1.5\,\tau_{RT}$, and $3.0\,\tau_{RT}$.

The fluid interface diffuses in all cases due to finite collisionality. As mean-free-path increases from the limit of infinite collisionality, particles stream past one another over longer distances without interacting. The net result is a mixing of the fluid layers that speeds up as mean-free-path increases, as particles are not affected by the pressure gradient until a collision event. With no perturbation, the interface continues to diffuse until the fluid layers mix completely.

For the lowest collisionality case, the interface diffuses so quickly relative to the RT instability growth timescale that there is effectively no interface where the instability can form. As collisionality increases by an order of magnitude, the interface diffuses slowly enough that the RT instability is able to grow. At the end time, the expected bubble and spike structures are present with diffuse edges. The most collisional case approaches the expected fluid result,

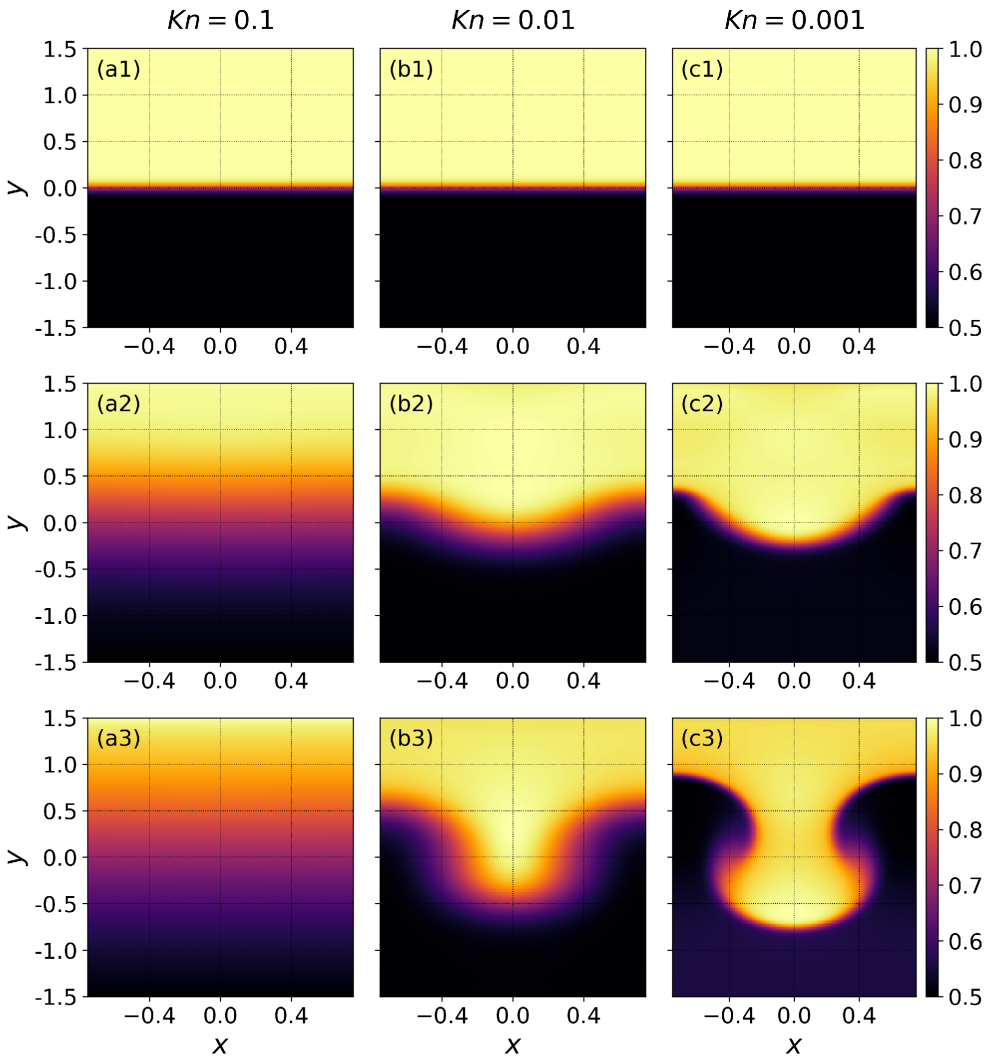

Figure 3.2: Time evolution of number density for varying collisionality. Left to right is Kn of 0.1 (a), 0.01 (b), and 0.001 (c). Top to bottom is time 0.0, $1.5\tau_{RT}$, and $3.0\tau_{RT}$. Note that the low collisionality case (left column) presents no RT instability growth, and the intermediate collisionality case (middle column) presents significantly altered RT instability growth compared to the high collisionality case (right column) which approaches the fluid limit.

with minimal diffusion of the interface and mushroom structures on the bubble and spike as secondary Kelvin-Helmholtz instabilities form. The temperature distribution exhibits identical behavior to the density evolution. Note that the growth of the RT instability for the intermediate case is slower than that of the highly collisional case.

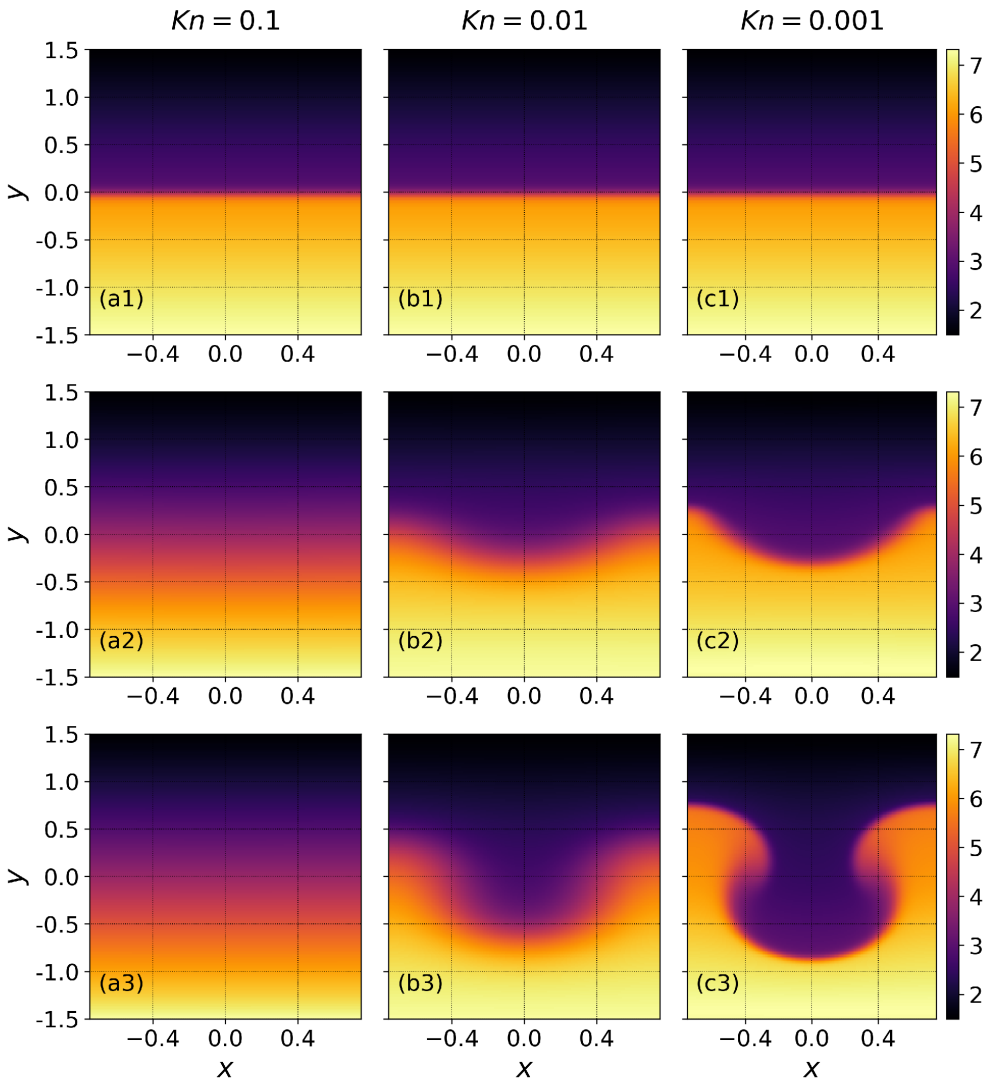

Figure 3.3: Time evolution of temperature for varying collisionality. Left to right is Kn of 0.1 (a), 0.01 (b), and 0.001 (c). Top to bottom is time 0.0, $1.5\tau_{RT}$, and $3.0\tau_{RT}$.

In order to quantify the effects of collisionality on RT instability growth, an approach similar to [102] and [34] is used to calculate a growth rate, γ_0, that includes viscous and diffusive effects,

$$\gamma_0 = \sqrt{kgAt + \nu_v^2 k^4} - (\nu_v + \xi)k^2, \tag{3.8}$$

where $\nu_v = v_{th,c}\lambda_m/2$ is the kinematic viscosity, and $\xi = \nu_v$ is the diffusion coefficient. Note that [102] and [34] include an additional factor for dynamic diffusion effects to calculate a

time-dependent growth rate, which has been neglected here. Because the primary dynamic diffusion effect is the diffusion of the interface, which occurs exclusively early in the simulation, those early data points are excluded from the growth rate calculation to achieve a constant linear growth rate that describes RT instability growth for the majority of the simulation. Growth rates are calculated using h, the difference between the top of the bubble and the bottom of the spike, and are presented in Figure 3.4 for the case of $Kn = 0.01$ and 0.001, compared with a neutral fluid simulation using the Euler equations. It is assumed that kinetic simulations converge to those of the Euler equations in the limit of infinite collisionality as non-ideal transport becomes negligible. Early data points are also ignored for the fluid simulations, as the perturbation to u_y causes waves to be launched that interfere with RT instability growth early in time. Growth rates calculated from the linear fits in Figure 3.4 are compared with theoretical growth rates in Table 3.1. There is good agreement between the calculated growth rates and the theoretical growth rates with static diffusion, and as Kn increases, γ and γ_0 approach the fluid result. The slight decrease in agreement from the 0.01 Kn case to the 0.001 case is likely due either to the presence of diffusion in the kinetic case or not capturing the transition from time-varying growth to linear growth as well in the data output frames (i.e., the transition is between data points 3 and 4 for the 0.001 case). While fluid simulations of the RT instability have been performed with viscosity [7, 84, 117, 121], the presence of a fluid viscosity alone is insufficient to explain the diffusion of the interface seen here (including the dynamic diffusion effects early in time). To explain the kinetic parameter regime of the intermediate collisionality case, this work is a first to

Table 3.1: Values of RT instability growth rates, calculated from simulation (γ) and theory (γ_0).

Case	γ	γ_0
$Kn = 0.01$	0.9635	0.9723
$Kn = 0.001$	1.1369	1.1603
Fluid	1.1708	1.1816

probe into a detailed kinetic interpretation of the RT instability.

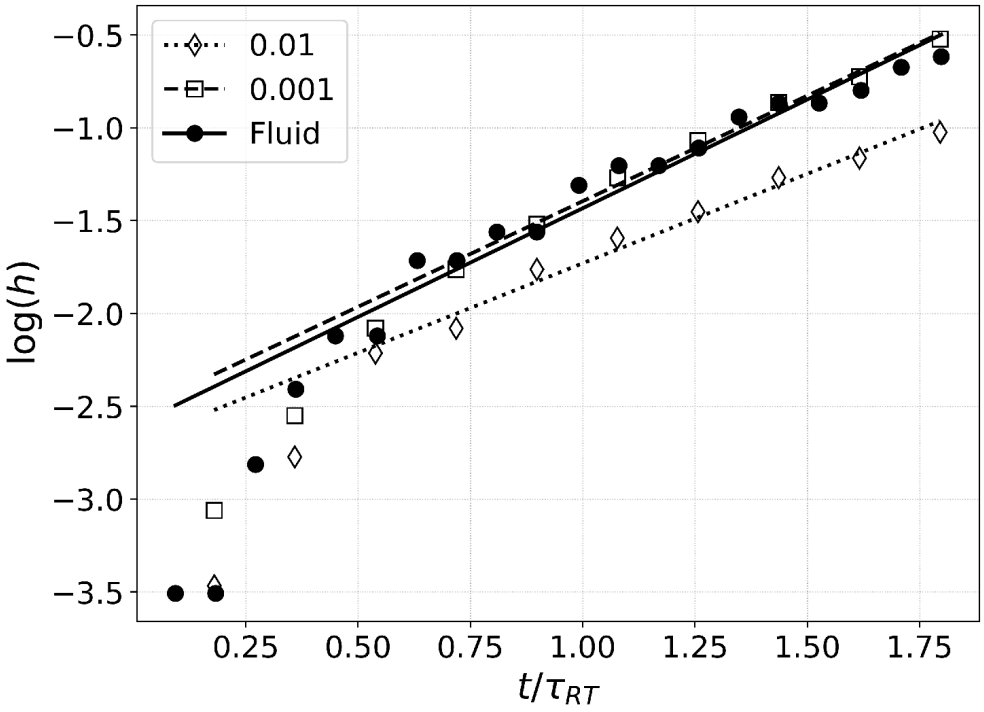

Figure 3.4: Logarithm of h, the difference between spike and bubble heights, as a function of time for $Kn = 0.01$ and 0.001 and a fluid simulation using the Euler equations. Data points early in time are excluded from the fit due to dynamic diffusion of the interface for the kinetic cases and wave launching for the fluid case. Note as Kn decreases, the RT instability growth rate approaches the fluid simulation result.

While highly collisional regimes asymptoting to fluid results are reasonably well-understood for the neutral fluid RT instability, intermediate collisional regimes require kinetic simulations since the fluid model is no longer valid in these regimes. Variation in RT instability growth as a function of collisionality implies the emergence of kinetic effects as collisionality decreases and distribution functions are allowed to become less Maxwellian. A metric to quantify non-Maxwellian distributions spatially can aid in probing the 5-dimensional distribution function by highlighting potential areas of variation from equilibrium. In an attempt to capture the spatial distribution of such variations, a density analogue is constructed from the distribution function and a constructed Maxwellian as follows,

$$n_N(\mathbf{x}) = \int |f(\mathbf{x},\mathbf{v}) - f_M(\mathbf{x},\mathbf{v})| \mathrm{d}^3\mathbf{v}. \tag{3.9}$$

This non-Maxwellian density allows for spatial representation of non-Maxwellian distribution functions and has units of density, allowing for simple comparison to the density profiles in Figure 3.2. Non-Maxwellian density for each Kn is presented below in Figure 3.5. Note that recent work by Cagas et al. [19] shows that a boundary layer forms at reservoir boundaries for the Vlasov-BGK model, Eqs. (1) and (2). The boundary layer is approximately one mean-free-path wide and non-Maxwellian. Therefore, three layers of cells at the top and bottom of the domain are omitted in Figure 3.5 in order to maintain a useful color scale for the regions of interest. As expected, high collisionality leads to a decrease in n_N by approximately an order of magnitude between the most and least collisional cases. For the case where the RT instability does not develop, n_N simply follows an almost identical distribution to density, comparing Figure 3.2 (a3) to the center plot of Figure 3.5. However, for the cases where the RT instability develops, the interfaces appear as regions of peak n_N. Magnitudes of n_N are small relative to n, even for the least collisional case that has the highest peak n_N.

To further characterize the effect of varying collisionality, two higher moments of the distribution function are defined,

$$\mathcal{P}_{ij} = m \int v_i v_j f \mathrm{d}^3\mathbf{v}, \tag{3.10}$$

$$\mathcal{Q}_{ijk} = m \int v_i v_j v_k f \mathrm{d}^3\mathbf{v}. \tag{3.11}$$

As in Wang et al. [137], by defining $w_i = v_i - u_i$, Eq. 3.11 can be expanded and tensor contracted to get the particle energy-flux (using Einstein's summation convention),

$$\frac{1}{2}\mathcal{Q}_{iik} = \underbrace{\frac{5}{2}u_k p + \frac{1}{2}mnu_k\mathbf{u}^2}_{\mathrm{I}} + \underbrace{q_k + u_i\Pi_{ik}}_{\mathrm{II}}, \tag{3.12}$$

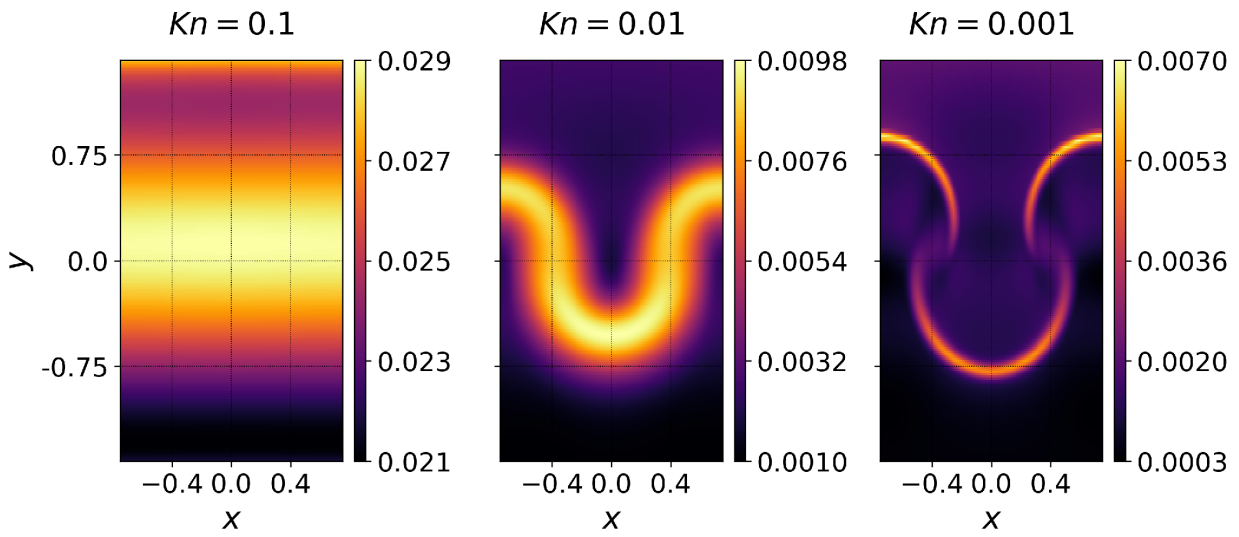

Figure 3.5: Density of non-ideal distribution, according to Eq. 3.9, for Kn of 0.1 (left), 0.01 (center), and 0.001 (right), normalized to number density. Note the varying color scale for each subplot.

where

$$q_k = \frac{1}{2} m \int w_i w_i w_k f \mathrm{d}^3\mathbf{v}, \tag{3.13}$$

is the heat flux vector in the gas frame, and the stress tensor Π_{ij} is related to the pressure tensor,

$$P_{ij} = m \int w_i w_j f \mathrm{d}^3\mathbf{v}, \tag{3.14}$$

by $\Pi_{ij} = P_{ij} - p\delta_{ij}$ with scalar pressure $p = P_{ii}/3$. The pressure tensor is also related to the second moment by $\mathcal{P}_{ij} = P_{ij} + mnu_i u_j$. Note that the use of collision frequency that is independent of particle velocity leads to an overestimation of energy fluxes in the high-energy tails of the distribution if charged species are considered instead of neutral species. In the case of charged species, the energy-fluxes presented here will be greater in magnitude than those calculated with a collision frequency that varies with velocity. This work considers neutral species. Individual terms are grouped in Eq. 3.12 by whether they arise from Maxwellian parts of the distribution (group I), or non-Maxwellian parts (group II). Group I will be

referred to as ideal terms, while group II are the non-ideal terms. The y-component of each term of Eq. 3.12 (normalized to $n_0 v_{th,c}^3$) is plotted in Figure 3.6 for the cases of $Kn = 0.01$ and 0.001 for times with similar instability amplitude. Note that magnitudes of each column (term) vary by orders of magnitude, so color scales are distinct by column to show spatial features. The first ideal term is the dominant term by several orders of magnitude at its peak for both cases. As collisionality increases, all terms increase in magnitude, though the ideal terms increase more than the non-ideal terms. This can be seen by taking the ratio of the average of the absolute values of the ideal terms to that of the non-ideal terms. The ratio is 21.5 for the less collisional case and 283.8 for the more collisional case, indicating the particle energy-flux becomes less dominated by the ideal terms as collisionality decreases. This is an important and impactful result as it is the first to present an order of magnitude increase in the importance of the non-ideal terms for the less collisional (more kinetic) case of the RT instability. The overall increase in energy-flux with increased collisionality, even when comparing similar amplitudes of RT instability growth, relates to the increase in growth rate shown in Table 3.1, as larger total flux leads to faster instability growth.

By taking moments of a first-order Chapman-Enskog expansion of the BGK collision operator, expressions for the heat flux, $q_{i,\mathrm{BGK}}$, and stress tensor $\Pi_{ij,\mathrm{BGK}}$, can be obtained assuming a nearly Maxwellian distribution,

$$\Pi_{ij,\mathrm{BGK}} = -\frac{p}{\nu}\left(\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right), \tag{3.15}$$

$$q_{i,\mathrm{BGK}} = -\frac{5p}{2m\nu}\frac{\partial T}{\partial x_i}. \tag{3.16}$$

Figure 3.7 presents the non-ideal terms of the particle energy-flux calculated directly from the distribution function with those calculated from the expansion. Note color scale is held

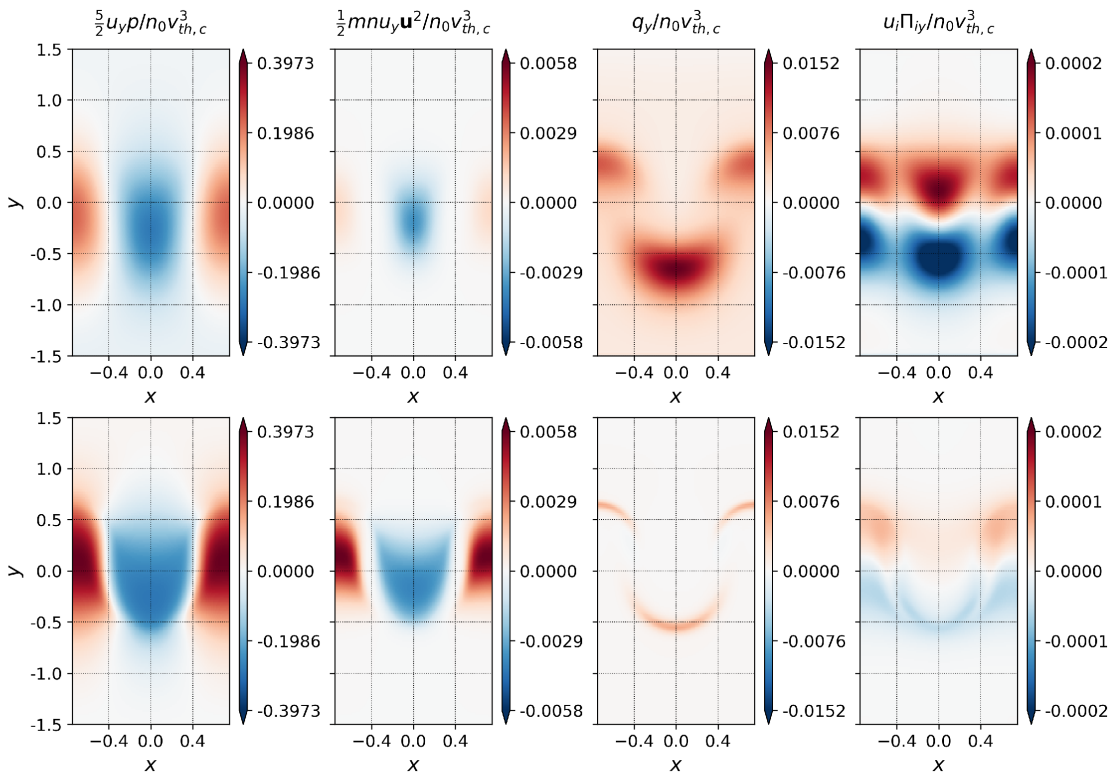

Figure 3.6: Terms of the expanded particle energy-flux, Eq. 3.12, in the y-direction for Kn of 0.01 (top) and 0.001 (bottom), normalized to $n_0 v_{th,c}^3$. Energy-flux is calculated at normalized time $3.0\tau_{RT}$ for the 0.01 case and $2.1\tau_{RT}$ for the 0.001 case to have similar amplitudes. Note the varying color scale of each column.

constant for each term compared across the two different values of collisionality. For both degrees of collisionality, the heat flux terms are similar in both magnitude and spatial distribution. The stress terms show more deviation between true and approximate results likely due to the fact that two stress tensor elements are involved in the calculation, so errors from the first-order approximation compound. As collisionality decreases from $Kn = 0.001$ to 0.01, the approximate stress term deviates more from the direct calculation because the assumption of near-equilibrium distribution becomes less accurate with decreasing collisionality.

Higher moments of the distribution function are also measures of non-ideal distribution, so

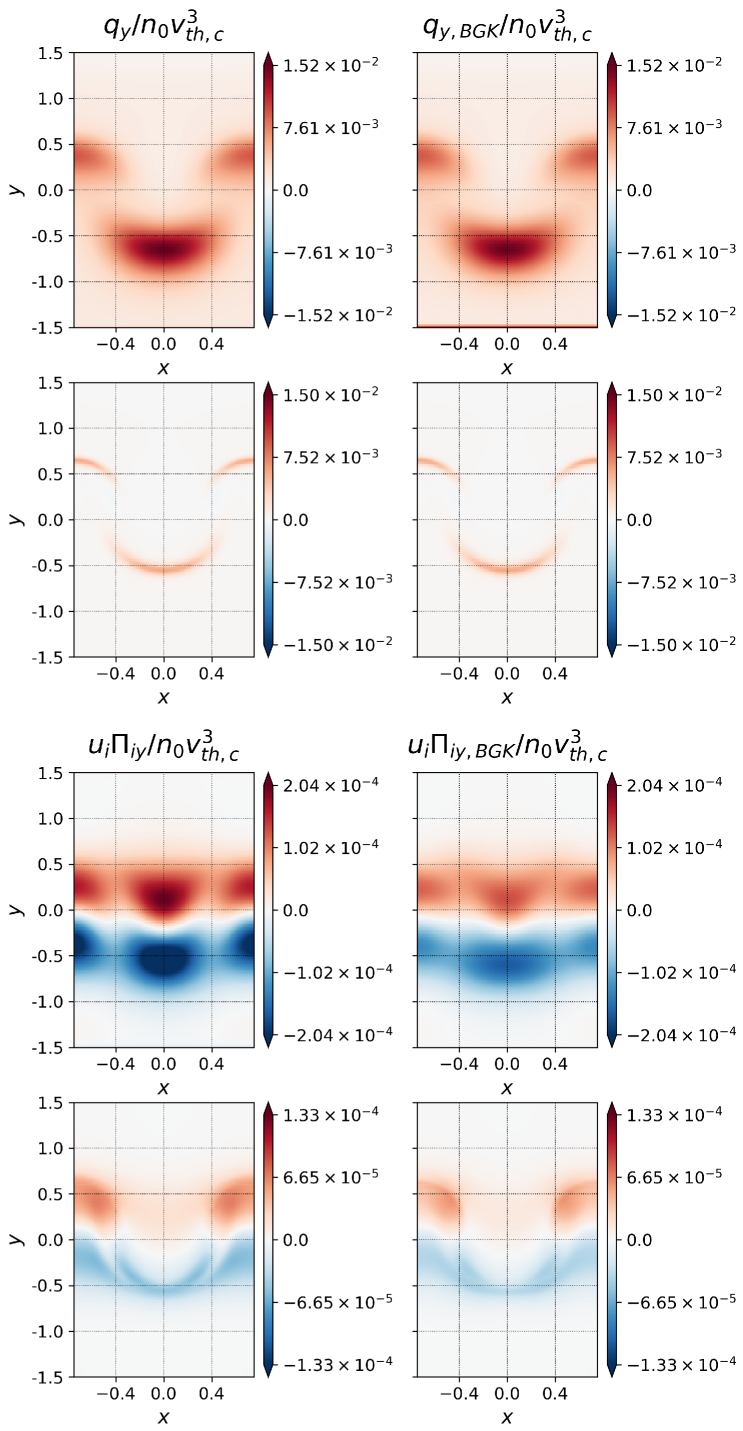

Figure 3.7: Comparison of energy-flux non-ideal terms, Eq. 3.12, calculated from distribution function and those calculated from a first-order Chapman-Enskog expansion of the collision operator. Top and bottom rows of each comparison are $Kn = 0.01$ and 0.001, respectively. Note the similarities in spatial distribution and magnitude and that color scales are constant by term and row (collisionality). Stress terms show more discrepancy because they are calculated from several stress tensor elements, so errors compound.

the spatial distribution of gas-frame higher moments should correlate with n_N. Presented in Figure 3.8 are comparisons of n_N, y-direction vector skewness, q_y, and y-direction excess kurtosis,

$$\delta K_y = \int w_i^4 f \mathrm{d}^3\mathbf{v} - \int w_i^4 f_M \mathrm{d}^3\mathbf{v}. \tag{3.17}$$

As expected, the distribution of n_N aligns with those of the higher moments. Additionally, the magnitudes of the normalized higher moments increase as collisionality decreases, which is expected as decreased collisions deviate from a Maxwellian distribution function towards a more kinetic regime. The evolution of the intermediate collisionality RT instability is clearly distinguished from the high collisionality regime to explain the kinetic effects that produce the difference in growth rates and morphology. These are the first results to present a high-fidelity kinetic interpretation of the classical RT instability in low and intermediate collisionality regimes where fluid models are inadequate.

3.1.3 Conclusion

Single-mode Rayleigh-Taylor instabilities are successfully simulated in 2x3v using the continuum-kinetic capabilities of `Gkeyll` for a range of collisionalities. As mean-free-paths become smaller relative to the width of the simulation domain, the resulting instability approaches the classical fluid result, as expected. Growth rates estimated using static viscosity and diffusion agree well with calculated growth rates when early dynamic diffusion of the interface is left out of the fit. Non-Maxwellian density, the velocity space integration of the difference between a local particle distribution function and a corresponding Maxwellian distribution calculated from the first three moments, shows that, as collisionality increases, the distribution function approaches a Maxwellian (fluid) distribution. Local maxima in non-Maxwellian density also occur around the primary areas of transport, i.e., the edges of the bubble and

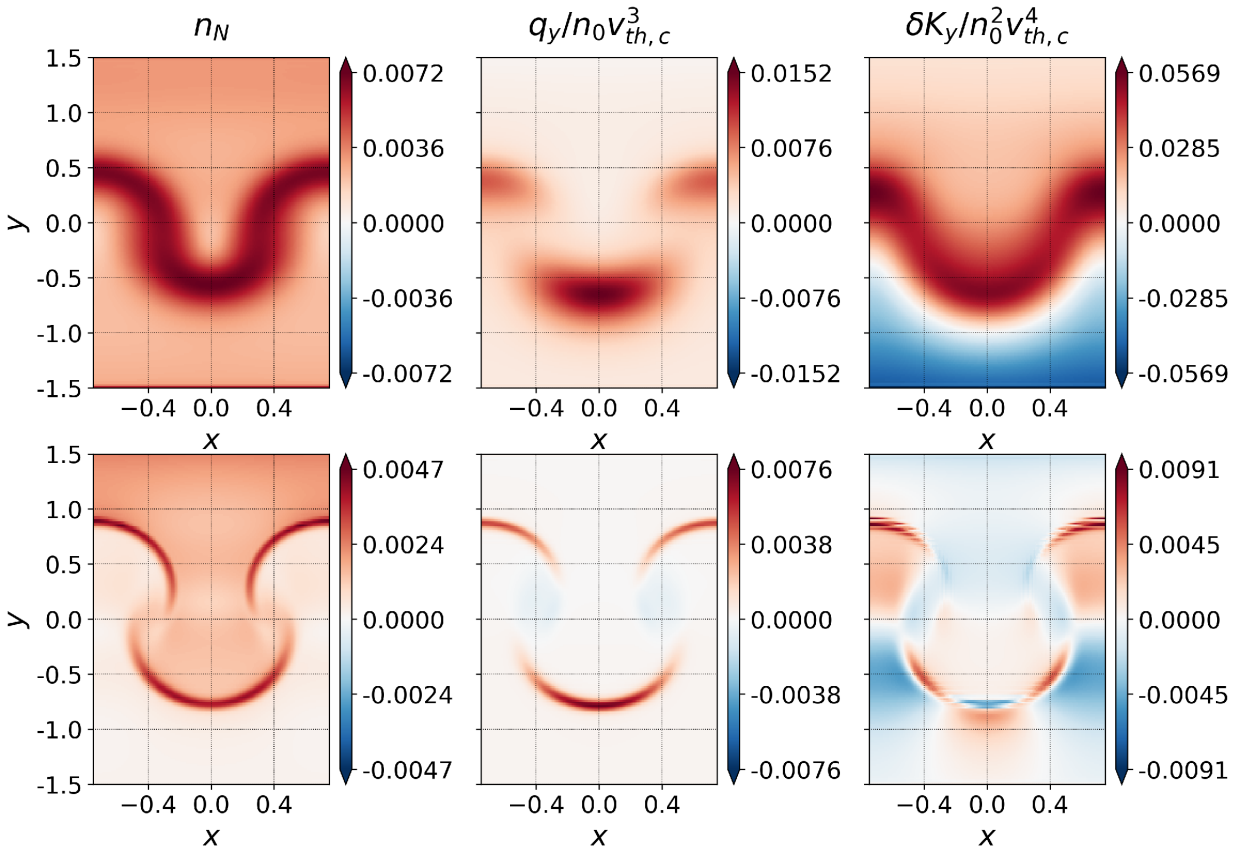

Figure 3.8: Non-Maxwellian density, n_N, compared with gas-frame y-direction skewness, q_y, and excess kurtosis, δK_y. Note the presence of local extrema for all quantities around the RT instability interface.

spike. A decomposition of the particle energy-flux shows that transport is dominated by terms that arise from the Maxwellian parts of the distribution, and the ideal terms of the expansion become more dominant as collisionality increases toward the fluid limit.

An important and novel contribution of this work is in the intermediate collisional cases that are not accessible with traditional fluid models and require kinetic modeling. The continuum-kinetic model used in this work provides unique access to the full noise-free distribution function to investigate the kinetic regime. Simulations of intermediate collisional cases show significantly altered RT instability evolution compared to the high collisionality fluid-like cases highlighting the importance of kinetic physics through higher moments of the distribution function. These higher moments include the heat flux vector, which is the third moment indicating the skewness of the distribution, and the fourth moment indicating the kurtosis of the distribution. The heat flux vector plays a more significant role relative to inertial terms in the intermediate collisional cases compared to the highly collisional cases. A quantitative comparison shows an order of magnitude difference in the ratio of the non-ideal terms to the ideal terms when comparing the intermediate collisional cases to the highly collisional fluid-like cases. These kinetic effects are primarily noted in the region of the RT instability interface. Regimes of intermediate collisionality often occur in astrophysical and laboratory plasmas requiring a kinetic model due to the invalidity of the fluid model for these cases, highlighting the significance and relevance of the results presented here.

3.2 Rayleigh-Taylor Instability with Varying Collision Frequency

Rayleigh-Taylor (RT) instabilities occur when a dense fluid is accelerated into a lighter one, for example under the influence of gravity [93, 128]. Traditionally, RT instabilities are studied using fluid models [92, 106], yet fully-kinetic simulations demonstrate the existence of regimes of finite collisionality that are RT unstable and exhibit significantly altered dynamics as compared to ideal fluid results [97, 102]. Previous simulations investigate the role of kinetic effects such as viscosity, resistivity, and thermal conductivity on the growth of the RT instability [7, 117, 121], but studies of these effects with a fully-kinetic model with a full nonlinear Fokker-Planck collision operator have yet to be performed.

The RT instability appears in high-energy-density regimes such as pulsar wind nebulae like the Crab nebula as a source of large-scale structure and mixing at the surface of the supernova shell [26, 90], laser implosions at the OMEGA laser facility [12, 69, 115], and early stages of supernova explosions [8, 27]. Supernova remnants expanding into the interstellar medium can also give rise to the Rayleigh-Taylor instability [28, 49]. Low collisionality and a weak magnetic field serve to reduce classical transport coefficients in the intergalactic medium [140], so kinetic simulations may be warranted to completely capture interactions of the supernova shell and the interstellar medium. In the Crab nebula, the interaction between the wind accelerated by the pulsar and the cold supernova shell is an RT-unstable configuration with orders of magnitude of variation in density and pressure across the interface [62, 91]. In general, collision frequency is a function of density and temperature [15]. Densities and temperatures can vary greatly in astrophysical regimes that are RT unstable [28, 49, 90], so collision frequencies are expected to vary similarly. Previous fully-kinetic RT instability simulations assumed spatially and temporally constant collision frequency [97]. This

work explores a continuum-kinetic, neutral species RT instability simulations with spatially-varying collisionality. It is the purpose of this work to explore a situation where the collision frequency varies strongly across the interface.

The rest of this paper is organized as follows. Section 3.2.1 details the governing equation and initial conditions for these simulations. Results for three cases of collisional variation across the interface are presented in Section 3.2.2 for an Atwood number of 1/3. The impact of increasing the Atwood number to 2/3 is discussed in Section 3.2.3, and results are compared to the lower Atwood number cases, including growth rates and magnitudes of interface widening. An expansion of the particle energy-flux is presented in Section 3.2.4, leveraging the information contained in the distribution function to quantify the importance of kinetic models in these collisional regimes. Finally, Section 3.2.5 summarizes all simulation results and discusses the necessity of kinetic models to accurately model the RT instability in these conditions.

3.2.1 Problem Description

Simulations in this work are performed using the continuum-kinetic capabilities of the plasma simulation framework `Gkeyll` [45]. `Gkeyll` utilizes a discontinuous Galerkin discretization scheme [29, 30, 94] with serendipity basis [4] to evolve the Boltzmann equation [50, 63],

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f + \boldsymbol{g} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f = \left(\frac{\partial f}{\partial t} \right)_C , \tag{3.18}$$

where $f = f(\boldsymbol{x}, \boldsymbol{v}, t)$ is the particle distribution function defined in phase space, $\boldsymbol{g}$ is acceleration due to gravity, and the right-hand-side is the collision operator. Where a traditional fluid model assumes particles always follow a thermalized Maxwellian velocity distribution, the continuum-kinetic model allows the velocity space distribution function f to deviate

from Maxwellian. The collision operator relaxes the distribution function to Maxwellian and contains much of the physics that must be explicitly included in fluid models, like viscosity and thermal conduction. While a full nonlinear Fokker-Planck collision operator [101] is required to accurately capture the physics of small-angle collisions between charged species, including collision-induced velocity space advection and diffusion and collision frequency that varies in velocity space as $1/v^3$, reduced collision models can be constructed to retain features relevant to the chosen problem. For example, the Dougherty or Lenard-Bernstein operator [52], explicitly includes velocity space advection and diffusion of the distribution function but utilizes a collision frequency that is constant in velocity space, overestimating the impact of collisions in the high-energy tail of the distribution. Collisions are modeled in this work by the Bhatnagar-Gross-Krook (BGK) operator [9],

$$\left(\frac{\partial f}{\partial t}\right)_C = \nu(f_M - f), \tag{3.19}$$

where ν is the collision frequency and f_M is a Maxwellian distribution constructed from moments of f. Where previous work utilized the BGK model with a collision frequency that was constant spatially, collision frequency of a single species generally varies spatially with number density and thermal velocity as [15],

$$\nu \propto \frac{n}{v_{th}^3}. \tag{3.20}$$

The BGK operator is well-suited to large-angle binary collisions between neutral species and is guaranteed to conserve particle number density, momentum, and energy when collision frequency is constant in velocity space. However, similarly to the Dougherty operator, the use of mean collision frequency results in an overestimation of collision frequency in the high-energy tail of the distribution.

Simulations in this work are 5-dimensional, with 2 physical space dimensions and 3 velocity space dimensions. Initial conditions are derived from hydrostatic equilibrium,

$$\boldsymbol{\nabla} p = -nm\boldsymbol{g}, \tag{3.21}$$

where p is pressure, n is number density, $m = 1.0$ is mass, and $g = 1.0$ is gravitational acceleration. Initial number density and pressure profiles are,

$$n(y) = \frac{n_0}{2} \tanh\left(\frac{\alpha y}{L_y}\right) + \frac{3}{2} n_0, \tag{3.22}$$

$$p(y) = -\frac{mgn_0}{2}\left[\frac{L_y}{\alpha} \ln\left(\cosh\left(\frac{\alpha y}{L_y}\right)\right) + 3y\right] + \frac{3}{2} n_0 T_0, \tag{3.23}$$

where $L_y = 1.0$ is half the domain length in y, n_0 and n_1 are the number density at the bottom and top of the domain respectively, and T_0 is an arbitrary integral constant chosen such that temperature and pressure remain positive in the domain. The width of the density gradient at the center of the domain is proportional to the constant α, which is set to 25 to ensure a small interface width relative to the domain size. The exact method of determining the bounds of the interface for calculating the growth rate is described in Section 4. Note that quantities at the lower boundary are denoted with a subscript 0, while those at the upper boundary are denoted with a subscript 1. This initial density profile corresponds to an Atwood number, $A = (n_1 - n_0)/(n_1 + n_0)$, of 1/3. Boundary conditions are periodic in x and static reservoir in y, where the boundary cells are a continuation of the initial conditions and do not evolve in time. Initial distribution functions are Maxwellian,

$$f(\boldsymbol{v}) = \frac{n}{(2\pi v_{th}^2)^{3/2}} \exp\left(-\frac{(\boldsymbol{v} - \boldsymbol{u})^2}{2v_{th}^2}\right), \tag{3.24}$$

where $\boldsymbol{u}$ is bulk velocity, and $v_{th} = \sqrt{T/m}$ is thermal velocity with temperature $T = p/n$.

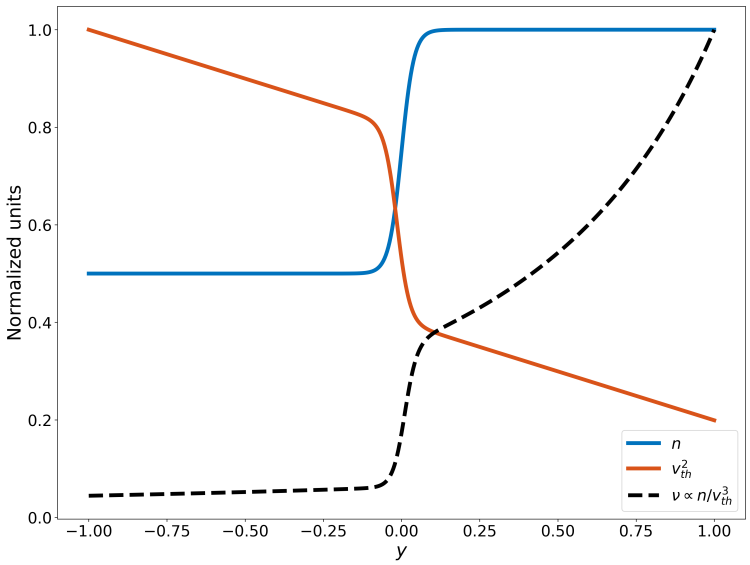

Figure 3.9: Initial conditions of number density, n, thermal velocity squared, v_{th}^2, and common collision frequency profile, n/v_{th}^3. Note the steep increase in collision frequency approaching the upper boundary. A collision frequency profile proportional to a power of number density maintains similar variation across the interface while avoiding an excessively small time-step for dynamics away from the interface.

The RT instability is seeded by a single-mode sinusoidal perturbation with wavenumber k applied to the y-direction bulk velocity, u_y, according to

$$u_y = -0.1 v_{th,c} \cos(kx) \exp\left(-\frac{y^2}{2y_r^2}\right), \tag{3.25}$$

where $k = \pi/(2L_x)$, $v_{th,c}$ is initial thermal velocity at the center of the domain, $L_x = 0.5$ is half the simulation domain length in x, and $y_r = L_y/10$ is the characteristic decay length for the perturbation.

For the hydrostatic equilibrium chosen in this work, the collision frequency profile defined in Eq. 3.20 increases dramatically near the upper boundary due to the temperature gradient, as shown in Figure 1.

Therefore, to capture similar collisional variation across the interface while maintaining a

reasonable time-step, collision frequency in this work varies according to

$$\nu = \nu_0 n^\beta, \tag{3.26}$$

where ν_0 is an arbitrary scaling constant and β is a parameter chosen to adjust the degree of variation of collisionality across the interface. As a result of this simplification, collision frequency near the upper boundary will be underestimated by up to a factor of approximately 5. This underestimation is acceptable because the high collisionality chosen in this upper boundary is already in a fluid-like regime, and an even higher collisionality would asymptote to a fluid regime without significantly impacting the results at the Rayleigh-Taylor-unstable interface. Relative collisionality is quantified by the Knudsen number $Kn = \lambda_m / L_x$ where $\lambda_m = v_{th}/\nu$.

3.2.2 Effect of Varying Collisionality

Three cases of varying collisionality are selected to cover regimes previously studied with constant collisionality [97]. Table 1 shows the collisional variation across the interface for each case. Note with the same equilibrium profile, the case of constant collisionality with Kn of 0.1 exhibits no RT instability growth, Kn of 0.01 showed diffusion of the interface and limited instability growth, and Kn of 0.001 showed limited interface diffusion and growth similar to an ideal fluid result.

Case	Lower Kn	Upper Kn
1	0.1	0.01
2	0.01	0.001
3	0.1	0.001

Table 3.2: Knudsen numbers defined at the lower and upper ends of the Rayleigh-Taylor interface for each of the three cases.

Evolution of the number density to the final time of 3 classical RT growth periods, $3\tau_{\mathrm{RT}} = 3/\sqrt{kgA}$, for each case is presented in Figure 3.10. Case 1 exhibits no instability growth and is dominated by diffusion of the interface, similar to the constant collisionality case with $Kn = 0.1$. In case 2, there is early-time diffusion of the interface as the characteristic bubble and spike structures of the RT instability begin to form, but late in time, diffusion appears to be limited as instability growth becomes dominant. At the end time (b4), there is clear development of the RT instability, and the average center position between the bubble and spike has moved upwards due to diffusion in the lower, less collisional region. Additionally, the interface has variable width, with the interface appearing thinner at the peaks of the bubble and spike than in the intermediate vertical regions. As will be discussed in Section 3.2.3, interface width varies between the bubble and spike, and Fig. 3.16 highlights the evolution of interface width in time. In case 3, the factor of 100 variation in collisionality between regions drives the interface upwards immediately, with diffusion being strongly biased on the lower side of the interface. Instability growth is greatly limited relative to case 2 as the lower collisionality region damps the growth of the downward spike. This is most clearly seen comparing the end time number densities of case 2 (b4) and case 3 (c4), where the bubble reaches approximately the same position at $y \approx 0.5$, while the spike in case 3 is well above $y = 0$ compared to $y \approx -0.25$ in case 2. Variation in interface width in case 3 also appears to be less substantial than in case 2, likely due to the limited instability growth. Figure 3.11 shows the evolution of temperature with time. The temperature distribution is nearly identical to the number density distribution at each time, though compressibility effects are less prevalent due to heat flux, yielding a smoother profile.

In a similar manner to [89] and [48], non-equilibrium kinetic effects can be quantified using a non-Maxwellian density, n_N, constructed from the distribution function and its associated

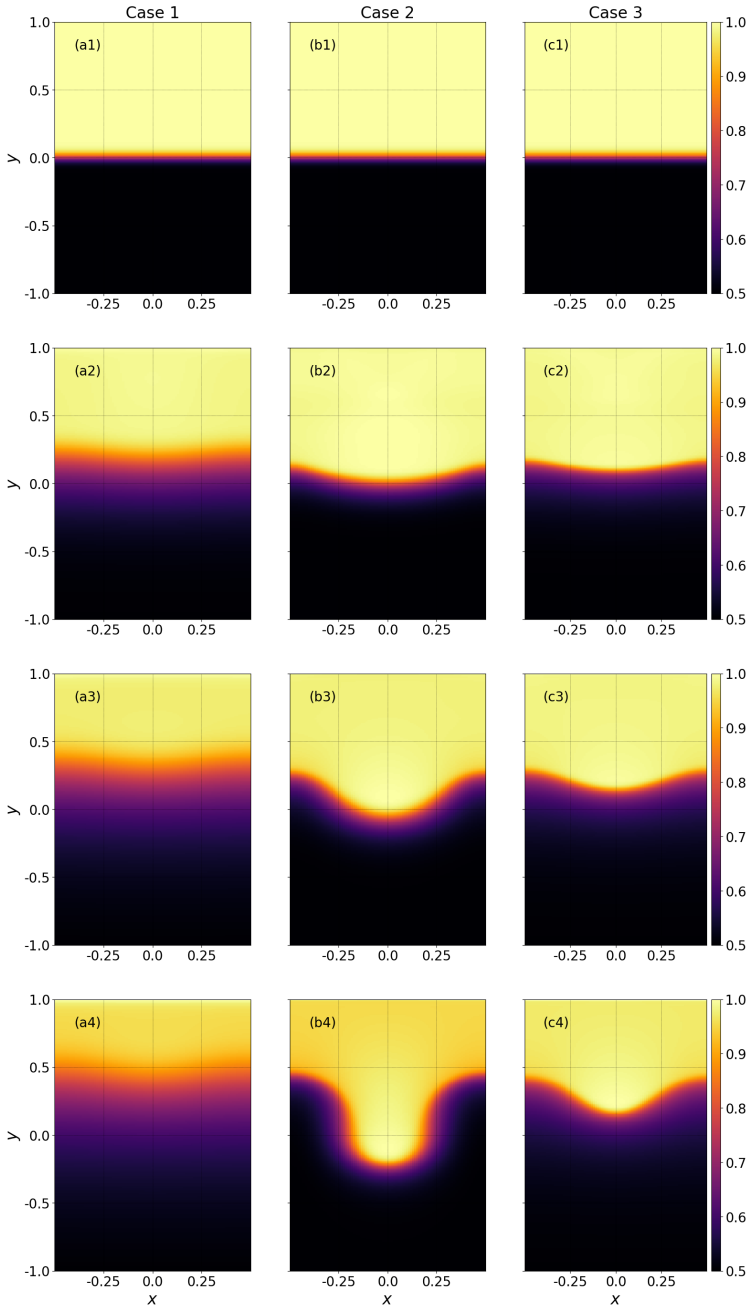

Figure 3.10: Evolution of number density for three Rayleigh-Taylor instability simulations with Knudsen number varying from: 0.1 to 0.01 (a, left), 0.01 to 0.001 (b, center), 0.1 to 0.001 (c, right). Knudsen numbers are calculated at the lower and upper ends of the interface, respectively. Note the lack of instability growth in case 1, fluid-like growth in case 2, and damped growth in case 3.

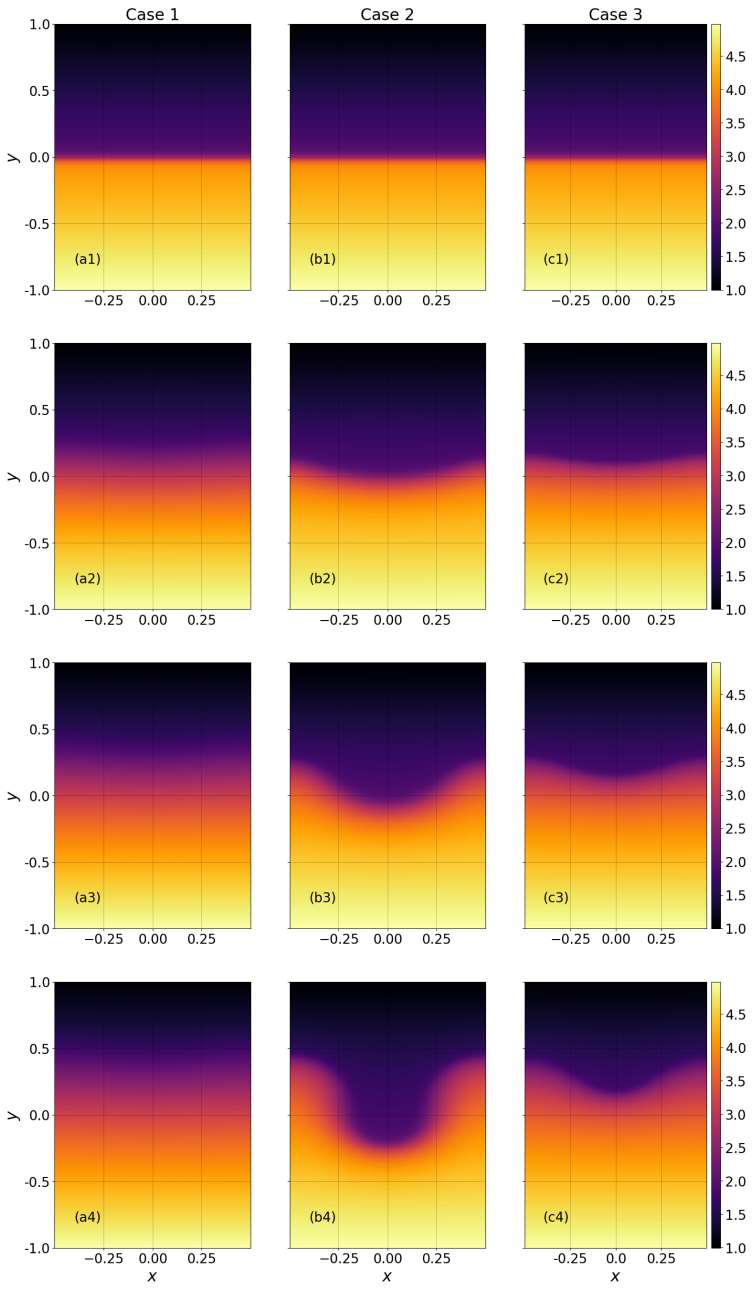

Figure 3.11: Evolution of v_{th}^2 for three Rayleigh-Taylor instability simulations with Knudsen number varying from: 0.1 to 0.01 (a, left), 0.01 to 0.001 (b, center), 0.1 to 0.001 (c, right). Knudsen numbers are calculated at the lower and upper ends of the interface, respectively. Note the magnitudes of v_{th}^2 remain stable in time, without the compressive effects seen in Figure 3.10.

Maxwellian, f_M as,

$$n_N(\boldsymbol{x}) = \int |f_M(\boldsymbol{x}, \boldsymbol{v}) - f(\boldsymbol{x}, \boldsymbol{v})| \mathrm{d}^3\boldsymbol{v}. \tag{3.27}$$

This diagnostic has units of density and can be interpreted as the density of non-Maxwellian distribution function. Figure 3.12 shows the distribution of n_N/n, the fraction of non-Maxwellian distribution, at the final time $3.0\tau_{\mathrm{RT}}$. In each case, n_N/n has higher magnitudes

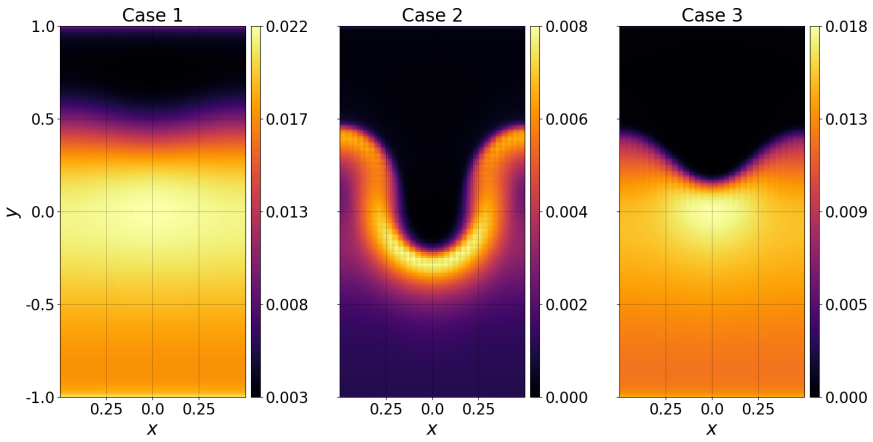

Figure 3.12: Density of non-Maxwellian distribution normalized to local number density, n_N/n, for each case at time $3.0\tau_{\mathrm{RT}}$. Around the interface in case 2 and in regions of low collision frequency in cases 1 and 3, n_N/n reaches maximum value.

in the regions of lower collisionality, and global magnitudes of n_N are small relative to number density, on the order of 1%. There is correspondence between collision frequency and n_N/n because lower collision frequencies will not be able to thermalize the distribution function as quickly. The maximum values of n_N/n are similar in cases 1 and 3, as expected due to the lower regions having the same collision frequency. Similarly, the minimum values of n_N/n approach 0 in cases 2 and 3, as the highly-collisional regions in those cases are similar to an ideal fluid-like regime. Non-Maxwellian density reaches its maximum around the interface in each case, implying the presence of kinetic effects around the bubble and spike of the RT instability. The peaks of the bubble and spike do not have equal magnitudes of n_N/n; the center of the spike is the absolute maximum of n_N/n in both cases 2 and 3. This is likely

connected to the damping of downward instability growth in the low-collisionality lower region due to diffusion.

3.2.3 Effect of Varying Atwood Number

Previous simulations have focused on varying collision frequency with a given equilibrium profile and Atwood number. Atwood number is known to vary greatly across astrophysically-relevant regimes [36], so the effect of a different Atwood number on RT instability growth is worth investigating. For these simulations, A is increased to 2/3 with a similar equilibrium profile by adjusting equations 6 and 7 while maintaining the same collision profile. Figure 3.13 shows the evolution of number density to the same normalized time $3.0\tau_{\mathrm{RT}}$ for the two cases where the RT instability develops with Atwood number of 2/3. Note the fluctuations in density early in time are caused by waves launched early in time from the initial perturbation in bulk velocity and appear to be more significant in magnitude relative to those in the lower Atwood number cases. Relative to the $A = 1/3$ cases, there is less upward movement of the interface due to diffusion, yet formation of the downward spike is still damped. In both cases, the bubble does not move as far upward as the corresponding $A = 1/3$ cases, but the spike reaches further downward, yielding larger total instability amplitude. The difference in spike position is especially clear when comparing the case 3 simulations, as the spike in the lower Atwood number case is in the upper half of the domain, while the spike remains around or below $y = 0$ in the higher Atwood number case. Normalized non-Maxwellian density n_N/n for these cases is shown in Figure 3.14. Spatial distributions of n_N/n follow similar patterns to the lower A cases. However, magnitudes of n_N/n for the higher A simulations are approximately twice those of the lower A cases. This is likely due to the proportionality of the kinematic viscosity and diffusion coefficient to thermal velocity 3.28 and the fact that the higher Atwood number cases are less dense,

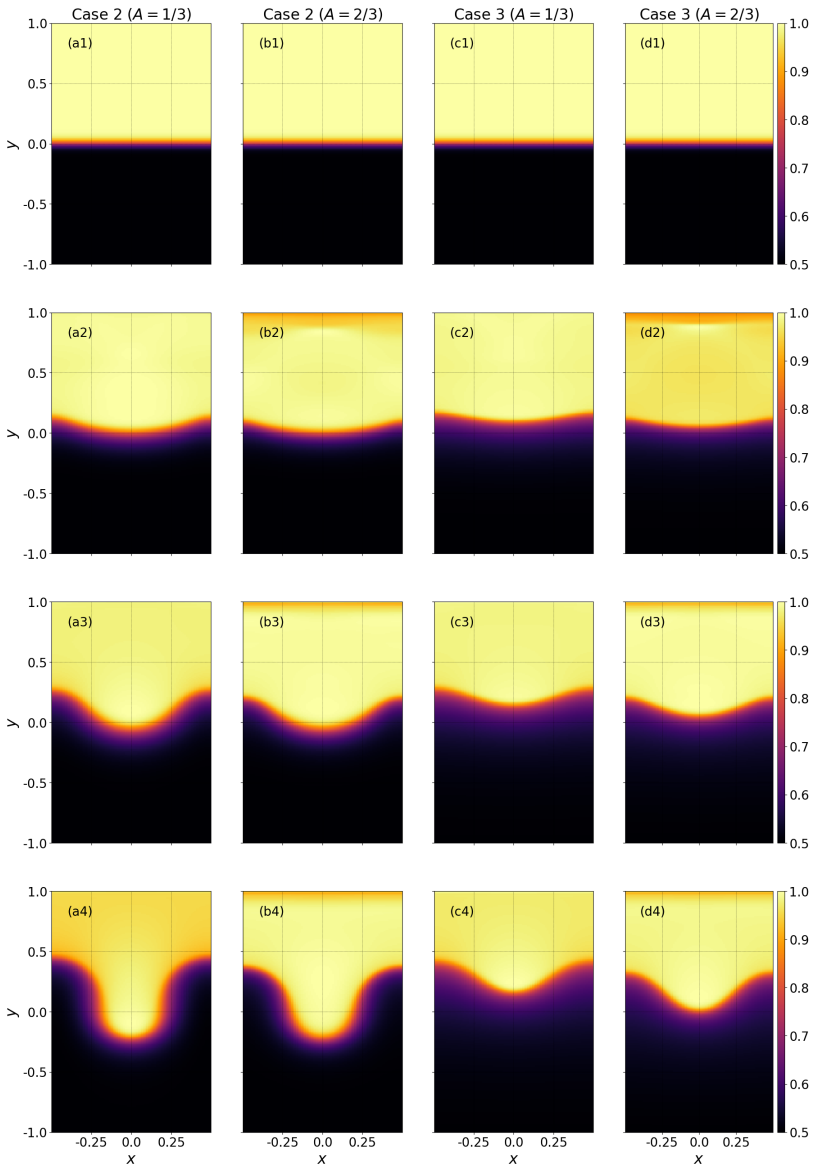

Figure 3.13: Evolution of number density for Rayleigh-Taylor instability simulations with Knudsen number varying from: 0.01 to 0.001 (left two columns) and 0.1 to 0.001 (right two columns). Knudsen numbers are calculated at the lower and upper ends of the interface, respectively. Results are included for Atwood numbers of 1/3 (a, c) and 2/3 (b, d). Relative to the lower Atwood number cases, in the 2/3 Atwood number cases, the bubble does not move as far upward, but the spike extends further down into the low-density region. There is also less upward movement of the interface due to particle streaming diffusion in the lower, less collisional region.

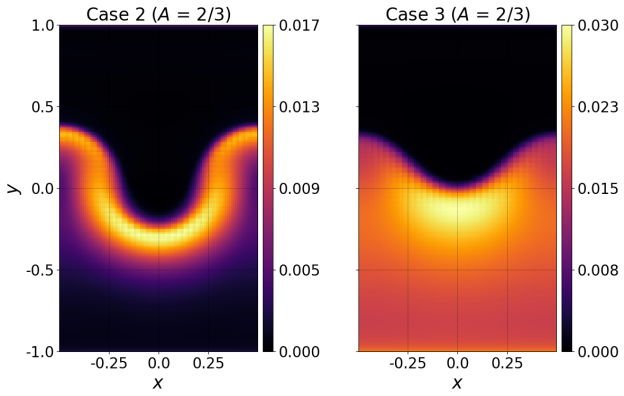

Figure 3.14: Density of non-Maxwellian distribution normalized to local number density, n_N/n, for cases 2 and 3 with Atwood number $A = 2/3$ at time $3.0\tau_{\mathrm{RT}}$. Similar to the $A = 1/3$ cases, the peaks in n_N/n appear at the interface in case 2 and in the lower collision frequency region in case 3.

yielding a higher temperature for the same hydrostatic equilibrium.

Growth of the RT instability can be tracked in time by calculating the amplitude between the center of the interface at the peaks of the bubble and spike. In order to calculate the amplitude, the y-location of a reference density, chosen as the initial value of n at the center of the interface, is determined for each frame at $x = 0$ and $x = L_x$ for the spike and bubble, respectively. As will be discussed at the end of this section, compressibility leads to buildup of density on either side of the interface, but the chosen reference density generally remains at the center of the interface in time. The vertical displacement between those two locations is taken as the amplitude, h.

Figure 3.15 shows the logarithm of the amplitude for cases of constant and varying collisionality that exhibits RT instability growth. The classical RT growth rate, $\gamma_{\mathrm{RT}} = 1/\tau_{\mathrm{RT}} = \sqrt{kgA}$ is not expected to capture the kinetic dynamics included in these simulations, so a growth rate is calculated similarly to [34, 102], including viscous and diffusive effects,

$$\gamma_0 = \sqrt{kgA + \nu_v^2 k^4} - (\nu_v + \xi)k^2, \tag{3.28}$$

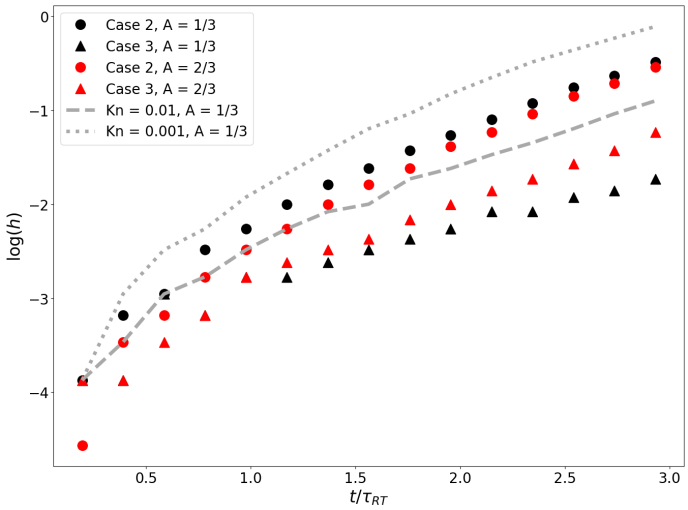

Figure 3.15: Rayleigh-Taylor amplitude growth. The most fluid-like kinetic case (grey dotted line) has the greatest overall instability growth. Case 2 for both Atwood numbers sits between the fluid-like and intermediate (dashed grey line) cases, while case 3 exhibits the least growth.

where $\nu_v = v_{th,c}\lambda_m/2$ is the kinematic viscosity, and $\xi = \nu_v$ is the diffusion coefficient. Dynamic diffusion effects may also be included in this calculation to give a time-dependent growth rate, but this is excluded for this calculation for simplicity. Dynamic diffusion dominates early in time, before the linear phase, [80] when the interface is diffusing with no instability growth, leading to nonlinear interface amplitude, so early-time points are excluded from the linear fit. Table 2 shows the growth rates for each case as calculated from the linear fit, γ, and from equation 11, γ_0. Note the classical growth rates are $\gamma_{\mathrm{RT},1} = 1.023$

Case	γ	γ_0	γ/γ_0
Kn = 0.01 (A = 1/3)	0.789	0.911	0.866
Kn = 0.001 (A = 1/3)	0.903	1.012	0.892
Case 2 (A = 1/3)	0.879	0.989	0.889
Case 3 (A = 1/3)	0.574	0.981	0.585
Case 2 (A = 2/3)	1.396	1.408	0.991
Case 3 (A = 2/3)	1.130	1.373	0.823

Table 3.3: Growth rates of the Rayleigh-Taylor instability as calculated from linear fit, γ, and from theory accounting for diffusion and viscosity (Eq. 3.28), γ_0.

and $\gamma_{\mathrm{RT},2} = 1.447$ for $A = 1/3$ and $2/3$, respectively.

Theoretical growth rates γ_0 increase with average collision frequency, approaching the classical growth rates $\gamma_{\mathrm{RT},1}$ and $\gamma_{\mathrm{RT},2}$. Regardless of Atwood number, as average collisionality increases, i.e. Case 3 to Case 2 and $\mathrm{Kn} = 0.01$ to $\mathrm{Kn} = 0.001$, the ratio γ/γ_0 increases as the calculated growth rate approaches the theoretical result. For the cases of spatially-varying collisionality, the degree to which the ratio increases with average collision frequency is dependent on the Atwood number. The ratio γ/γ_0 for the $A = 1/3$ cases increases from 0.585 to 0.889 from Case 3 to Case 2, an increase in agreement of approximately 34%. Similarly, the $A = 2/3$ cases increase from 0.823 to 0.991, an increase of approximately 17%. Therefore, for the same given collisionality profile, the agreement of the calculated growth rate with the theoretical growth rate increases with Atwood number, but as collisionality increases, the relative increase in agreement is greater in the lower Atwood number simulations.

A primary distinguishing factor between simulations is the magnitude of interface diffusion, which can be quantified by tracking the width of the interface in time. Traditionally in fluid simulations, the moving interface can easily be tracked in time using fluid mass fraction or by following the constant density values that define the bounds of the interface. However, the number density profile is not constant around the interface in these simulations due to compressibility, making it difficult to exactly define and track the interface. Therefore, in an approach similar to [73], the interface is instead tracked using the temperature, which remains relatively smooth in time because of heat fluxes smoothing perturbations. The bounds of the interface are determined from the initial conditions by first retrieving the location where n reaches 99% of the global maximum as the upper bound and defining that position as $y = L_{\mathrm{int}}$. The initial upper and lower bounds of the interface are then located at $y = L_{\mathrm{int}}$ and $y = -L_{\mathrm{int}}$, respectively. Then v_{th}^2 is evaluated at each of these points to determine the reference values that are used to track the interface. For each data frame, the reference values of v_{th}^2 are matched at $x = 0, L_x$ for the spike and bubble, respectively.

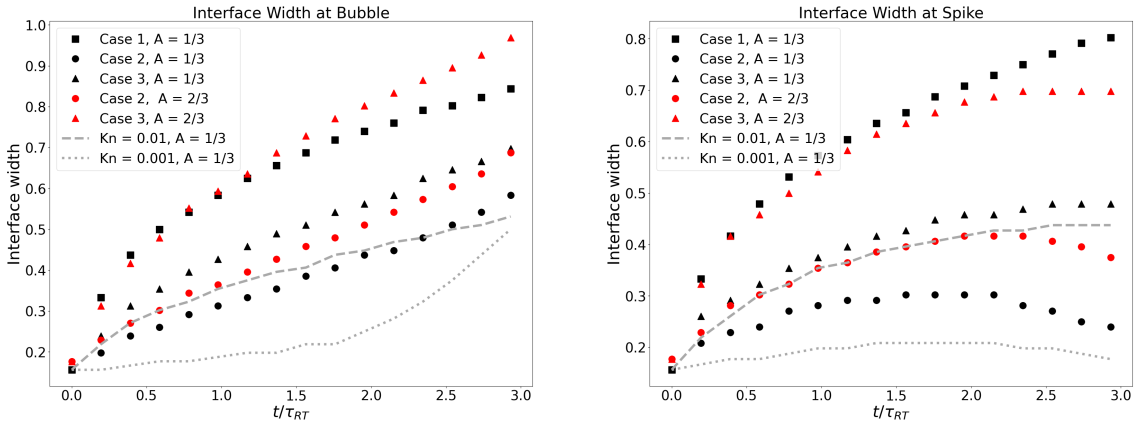

Figure 3.16: Evolution of the width of the interface between regions as measured at the location of the peak of the bubble (left) and spike (right). Higher Atwood number cases exhibit more interface diffusion than corresponding cases with lower Atwood numbers. The interface width at the bubble is in general larger than at the spike due to diffusion in the low collision frequency region diffusing the lower end of the interface away from the upward moving spike. Note the increase in constant $Kn = 0.001$ (gray dotted line) is due to secondary instabilities.

Figure 3.16 shows the evolution of the interface width as calculated at the peak of the bubble and spike. Differences in interface width between the bubble and spike can be attributed to the difference in collision frequency between the upper and lower regions. In the upper region, collision frequency is greater, so it is expected that the upward moving bubble exhibits more ideal fluid-like behavior, i.e. faster instability growth and less diffusion. Conversely, the lower region with lower collision frequency is dominated by diffusion and damps instability growth, as can be most easily seen in the final number density distributions of case 3. Therefore, diffusion is in general biased toward the less collisional lower region. As the bubble and upper end of the interface move upward, the lower end of the interface diffuses downward, resulting in generally larger interface width relative to the spike, which pushes into the lower region as the interface diffuses in the same direction. This can be seen by comparing case 1 with $A = 1/3$ (black squares) and case 3 with $A = 2/3$ (red triangles) in Figure 3.16, which have the same collisionality in the lower region. At the bubble location, the interface width of case 3 is increased by the upward movement of the bubble, yielding a wider interface than

case 1, which is effectively pure diffusion. Conversely, the spike in case 3 moves slightly below $y = 0$, somewhat offsetting the interface width gain due to diffusion, so case 1 has a wider interface at that location. Additionally, in some cases the lower end of the interface diffuses enough that it reaches the lower boundary, so the interface width late in time reaches a maximum. If the domain was larger, diffusion would likely continue to follow the early time trend and increase as a similar rate.

Constant collisionality cases are included as dashed lines in Figure 3.16. The most collisional and fluid-like case, constant $Kn = 0.001$, has substantially less interface diffusion than any of the other cases and has effectively a constant interface width in time. Note the increase at the bubble late in time is an artifact due to the development of secondary instabilities. At the bubble, the intermediate constant $Kn = 0.01$ case closely matches case 2 for both Atwood numbers, which have a Knudsen number of 0.01 in the upper region. However, at the spike, the intermediate constant Kn case still matches the $A = 2/3$ case well, while the lower A case deviates from both, exhibiting much less interface widening. Cases with $A = 2/3$ show greater interface width than the corresponding case with $A = 1/3$, likely due to the higher thermal velocity and therefore diffusion coefficient. Comparable fluid simulations with varying viscosity exhibit no interface diffusion, similar to the fluid-like $Kn = 0.001$ case [7, 116, 117]. Viscous fluid simulations also exhibit stabilization of short-wavelength modes, an effect that is not seen in these large single-mode kinetic simulations. Stabilization in small RT instability modes in kinetic simulations with finite collision frequency is expected to differ from previous fluid simulations, as Braginskii viscosity only applies in the high-collisionality limit.

3.2.4 Expansion of the Particle Energy-Flux

Higher moments of the distribution function can also be utilized to characterize the impact of kinetic effects. Beginning with the laboratory-frame second and third moment,

$$\mathcal{P}_{ij} = m \int v_i v_j f \mathrm{d}^3\boldsymbol{v}, \tag{3.29}$$

$$\mathcal{Q}_{ijk} = m \int v_i v_j v_k f \mathrm{d}^3\boldsymbol{v}. \tag{3.30}$$

As in Wang et al. [137], by defining $w_i = v_i - u_i$, Eq. 3.30 can be expanded and tensor contracted to get the particle energy-flux (using Einstein's summation convention),

$$\frac{1}{2}\mathcal{Q}_{iik} = \underbrace{\frac{5}{2}u_k p + \frac{1}{2}mnu_k\boldsymbol{u}^2}_{\mathrm{I}} + \underbrace{q_k + u_i\Pi_{ik}}_{\mathrm{II}}, \tag{3.31}$$

where

$$q_k = \frac{1}{2}m \int w_i w_i w_k f \mathrm{d}^3\boldsymbol{v}, \tag{3.32}$$

is the heat flux vector in the gas frame, and the stress tensor Π_{ij} is related to the pressure tensor,

$$P_{ij} = m \int w_i w_j f \mathrm{d}^3\boldsymbol{v}, \tag{3.33}$$

by $\Pi_{ij} = P_{ij} - p\delta_{ij}$ with scalar pressure $p = P_{ii}/3$. The pressure tensor is also related to the second moment by $\mathcal{P}_{ij} = P_{ij} + mnu_i u_j$. The four terms in the expanded particle energy-flux, Eq. 3.31, can be grouped into terms that arise from the Maxwellian parts of f, group I, and from the non-Maxwellian parts, group II. Comparing the magnitudes of group I and group II terms quantifies the relative contributions to the total energy-flux of effects that would not be captured by pure fluid models. Figure 3.17 shows the y-components of the expanded energy-flux for cases 1, 2, and 3 for $A = 1/3$, normalized to $n_0 v_{th}^3$. Beginning with case

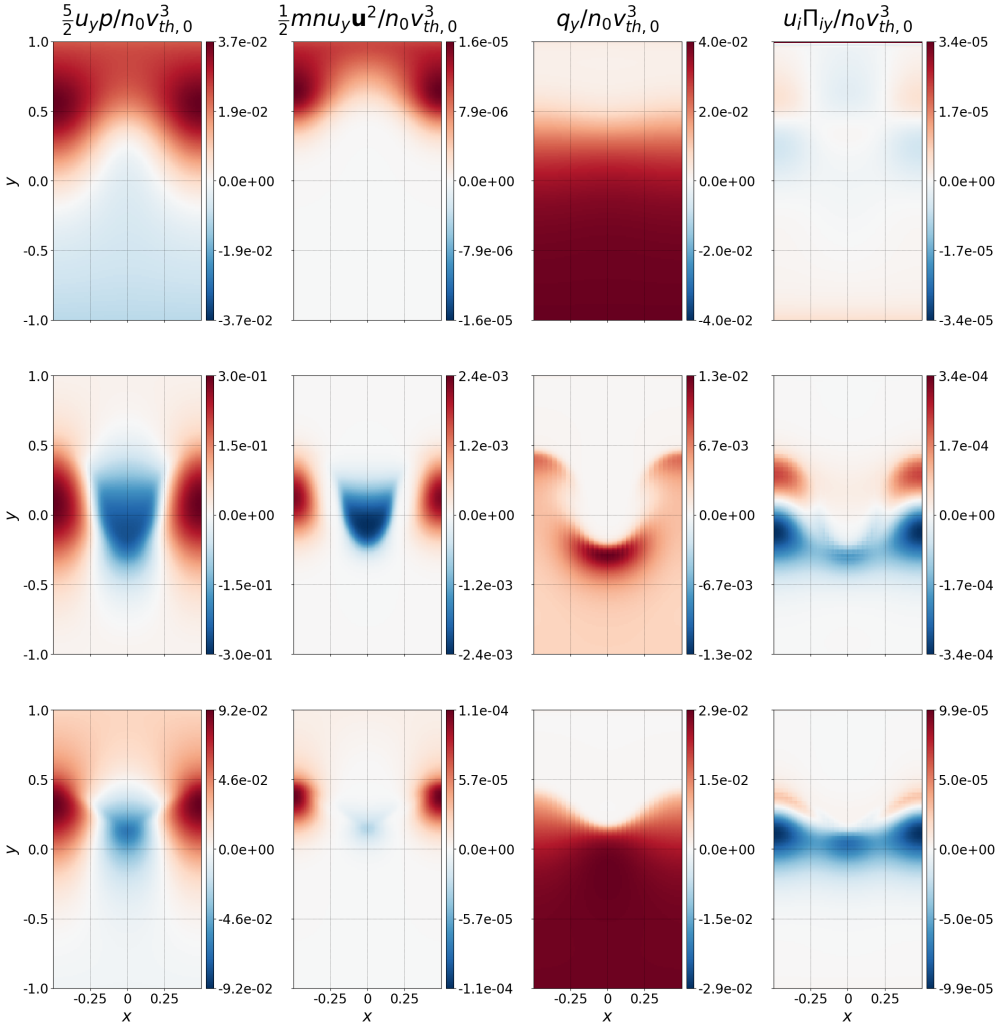

Figure 3.17: Terms in the expanded particle energy-flux, Eq. 3.31, for cases 1 (top), 2 (center), and 3 (bottom). Ideal terms (left two columns) are concentrated in the highly-collisional region in case 1 and in the bubble and spike in cases 2 and 3. The non-ideal terms (right two columns) are concentrated in the low-collisional regions in all cases with extrema present around the interface in cases 2 and 3.

1 (Figure 3.17, top row), the ideal terms (left two columns) reach maximum values in the more collisional upper region, while the dominant non-ideal term, q_y reaches its maximum in the low-collisional lower region. All energy-flux terms in case 2 (Figure 3.17, middle row) have structure corresponding to the RT instability bubble and spike. The ideal terms are concentrated in the bubble and spike, with the negative and positive regions in the

spike and bubble, respectively. However, the non-ideal terms are concentrated in the lower collisionality low-density fluid. The heat flux, q_y, similar to n_N/n, reaches a global maximum at the center of the spike, with a local maximum at the tip of the bubble. Whereas the stress term, $u_i\Pi_{iy}$, reaches its maximum magnitude in the bulk of the bubble rather than the tip, where it also flips sign. This is likely due to dominance of tangential stress terms in the lower region, compared to dominant positive vertical flux at the tip of the bubble. Case 3 (Figure 3.17, bottom row) shows similar characteristics to both cases 1 and 2, where the ideal terms are largely concentrated in the RT instability structures and the more collisional upper region, while the non-ideal terms are concentrated in the less collisional lower region. The vertical asymmetry in bubble and spike formation is easily seen through the energy-flux, as the values reached by the ideal terms within the bubble are substantially greater than those reached in the spike. Differences between the pressure term, $5/2u_yp$, and the inertial term, $1/2mnu_y\boldsymbol{u}^2$, at the spike position show energy tends to go into compression rather than downward movement of the gas. Relative importance of the ideal and non-ideal terms can be quantified by taking the ratio of the averages of the absolute values of the non-ideal terms to the ideal terms,

$$\text{ratio} = \frac{\text{avg}(|q_k + u_i\Pi_{ik}|)}{\text{avg}(\left|\frac{5}{2}u_kp + \frac{1}{2}mnu_k\boldsymbol{u}^2\right|)}. \tag{3.34}$$

For case 1, the low collisional case dominated by kinetic non-ideal behavior, the ratio is approximately 1.952. Cases 2 and 3 are expected to have more relevant ideal term dynamics related to instability growth, and the ratios for those cases are 0.037 and 0.873, respectively. Case 3, while still dominated by the ideal terms, has a higher ratio than case 2 due to the lower average collisionality.

A similar analysis can be applied to the higher Atwood number cases, shown in Figure 3.18. In case 2 (Figure 3.18, top row), each term of the expanded particle energy flux is of the

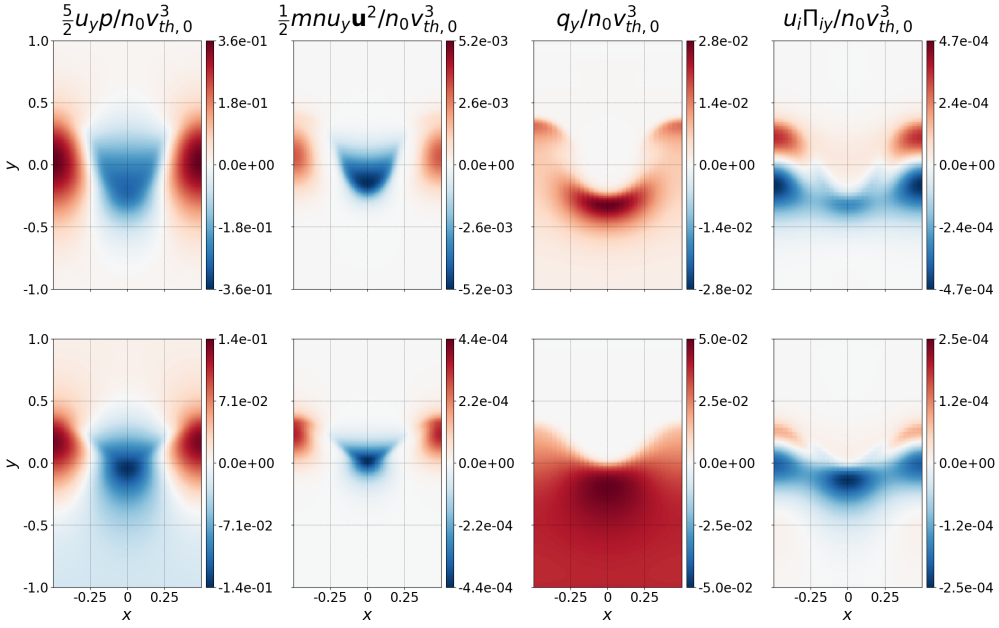

Figure 3.18: Terms in the expanded particle energy-flux, Eq. 3.31, for cases 2 (top) and 3 (bottom) with Atwood number $A = 2/3$. Similar to the lower Atwood number cases, the ideal terms (left two columns) are concentrated in the bubble and spike, while the non-ideal terms (right two terms) are concentrated in the low-collisionality regions with extrema around the interface.

same order of magnitude as the lower A cases. However, one notable change is the increase in magnitude of the non-ideal terms by approximately a factor of 2, similar to the increase present in n_N/n. The ratio of ideal to non-ideal terms for this case is 17.81, a similar magnitude to the lower Atwood number case and still dominated by ideal behavior.

Case 3 (Figure 3.18, bottom row) has a ratio of 1.58 and shows substantial differences to the lower A case, primarily centered around the spike. Both ideal terms increase in magnitude as A increases, and the maxima reached in the spike are comparable to those in the bubble, whereas the lower A case shows much smaller magnitudes in each term in the spike relative to the bubble. This is especially true in the inertial term $mnu_y \boldsymbol{u}^2/2$, where the ratio of peak magnitude in the bubble to the spike is 0.789 in the $A = 2/3$ case, compared to 3.389 in the $A = 1/3$ case. Similar to case 2, the non-ideal terms both approximately double in

magnitude, contributing to the similar increase in n_N/n. The non-ideal terms of the particle energy-flux are purely kinetic effects, and spatial variation of collision frequency contributes further to non-ideal behavior by varying effects such as diffusion and viscosity. Specifically, the presence of these non-ideal terms in the expansion of the third moment of f imply the presence of skewness in the distribution function that would not be captured by fluid models.

3.2.5 Summary of Results

Simulations of the single-mode Rayleigh-Taylor (RT) instability in 2X3V (2 spatial dimensions, 3 velocity space dimensions) are successfully conducted using a continuum-kinetic model implemented within `Gkeyll` with a more realistic, spatially-varying collision frequency. Three cases are explored with an Atwood number $A = 1/3$, enumerated in Table 1, covering regimes previously studied with spatially constant collisionality. Case 1 exhibits no Rayleigh-Taylor instability growth, despite the upper region being in the intermediate collisional regime that shows instability growth for previous constant collision frequency simulations. Case 2 varies between intermediate and highly-collisional regimes, and the RT instability growth is the most fluid-like of the three cases. The fully-varying case 3 that covers low and high collisional regimes exhibits limited RT instability growth with a high degree of diffusion in the low collisional region that results in an upward movement of the interface.

Variations from Maxwellian distribution function can be quantified by taking the zeroth velocity moment of the difference between the local distribution function and a Maxwellian constructed from its moments to calculate the non-Maxwellian density, n_N. In all cases, the normalized non-Maxwellian density, n_N/n, is localized to regions of low collisionality. Additionally, there are extrema in n_N/n around the interface in cases 2 and 3, where the instability develops, likely related to compressibility effects. Higher relative magnitudes in

n_N/n at the spikes than the bubbles of cases 2 and 3 are connected to the damping of the downward growth of the spike.

Increasing the Atwood number from 1/3 to 2/3 for cases 2 and 3 yields larger instability amplitude stemming from greater downward movement of the spike into the low density region. Diffusion in the low collisionality region also moves the interface upwards to a lesser degree than the $A = 1/3$ cases. Non-Maxwellian density follows the same trends as the lower Atwood number cases, though the magnitudes are higher in the higher Atwood number cases. This is likely due to the lower collision frequency and globally lower density and higher temperature in the latter cases and the proportionality of kinematic viscosity and diffusion coefficient to thermal velocity.

Growth rates calculated from tracking the instability amplitude generally match the modified theoretical growth rates, γ_0, that include the effects of diffusion and viscosity. Case 2, which has the highest average collision frequency, agrees well with γ_0 for each Atwood number. However, case 3, which has the greatest collisional variation, matches the theoretical prediction much more closely for $A = 2/3$ than for $A = 1/3$, implying a sensitivity to Atwood number for the highly collisionally varying cases.

As the instability evolves, the width of the interface changes in time due to free-streaming particle diffusion. In a traditional inviscid fluid model, numerical diffusion, rather than a physical mechanism, is the primary source of interface widening, so the continuum-kinetic model is expected to capture the interface width evolution more accurately. Further, there is a difference between the interface width at the center of the spike and the center of the bubble. The lower, less-collisional region is the region where most diffusion occurs in the domain, so as the bubble moves upward, the lower end of the interface diffuses away downward, spreading the interface. At the center of the spike, the upper end of the interface moves downward as the instability grows, limiting the spread of the interface due to diffusion. For

both cases, the higher A runs show greater widening of the interface than the corresponding lower A runs. Case 2 for both Atwood numbers exhibits similar interface widening to the spatially constant intermediate collisionality $Kn = 0.01$ case. With $A = 2/3$, case 3 most closely resembles case 1 with $A = 1/3$, which is effectively pure diffusion, while case 3 with $A = 1/3$ is similar to case 2 and $A = 2/3$, further showing the sensitivity of case 3 to Atwood number.

An expansion of the particle energy-flux is utilized to highlight the presence of kinetic effects and the presence of non-Maxwellian dynamics. While the dominant components of the energy-flux come from the ideal terms arising from the Maxwellian components of the distribution function, there are concentrations of non-ideal terms that only appear when using a distribution function that can deviate from Maxwellian. In general, the non-ideal components of the energy-flux are primarily present in regions of low collisionality, i.e. the lower region, and are inversely proportional in magnitude to the collision frequency. The global extrema of each term occur around the interface when the RT instability grows, with the maximum magnitude occurring in the spike, where corresponding density compression and maxima in n_N/n are also present. Non-ideal terms of the particle energy-flux, the laboratory-frame third moment of the distribution function, implies the presence of skewness away from Maxwellian, which would not be inherently captured by fluid models. Additionally, simple high collisional limit models of viscosity do not account for intermediate and low collisionality effects.

While fluid models generally are well-suited to study the RT instability, collisional effects can be relevant in astrophysical and laboratory plasmas. Previous continuum-kinetic simulations with constant collision frequency demonstrate the existence of intermediate regimes that are not accessible to traditional fluid models. Simulations with spatially-varying collision frequency offer further variations from the high-collisionality fluid limit, emphasizing the

necessity of kinetic models to capture dynamics that would be missed with traditional fluid models. Further simulations are necessary to expand a kinetic study into two-component plasmas with self-consistent electromagnetic fields both with and without collisions and externally applied fields. Introducing such complications adds to the number of relevant spatial and temporal scales in the system, increasing the likelihood that kinetic physics will be relevant.

3.3 Summary of Chapter 3

In this chapter, we demonstrated an application of the discontinuous Galerkin method introduced in Chapter 2 to the Vlasov-BGK model from Chapter 1. While the RT instability is generally studied with ideal fluid models, we note that the ubiquity of the instability implies regimes exist where kinetic effects may be relevant, an idea that is backed up by the literature in astrophysical and inertial fusion contexts. We began by studying the RT instability with neutral species by performing kinetic simulations with collision frequency that was spatially constant. Three collisionality regimes were studied: low-collisional, fluid-limit, and intermediate. The primary finding from the first set of simulations is that there is an intermediate regime between the low-collisional regime, where no RT instability grows, and the fluid-limit, where RT instability behavior is well-described by ideal fluid models. This intermediate regime exhibits RT instability growth that is distinct from the fluid result: the interface between fluids diffuses as the instability grows and growth rates are slowed compared to the fluid case. Building from this result, we then explored allowing the collision frequency to vary spatially as a function of density to recover a more physical result than the spatially constant case. By adapting the three cases from the constant collisionality simulations, we arrived at three comparable cases with varying collisionality: low to

intermediate (case 1), intermediate to high (case 2), and low to high (case 3), where the minimum collision frequency is at the bottom of the domain and increases with density to the top of the domain. While the average collision frequencies were similar to the constant collisionality cases, the RT instability behavior was substantially altered. In the cases where there was instability growth, there is significant upward movement of the interface driven by free-streaming diffusion in the lower region. The net effect is a time-dependent widening of the interface that varies in the x-direction, a phenomenon that does not appear in fluid models, as interface diffusion in a fluid model is generally the result of numerical diffusion rather than a physical process. Increasing the Atwood number yields greater amplitude growth and reduced upward interface movement.

Application of the continuum-kinetic method to hydrodynamic instability simulations has demonstrated the importance of capturing collisional physics in certain regimes. While fluid methods typically work well enough for the RT instability, we have shown that conditions exist where collisional dynamics are relevant while still maintaining the overall hydrodynamic behavior. The results we discussed compared the simple BGK model to traditional fluid models, and the differences in behavior are significant. However, these simulations considered a simple setup of neutral particles experiencing a single body force and interacting through collisions. As the system of interest gets more complex, we need a more accurate collision operator to capture the relevant physics. In the final chapter, we describe the process of implementing one of the most accurate collision operators, the Fokker-Planck operator introduced in Chapter 1, in `Gkeyll`.

Chapter 4

Discontinuous Galerkin Discretization of the Rosenbluth/Fokker-Planck Operator

If one wishes to model collisional kinetic physics in a plasma, there are few options more accurate than the full nonlinear Fokker-Planck collision operator. As shown in Chapter 1, the only major assumption included in the derivation of the FPO is that the plasma is weakly coupled, $\ln \Lambda = \ln(b_{max}/b_{min}) = \ln(\lambda_D/b_{min}) \gg 1$. Weakly coupled plasmas generally have many particles within the Debye sphere, as $\Lambda \sim n^3\lambda_D$. As the Coulomb force is long-range, the majority of collision events occur between the many particles at distances approximately one Debye length, λ_D, away from a particle of interest, which is the maximum distance beyond which electric forces are shielded by collective particle movement in the plasma [54]. Collisions at large distances result in a small deflection to the velocity vector, and the dominance of the effect of many of these small-angle collisions within the Debye sphere is the assumption used to arrive at the FPO.

Weakly coupled plasmas are generally hot and diffuse, a description matching many of the applications we would hope to apply these models to, e.g. tokamak fusion reactors, the solar wind, and the ionosphere [41]. Why then do we not always use the FPO to model collisions when investigating behavior in these regimes? The complication is made clear by writing

the FPO for a single species, (in advection-diffusion form here)

$$C[f(\mathbf{z},t)] = -\frac{1}{2}\nabla_{\mathbf{v}} \cdot (\mathbf{a}(\mathbf{z},\mathbf{t})f(\mathbf{z},t) - \mathbf{D}(\mathbf{z},\mathbf{t}) \cdot \nabla_{\mathbf{v}} f(\mathbf{z},t)), \tag{4.1}$$

where we have dropped the species indices, and substituting in the Rosenbluth potentials for the drag coefficient and diffusion tensor,

$$\mathbf{a} = 2\Gamma\nabla_{\mathbf{v}} H \tag{4.2}$$

$$\mathbf{D} = \Gamma_s \nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} G, \tag{4.3}$$

before writing the potentials in their integral forms,

$$H(\mathbf{v}) \equiv \int \frac{1}{|\mathbf{v}-\mathbf{v}'|} f(\mathbf{v}')\, d^3v' \tag{4.4}$$

$$G(\mathbf{v}) \equiv \int |\mathbf{v}-\mathbf{v}'| f(\mathbf{v}')\, d^3v'. \tag{4.5}$$

Making these substitutions, it becomes clear that the FPO is, in fact, a complex nonlinear integro-differential equation for the distribution function defined in six-dimensional phase space, $f(\mathbf{z},t)$,

$$\begin{aligned} C[f(\mathbf{z},t)] = -\frac{\Gamma}{2}\nabla_{\mathbf{v}} \cdot \Bigg\{ 2f(\mathbf{z},t)\nabla_{\mathbf{v}} \left(\int \frac{1}{|\mathbf{v}-\mathbf{v}'|} f_b(\mathbf{v}')\, d^3v' \right) - \\ \nabla_{\mathbf{v}} f(\mathbf{z},t) \cdot \left[\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} \left(\int |\mathbf{v}-\mathbf{v}'| f(\mathbf{v}')\, d^3v' \right) \right] \Bigg\}. \end{aligned} \tag{4.6}$$

While the numerical techniques introduced in Chapter 2 are all valid, a high-dimensional nonlinear model like the FPO is generally prohibitively computationally expensive and difficult to implement as-is.

Vlasov-Fokker-Planck (VFP) codes have applied a variety of techniques to simplify the FPO

to achieve a more accurate treatment of collisions than approximate models like the LBO, Eq. 1.59. It is worth mentioning particle-in-cell (PIC) methods [10] here as well, as collision models in PIC codes avoid the statistical averaging performed in deriving the FPO by handling particle collisions directly e.g. by Monte Carlo methods [127]. A review of these methods up to 2012 was conducted by Thomas et al. [129], and more recent developments are discussed in [5]. We now briefly discuss the basics of some of these common techniques applied by existing VFP codes for context.

The first approach involves decomposing the distribution function into an isotropic part summed with anisotropic contributions that are expressed in terms of some angular basis set in order to reduce the dimensionality of the problem. There are two common choices for the form of this expansion: the *Cartesian tensor expansion* and the *spherical harmonic expansion* [129]. In the Cartesian tensor expansion, the anisotropic contributions of the distribution function are written in terms of the direction cosines,

$$f(\mathbf{v}) = f_0 + \widehat{\mathbf{v}} \cdot \mathbf{f}_1(v) + \widehat{\mathbf{v}}\widehat{\mathbf{v}} : \mathbf{f}_2(v) + \cdots . \tag{4.7}$$

With this approximation, the anisotropic parts of the distribution function are functions of the velocity magnitude v rather than the full velocity vector. When the Cartesian tensor expansion is truncated after the first term, the method is also known as the *diffusive approximation* [37], used in codes such as IMPACT [67] and SPARK [39]. One noteworthy fact about the diffusive approximation is that the two components of the expansion, f_0 and $\mathbf{f}_1$, give different macroscopic quantities: the zeroth and second velocity moments of f_0 are the number density and energy density, while the same moments of $\mathbf{f}_1$ yield the current density and heat flux [129].

One can alternatively choose to expand the distribution function as a series of spherical

harmonics as in the KALOS formalism [6],

$$f(\mathbf{v}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_l^m(v) P_l^m(\cos\theta) e^{im\phi}, \tag{4.8}$$

where $f_l^{-m} = (f_l^m)$. In a similar vein to the Cartesian tensor expansion, the use of spherical harmonics in this way reduces the dimensionality of the evolution equations of the coefficients f_l^m to be only dependent on velocity magnitude, rather than each velocity component. The spherical harmonic expansion offers several benefits, detailed by Bell et al. [6], such as the ability to arbitrarily increase the number of included harmonics in the expansion to capture higher degrees of anisotropy. Additionally, approximations for the Fokker-Planck collision terms can be made naturally by including only certain harmonics in the computation of the Rosenbluth potentials, such as in the case of electron-electron collisions, which can be well-approximated by only including the isotropic term in the computation when the electrons can be assumed to be thermalized. Examples of codes that utilize the spherical harmonic expansion include *fast*VFP [5], OSHUN [130, 131], and K2 [107]. Note that while we have introduced these two approximations as distinct, they are deeply related and can be transformed between [105].

Regardless of which decomposition method is chosen, the next step is to insert the approximate distribution function into the Boltzmann equation to arrive at the evolution equations for each coefficient. As an example, consider the KALOS equations for the f_l^m spherical harmonic coefficients in 2X3V [6],

$$\frac{\partial f_l^m}{\partial t} - \mathcal{C}_{l,i}^m - \mathcal{C}_{l,e}^m - \mathcal{B}_l^m = \mathcal{A}_{l,x}^m + \mathcal{A}_{l,y}^m + \mathcal{E}_{l,x}^m + \mathcal{E}_{l,y}^m + \mathcal{E}_{l,z}^m. \tag{4.9}$$

The first two terms, $\mathcal{C}_{l,i}^m$ and $\mathcal{C}_{l,e}^m$, are the contributions from collisions with ions and electrons, respectively, and $\mathcal{B}_l^m$ handles velocity-space rotation of the distribution function due to

magnetic field. The right-hand-side consists of configuration space advection terms, $\mathcal{A}^m_{l,x}$ and $\mathcal{A}^m_{l,y}$, and velocity space advection terms due to electric field, $\mathcal{E}^m_{l,x}$, $\mathcal{E}^m_{l,y}$, and $\mathcal{E}^m_{l,z}$. These terms are then discretized in phase space, generally using a finite difference method [5, 67, 130]. Time discretization is also done by term with either an implicit or explicit method; as noted by Bell et al., the velocity space terms on the left-hand side are generally solved implicitly while the configuration space terms on the right-hand-side are solved explicitly, such as in K2 (though they treat anisotropic collisions semi-implicitly) [107]. Applying an implicit method in this way can serve to maintain a reasonable timestep when the high collision frequencies at low velocity could dramatically reduce the timestep with an explicit method [5]. An alternative to this split implicit-explicit method is known as the alternating direction implicit (ADI) method, where the configuration and velocity space derivatives swap between being handled implicitly and explicitly each timestep, as in the SPARK [39] and IMPACT [67] codes.

When considering a numerical scheme for a conservation law like the Vlasov-Fokker-Planck system, one must assess whether that scheme maintains the conservation properties of the original equation. In our case, we know the Boltzmann equation and Fokker-Planck operator conserve number density, momentum, and energy in their continuous forms. The codes we have discussed thus far apply the Cartesian tensor or spherical harmonic expansion to simplify the continuous form of the VFP equation and a finite difference scheme to the configuration and phase space derivatives. Differencing schemes for the isotropic equations can be constructed relatively easily to guarantee conservation of density and energy in the advection portions of the scheme [67, 131]. For the collision terms, a commonly-applied implicit method was developed by Chang and Cooper [23] that enforces positivity of f and conservation of number density, and this method was later modified to conserve energy as well [11, 40]. While number density and energy are expected to be conserved by these models, the

typical simplifications on the collision operator, e.g. removing anisotropic electron-electron contributions and replacing them with an approximate model [38], can result in momentum nonconservation. Should the ion species be treated as static, removal of collision terms in this manner does not directly violate momentum conservation, but in the case of mobile ions, the missing momentum can simply be injected directly to the ions [6, 130]. This is one example how an expansion-based VFP scheme can be made fully-conservative to account for errors that arise from the chosen approximations.

The diffusive approximation and KALOS method spherical harmonic expansion are both valid and well-studied techniques for VFP simulations that have been successfully implemented in an assortment of codes and utilized for novel studies of transport [16, 17, 107]. However, these methods all begin by simplifying the Vlasov-Fokker-Planck equation through decomposing the distribution function into an isotropic portion and a truncated series representing the anisotropic portion, removing some information from the "true" distribution function. Should we desire a complete treatment of the complete FPO, we require a method robust and efficient enough to handle the original highly-dimensional nonlinear integro-differential equation while retaining its inherent conservation properties. One method proposed by Taitano et al. is a fully-implicit, fully-conservative, finite volume discretization, first in 0X2V [124] and later in 1X2V [126]. This method begins by considering the FPO in conservative advection-diffusion form (similar to that used in the conservation proofs in Chapter 1),

$$C_{\alpha\beta} = \nabla_{\mathbf{v}} \cdot \mathbf{J}_{\alpha\beta} = \nabla_{\mathbf{v}} \cdot \Gamma_{\alpha\beta} \left(\mathbf{D}_{\beta} \cdot \nabla_{\mathbf{v}} f_{\alpha} - \frac{m_{\alpha}}{m_{\beta}} \mathbf{A}_{\beta} f_{\alpha} \right). \tag{4.10}$$

Just as described in Chapter 2, a finite volume discretization is applied to this equation (in two velocity dimensions, $v_{\parallel}$ and $v_{\perp}$) to arrive at a discrete solution written in terms of cell interface fluxes. There are then four total fluxes that must be handled carefully in order to ensure conservation and positivity of f: advection and diffusion fluxes at $v_{\parallel}$ and

$v_\perp$ cell boundaries. In summary, Taitano et al. apply the SMART limiter scheme to the advective fluxes [43], and a symmetric, positivity-preserving flux for the diffusion tensor fluxes [21]. While this scheme guarantees conservation of number density with zero-flux boundary conditions in velocity space, energy conservation requires the use of a scaling factor for a term in the FPO to account for numerical imbalance arising from truncation error (they chose to apply this to the diffusion term, as the advective fluxes are expected to be more accurate). Similarly, momentum conservation requires another scaling factor to be applied to the original equation, and it is again applied to the diffusion term. The final modified FPO used by Taitano et al. in their numerical implementation is then

$$C_{\alpha\beta} = \nabla_{\mathbf{v}} \cdot \Gamma_{\alpha\beta} \left((\gamma_{\alpha\beta}\mathbf{I} + \epsilon_{\alpha\beta}) \cdot \mathbf{D}_\beta \cdot \nabla_{\mathbf{v}} f_\alpha - \frac{m_\alpha}{m_\beta} \mathbf{A}_\beta f_\alpha \right), \tag{4.11}$$

where $\gamma_{\alpha\beta}$ is the energy conservation factor and $\epsilon_{\alpha\beta}$ is the momentum conservation factor, both of which are computed as the solutions to a linear system arising from the conservation constraint equations. Timestepping is then performed using a fully-implicit multigrid-preconditioned Jacobian-free Newton-Krylov (JFNK) method [65]. In their extension from 0X2V to 1X2V with an adaptive velocity grid [125, 126], Taitano et al. apply a fluid treatment to the electrons while handling the ions with essentially the same implicit finite volume VFP method. One major difference is the adjustment of the conservation corrections to account for an adaptive grid, introducing several new numerical constraint factors that enforce exact conservation of number density, momentum, and energy in these normalized velocity cases.

This fully-implicit finite volume implementation of the VFP is robust and has many attractive features such as large timestep due to the implicit time integrator, guaranteed conservation of mass, momentum, and energy, and adaptibility allowing for large variations in bulk plasma properties. However, the implicit nature of the JFNK method results in a numerical

scheme that requires inverting large matrices each timestep, and the preconditioner limits the robustness of the conservation constraints [124]. In an effort to tackle the full nonlinear FPO in a similarly accurate and robust manner, we note that Eq. 4.6 is still in the form of a hyperbolic conservation law, which we noted in Chapter 2 are well-handled by discontinuous Galerkin methods. DG discretizations are, in general, highly-parallelizeable, accurate to arbitrarily high order, and, in the case of a modal expression, can be made efficient and free of aliasing errors and matrix inversions via pre-computation of integrals [52, 63]. A local DG scheme for the multispecies Rosenbluth/Fokker-Planck operator was proposed by Shiroto et al. [109] that is designed to preserve the skew-symmetry of the thermal relaxation term to enforce mass and energy conservation with an isotropic distribution function. It is also worth noting that DG methods have also successfully been applied to the gyroaveraged FPO to form a scheme that is fully-conservative and utilizes a finite element method solver for the Rosenbluth potentials [66]. We will now discuss the application of the discontinuous Galerkin method introduced in Chapter 2 to the advection-diffusion form of the full nonlinear Rosenbluth/Fokker-Planck collision operator and address the details of the implementation of this scheme in `Gkeyll`.

4.1 Semi-Discrete Form of the Rosenbluth/Fokker-Planck Operator

Beginning with the advection-diffusion form of the FPO, Eq. 4.1, we follow the standard DG process by replacing quantities with their approximations, multiplying by a basis function $\psi_i(\mathbf{z})$, and integrating over a cell K_j,

$$\int_{K_j} \frac{\partial f_h(\mathbf{z})}{\partial t} \psi_i(\mathbf{z}) d\mathbf{z} = -\frac{1}{2} \int_{K_j} \psi_i(\mathbf{z}) \nabla_{\mathbf{v}} \cdot \left[\mathbf{a}_h(\mathbf{z}) f_h(\mathbf{z}) - \mathbf{D}_h(\mathbf{z}) \cdot \nabla_{\mathbf{v}} f(\mathbf{z}) \right] d\mathbf{z}.$$

To keep notation as clean as possible, we will omit the phase space dependence and handle the drag and diffusion terms separately.

Integrating the drag contribution of the FPO by parts yields

$$\int_{K_j} \psi_i \nabla_{\mathbf{v}} \cdot \mathbf{a}_h f_h d\mathbf{z} = \oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}} \cdot \hat{\mathbf{F}}_{\text{drag}} dS - \int_{K_j} \nabla_{\mathbf{v}} \psi_i \cdot \mathbf{a}_h f_h d\mathbf{z}. \tag{4.12}$$

A choice must be made for the flux term, $\hat{\mathbf{F}}_{\text{drag}}$, and a robust option for advection equations is a local Lax-Friedrichs flux that penalizes large jumps in f_h [64],

$$\hat{\mathbf{F}}_{\text{drag}} = \frac{1}{2}(\widetilde{\mathbf{a}}_h f_h^- - \widetilde{\mathbf{a}}_h f^+) - \frac{|\max(\widetilde{\mathbf{a}}_h)|}{2}(f_h^+ - f_h^-), \tag{4.13}$$

though a simpler upwind flux can be used if $\text{sign}(\widetilde{\mathbf{a}}_h)$ is constant along a cell boundary,

$$\widehat{\mathbf{n}} \cdot \hat{\mathbf{F}}_{\text{drag}} = \begin{cases} \widehat{\mathbf{n}} \cdot \widetilde{\mathbf{a}}_h f_h^- & \text{sign}(\widetilde{\mathbf{a}}_h) > 0, \\ \widehat{\mathbf{n}} \cdot \widetilde{\mathbf{a}}_h f_h^+ & \text{sign}(\widetilde{\mathbf{a}}_h) < 0. \end{cases}$$

The advection velocity, $\mathbf{a}$, must be continuous across cell boundaries for stability, so we apply the recovery scheme from Chapter 2 to recover a continuous value for the drag coefficient across each cell boundary.

Applying the same treatment to the diffusion term, we obtain

$$\int_{K_j} \psi_i \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}} f_h) \mathbf{z} = \oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}}_{\text{diff}} \, dS - \int_{K_j} \nabla_{\mathbf{v}} \psi_i \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}} f_h) d\mathbf{z}, \tag{4.14}$$

where the diffusion flux term is

$$\widehat{\mathbf{F}}_{\text{diff}} = \widetilde{\mathbf{D}}_h \cdot \nabla_{\mathbf{v}} \widetilde{f}_h. \tag{4.15}$$

Similar to the drag flux, $\mathbf{D}$ must be continuous across cell boundaries, so it is recovered in the surface term. Recall the treatment of the LBO, where a single integration by parts

resulted in a nonconservative scheme due to the $\nabla_{\mathbf{v}} f$ in the volume term. That issue arises in this case as well, so we integrate by parts again to arrive at the semi-discrete form of the diffusion term,

$$\int_{K_j} \psi_i \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}} f_h) \mathbf{z} = \oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}}_{\text{diff}} \, dS - \oint_{\partial K_j} \nabla_{\mathbf{v}} \psi^- \cdot \widetilde{\mathbf{D}}_h \widetilde{f}_h \cdot \widehat{\mathbf{n}} \, dS + \int_{K_j} f_h \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}} \psi_i) \, d\mathbf{z}. \tag{4.16}$$

Substituting the semi-discrete forms of the drag and diffusion terms into the original equation leads us to the DG semi-discrete advection-diffusion form of the FPO,

$$\int_{K_j} \frac{\partial f_h}{\partial t} \psi_i d\mathbf{z} = -\frac{1}{2} \oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}} \, dS - \frac{1}{2} \oint_{\partial K_j} \nabla_{\mathbf{v}} \psi^- \cdot \widetilde{\mathbf{D}}_h \widetilde{f}_h \cdot \widehat{\mathbf{n}} \, dS + \frac{1}{2} \int_{K_j} \nabla_{\mathbf{v}} \psi_i \cdot \mathbf{a}_h f_h d\mathbf{z} + \frac{1}{2} \int_{K_j} f_h \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}} \psi_i) \, d\mathbf{z}, \tag{4.17}$$

where the drag and diffusion fluxes have been combined into a cumulative numerical flux,

$$\hat{\mathbf{F}} = \widehat{\mathbf{F}}_{\text{diff}} + \widehat{\mathbf{F}}_{\text{drag}} = \frac{1}{2}(\widetilde{\mathbf{a}}_h f_h^- - \widetilde{\mathbf{a}}_h f^+) - \frac{|\max(\widetilde{\mathbf{a}}_h)|}{2}(f_h^+ - f_h^-) - \widetilde{\mathbf{D}}_h \cdot \nabla_{\mathbf{v}} \widetilde{f}_h. \tag{4.18}$$

This is the formalism that is implemented in `Gkeyll`. Before progressing with the implementation details, there is one important subtlety in the diffusion flux term that we have glossed over. In the DG discretization presented in Chapter 2, when surface terms required a derivative, we utilized the recovery process to reconstruct a polynomial that is higher-order and continuous across cell boundaries. However, those quantities were only vectors, so the derivatives were taken across cell boundaries. For the FPO, our diffusive flux is a tensor

quantity, introducing some complication. Consider the diffusion flux component of Eq. 4.17,

$$\oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}} \cdot \widetilde{\mathbf{D}}_h \cdot \nabla_{\mathbf{v}} \widetilde{f}_h \, dS.$$

The contribution of this term at a $\widehat{\mathbf{v}}_x$ surface would be

$$\begin{aligned}\oint_{\partial K_j} \psi_i^- \widehat{\mathbf{v}}_x \cdot \widetilde{\mathbf{D}}_h \cdot \nabla_{\mathbf{v}} \widetilde{f}_h \, dS_{vx} &= \oint_{\partial K_j} \psi_i^- \widetilde{D}_{xi} \frac{\partial f}{\partial v_i} \, dS_{vx} \\ &= \oint_{\partial K_j} \psi_i^- \left(\widetilde{D}_{xx} \frac{\partial \widetilde{f}}{\partial v_x} + \widetilde{D}_{xy} \frac{\partial \widetilde{f}}{\partial v_y} + \widetilde{D}_{xz} \frac{\partial \widetilde{f}}{\partial v_z} \right) dS_{vx}.\end{aligned}$$

For the v_x derivative, we can simply apply the same recovery process as before, yielding a representation that is continuous between the cell neighbors and higher-order in v_x. But what of the v_y and v_z derivatives evaluated at this v_x surface? We require a derivative in the transverse directions *along* a cell boundary for the off-diagonal contributions of this surface term. While we could simply take the transverse derivatives of the distribution function recovered in the v_x direction, we will lose accuracy (we will see later this it does in fact make sense to do when considering certain domain boundary cells). Instead, we extend the simple recovery scheme described earlier to reconstruct polynomials that are continuous across cell boundaries and higher-order *along* them.

4.2 Extending the Recovery Process: 1-cell and 6-cell Recovery

In this section, we extend the recovery scheme introduced in Chapter 2 to arrive at two new recovery processes: (a) recovery of a higher-order representation defined within a cell volume that is continuous at both cell boundaries and (b) recovery of a representation that is con-

tinuous across a boundary and higher order along said boundary. Regarding nomenclature, our original recovery operation is referred to as *2-cell recovery*, process (a) will be referred to as *1-cell recovery*, and process (b) is known as *6-cell recovery*. A complete introduction to recovery discontinuous Galerkin (RDG) schemes such as these, including convergence and Fourier analyses and several other multidimensional recovery algorithms, is presented in [78].

To begin, recall the 2-cell recovery process as applied to some DG discretized quantity defined in two dimensions, $q(x, y)$. If we require a high order continuous representation between two adjacent cells in the x-direction, I_L and I_R, we seek a polynomial, $\widetilde{q}$, that is defined across both cells and is weakly equal to each quantity in its respective cell. That is,

$$\int_{I_L} \psi_L(x,y)\widetilde{q}(x,y)\,dx = \int_{I_L} \psi_L(x,y)q_L(x,y)\,dx,$$
$$\int_{I_R} \psi_R(x,y)\widetilde{q}(x,y)\,dx = \int_{I_R} \psi_R(x,y)q_R(x,y)\,dx,$$

where the basis functions, $\psi_L(x, y)$ and $\psi_R(x, y)$, are mapped from the logical space reference element to be defined within each cell. Choosing $\widetilde{q}(x, y)$ to be a polynomial, the weak equality conditions form a linear system that is solved to determine the coefficients of the recovery polynomial. In the one-dimensional case, the recovery polynomial is a simple $2p + 1$-order polynomial in x, but adding another spatial dimension naturally complicates this. One could choose to construct a recovery polynomial using a tensor product of the one-dimensional recovery polynomial and standard basis set, matching the number of degrees of freedom to achieve a multidimensional recovery basis set that is higher order in the recovery direction, and performing the recovery using that larger basis set directly. In `Gkeyll`, we instead choose to project the multidimensional expansions in the neighboring cells onto the one-dimensional basis set and perform the 2-cell recovery as normal in one dimension. The recovery polynomial in this case is still of order $2p + 1$ in the recovery direction, and while

the coefficients $\widetilde{q}_i$ are constant in the recovery direction, they are still functions of the other spatial variables. Considering our two-dimensional example recovering in x, the original expansions $q_{L/R}(x,y)$ are projected onto the reduced one-dimensional basis set,

$$\hat{u}_{i,L}(y) = \int_{I_L} q_L(x,y)\psi_{L,i}^{(1\mathrm{D})}(x)\,dx,$$
$$\hat{u}_{i,R}(y) = \int_{I_R} q_R(x,y)\psi_{R,i}^{(1\mathrm{D})}(x)\,dx,$$

The recovered polynomial is then computed as before, enforcing weak equality in each cell with these new projections, $u_{L/R}(x,y) = \hat{u}_{i,L/R}(y)\psi_i^{(1\mathrm{D})}(x)$, yielding

$$\widetilde{q}(x,y) = \widetilde{q}_0(y) + \widetilde{q}_1(y)x + \widetilde{q}_2(y)x^2 + \widetilde{q}_3(y)x^3.$$

Having extended the 2-cell recovery process to multiple dimensions, we can now address the first of the two extended recovery schemes: 1-cell recovery. 1-cell recovery is actually a one-dimensional process like 2-cell recovery, but as it is exclusively applied to multidimensional problems in this work, it is worth noting that the steps outlined above for multidimensional 2-cell recovery are equally applicable to 1-cell recovery. For simplicity however, we again return to one dimension to set up an example problem where 1-cell recovery may be useful. Consider for a moment the weak form of the diffusion tensor,

$$\int_{K_j} \mathbf{D}_h\psi_n\,d\mathbf{z} = \Gamma\int_{K_j} \nabla_{\mathbf{v}}\otimes\nabla_{\mathbf{v}}G_h\psi_n\,d\mathbf{z}. \tag{4.19}$$

The derivatives of G_h in this integral can cause issues like nonconservation due to discontinuities at the cell boundaries, and taking a derivative like this lowers the expected order of our scheme. Therefore, we seek a recovered representation that is continuous at *both* cell boundaries and is high-order in the gradient direction within the volume of the cell. 1-cell

recovery addresses these needs through a two-step process:

1. Perform 2-cell recovery at the upper and lower boundaries, evaluating the recovered quantities at the cell interface.

2. Use these values at the cell interfaces as boundary conditions for a third recovery in the center cell.

The second step is where we deviate from 2-cell recovery, as the constraints on the final recovered polynomial are simply: 1) it is weakly equal to the original expansion in the central cell and 2) it matches the values prescribed by the previous step at the cell boundaries. Figure 4.1 demonstrates these two steps on the simple Gaussian distribution from Chapter 2.

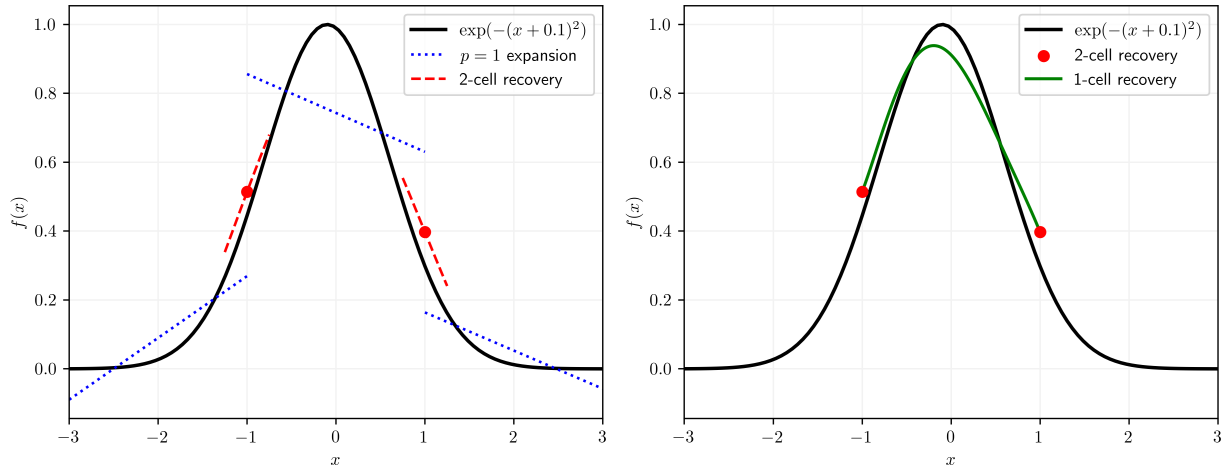

Figure 4.1: Example of the two-step 1-cell recovery process. The first step (left) is to perform 2-cell recovery on both sides of the central cell and evaluate the recovered polynomial at the interface. The second step (right) is to construct a third recovery using the two 2-cell recovered interface values as boundary conditions.

1-cell recovery is utilized in our algorithm to compute the drag coefficient and diagonal components of the diffusion tensor. However, due to the conservation relations, we instead choose to integrate the diffusion term of the FPO by parts twice rather than utilizing 1-cell

recovery. Further, 1-cell recovery is implicitly present throughout the FPO implementation as it is one step in the 6-cell recovery process.

In order to recover the necessary polynomial that allows us to take derivatives along cell boundaries, we now combine the 2-cell and 1-cell processes with a 6-cell stencil around the interface of interest. As this is an inherently multidimensional problem, it is worth noting that all recovery operations use the projection method discussed previously to recover along a given direction. With the details of 2-cell and 1-cell recovery settled, few theoretical leaps are required to arrive at the 6-cell recovery process. An example 6-cell stencil is presented in Figure 4.2, where we seek a high-order continuous solution in the x-direction at the y-normal cell interface between cells TC and BC. 6-cell recovery consists of three steps:

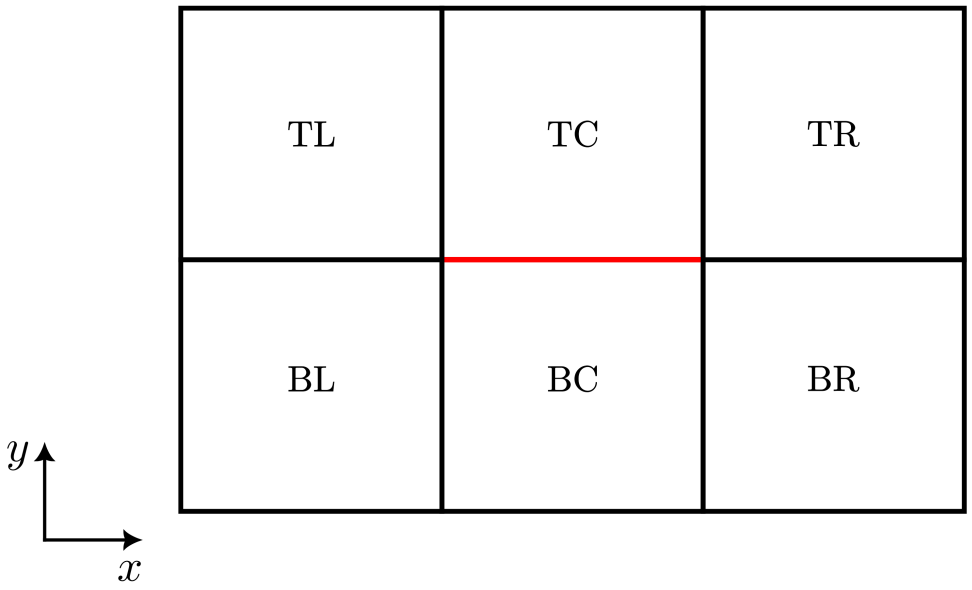

Figure 4.2: 6-cell stencil used for interface-tangential recovery.

1. 2-cell recovery between each of the three pairs of cell neighbors in the interface normal direction.

2. Evaluation of each recovered quantity at the cell interface and projection onto a surface basis set with one fewer dimension than the original set.

3. 1-cell recovery along the cell interface of interest, where the central volume expansion and edge boundary conditions are the surface projections of the three respective 2-cell recoveries.

To hopefully elucidate these steps, we now walk through an example of the 6-cell recovery process for a quantity q. In the case above, in which the interface normal is in the y-direction, the first step is to perform three 2-cell recoveries in the y-direction, between the following cell pairs: BL-TL, BC-TC, and BR-TR. These recovery operations yield a representation that is continuous and higher-order in y, e.g. recovering between $q_{BL}(x,y)$ and $q_{TL}(x,y)$ gives $\widetilde{q}_L(x,y)$. These three recovered quantities, which we call $\widetilde{q}_{L/C/R}(x,y)$, are then projected onto a so-called surface basis set. A surface basis set is useful when we are removing the variation in one direction, namely when we are evaluating a projected quantity at a cell boundary, and it is constructed by evaluating that independent variable at the corresponding boundary in logical space. In practice, this generally involves simply constructing a basis set with one fewer dimension than the volume basis. One must take care however, as the volume expansion should be evaluated at the proper location in logical space when projecting onto a surface basis set. For example, in `Gkeyll`, the Maxima CAS script that performs 2-cell recovery (see Appendix B.1)[1] maps the two cells to $\eta \in [-2,0]$ and $\eta \in [0,2]$ for logical space variable η, so our scripts evaluate the recovery polynomial at $\eta = 0$ before performing the inner products to project onto the lower dimensional surface basis.

With the recovered polynomials projected onto the surface basis, we now have three quantities: $\widetilde{q}_{L/C/R}(x, y=0)$, each of which is evaluated along the center horizontal cell boundary in Figure 4.2. These three quantities are still discontinuous in the x-direction, and we have yet

[1] All `Gkeyll` Maxima scripts can be found here: https://github.com/ammarhakim/gkylcas

to increase the order of our approximation in the x-direction. Therefore, we complete the 6-cell recovery by applying 1-cell recovery in the x-direction. In forming the linear system, we impose the following conditions on our final recovery polynomial, $\widetilde{q}^{(6)}$: weak equality with the central surface projection, $\widetilde{q}_C(x, y = 0)$, and value and/or derivative matching at the edges of the interface, with the constraints coming from $\widetilde{q}_{L/R}(x, y = 0)$. Mathematically, these conditions are (expressed in logical space),

$$\begin{aligned}
\widetilde{q}^{(6)}(\eta_x) &\overset{\circ}{=} \widetilde{q}_C(\eta_x, \eta_y = 0), \\
\widetilde{q}^{(6)}(\eta_x = -1) &= \widetilde{q}_L(\eta_x = 1), \\
\widetilde{q}^{(6)}(\eta_x = 1) &= \widetilde{q}_R(\eta_x = -1),
\end{aligned}$$

where $\eta_x \in [-1, 1]$. While we only enforce value matching in this final step, it is also valid, and sometimes necessary, to require that the derivatives match as well as, or instead of, the values. 6-cell recovery is a powerful tool that allows our scheme to completely capture the cross derivatives that arise from the use of the full diffusion tensor in the FPO in a highly accurate manner. Additionally, as with 2-cell recovery, the use of 6-cell recovery when computing numerical fluxes guarantees cell pair-wise conservation thanks to the symmetry of the operation. With this numerical background, we can proceed to the details of the DG FPO implementation in `Gkeyll`.

4.3 Calculation of the Drag and Diffusion Coefficients

Our implementation begins with computing the drag and diffusion coefficients from the Rosenbluth potentials. At this time, we have elected to simply project the analytic Rosenbluth potentials corresponding to collisions with a Maxwellian onto our basis set. Presented

in Chapter 1 (Eqs. 1.41 and 1.42), the analytic potentials for collisions with a Maxwellian have the form

$$H_b = \frac{n_b}{v}\text{erf}\left(\frac{v}{\sqrt{2}v_{th,b}}\right),$$

$$G_b = n_b\sqrt{2}v_{th,b}\left[\frac{1}{\sqrt{\pi}}\exp\left(-\frac{v^2}{2v_{th,b}^2}\right) + \text{erf}\left(\frac{v}{\sqrt{2}v_{th}}\right)\left(\frac{\sqrt{2}v_{th,b}}{v} + \frac{v}{\sqrt{2}v_{th,b}}\right)\right],$$

where the speed, v, is instead the relative speed when there is nonzero bulk velocity, $v \to |\mathbf{v} - \mathbf{u}|$. We calculate these potentials each timestep using the discrete moments of the distribution function at $(p+1)^{d-1}$ Gaussian quadrature nodes in each cell, for d phase space dimensions. Future development will involve a Poisson solve to calculate these potentials directly from the distribution function.

Regardless of how the modal expansions of the Rosenbluth potentials are computed (this method is agnostic to the potential calculation, so the future Poisson solve will hook in seamlessly), they are then used to compute the drag coefficient and diffusion tensor. We seek the expansions of $\mathbf{a}$ and $\mathbf{D}$ in our basis set,

$$\mathbf{a}_h \overset{\circ}{=} 2\Gamma\nabla_{\mathbf{v}}H_h, \tag{4.20}$$

$$\mathbf{D}_h \overset{\circ}{=} \Gamma_s\nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}}G_h, \tag{4.21}$$

so the question is of the best manner of taking these velocity derivatives. Beginning with the drag coefficient and the diagonal terms of the diffusion tensor,

$$\int_{K_j} a_{h,i}\psi_n d\mathbf{z} = 2\Gamma\int_{K_j}\frac{\partial H_h}{\partial v_i}\psi_n d\mathbf{z} \tag{4.22}$$

$$\int_{K_j} D_{h,ii}\psi_n d\mathbf{z} = \Gamma\int_{K_j}\frac{\partial^2 G_h}{\partial v_i^2}\psi_n d\mathbf{z}. \tag{4.23}$$

The right-hand-sides of these equations are reminiscent of the motivation we gave for the 1-cell recovery process, i.e. derivatives in a cell volume. By performing 1-cell recovery, the polynomial will be of high enough order that we can take multiple derivatives without dropping below the polynomial order of the basis set. Therefore, in the interior of the computational domain, we compute the drag coefficient and diagonal terms simply by performing 1-cell recovery on the potentials and differentiating before projecting back onto the phase basis. Complications arise when we consider cells at the boundaries of velocity space, however. When at a domain boundary in the gradient direction, because `Gkeyll` does not employ ghost cells in velocity space, the last cell of the domain (the *skin* cell) only has one neighbor (the *edge* cell), so our general 1-cell recovery scheme requiring a 3-cell stencil is not applicable. For that reason, within the same routine that calculates the Rosenbluth potentials, when handling a skin cell in a given direction, the analytic derivatives of the Rosenbluth potentials are also evaluated and projected onto the corresponding surface basis using Gaussian quadrature. We have found that specifying only the derivative corresponding to the respective coefficient yields the most accurate results in the final coefficients. That is, the analytic derivatives $\partial_v H$ and $\partial_v^2 G$ are what we compute in each skin cell. These analytic derivatives are then supplied as boundary conditions to the 1-cell recovery operation to compute $a_{h,i}$ and $D_{h,ii}$ at velocity domain boundaries.

Calculation of the off-diagonal elements of the diffusion term of the FPO is a novel contribution of this work and is naturally more complex than the diagonal elements, requiring a distinct treatment. Conveniently, computation of the diffusion tensor introduces similar complications while having substantially less bulk, so implementation of its evaluation serves as a simplified version of the full FPO diffusion term. For these off-diagonal elements, we

are now taking cross derivatives of G,

$$\int_{K_l} D_{h,ij}\psi_n d\mathbf{z} = \Gamma \int_{K_l} \frac{\partial^2 G_h}{\partial v_i \partial v_j}\psi_n d\mathbf{z}. \tag{4.24}$$

As with the diffusion terms of the LBO and full FPO, we will integrate by parts twice,

$$\frac{1}{\Gamma}\int_{K_l} D_{h,ij}\psi_n d\mathbf{z} = \oint_{\partial K_l} \frac{\partial \widetilde{G}_h}{\partial v_j}\psi_n^- dS_{v_i} - \oint_{\partial K_l} \widetilde{G}_h \frac{\partial \psi_n^-}{\partial v_i} dS_{v_j} + \int_{K_l} G_h \frac{\partial^2 \psi_n}{\partial v_i \partial v_j} d\mathbf{z}, \tag{4.25}$$

yielding two surface terms: one that requires a transverse derivative of G_h along a cell interface and one that includes a derivative of G_h across a cell interface. At this point, we are well aware of how to handle these types of surface terms: 6-cell recovery for the first term and 2-cell recovery for the second term. However, the domain boundary cells again introduce issues. For the diagonal components of the diffusion tensor, we simply needed to prescribe boundary conditions for the boundary side of the 1-cell recovery through the analytic derivatives of the potentials. The off-diagonal terms instead have two surface terms over different cell boundaries. Consider the following three examples that arise when computing D_{xy},

$$\frac{1}{\Gamma}\int_{K_l} D_{h,xy}\psi_n d\mathbf{z} = \oint_{\partial K_l} \frac{\partial \widetilde{G}_h}{\partial v_y}\psi_n^- dS_{v_x} - \oint_{\partial K_l} \widetilde{G}_h \frac{\partial \psi_n^-}{\partial v_x} dS_{v_y} + \int_{K_l} G_h \frac{\partial \psi_n}{\partial v_x \partial v_x} d\mathbf{z},$$

within the cell K_l:

(i) K_l is at the domain boundary in v_x and in the interior in v_y. In this case, the boundary consideration emerges when addressing the first surface term. 2-cell recovery in the v_y direction for the second surface term can be performed with no problem at both v_y cell boundaries. The v_x boundary on the interior side presents no problems, as there are 6 neighboring cells to use for the 6-cell recovery. However, the boundary-side cell surface

obviously does not have the required neighbors for this operation. Similarly to before, we choose to project an analytic derivative of G onto a surface basis at the boundary-side cell surface for this cell for use in computing this surface term. Note that we have elected to use a different projection method for this quantity, however: rather than Gaussian quadrature, the derivative is evaluated at the DG node locations and converted to the modal representation. This method enforces continuity of the derivative between cells, which we found aids stability. In short: for domain boundaries in v_i (the direction of the first surface term integral), project an analytic cross derivative of G for use in the boundary-side surface term contribution.

(ii) K_l is at the domain boundary in v_y and in the interior in v_x. In a similar vein as the prior case, in this scenario we can infer that the issue arises in boundary side of the second surface term. The derivative of the basis function can be taken without issue, so all we need is to again prescribe the value of G at that surface. One subtlety here is that while we can easily project the potential onto the surface basis, a simple projection of G does *not* correspond to this second surface term. Rather, we must take the inner product between that surface expansion and the derivative of the surface basis set to achieve the proper surface term.

(iii) K_l is in a corner, at the boundary in both v_x and v_y. Finally, we have the most inconvenient case. A corner cell in velocity space has only three neighbors. For the boundary-side surfaces, we simply repeat the previous two cases, utilizing the surface projections of G and $\partial_{v_y} G$ to compute the respective surface term contributions. For the second surface term, we thankfully have a neighboring cell with which to perform 2-cell recovery as normal. The first surface term again requires more consideration. One can perform a modified 6-cell recovery using the four available cells and use the analytic surface projection as a boundary condition for the final 1-cell recovery stage. Alternatively, and more simply, a 2-cell recovery can be performed instead, and we can accept the loss of a polynomial order of accuracy in the corner cells. More study is necessary to determine the impact of these choices, but we

naturally select the latter method for this purpose.

These three cases actually represent eight different scenarios (upper/lower in each velocity direction, plus four corners) that must each be handled separately. As discussed in Chapter 2, a strength of the modal DG method employed in this work is the ability to precompute the necessary integrals to machine precision, reducing them to kernel functions that are called at runtime. This separation of the majority of the numerics from the rest of the code infrastructure aids us greatly in handling these many boundary conditions. When computing $\mathbf{a}_h$ and $\mathbf{D}_h$, we precompute and write the kernels for each of these domain locations separately. At runtime, during the loop over all phase space cells, we simply determine whether a given cell is in the interior, at an edge, etc. and call the appropriate kernel. The ability of this scheme to capture cross derivatives with such a high order of accuracy is one of the primary achievements of this work.

4.4 Transition to the Full FPO Implementation

The reason we began this discussion with the implementation of the drag and diffusion coefficients is that many of the same concepts transfer directly to the FPO, Eq. 4.17,

$$\int_{K_j} \frac{\partial f_h}{\partial t}\psi_i d\mathbf{z} = -\frac{1}{2}\oint_{\partial K_j} \psi_i^- \widehat{\mathbf{n}}\cdot\widehat{\mathbf{F}}\, dS - \frac{1}{2}\oint_{\partial K_j} \nabla_{\mathbf{v}}\psi^-\cdot\widetilde{\mathbf{D}}_h\widetilde{f}_h\cdot\widehat{\mathbf{n}}\, dS \\ +\frac{1}{2}\int_{K_j}\nabla_{\mathbf{v}}\psi_i\cdot\mathbf{a}_h f_h d\mathbf{z} + \frac{1}{2}\int_{K_j} f_h\nabla_{\mathbf{v}}\cdot\left(\mathbf{D}_h\cdot\nabla_{\mathbf{v}}\psi_i\right) d\mathbf{z},$$

with numerical flux

$$\hat{\mathbf{F}} = \widehat{\mathbf{F}}_{\text{diff}} + \widehat{\mathbf{F}}_{\text{drag}} = \frac{1}{2}(\widetilde{\mathbf{a}}_h f_h^- - \widetilde{\mathbf{a}}_h f^+) - \frac{|\max(\widetilde{\mathbf{a}}_h)|}{2}(f_h^+ - f_h^-) - \widetilde{\mathbf{D}}_h\cdot\nabla_{\mathbf{v}}\widetilde{f}_h.$$

No major theoretical leaps are necessary to reach a full implementation of the FPO given what we have already discussed: 2-cell recovery for surface terms unless we need a transverse derivative, in which case we use 6-cell recovery. One note is that the penalty term in the Lax-Friedrichs flux, the factor $\max(\widetilde{\mathbf{a}}_h)$ is computed by evaluating $\widetilde{\mathbf{a}}_h$ at quadrature points along the cell boundary. If $\widetilde{\mathbf{a}}_h$ has the same sign at all quadrature points, the advective numerical flux simplifies to a simple upwind flux. No 1-cell recovery is performed for the volume terms, rather we just compute volume contributions with the necessary volume expansions. In the same vein as the drag and diffusion coefficients, we decompose the computational grid into three regions for the drag term and diagonal components of the diffusion term: domain volume and the upper and lower domain boundaries. The off-diagonal components of the diffusion term require the same treatment as the diffusion coefficient, so since we have already covered the process in the simpler case, we know to handle all 9 regions of the domain separately for the diffusion update: four edges, four corners, and the domain volume. In fact, handling the assorted domain boundary regions is generally easier for the full FPO because we impose zero-flux boundary conditions in velocity space, $\widehat{\mathbf{n}}\cdot\widehat{\mathbf{F}}(\mathbf{x}, \mathbf{v}_{max}) = \widehat{\mathbf{n}}\cdot\widehat{\mathbf{F}}(\mathbf{x}, \mathbf{v}_{min}) = 0$, causing the majority of those surface terms to simply vanish.

4.5 Conservation Corrections

As demonstrated in Chapter 1, the continuous FPO is conservative in particles, momentum, and energy, but does that hold true for the DG semi-discrete form? To study the conservation properties of the semi-discrete FPO, we simply replace the basis function with 1, $\mathbf{v}$, and $\mathbf{v}^2/2$ for density, momentum, and energy conservation, respectively, and sum over all velocity cells

N_j. Conservation of particles is the simplest to prove,

$$\sum_j^{N_v} \int_{K_j} \frac{\partial f_h}{\partial t}(1) d\mathbf{z} = -\frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} (1)\widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}}\, dS - \frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} \nabla_{\mathbf{v}}(1) \cdot \widetilde{\mathbf{D}}_h \widetilde{f}_h \cdot \widehat{\mathbf{n}}\, dS$$
$$+\frac{1}{2}\sum_j^{N_v} \int_{K_j} \nabla_{\mathbf{v}}(1) \cdot \mathbf{a}_h f_h d\mathbf{z} + \frac{1}{2}\sum_j^{N_v} \int_{K_j} f_h \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot \nabla_{\mathbf{v}}(1))\, d\mathbf{z},$$

as all terms but the first surface term are identically 0 from the gradient, and because $\widehat{\mathbf{F}}$ is continuous across cell boundaries, the only remaining contributions are the domain boundaries, which are 0 from the zero-flux boundary condition. Therefore, this scheme conserves particles,

$$\sum_j^{N_x} \int_{K_j} \frac{\partial n}{\partial t}\, d\mathbf{x} = \sum_j^{N_v} \int_{K_j} \frac{\partial f_h}{\partial t} d\mathbf{z} = 0.$$

For momentum conservation, the vector equation is

$$\sum_j^{N_v} \int_{K_j} \frac{\partial f_h}{\partial t} \mathbf{v} d\mathbf{z} = -\frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} \mathbf{v} \cdot \widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}}\, dS - \frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} (\nabla_{\mathbf{v}} \cdot \mathbf{v}) \cdot \widetilde{\mathbf{D}}_h \widetilde{f}_h \cdot \widehat{\mathbf{n}}\, dS$$
$$+\frac{1}{2}\sum_j^{N_v} \int_{K_j} (\nabla_{\mathbf{v}} \cdot \mathbf{v}) \cdot \mathbf{a}_h f_h d\mathbf{z} + \frac{1}{2}\sum_j^{N_v} \int_{K_j} f_h \nabla_{\mathbf{v}} \cdot (\mathbf{D}_h \cdot (\nabla_{\mathbf{v}} \cdot \mathbf{v}))\, d\mathbf{z},$$

and because $\nabla_{\mathbf{v}} \cdot \mathbf{v} = \mathbf{I}$, the equation before the sum becomes

$$\sum_j^{N_v} \int_{K_j} \frac{\partial f_h}{\partial t} \mathbf{v} d\mathbf{z} = -\frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} \mathbf{v} \cdot \widehat{\mathbf{n}} \cdot \widehat{\mathbf{F}}\, dS - \frac{1}{2}\sum_j^{N_v} \oint_{\partial K_j} \mathbf{I} \cdot \widetilde{\mathbf{D}}_h \widetilde{f}_h \cdot \widehat{\mathbf{n}}\, dS$$
$$+\frac{1}{2}\sum_j^{N_v} \int_{K_j} \mathbf{a}_h f_h d\mathbf{z} + \frac{1}{2}\sum_j^{N_v} \int_{K_j} f_h \nabla_{\mathbf{v}} \cdot \mathbf{D}_h\, d\mathbf{z}.$$

Performing the summation over velocity space cells yields the equation that *must* hold for momentum to be conserved,

$$-\frac{1}{2}\left[\sum_j^{N_v}\oint_{\partial V_{\max_j}}\mathbf{I}\cdot\mathbf{D}_h f_h\cdot\widehat{\mathbf{n}}\,dS_{V_{\max}}-\sum_j^{N_v}\oint_{\partial V_{\min_j}}\mathbf{I}\cdot\mathbf{D}_h f_h\cdot\widehat{\mathbf{n}}\,dS_{V_{\min}}\right]$$
$$+\frac{1}{2}\sum_j^{N_v}\int_{K_j}\mathbf{a}_h f_h d\mathbf{z}+\frac{1}{2}\sum_j^{N_v}\int_{K_j}f_h\nabla_{\mathbf{v}}\cdot\mathbf{D}_h\,d\mathbf{z}\stackrel{\circ}{=}0. \tag{4.26}$$

The first term of this equation is the contribution to total momentum stemming from our introduction of finite velocity extents. One major advantage of DG is that our solutions are discontinuous across cell boundaries, however, when considering conservation, the summation of the volume term over all velocity cells picks up all the jumps at cell boundaries, leading to the two nonzero volume moments in Eq. 4.26.

Repeating this process for energy conservation,

$$\sum_j^{N_v}\int_{K_j}\frac{\partial f_h}{\partial t}\frac{\mathbf{v}^2}{2}d\mathbf{z}=-\frac{1}{2}\sum_j^{N_v}\oint_{\partial K_j}\frac{\mathbf{v}^2}{2}\widehat{\mathbf{n}}\cdot\widehat{\mathbf{F}}\,dS-\frac{1}{2}\sum_j^{N_v}\oint_{\partial K_j}\nabla_{\mathbf{v}}\frac{\mathbf{v}^2}{2}\cdot\widetilde{\mathbf{D}}_h\widetilde{f_h}\cdot\widehat{\mathbf{n}}\,dS$$
$$+\frac{1}{2}\sum_j^{N_v}\int_{K_j}\nabla_{\mathbf{v}}\frac{\mathbf{v}^2}{2}\cdot\mathbf{a}_h f_h d\mathbf{z}+\frac{1}{2}\sum_j^{N_v}\int_{K_j}f_h\nabla_{\mathbf{v}}\cdot(\mathbf{D}_h\cdot\nabla_{\mathbf{v}}\frac{\mathbf{v}^2}{2})\,d\mathbf{z},$$

we arrive at the conservation constraint for energy,

$$-\frac{1}{2}\left\{\sum_j^{N_v}\oint_{\partial V_{\max_j}}\mathbf{v}_{\max}\cdot\mathbf{D}_h f_h\cdot\widehat{\mathbf{n}}\,dS_{V_{\max}}-\sum_j^{N_v}\oint_{\partial V_{\min_j}}\mathbf{v}_{\min}\cdot\mathbf{D}_h f_h\cdot\widehat{\mathbf{n}}\,dS_{V_{\min}}\right\}_{\text{surf}}$$
$$+\frac{1}{2}\sum_j^{N_v}\int_{K_j}\mathbf{v}\cdot\mathbf{a}_h f_h d\mathbf{z}+\frac{1}{2}\sum_j^{N_v}\int_{K_j}f_h\nabla_{\mathbf{v}}\cdot(\mathbf{D}_h\cdot\mathbf{v})\,d\mathbf{z}\stackrel{\circ}{=}0. \tag{4.27}$$

where the boundary surface integral is a sum over the contributions from the upper and lower surfaces in each direction.

For the FPO to conserve energy, Eqs. 4.26 and 4.27 must be true. However, the emergence of the boundary integrals and volume moments is a direct result of finite velocity extents and discontinuous representations within cells, so they are unavoidable in this scheme. Rather than reformulating our scheme in some way to be inherently conservative, we instead follow the same process as the LBO in `Gkeyll`: using these conservation constraint equation to solve for corrections to $\mathbf{a}$ and $\mathbf{D}$. As Eqs. 4.26 and 4.27 are summed over velocity space, we apply corrections to the drag and diffusion coefficients that are functions of configuration space only,

$$\mathbf{a}_h = \widetilde{\mathbf{a}}_h(\mathbf{x},\mathbf{v}) + \delta\mathbf{a}_h(\mathbf{x}), \tag{4.28}$$

$$\mathbf{D}_h = \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) + \delta D_h(\mathbf{x})\mathbf{I}, \tag{4.29}$$

where $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{D}}$ are the drag and diffusion coefficients as calculated from the Rosenbluth potentials. Inserting these forms of the coefficients into the conservation corrections yields

$$\begin{aligned}
&-\sum_j \oint_{\partial V_{max_j}} f_h \left(\mathbf{I}\cdot\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v})\cdot\mathbf{n} + \delta D_h(\mathbf{x})\right) dS_{V_{\max}} \\
&+\sum_j \oint_{\partial V_{min_j}} f_h \left(\mathbf{I}\cdot\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v})\cdot\mathbf{n} + \delta D_h(\mathbf{x})\right) dS_{V_{\min}} \\
&+\sum_j \int_{K_j} (\widetilde{\mathbf{a}}_h(\mathbf{x},\mathbf{v}) + \delta\mathbf{a}_h(\mathbf{x})) f_h \, d\mathbf{z} + \sum_j \int_{K_j} \nabla_{\mathbf{v}}\cdot\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) f_h \, d\mathbf{z} \overset{\circ}{=} 0,
\end{aligned} \tag{4.30}$$

$$\begin{aligned}
&-\sum_j \oint_{\partial V_{max_j}} f_h \left(\mathbf{v}_{\max}\cdot\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v})\cdot\mathbf{n} + \delta D_h(\mathbf{x})\mathbf{v}_{\max}\cdot\hat{\mathbf{n}}\right) dS_{V_{\max}} \\
&+\sum_j \oint_{\partial V_{min_j}} f_h \left(\mathbf{v}_{\min}\cdot\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v})\cdot\mathbf{n} + \delta D_h(\mathbf{x})\mathbf{v}_{\min}\cdot\hat{\mathbf{n}}\right) dS_{V_{\min}} \\
&+\sum_j \int_{K_j} \mathbf{v}\cdot(\widetilde{\mathbf{a}}_h(\mathbf{x},\mathbf{v}) + \delta\mathbf{a}_h(\mathbf{x})) f_h \, d\mathbf{z} \\
&+\sum_j \int_{K_j} \nabla_{\mathbf{v}}\cdot(\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v})\cdot\mathbf{v}) f_h \, d\mathbf{z} + 3\sum_j \int_{K_j} f_h \delta D_h(\mathbf{x}) \, d\mathbf{z} \overset{\circ}{=} 0,
\end{aligned} \tag{4.31}$$

which we can then rearrange into a linear system of four equations for the four unknown correction quantities,

$$
\begin{aligned}
&-\sum_j \oint_{\partial V_{max_j}} f_h \delta D_h(\mathbf{x})\, dS_{V_{\max}} + \sum_j \oint_{\partial V_{min_j}} f_h \delta D_h(\mathbf{x})\, dS_{V_{\min}} + M_{0,h}\delta \mathbf{a}_h(\mathbf{x}) \\
&\overset{\circ}{=} \sum_j \oint_{\partial V_{max_j}} f_h \mathbf{I} \cdot \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \cdot \mathbf{n}\, dS_{V_{\max}} - \sum_j \oint_{\partial V_{min_j}} f_h \mathbf{I} \cdot \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \cdot \mathbf{n}\, dS_{V_{\min}} \\
&+ \sum_j \int_{K_j} \left(\widetilde{\mathbf{a}}_h(\mathbf{x},\mathbf{v}) + \nabla_{\mathbf{v}} \cdot \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \right) f_h\, d\mathbf{z}
\end{aligned} \tag{4.32}
$$

$$
\begin{aligned}
&-\sum_j \oint_{\partial V_{max_j}} f_h \delta D_h(\mathbf{x}) \mathbf{v}_{\max} \cdot \hat{\mathbf{n}}\, dS_{V_{\max}} \\
&+ \sum_j \oint_{\partial V_{min_j}} f_h \delta D_h(\mathbf{x}) \mathbf{v}_{\min} \cdot \hat{\mathbf{n}}\, dS_{V_{\min}} + \delta \mathbf{a}_h(\mathbf{x}) \cdot \mathbf{M}_1 + 3 M_0 \delta D_h \\
&\overset{\circ}{=} \sum_j \oint_{\partial V_{max_j}} f_h \mathbf{v}_{\max} \cdot \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \cdot \mathbf{n}\, dS_{V_{\max}} - \sum_j \oint_{\partial V_{min_j}} f_h \mathbf{v}_{\min} \cdot \widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \cdot \mathbf{n}\, dS_{V_{\min}} \\
&+ \sum_j \int_{K_j} \left(\mathbf{v} \cdot \widetilde{\mathbf{a}}_h(\mathbf{x},\mathbf{v}) + \nabla_{\mathbf{v}} \cdot (\widetilde{\mathbf{D}}_h(\mathbf{x},\mathbf{v}) \cdot \mathbf{v}) \right) f_h d\mathbf{z}.
\end{aligned} \tag{4.33}
$$

The following procedure results in drag and diffusion coefficients that, when used in the FPO update, ensures our scheme is fully conservative in number density, momentum, and energy:

1. Compute the necessary volume moments and boundary corrections in Eqs. 4.32 and 4.33 given $\mathbf{a}_h$ and $\mathbf{D}_h$.
2. Solve the linear system for the corrections $\delta \mathbf{a}_h$ and δD.
3. Accumulate the corrections onto the original $\mathbf{a}_h$ and $\mathbf{D}_h$.

Note that this procedure is strictly valid for *same-species* collisions. Future work will address the addition of cross-species collisions, and the primary hurdle in that implementation will

likely be enforcing conservation in a similar manner. In summary, this process involves accounting for the conservation constraints from one species when correcting the other's drag and diffusion coefficients, such that total momentum and energy are conserved.

4.6 DG FPO Algorithm Overview

In this chapter, we have introduced several new concepts and complications as we developed the conservative, modal DG algorithm for the Fokker-Planck collision operator. It is therefore worthwhile at this point to step back and reconsider the overall picture by summarizing the collision update. The algorithm as implemented in `Gkeyll` can be summarized in the following set of steps that are conducted alongside the Vlasov-Maxwell update [50, 63] each timestep:

1. Compute the necessary velocity space moments (M_0, $\mathbf{M}_1$, and M_2) and primitive moments (n, $\mathbf{u}$, and v_{th}^2) from the input distribution function.
2. Iterate over all cells in phase space to project the analytic Rosenbluth potentials for collisions with a Maxwellian 1.41-1.42.
   (a) Evaluate the analytic potentials H and G in each cell at Gaussian quadrature points to perform the necessary integrations to project them onto the modal basis set.
   (b) In skin cells at domain boundaries, compute H, G, $\partial_v H$, and $\partial_v^2 G$ evaluated at the domain edge and project onto the respective surface basis, again with Gaussian quadrature (only calculate the gradients normal to the boundary).
   (c) Iterate over transverse directions in skin cells to compute $\partial_v G$ and project onto surface basis. This quantity is evaluated at the DG nodes of the surface basis

set and transformed to the modal representation, rather than integrated with Gaussian quadrature, to enforce continuity along the domain boundary.

3. Iterate over phase space cells again to compute $\mathbf{a}_h$ and $\mathbf{D}_h$, Eqs. 4.20 and 4.21. Determine the location of each cell in the computational domain (interior, edge, corner) to fetch the required neighbor cell pointers and call the correct kernel. Additionally, compute the surface projections of $\mathbf{a}$ and $\mathbf{D}$ at cell boundaries required for the surface terms in the FPO update. Example velocity space slices of $\mathbf{a}$, and $\mathbf{D}$ are presented along with H and G in Figure A.1.

4. Compute the moments and boundary corrections required by the conservation correction equations, Eqs. 4.32 and 4.33.

5. Form the matrix for the linear system of conservation equations and solve via LU decomposition. Accumulate the corrections, $\delta\mathbf{a}_h$ and $\delta D_h\mathbf{I}$ onto the drag coefficient and diffusion tensor and their surface expansions.

6. Compute sign information for the corrected drag coefficient surface expansion to simplify the Lax-Friedrichs flux computation.

7. Advance the full FPO, iterating over all phase space cells to apply the drag update then the diffusion update, accumulating CFL frequency contributions in each cell from both updates.

8. Compute the CFL using the most restrictive timestep and return to the main Vlasov-Maxwell algorithm.

Again, each of these steps is carried out every timestep as part of the Vlasov-Fokker-Planck update in `Gkeyll`, and the timestepper of choice is an explicit strong-stability preserving Runge-Kutta method. While this algorithm may appear to be prohibitively expensive to

perform every timestep, the efficient modal DG algorithm makes its application to high-dimensional problems achievable. This DG scheme is highly parallelizeable and can be run in parallel on CPUs with OpenMPI[2], decomposing over configuration space cells. Velocity space parallelization requires a complex shared memory treatment [50]. To this end, the FPO algorithm outlined above is also implemented on GPUs with CUDA [86] for substantial speedup through shared memory parallelization, and multiple-GPU simulations are made possible through NCCL[3].

4.7 Testing the DG Implementation of the FPO

Having outlined the details of our fully-conservative modal discontinuous Galerkin algorithm for the Rosenbluth/Fokker-Planck collision operator, we proceed to testing. The Vlasov-Maxwell infrastructure in `Gkeyll` is fully capable of including all six phase space dimensions, but for this work, each of the test cases in this chapter is performed in 1X3V (one spatial, three velocity space dimensions). We are required to include all velocity space dimensions when using our algorithm, as the FPO is an inherently 3V model, and `Gkeyll` requires at least one configuration space dimension. In some of these studies, we restrict configuration space to a single cell, emulating a 0X3V setup.

4.7.1 Convergence of the Drag and Diffusion Coefficient Routines

The first set of tests seek to evaluate the efficacy of the recovery-based differentiation algorithm utilized in both the drag and diffusion coefficient calculation and the FPO update. To this end, we begin by studying the convergence of our computation of $\mathbf{a}_h$ and $\mathbf{D}_h$. While

[2] https://www.open-mpi.org/
[3] https://developer.nvidia.com/nccl

we are utilizing an analytic form of the Rosenbluth potentials and could similarly take the analytic derivatives of H and G for comparison with the computed $\mathbf{a}_h$ and $\mathbf{D}_h$, the analytic cross derivative of G introduces a strong peak at $v = 0$ that is not resolved when using an even number of velocity space cells (as is the norm). We instead utilize a sinusoidal function,

$$\mathcal{R}(v_x, v_y) = \cos(k_x v_x)\cos(k_y v_y),$$

where $k_x = k_y = \pi/v_{max}$ for maximum velocity space extent $v_{max} = 5.0$.

The drag and diffusion coefficients are then computed according to,

$$\mathbf{a} = 2\Gamma\nabla_{\mathbf{v}} f$$
$$\mathbf{D} = \Gamma_s \nabla_{\mathbf{v}} \otimes \nabla_{\mathbf{v}} f,$$

with the recovery scheme presented in Section 4.3. Note that for these convergence tests, the conservation correction accumulation is omitted. Tests are performed using resolutions of 2^2 to 2^6 cells in each velocity space dimension, and results are presented in Figure 4.3. These tests are restricted to a $p = 2$ Serendipity basis, as the FPO has not yet been implemented for the full tensor basis or $p = 3$ Serendipity (this is a relatively simple matter of generating the kernels). In addition, while `Gkeyll` lists support for $p = 1$ basis, this is generally implemented as a so-called *hybrid* basis that is $p = 1$ in configuration space but $p = 2$ in velocity space. One motivation for the use of this basis is energy conservation: when proving discrete conservation of energy, one integrates over v^2, which is not a member of a $p = 1$ basis set, so a projection operation must be utilized to map onto the $p = 1$ basis [52].

We demonstrate the remarkable convergence order of a recovery-based DG scheme [78], with the drag coefficient computation achieving 6^{th}-order convergence (calculated order 6.14). The diagonal elements of the diffusion tensor, which also utilize 1-cell recovery, naturally

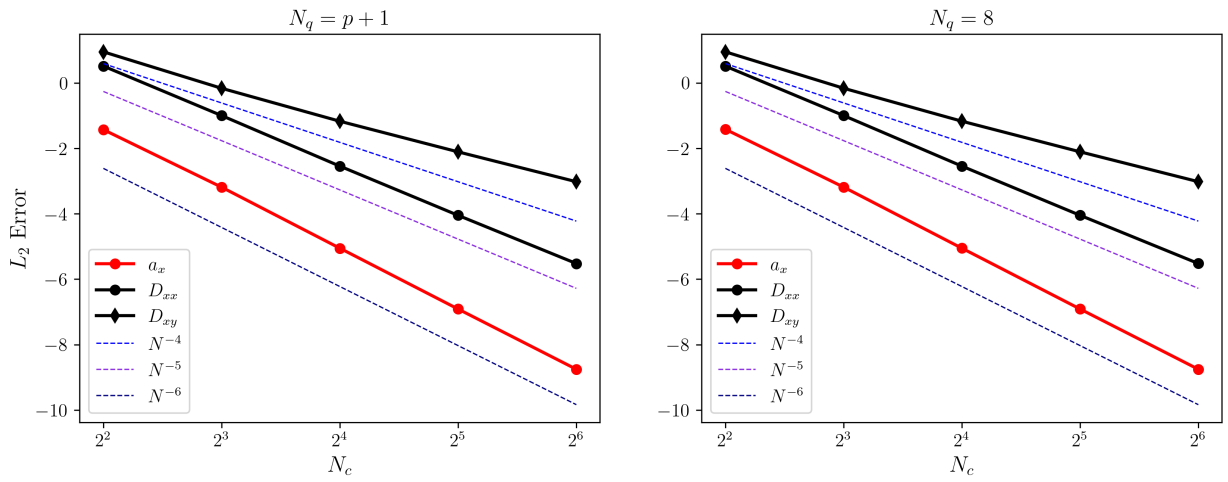

Figure 4.3: Convergence of the L^2-error of the 0^{th} component (cell average) for several derivatives computed using the recovery-based DG schemes outlined in Section 4.3. The left plot uses $p + 1$ quadrature points, N_q, in each direction for the Rosenbluth potential projection, while the right plot uses 8 points in each direction (the maximum in `Gkeyll`), verifying that our reduced number of points still gives an accurate result. The drag coefficient computation and diffusion tensor diagonal terms utilize 1-cell recovery, and their orders of accuracy nearly match the expected values of 6 and 5, respectively. The off-diagonal terms of the diffusion tensor exhibit a lower order of convergence (around 3).

have about one lower convergence order than the drag coefficient, as we are taking another derivative (calculated order 4.95). Off-diagonal terms of the diffusion tensor are instead computed using two integrations by parts, and we are left with an even lower order of accuracy (calculated order 3.07). This may be in part due to the boundary conditions, especially at the corners, where we elected to perform 2-cell recovery rather than a modified 6-cell recovery. Additionally, the 6-cell recovery procedure may be out-performed by a method utilizing a single integration parts, and this is an area of active investigation. To dive a bit further, we compare in Figure 4.4 the global L^2 error for D_{xy} computed only in the interior cells with the calculation over the whole domain. Note the overall decrease in L^2 error when the domain boundary cells are removed from the computation, demonstrating their large relative contribution to the total error. The observed order of accuracy also increases, but not substantially: from 3.07 to 3.64. Overall, further study of the efficacy of

the two integration by parts, 6-cell recovery method relative to other recovery-based methods is warranted.

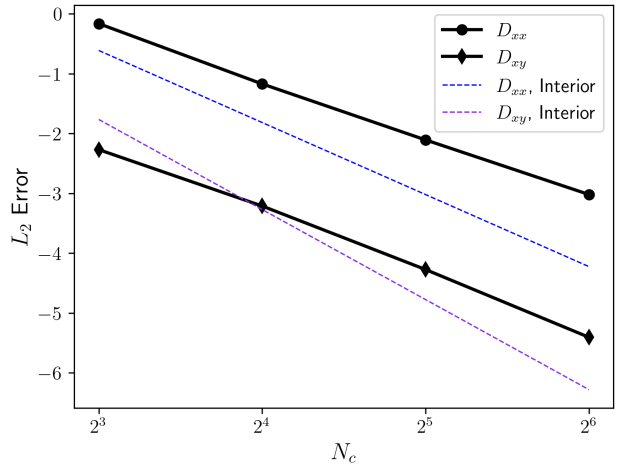

Figure 4.4: Comparison of the L^2 error convergence of the 6-cell recovery-based computation of D_{xy} when considering only the interior of the domain vs. the entire domain. Without the boundary layer of cells, the observed order of accuracy increases slightly, from 3.07 to 3.64. Additionally, the errors are several orders of magnitude smaller overall.

4.7.2 Square Distribution

We now apply the FPO solver to the simple test case of an initially square distribution function. The square distribution offers a case where the collision operator must handle a discontinuity in phase space and an initial condition that is far from the discrete Maxwellian we expect for the steady-state solution. We define the initial distribution function as,

$$f_{\text{square}}(x, v_x, v_y, v_z) = \begin{cases} 0.5 & |v_x| \leq 1.0 \text{ and } |v_y| \leq 1.0 \text{ and } |v_z| \leq 1.0, \\ 0 & \text{otherwise.} \end{cases}$$

For the collision operator, we set $\Gamma = 1.0$, and the characteristic collision frequency is $\nu = n\Gamma\sqrt{2/\pi}/(3v_{th}^3)$ (computed from Eq. 1.30). This polynomial order $p = 2$ simulation is effectively 0X3V, with a single cell in configuration space, 32 cells in each velocity dimension,

and velocity extents ± 4.0 in each direction. The simulation is run to an end time of $5/\nu$ using both the FPO and fully-conservative LBO [52] collision operators, where the LBO collision frequency is ν to be consistent with the FPO.

Below is a slice along $x = v_y = v_z = 0$ of the time evolution of the distribution functions in time. The two collision operators relax at slightly different rates because of the inherent velocity-dependent collision frequency of the FPO, and the greatest discrepancies occur when f is far from Maxwellian. However, both cases reach an identical final Maxwellian when the simulation reaches several collision times.

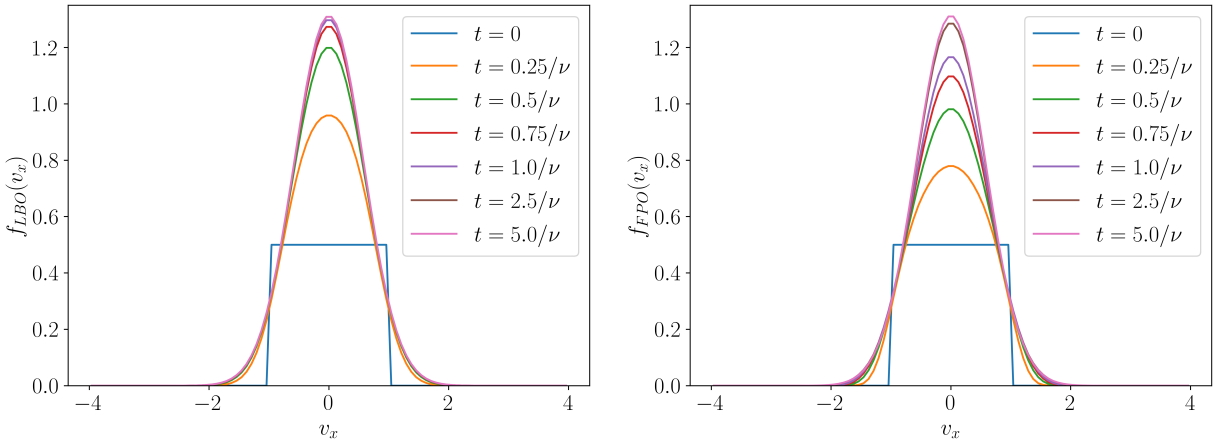

Figure 4.5: Relaxation of initial square distributions to Maxwellian through collisions via the Fokker-Planck and Dougherty-Lenard-Bernstein operators.

This test case also presents an opportunity to address the conservation and stability of our scheme. While there is no bulk velocity, and therefore no momentum, number density and energy conservation can be verified for the FPO by tracking the domain-integrated 0^{th} and 2^{nd} moments of f, as in Figure 4.6. These quantities are conserved to machine precision, with the small decay in magnitude arising due to the damping in the SSP-RK timestepper [52].

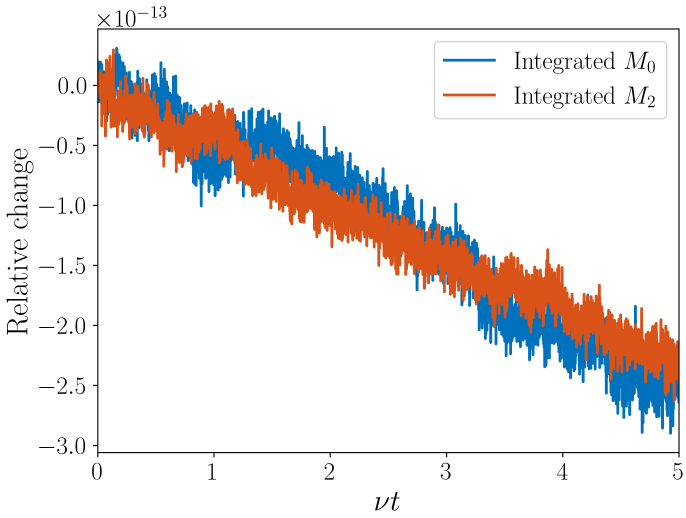

Figure 4.6: Behavior of domain-integrated number density and energy in time for the initial square distribution.

4.7.3 Bump-on-Tail Distribution

In the square distribution case, we find that the velocity-dependent collision frequency of the FPO results in a discrepancy in the rate of relaxation to Maxwellian when compared to the average collision frequency used in the LBO. As discussed in Chapter 1, many experimental regimes are concerned with the fastest particles, that is, the tails of the distribution function. The v^{-3} scaling of the effective collision frequency implies we can expect substantial differences between the two collision models when the deviations from Maxwellian occur at high velocities. To this end, we introduce the bump-on-tail distribution,

$$f(\mathbf{x}, \mathbf{v}, t) = f_M(\mathbf{x}, \mathbf{v}, t) + f_b(\mathbf{x}, \mathbf{v}, t),$$

with

$$f_M(\mathbf{x},\mathbf{v},t) = \frac{n}{\sqrt{2\pi v_{th}^2}^3}\exp\left(-\frac{v_x^2+v_y^2+v_z^2}{2v_{th}^2}\right),$$
$$f_b(\mathbf{x},\mathbf{v},t) = \frac{n}{\sqrt{2\pi v_{th,b}^2}^3}\frac{A_b^2}{v^2+S^2}\exp\left(-\frac{(v_x-u_{x,b})^2+v_y^2+v_z^2}{2v_{th,b}^2}\right),$$

where $n = v_{th} = 1.0$, $A_b = \sqrt{0.15}$, $u_{x,b} = 4v_{th}$, $S_b = 0.14$, and $v_{th,b} = 3.0v_{th}$. Simulations are again 1X3V, $p = 2$, with 16 cells in velocity space. The end time is $5.0/\nu$ with $\Gamma = 1.0$, and $\log f$ slices at $x = v_y = v_z = 0$ are presented at select times in Figure 4.7. As expected, the

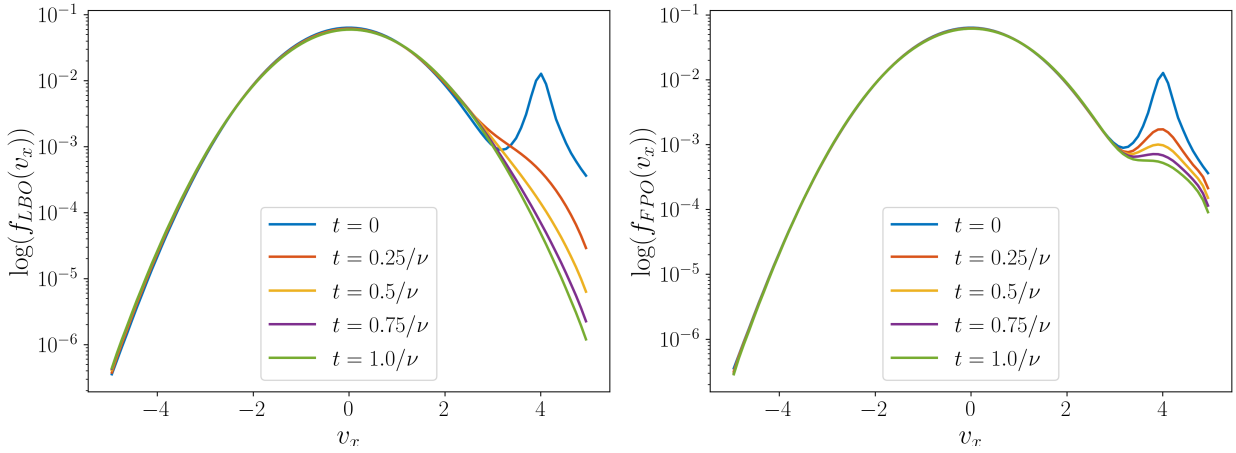

Figure 4.7: Logarithmic plots of f for relaxation of an initial square distribution to Maxwellian through collisions with the Fokker-Planck and Dougherty-Lenard-Bernstein operators. Here we see the effect of the v^{-3} scaling of the collision frequency in the FPO, where the LBO substantially overestimates the effect of collisions in the high-energy tail of the distribution function.

LBO overestimates the relaxation of the high-velocity perturbation relative to the FPO. The distribution functions are plotted here logarithmically to emphasize the high-energy portions of the distribution, but it is still clear that the perturbation has not fully equilibrated even after $5\nu t$ for the FPO. In fusion plasmas, particles moving at several thermal velocities are the ones undergoing fusion reactions and are responsible for a majority of the heat transport,

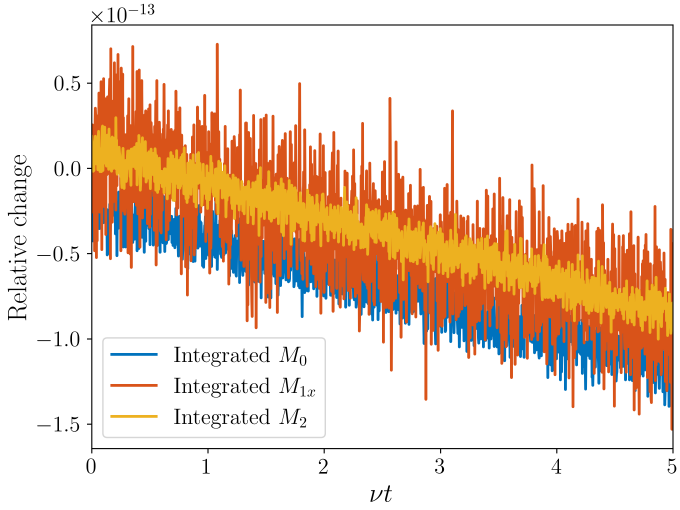

Figure 4.8: Behavior of domain-integrated number density, x-momentum, and energy in time for the bump-on-tail distribution.

so proper treatment of these populations is vital [16].

These simulations have nonzero total momentum, so we can verify momentum conservation here as well. Relative changes in integrated number density, x-direction momentum, and energy are presented in Figure 4.8. Momentum in y and z is initially 0, and both remain at 0 to machine precision for the entire simulation.

We now briefly address the stability of our scheme. While a full linear stability analysis is well beyond the scope of this work, we can "verify" stability experimentally. For an arbitrary distribution function subject to thermalizing collisions, entropy density, $\sigma = -f \log f$, is expected to increase as the f isotropizes to the maximum entropy distribution, i.e. a Maxwellian. Entropy then remains constant for a Maxwellian distribution function in the continuous limit, but numerical dissipation from the use of upwinding fluxes in this DG algorithm causes the entropy to continue to increase to a small degree [52]. Therefore, we can characterize the stability of our scheme by computing the time evolution of the total phase-space integrated entropy. The relative change in total entropy is presented in Figure 4.9 for a

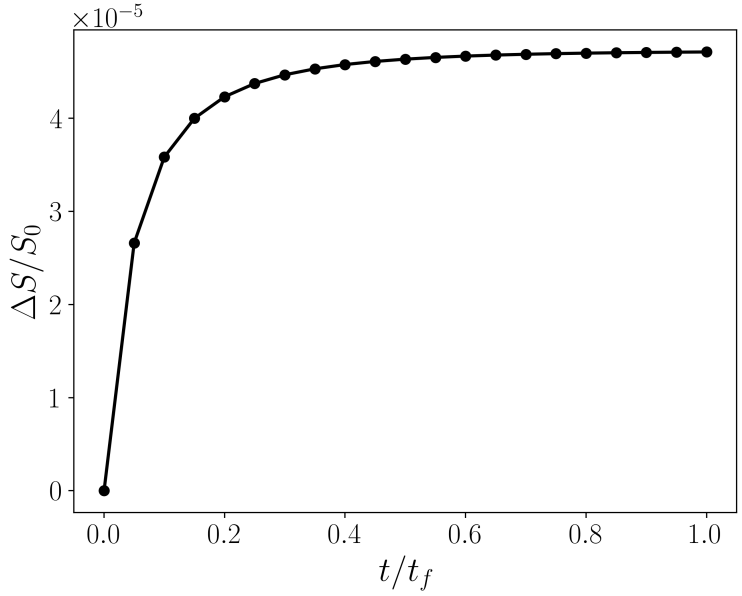

Figure 4.9: Evolution of domain-integrated entropy density, $S = -\int f \log f \, d\mathbf{z}$, for a bump-on-tail distribution undergoing collisional relaxation via the Fokker-Planck operator. As collisions thermalize the distribution function to Maxwellian, entropy increases. When f reaches equilibrium as a discrete Maxwellian, the entropy reaches a peak. Numerical dissipation from the upwinding fluxes in the DG scheme continues to increase the entropy to a small degree.

bump-on-tail distribution. We demonstrate that entropy behaves as expected, asymptoting to a maximum value as the distribution function relaxes to a discrete Maxwellian.

4.7.4 Heat Bath Problem

Next, we apply the FPO to the classic heat bath problem in 1X3V. In the famous work by Braginskii [15], transport equations for momentum and energy in a magnetized hydrogen plasma are derived directly from the Fokker-Planck collision operator using a perturbative approach (similar, in fact, to the numerical approaches introduced at the start of this chapter). By expanding the distribution function as a sum of an isotropic term and a first-order correction term, Braginskii takes moments of the collision operator to arrive at the model

equations for friction forces, heat fluxes, viscosity, and interspecies energy exchange. Of these equations, the portion relevant to this example is the free-streaming temperature-driven ion heat flux due to self-collisions, which is shown to obey Fourier's law [54],

$$\mathbf{q}_T^i = -\kappa_i \nabla_{\mathbf{x}} T_i, \tag{4.34}$$

with thermal conductivity given by

$$\kappa_i = 3.9 \frac{n_i T_i \tau_i}{m_i}. \tag{4.35}$$

This equation for the heat flux that results from temperature variation in a highly collisional plasma dominated by small-angle Coulomb collisions can be applied to our single-species FPO model.

In this setup, we initialize the distribution as a linear variation in temperature between two regions of disparate constant temperatures,

$$T(x) = \begin{cases} T_C & x < -L_x/4, \\ 2\dfrac{T_H - T_C}{L_x} x + \dfrac{T_H + T_C}{2} & -L_x/4 \leq x \leq L_x/4, \\ T_H & x > L_x/4. \end{cases} \tag{4.36}$$

The simulation domain spans $\pm L_x/2$ in x, and $\pm\sqrt{T_H/m}$ in each velocity space direction, and the distribution function is initially Maxwellian everywhere. For Braginskii's model to be valid, the system must be in a highly collisional regime. Therefore, we specify the Knudsen number to be small, $Kn = \lambda_{\text{mfp}}/L_x = 10^{-3}$, and compute the parameter Γ from the corresponding collision frequency. Using a grid with 32 cells in x and N_v^3 cells in velocity

space, we allow the simulation to evolve until the heat flux,

$$\mathbf{q} = \frac{1}{2} m \int \mathbf{v} v^2 f(\mathbf{x}, \mathbf{v}, t)\, d^3 v,$$

reaches a steady state. The thermal conductivity is then computed as $q_x / \partial_x T$ for comparison with theory. Two cases are performed to qualitatively test the velocity space convergence of the FPO: one with $N_v = 8$ and one with $N_v = 16$. The computed thermal conductivity, κ_c in each case is close to the theoretical result, $\kappa_i = 0.0039$: for $N_v = 8$, $\kappa_c = 0.00426$ and for $N_v = 16$, $\kappa_c = 0.00385$. The lower resolution case did exhibit oscillations in q_x, likely because of the boundary layer that forms when using fixed boundary conditions as we did here [19]. It is worth again noting that these simulations use a single uncharged species with only self collisions. When the FPO is implemented with cross collisions, more terms of the heat flux will need to be considered depending on if we study ions or elections, especially in the presence of a magnetic field.

4.8 Summary of Chapter 4

This final chapter is a notable departure from the previous hydrodynamics-focused chapter. We began by introducing background about the state of the art of Vlasov-Fokker-Planck simulations. Three techniques were discussed: the Cartesian tensor expansion (and diffusive approximation) [67], the KALOS formalism spherical harmonic expansion [6], and an implicit finite volume scheme with specific choice of limiters [124]. The former two techniques are closely related and commonly applied VFP methods, but both are approximate methods that remove nearly all anisotropy from the distribution function and collision operator. For many purposes, especially in the presence of a magnetic field introducing a preferred direction, these approximations are more than satisfactory. However, a generalized approach like the

implicit finite volume method is applicable to a wider range of problems. Additionally, a scheme of this type allows for relatively straightforward enforcement of conservation of mass, momentum, and energy simultaneously through a moment-based correction approach.

We then followed a similar trajectory by applying the modal discontinuous Galerkin method introduced in Chapter 2 to the advection-diffusion form of the FPO. When constructing the numerical fluxes, it became apparent that the 2-cell recovery scheme is insufficient for the semi-discrete form of the FPO, as one numerical flux from the diffusion term includes derivatives along cell boundaries, rather than across. To that end, we introduced two multi-stage recovery operations, 1-cell and 6-cell recovery, that allow for polynomials that are high order along cell boundaries to be constructed. 1-cell recovery is applied to compute the drag coefficient and diagonal elements of the diffusion tensor from input Rosenbluth potentials (which are computed analytically and projected onto the basis set). The off-diagonal elements of the diffusion tensor are calculated through a nearly identical process to the FPO drag term, with two integrations by parts and 2- and 6-cell recovery. Boundary conditions for the drag coefficient and diffusion tensor introduced a complication, and we utilize the analytic Rosenbluth potentials and their velocity derivatives as inputs to the necessary recovery operations at the domain boundary.

Implementing the semi-discrete FPO required nearly the same considerations as the drag coefficient and diffusion tensor, so the same general process was adapted. With the main FPO infrastructure accounted for, we then addressed conservation. Conservation of density is guaranteed by the scheme owing to symmetric fluxes and zero-flux boundary conditions in velocity space, but momentum and energy conservation were shown to be subject to constraint equations. We chose to add correction quantities onto $\mathbf{a}_h$ and $\mathbf{D}_h$ that are functions of only configuration space. The constraint equations were then reformulated as a linear system that is solved each timestep for the corrections, which, when accumulated onto $\mathbf{a}_h$

and $\mathbf{D}_h$, guarantee conservation of momentum and energy.

Finally, we enumerated the steps in the FPO algorithm before progressing onto testing. The first test was for convergence of the recovery-based DG scheme for differentiation required by the drag and diffusion coefficient calculation and the full FPO. By preparing an exact solution, we found that the drag coefficient and diagonal components of the diffusion tensor demonstrated 6^{th} and 5^{th} order convergence for a $p = 2$ Serendipity basis set, as expected. However, the off-diagonal diffusion tensor elements converge only to 3^{rd} order. It is likely that this reduced order of accuracy is due to the 6-cell recovery operation, and further study is warranted.

Following the convergence study, we initialized two simulations to investigate the differences between the FPO and LBO when relaxing a distribution to Maxwellian. The first of these cases was an initial square distribution, which showcases the ability of the recovery-based DG scheme to handle discontinuities in velocity space. We chose an average collision frequency for the LBO model that gives comparable results to the FPO, but there was still a discrepancy in the relaxation rate due to the velocity-dependent collision frequency of the FPO. At several collision times however, the two models reach identical Maxwellian distributions. As part of this test, we also verified that density and energy are conserved to machine precision.

The second test applied a bump-on-tail perturbation at high velocity to a stationary Maxwellian distribution function to emphasize the importance of velocity-dependent collisions when considering high-velocity particle dynamics. Choosing the same characteristic collision frequency as in the previous case, we see a significantly greater overestimation of the relaxation rate in the LBO for the high-velocity perturbation. This case also has nonzero momentum, so we were able to show that density, momentum, and energy are all conserved simultaneously. As a final assessment, we demonstrated the stability of the DG FPO algorithm by tracking the relative change of total entropy in time for the bump-on-tail distribution. Collisions drive

the distribution function to the maximum entropy solution (the Maxwellian), and entropy is expected to remain constant once equilibrium is reached. We verified that the entropy increases monotonically as the distribution function thermalizes, which is a strong case for the stability of the scheme.

We then finally introduced a physics problem. By initializing a Maxwellian distribution with a linear temperature variation between to constant temperature regions, we allowed the distribution function to evolve with collisions until the heat flux reached steady state. This steady state heat flux was then divided by the temperature gradient according to Fourier's law to compute a thermal conductivity. Braginskii [15] derived transport equations directly from the Fokker-Planck collision operator for a singly-ionized plasma, and the ion heat flux and thermal conductivity are valid approximations for our neutral single species case here. Simulations were performed with 8^3 and 16^3 cells in velocity space to get a qualitative sense of the convergence of our scheme. The lower resolution case had an error of approximately 11% off the theoretical value, while the higher resolution case was approximately 1% away.

Chapter 5

Summary and Conclusions

Collisional dynamics are relevant to plasmas in many regimes of interest, from astrophysical plasmas to the ionosphere to nuclear fusion reactors.

In this work, we utilized the novel modal discontinuous Galerkin Vlasov-Maxwell-Fokker-Planck (VFP) capabilities of the `Gkeyll` simulation framework to investigate hydrodynamic phenomena with a continuum-kinetic model. We demonstrated that the Rayleigh-Taylor (RT) instability, while almost exclusively studied with fluid models, exists in regimes of finite collisionality that results in altered instability behavior. For this purpose, we utilized the simple Bhatnagar-Gross-Krook collision operator with the Vlasov equation for a neutral species. First considering the simple case of a spatially constant collision frequency, the continuum-kinetic model was able to reproduce expected behavior in the low- and high-collisionality limits, i.e. particle interpenetration and the classical fluid result, respectively. We encountered an intermediate regime between these two limits in which the RT instability grew while displaying interface diffusion that would not be captured in an ideal fluid model. The phase space information contained in the distribution function was also leveraged to provide a unique view of the instability in terms of the particle energy-flux in which we directly saw the effect of particle collisions decreasing non-ideal transport. By computing the velocity-space moment of the deviation of the distribution function from Maxwellian, it was shown that the kinetic behavior was most prominent around the interface. Following this study, results were presented of RT instability evolution with spatially varying collision

frequency, as collision frequency is generally a function of density and temperature. Simulations demonstrated that the asymmetric collisionality about the fluid interface yield further departure from classical RT instability behavior. The lower collisionality region exhibits sufficient streaming diffusion to move upward and widen the interface even as the instability grows. These simulations of the Rayleigh-Taylor instability demonstrated the utility of the continuum-kinetic method in the context of a hydrodynamic regime, but we noted that the numerical method used in `Gkeyll` is equally applicable to more advanced models.

Building upon that idea, we presented an algorithm for the full nonlinear Rosenbluth/Fokker-Planck operator (FPO) using a recovery-based modal discontinuous Galerkin (DG) method and discussed the details of its implementation in `Gkeyll`. We noted that currently a predominant tactic used when performing VFP simulations is to expand the distribution function in a spherical basis set, reducing the degree of anisotropy of the problem and simplifying the FPO [6, 129]. One code of note has instead applied a finite volume scheme to the full nonlinear FPO and demonstrates flexibility and robustness while guaranteeing conservation [126]. We took a similar approach to the latter, applying the DG method to the advection-diffusion form of the FPO. Our implementation utilizes the recovery process to guarantee continuity across cell boundaries and increase the order of accuracy, and we demonstrated two extensions of the basic operation that are required for this algorithm. We discussed the details of the implementation, first for computing the necessary vector and tensor quantities, then for applying that same process to the semi-discrete FPO. By applying corrections to the drag coefficient and diffusion tensor, this algorithm guarantees conservation of number density, momentum, and energy as well.

Finally, we tested this DG FPO implementation in `Gkeyll` by applying it to a set of test cases. The high convergence order of the recovery-based scheme was confirmed for the drag coefficient and diagonal components of the diffusion tensor, though the off-diagonal portions

were lower order, even when not including the domain boundaries. For initial square and bump-on-tail distributions, it was shown that the FPO stably relaxes the distribution function to Maxwellian while being fully conservative and monotonically increasing the entropy. We saw that while the approximate Dougherty-Lenard-Bernstein operator (LBO) gave a similar result to the FPO for collisions at low velocity, it dramatically overestimated the effect of collisions in the high-energy tail, as expected. We then applied this model to a classic heat bath problem, where collisions maintain a temperature gradient between two static regions. The FPO simulation agreed well with the expected result, and we got a qualitative sense of convergence by noting that the relative error in thermal conductivity decreased from 11% to 1% by doubling the velocity space resolution.

The work covered in this thesis presents many potential directions for further research. Regarding the Rayleigh-Taylor simulations, further neutral species studies can be performed to investigate the dynamics of multiple initialized modes in the presence of kinetic interface diffusion, including the effect of various Atwood numbers. The continuum-kinetic model in `Gkeyll` can be applied to the RT instability in plasma regimes as well, as the efficient DG implementation allows for highly-parallelized simulations that reduce the computational expense of previously inaccessible regimes such as these. Research has been conducted on kinetic-scale physics in plasma RT instabilities, e.g. [132, 133], and it is worth exploring the effect of including even weak collisions into these regimes.

The novel algorithm developed in this work for the full nonlinear Rosenbluth/Fokker-Planck operator also provides an array of opportunities for future study. Algorithmically, further research on different recovery schemes to handle the cross derivatives in the diffusion term is warranted. A linear stability analysis for this system can also be conducted to verify and characterize this scheme more concretely. Continued development of efficient DG Poisson solvers will also serve to improve this method, as computation of the Rosenbluth potentials

directly from the distribution function itself will increase the physical accuracy for highly non-Maxwellian distributions.

This numerical scheme will also be extended to cross-species collisions in the near future. For multi-species collisions, considerations must be made on effective representation of the Rosenbluth potentials of the colder ions on the larger electron velocity space grid and enforcement of conservation properties, perhaps through utilization of an LBO for ion-electron collisions. A gyroaveraged FPO can be implemented in much the same way as the process presented here, but the Rosenbluth potentials will have to be computed via Poisson solve.

Further studies based on this work are certainly not limited to numerics, however. Because of the power of the matrix-free discontinuous Galerkin method, the continuum-kinetic model implemented in `Gkeyll` is able to perform simulations of previously unrealistic cases in a reasonable amount of wall clock time. This is especially true for this FPO implementation, as Fokker-Planck modeling has historically been generally inaccessible due to computational expense. Only relatively recently have numerical developments such as those discussed in Chapter 4 allowed for VFP simulations in reasonable time. Utilizing the algorithm and implementation outlined in this work, VFP simulations with high effective resolution can be performed quickly and with a high degree of parallelization. Potential applications of this work include studies of the impact of nonlocal electron transport in laser plasmas [16, 17], sheath formation [14, 114], ionospheric instabilities [2, 3], Rayleigh-Taylor in inertial confinement fusion [20, 103], and recycling in tokamak divertors [76, 112].

In conclusion, discontinuous Galerkin methods have dramatically increased the accessibility of continuum-kinetic methods for simulations of collisional dynamics. We have demonstrated that these methods can now be applied in certain hydrodynamic regimes to study the effects of finite collisionality on fluid-like behavior. By applying a novel DG method to the full nonlinear Fokker-Planck collision operator, we are now able to perform highly accurate

studies of collisional systems that were previously inaccessible.

Bibliography

[1] IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, 2019. doi: 10.1109/IEEESTD.2019.8766229.

[2] Lujain Almarhabi, Chirag R. Skolar, Wayne Scales, and Bhuvana Srinivasan. Investigating the impact of the latitudinal velocity profile on nonlinear gradient drift instability development in the subauroral ionosphere. *Radiation Effects and Defects in Solids*, 177(1-2):2–14, 2022. doi: 10.1080/10420150.2022.2063126. URL https://doi.org/10.1080/10420150.2022.2063126.

[3] Lujain Almarhabi, Chirag R. Skolar, Wayne Scales, and Bhuvana Srinivasan. Nonlinear Three-Dimensional Simulations of the Gradient Drift and Secondary Kelvin–Helmholtz Instabilities in Ionospheric Plasma Clouds. *Atmosphere*, 14(4), 2023. ISSN 2073-4433. doi: 10.3390/atmos14040676. URL https://www.mdpi.com/2073-4433/14/4/676.

[4] Douglas N. Arnold and Gerard Awanou. The Serendipity Family of Finite Elements. *Found. Comput. Math.*, 11(3):337–344, Mar 2011. ISSN 1615-3383. doi: 10.1007/s10208-011-9087-3.

[5] A R Bell and M Sherlock. The fastVFP code for solution of the Vlasov–Fokker–Planck equation. *Plasma Physics and Controlled Fusion*, 66(3):035014, feb 2024. doi: 10.1088/1361-6587/ad2278.

[6] A R Bell, A P L Robinson, M Sherlock, R J Kingham, and W Rozmus. Fast electron transport in laser-produced plasmas and the KALOS code for solution of the Vlasov–Fokker–Planck equation. 48(3):R37. doi: 10.1088/0741-3335/48/3/R01.

[7] Ratan Kumar Bera, Yang Song, and Bhuvana Srinivasan. The effect of viscosity and resistivity on Rayleigh-Taylor instability induced mixing in magnetized high-energy-density plasmas. *Journal of Plasma Physics*, 88(2):905880209, 2022. doi: 10.1017/S0022377821001343.

[8] H. A. Bethe. Supernova mechanisms. *Reviews of Modern Physics*, 62(4):801–866, October 1990. doi: 10.1103/RevModPhys.62.801.

[9] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.*, 94:511–525, May 1954. doi: 10.1103/PhysRev.94.511.

[10] Robert Bird, Nigel Tan, Scott V. Luedtke, Stephen Lien Harrell, Michela Taufer, and Brian Albright. VPIC 2.0: Next Generation Particle-in-Cell Simulations. *IEEE Transactions on Parallel and Distributed Systems*, 33(4):952–963, 2022. doi: 10.1109/TPDS.2021.3084795.

[11] A.V. Bobylev and V.A. Chuyanov. On the numerical solution of Landau's kinetic equation. *USSR Computational Mathematics and Mathematical Physics*, 16 (2):121–130, 1976. ISSN 0041-5553. doi: 10.1016/0041-5553(76)90109-9. URL https://www.sciencedirect.com/science/article/pii/0041555376901099.

[12] T.R Boehly, D.L Brown, R.S Craxton, R.L Keck, J.P Knauer, J.H Kelly, T.J Kessler, S.A Kumpan, S.J Loucks, S.A Letzring, F.J Marshall, R.L McCrory, S.F.B Morse, W Seka, J.M Soures, and C.P Verdon. Initial performance results of the OMEGA laser system. *Optics Communications*, 133(1):495 – 506, 1997. ISSN 0030-4018. doi: 10.1016/S0030-4018(96)00325-2. URL http://www.sciencedirect.com/science/article/pii/S0030401896003252.

[13] Ludwig Boltzmann. *Further Studies on the Thermal Equilibrium of Gas Molecules*. doi: 10.1142/9781848161337_0015.

[14] Kolter Bradshaw and Bhuvana Srinivasan. Energy-dependent implementation of secondary electron emission models in continuum kinetic sheath simulations. *Plasma Sources Science and Technology*, 33(3):035008, mar 2024. doi: 10.1088/1361-6595/ad331c. URL https://dx.doi.org/10.1088/1361-6595/ad331c.

[15] S. I. Braginskii. Transport Processes in a Plasma. *Reviews of Plasma Physics*, 1:205, January 1965.

[16] J. P. Brodrick, R. J. Kingham, M. M. Marinak, M. V. Patel, A. V. Chankin, J. T. Omotani, M. V. Umansky, D. Del Sorbo, B. Dudson, J. T. Parker, G. D. Kerbel, M. Sherlock, and C. P. Ridgers. Testing nonlocal models of electron thermal conduction for magnetic and inertial confinement fusion applications. *Physics of Plasmas*, 24(9): 092309, 09 2017. ISSN 1070-664X. doi: 10.1063/1.5001079.

[17] J P Brodrick, M Sherlock, W A Farmer, A S Joglekar, R Barrois, J Wengraf, J J Bissell, R J Kingham, D Del Sorbo, M P Read, and C P Ridgers. Incorporating kinetic effects on Nernst advection in inertial fusion simulations. *Plasma Physics and Controlled Fusion*, 60(8):084009, jun 2018. doi: 10.1088/1361-6587/aaca0b.

[18] P. Cagas. *Continuum Kinetic Simulations of Plasma Sheaths and Instabilities*. PhD thesis, Virginia Tech, 2018.

[19] Petr Cagas, Ammar H. Hakim, and Bhuvana Srinivasan. A boundary value "reservoir problem" and boundary conditions for multi-moment multifluid simulations of sheaths. *Physics of Plasmas*, 28(1):014501, 2021. doi: 10.1063/5.0024510.

[20] Matthew James Carrier. *Magnetohydrodynamic Simulations of Fast Instability Development in Pulsed-Power-Driven Explosions and Implosions of Electrical Conductors*. PhD thesis, Virginia Tech, 2024.

[21] L. Chacón. A non-staggered, conservative, $\nabla\cdot B\rightarrow=0$, finite-volume scheme for 3D implicit extended magnetohydrodynamics in curvilinear geometries. *Computer Physics Communications*, 163(3):143–171, 2004. ISSN 0010-4655. doi: 10.1016/j.cpc.2004.08.005. URL https://www.sciencedirect.com/science/article/pii/S0010465504004369.

[22] Jesse Chan and T. Warburton. On the penalty stabilization mechanism for upwind discontinuous Galerkin formulations of first order hyperbolic systems. 74 (12):3099–3110, 2017. ISSN 0898-1221. doi: 10.1016/j.camwa.2017.08.003. URL https://www.sciencedirect.com/science/article/pii/S0898122117304819.

[23] J.S Chang and G Cooper. A practical difference scheme for Fokker-Planck equations. *Journal of Computational Physics*, 6(1):1–16, 1970. ISSN 0021-9991. doi: 10.1016/0021-9991(70)90001-X. URL https://www.sciencedirect.com/science/article/pii/002199917090001X.

[24] Sydney Chapman and T.G. Cowling. *The Mathematical Theory of Non-uniform Gases*. Cambridge University Press. ISBN 052140844X.

[25] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer Cham, 2015. doi: 10.1007/978-3-319-22309-4.

[26] R. A. Chevalier and T. R. Gull. The outer structure of the Crab nebula. *The Astrophysical Journal*, 200:399–401, September 1975. doi: 10.1086/153802.

[27] R. A. Chevalier and R. I. Klein. On the Rayleigh-Taylor instability in stellar explosions. *The Astrophysical Journal*, 219:994–1007, February 1978. doi: 10.1086/155864.

[28] Roger A. Chevalier. The Interaction of Supernovae with the Interstellar Medium. *Annual Review of Astronomy and Astrophysics*, 15(1):175–196, 1977. doi: 10.1146/annurev.aa.15.090177.001135.

[29] B. Cockburn and C.W. Shu. The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998. ISSN 0021-9991. doi: 10.1006/jcph.1998.5892. URL http://www.sciencedirect.com/science/article/pii/S0021999198958922.

[30] B. Cockburn and C.W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001. ISSN 0885-7474.

[31] Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. *Discontinuous Galerkin Methods: Theory, Compuration and Applications*. Springer-Verlag Berlin, 2000. doi: 10.1007/978-3-642-59721-3.

[32] J. P. Dougherty. Model Fokker-Planck Equation for a Plasma and Its Solution. *The Physics of Fluids*, 7(11):1788–1799, 11 1964. ISSN 0031-9171. doi: 10.1063/1.2746779.

[33] R. P. Drake, D. R. Leibrandt, E. C. Harding, C. C. Kuranz, M. A. Blackburn, H. F. Robey, B. A. Remington, M. J. Edwards, A. R. Miles, T. S. Perry, R. J. Wallace, H. Louis, J. P. Knauer, and D. Arnett. Nonlinear mixing behavior of the three-dimensional Rayleigh–Taylor instability at a decelerating interface. *Physics of Plasmas*, 11(5):2829–2837, 2004. doi: 10.1063/1.1651492.

[34] R. E. Duff, F. H. Harlow, and C. W. Hirt. Effects of Diffusion on Interface Instability between Gases. *The Physics of Fluids*, 5(4):417–425, 1962. doi: 10.1063/1.1706634.

[35] Dale R. Durran. *Numerical Methods for Fluid Dynamics*. Springer New York, NY.

[36] T Ebisuzaki, T Shigeyama, K Nomoto, and Tokyo Univ. Rayleigh-Taylor instability and mixing in SN 1987A. *Astrophysical Journal; (USA)*, 344, 9 1989. ISSN 0004-637X. doi: 10.1086/185532. URL https://www.osti.gov/biblio/5357268.

[37] E. M. Epperlein and M. G. Haines. Plasma transport coefficients in a magnetic field by direct numerical solution of the Fokker–Planck equation. *The Physics of Fluids*, 29 (4):1029–1041, 04 1986. ISSN 0031-9171. doi: 10.1063/1.865901.

[38] E. M. Epperlein and R. W. Short. A practical nonlocal model for electron heat transport in laser plasmas. *Physics of Fluids B: Plasma Physics*, 3(11):3092–3098, 1991. doi: 10.1063/1.859789.

[39] E. M. Epperlein, G. J. Rickard, and A. R. Bell. Two-Dimensional Nonlocal Electron Transport in Laser-Produced Plasmas. *Phys. Rev. Lett.*, 61:2453–2456, Nov 1988. doi: 10.1103/PhysRevLett.61.2453.

[40] E.M. Epperlein. Implicit and Conservative Difference Scheme for the Fokker-Planck Equation. *Journal of Computational Physics*, 112(2):291–297, 1994. ISSN 0021-9991. doi: 10.1006/jcph.1994.1101. URL https://www.sciencedirect.com/science/article/pii/S0021999184711016.

[41] Richard Fitzpatrick. *Plasma Physics: An Introduction*. CRC Press, 2023.

[42] M. A. Gallis, T. P. Koehler, J. R. Torczynski, and S. J. Plimpton. Direct simulation Monte Carlo investigation of the Rayleigh-Taylor instability. *Phys. Rev. Fluids*, 1: 043403, Aug 2016. doi: 10.1103/PhysRevFluids.1.043403.

[43] P. H. Gaskell and A. K. C. Lau. Curvature-compensated convective transport: SMART, A new boundedness- preserving transport algorithm. *International Journal for Numerical Methods in Fluids*, 8(6):617–641, 1988. doi: 10.1002/fld.1650080602.

[44] Gregor Gassner, Frieder Lörcher, and Claus-Dieter Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. 224(2):1049–1063. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.11.004. URL https://www.sciencedirect.com/science/article/pii/S0021999106005651.

[45] Gkeyll, 2022. URL https://gkeyll.readthedocs.io.

[46] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb., Nov. Ser.*, 47:271–306, 1959.

[47] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Review*, 43(1):89–112, 2001. doi: 10.1137/S003614450036757X.

[48] A. Greco, F. Valentini, S. Servidio, and W. H. Matthaeus. Inhomogeneous kinetic effects related to intermittent magnetic discontinuities. *Phys. Rev. E*, 86:066405, Dec 2012. doi: 10.1103/PhysRevE.86.066405.

[49] S. F. Gull. The X-ray, Optical and Radio Properties of Young Supernova Remnants. *Monthly Notices of the Royal Astronomical Society*, 171(2):263–278, 05 1975. ISSN 0035-8711. doi: 10.1093/mnras/171.2.263.

[50] A. Hakim and J. Juno. Alias-free, matrix-free, and quadrature-free discontinuous Galerkin algorithms for (plasma) kinetic equations, 2020.

[51] A. H. Hakim, G. W. Hammett, and E. L. Shi. On discontinuous Galerkin discretizations of second-order derivatives, 2014.

[52] Ammar Hakim, Manaure Francisquez, et al. Conservative discontinuous Galerkin schemes for nonlinear Dougherty–Fokker–Planck collision operators. *Journal of Plasma Physics*, 86(4):905860403, 2020. doi: 10.1017/S0022377820000586.

[53] Amiram Harten, Peter D. Lax, and Bram van Leer. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1): 35–61, 1983. doi: 10.1137/1025002.

[54] P. Helander and D. J. Sigmar. *Collisional Transport in Magnetized Plasmas*. Cambridge University Press, 2005.

[55] Jan Hesthaven and Tim Warburton. High-order nodal discontinuous Galerkin methods for the Maxwell eigenvalue problem: One contribution of 13 to a Theme 'Short-wave scattering'. *Philosophical Transactions of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 362:493–524, 03 2004. doi: 10.1098/rsta.2003.1332.

[56] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, 2008.

[57] Charles Hirsch. *Numerical Computation of Internal and External Flows*, volume 2. Butterworth-Heinemann.

[58] Charles Hirsch. *Numerical Computation of Internal and External Flows*. Butterworth-Heinemann, 2008.

[59] J. D. Huba. *NRL Plasma Formulary*. Naval Research Laboratory, 2023.

[60] J. D. Huba and D. Winske. Rayleigh–Taylor instability: Comparison of hybrid and non-ideal magnetohydrodynamic simulations. *Physics of Plasmas*, 5(6):2305–2316, 1998. doi: 10.1063/1.872904.

[61] H.T. Huynh. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. *AIAA Paper AIAA 20074079*, pages 1–42, 01 2007.

[62] Byung-Il Jun. Interaction of a Pulsar Wind with the Expanding Supernova Remnant. *The Astrophysical Journal*, 499(1):282, may 1998. doi: 10.1086/305627.

[63] J. Juno, A. Hakim, J. TenBarge, E. Shi, and W. Dorland. Discontinuous Galerkin algorithms for fully kinetic plasmas. *Journal of Computational Physics*, 353:110–147, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2017.10.009. URL https://www.sciencedirect.com/science/article/pii/S0021999117307477.

[64] James Juno, James Drake, Adil Hassam, and Jacob Bedrossian. *A Deep Dive into the Distribution Function: Understanding Phase Space Dynamics with Continuum Vlasov-Maxwell Simulations*. PhD thesis, University of Maryland, College Park, 2020. AAI27838261.

[65] Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

[66] Dongkyu Kim, Janghoon Seo, Gahyung Jo, Jae-Min Kwon, and Eisung Yoon. Nonlinear Fokker-Planck collision operator in Rosenbluth form for gyrokinetic simulations using discontinuous Galerkin method. *Computer Physics Communications*, 279: 108459, 2022. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2022.108459. URL https://www.sciencedirect.com/science/article/pii/S0010465522001783.

[67] R.J. Kingham and A.R. Bell. An implicit Vlasov–Fokker–Planck code to model nonlocal electron transport in 2-D with magnetic fields. 194(1):1–34. ISSN 0021-9991. doi: 10.1016/j.jcp.2003.08.017. URL https://www.sciencedirect.com/science/article/pii/S0021999103004583.

[68] Yu. L. Klimontovich. *The Statistical Theory of Non-Equilibrium Processes in a Plasma*. The M.I.T. Press.

[69] JP Knauer, R Betti, DK Bradley, TR Boehly, TJB Collins, VN Goncharov, PW McKenty, DD Meyerhofer, VA Smalyuk, CP Verdon, et al. Single-mode, Rayleigh-Taylor growth-rate measurements on the OMEGA laser system. *Physics of Plasmas*, 7(1): 338–345, 2000.

[70] H.J. Kull. Theory of the Rayleigh-Taylor instability. *Physics Reports*, 206(5):197 – 325, 1991. ISSN 0370-1573. doi: 10.1016/0370-1573(91)90153-D. URL http://www.sciencedirect.com/science/article/pii/037015739190153D.

[71] C. C. Kuranz, R. P. Drake, M. J. Grosskopf, A. Budde, C. Krauland, D. C. Marion, A. J. Visco, J. R. Ditmar, H. F. Robey, B. A. Remington, A. R. Miles, A. B. R. Cooper, C. Sorce, T. Plewa, N. C. Hearn, K. L. Killebrew, J. P. Knauer, D. Arnett, and T. Donajkowski. Three-dimensional blast-wave-driven Rayleigh–Taylor instability and the effects of long-wavelength modes. *Physics of Plasmas*, 16(5):056310, 2009. doi: 10.1063/1.3099320.

[72] C. C. Kuranz, R. P. Drake, E. C. Harding, M. J. Grosskopf, H. F. Robey, B. A. Remington, M. J. Edwards, A. R. Miles, T. S. Perry, B. E. Blue, T. Plewa, N. C. Hearn, J. P. Knauer, D. Arnett, and D. R. Leibrandt. Two-dimensional blast-wave-driven Rayleigh-Taylor Instability: Experiment and Simulation. *The Astrophysical Journal*, 696(1):749–759, apr 2009. doi: 10.1088/0004-637x/696/1/749.

[73] Huilin Lai, Aiguo Xu, Guangcai Zhang, Yanbiao Gan, Yangjun Ying, and Sauro Succi. Nonequilibrium thermohydrodynamic effects on the Rayleigh-Taylor instability in compressible flows. *Phys. Rev. E*, 94:023106, Aug 2016. doi: 10.1103/PhysRevE.94.023106.

[74] Lev Davidovich Landau. The kinetic equation in the case of Coulomb interaction. Technical report, General Dynamics/Astronautics San Diego Calif, 1958.

[75] A. Lenard and Ira B. Bernstein. Plasma Oscillations with Diffusion in Velocity Space. *Phys. Rev.*, 112:1456–1459, Dec 1958. doi: 10.1103/PhysRev.112.1456.

[76] Yuzhi Li, Bhuvana Srinivasan, Yanzeng Zhang, and Xian-Zhu Tang. The plasma-sheath transition and Bohm criterion in a high recycling divertor. *Physics of Plasmas*, 30(6):063505, 06 2023. ISSN 1070-664X. doi: 10.1063/5.0147580.

[77] Richard L. Liboff. *Kinetic Theory: Classical, Quantum, and Relativistic Descriptions*. Springer New York, NY. ISBN 978-0-387-21775-8. doi: 10.1007/b97467.

[78] Kwok Ho Marcus Lo. *A Space-Time Discontinuous Galerkin Method for Navier-Stokes with Recovery*. PhD thesis, University of Michigan, 2011.

[79] John Loverich, Ammar Hakim, and Uri Shumlak. A Discontinuous Galerkin Method for Ideal Two-Fluid Plasma Equations. *Communications in Computational Physics*, 9 (2):240–268, 2011. doi: 10.4208/cicp.250509.210610a.

[80] Tengfei Luo, Jianchun Wang, Chenyue Xie, Minping Wan, and Shiyi Chen. Effects of compressibility and Atwood number on the single-mode Rayleigh-Taylor instability. *Physics of Fluids*, 32(1):012110, 01 2020. ISSN 1070-6631. doi: 10.1063/1.5131585.

[81] N. R. Mandell, A. Hakim, G. W. Hammett, and M. Francisquez. Electromagnetic full-f gyrokinetics in the tokamak edge with discontinuous Galerkin methods. *Journal of Plasma Physics*, 86(1):905860109, 2020. doi: 10.1017/S0022377820000070.

[82] Noah Mandell. *Magnetic Fluctuations in Gyrokinetic Simulations of Tokamak Scrape-Off Layer Turbulence*. PhD thesis, Princeton University, 2021.

[83] Maxima. Maxima, a Computer Algebra System. Version 5.47.0, 2023. URL https://maxima.sourceforge.io/.

[84] R Menikoff, RC Mjolsness, DH Sharp, and C Zemach. Unstable normal mode for Rayleigh-Taylor instability in viscous fluids. *The Physics of Fluids*, 20(12):2000–2004, 1977.

[85] Dwight R. Nicholson. *Introduction to Plasma Theory*. John Wiley & Sons, 1983.

[86] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable Parallel Programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, March 2008. ISSN 1542-7730. doi: 10.1145/1365490.1365500.

[87] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.

[88] Felix Parra. Lecture Note: Fokker-Planck collision oparator. URL https://www-thphys.physics.ox.ac.uk/people/FelixParra/CollisionalPlasmaPhysics/notes/lecI_fokkerplanck.pdf.

[89] O Pezzi, H Liang, J L Juno, P A Cassak, C L Vásconez, L Sorriso-Valvo, D Perrone, S Servidio, V Roytershteyn, J M TenBarge, and W H Matthaeus. Dissipation measures in weakly collisional plasmas. *Monthly Notices of the Royal Astronomical Society*, 505 (4):4857–4873, 05 2021. ISSN 0035-8711. doi: 10.1093/mnras/stab1516.

[90] Oliver Porth, Serguei S. Komissarov, and Rony Keppens. Three-dimensional magneto-hydrodynamic simulations of the Crab nebula. *Monthly Notices of the Royal Astronomical Society*, 438(1):278–306, 12 2013. ISSN 0035-8711. doi: 10.1093/mnras/stt2176.

[91] Oliver Porth, Serguei S. Komissarov, and Rony Keppens. Rayleigh-Taylor instability in magnetohydrodynamic simulations of the Crab nebula. *Monthly Notices of the Royal Astronomical Society*, 443(1):547–558, 07 2014. ISSN 0035-8711. doi: 10.1093/mnras/stu1082.

[92] P. Ramaprabhu, Guy Dimonte, Yuan-Nan Young, A. C. Calder, and B. Fryxell. Limits of the potential flow approach to the single-mode Rayleigh-Taylor problem. *Phys. Rev. E*, 74:066308, Dec 2006. doi: 10.1103/PhysRevE.74.066308.

[93] Rayleigh. Investigation of the Character of the Equilibrium of an Incompressible Heavy Fluid of Variable Density. *Proc. London Math. Soc.*, s1-14(1):170–177, 1882. doi: 10.1112/plms/s1-14.1.170.

[94] W H Reed and T R Hill. Triangular mesh methods for the neutron transport equation. 10 1973. URL https://www.osti.gov/biblio/4491151.

[95] H. G. Rinderknecht, H. Sio, C. K. Li, A. B. Zylstra, M. J. Rosenberg, P. Amendt, J. Delettrez, C. Bellei, J. A. Frenje, M. Gatu Johnson, F. H. Séguin, R. D. Petrasso, R. Betti, V. Yu. Glebov, D. D. Meyerhofer, T. C. Sangster, C. Stoeckl, O. Landen, V. A. Smalyuk, S. Wilks, A. Greenwood, and A. Nikroo. First Observations of Nonhydrodynamic Mix at the Fuel-Shell Interface in Shock-Driven Inertial Confinement Implosions. *Phys. Rev. Lett.*, 112:135001, Apr 2014. doi: 10.1103/PhysRevLett.112.135001.

[96] Hans G. Rinderknecht, M. J. Rosenberg, C. K. Li, N. M. Hoffman, G. Kagan, A. B. Zylstra, H. Sio, J. A. Frenje, M. Gatu Johnson, F. H. Séguin, R. D. Petrasso, P. Amendt, C. Bellei, S. Wilks, J. Delettrez, V. Yu. Glebov, C. Stoeckl, T. C. Sangster, D. D. Meyerhofer, and A. Nikroo. Ion Thermal Decoupling and Species Separation in Shock-Driven Implosions. *Phys. Rev. Lett.*, 114:025001, Jan 2015. doi: 10.1103/PhysRevLett.114.025001.

[97] John Rodman, Petr Cagas, Ammar Hakim, and Bhuvana Srinivasan. Kinetic interpretation of the classical Rayleigh-Taylor instability. *Phys. Rev. E*, 105:065209, Jun 2022. doi: 10.1103/PhysRevE.105.065209.

[98] John Rodman, James Juno, and Bhuvana Srinivasan. The Effect of Spatially Varying Collision Frequency on the Development of the Rayleigh–Taylor Instability. *The Astrophysical Journal*, 965(2):173, apr 2024. doi: 10.3847/1538-4357/ad3359.

[99] P.L Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981. ISSN 0021-9991. doi: 10.1016/0021-9991(81)90128-5. URL https://www.sciencedirect.com/science/article/pii/0021999181901285.

[100] M. J. Rosenberg, H. G. Rinderknecht, N. M. Hoffman, P. A. Amendt, S. Atzeni, A. B. Zylstra, C. K. Li, F. H. Séguin, H. Sio, M. Gatu Johnson, J. A. Frenje, R. D. Petrasso, V. Yu. Glebov, C. Stoeckl, W. Seka, F. J. Marshall, J. A. Delettrez, T. C. Sangster, R. Betti, V. N. Goncharov, D. D. Meyerhofer, S. Skupsky, C. Bellei, J. Pino, S. C. Wilks, G. Kagan, K. Molvig, and A. Nikroo. Exploration of the Transition from the Hydrodynamiclike to the Strongly Kinetic Regime in Shock-Driven Implosions. *Phys. Rev. Lett.*, 112:185001, May 2014. doi: 10.1103/PhysRevLett.112.185001.

[101] Marshall N. Rosenbluth, William M. MacDonald, and David L. Judd. Fokker-Planck Equation for an Inverse-Square Force. *Phys. Rev.*, 107:1–6, Jul 1957. doi: 10.1103/PhysRev.107.1.

[102] Irina Sagert, Jim Howell, Alec Staber, Terrance Strother, Dirk Colbry, and Wolfgang Bauer. Knudsen-number dependence of two-dimensional single-mode Rayleigh-Taylor fluid instabilities. *Phys. Rev. E*, 92:013009, Jul 2015. doi: 10.1103/PhysRevE.92.013009.

[103] Camille Samulski. *A study of the Rayleigh-Taylor Instability during deceleration in inertial confinement fusion relevant conditions*. PhD thesis, Virginia Tech, 2024.

[104] Daniel V. Schroeder. *An Introduction to Thermal Physics*. Oxford University Press, 01 2021. ISBN 9780192895547. doi: 10.1093/oso/9780192895547.001.0001.

[105] Nils W. Schween and Brian Reville. Converting between the Cartesian tensor and spherical harmonic expansion of solutions to the Boltzmann equation. *Journal of Plasma Physics*, 88(5):905880510, 2022. doi: 10.1017/S002237782200099X.

[106] D.H. Sharp. An overview of Rayleigh-Taylor instability. *Physica D: Nonlinear Phenomena*, 12(1):3 – 18, 1984. ISSN 0167-2789. doi: 10.1016/0167-2789(84)90510-4. URL http://www.sciencedirect.com/science/article/pii/0167278984905104.

[107] M. Sherlock, J. P. Brodrick, and C. P. Ridgers. A comparison of non-local electron transport models for laser-plasmas relevant to inertial confinement fusion. *Physics of Plasmas*, 24(8):082706, 08 2017. ISSN 1070-664X. doi: 10.1063/1.4986095.

[108] E. L. Shi, G. W. Hammett, T. Stoltzfus-Dueck, and A. Hakim. Gyrokinetic continuum simulation of turbulence in a straight open-field-line plasma. *Journal of Plasma Physics*, 83(3):905830304, 2017. doi: 10.1017/S002237781700037X.

[109] Takashi Shiroto, Akinobu Matsuyama, Nobuyuki Aiba, and Masatoshi Yagi. A mass-energy-conserving discontinuous Galerkin scheme for the isotropic multispecies Rosenbluth-Fokker-Planck equation. *Journal of Computational Physics*, 449:110813, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110813. URL https://www.sciencedirect.com/science/article/pii/S0021999121007087.

[110] Chi-Wang Shu. Total-Variation-Diminishing Time Discretizations. 9(6):1073–1084, 1988. doi: 10.1137/0909073.

[111] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988. ISSN 0021-9991. doi: 10.1016/0021-9991(88)90177-5. URL https://www.sciencedirect.com/science/article/pii/0021999188901775.

[112] A. Shukla, J. Roeltgen, et al. Direct Comparison of Gyrokinetic and Fluid Simulations of a Prospective Spherical Tokamak Pilot Plant Scrape-Off Layer. In *US-EU Transport Task Force Workshop*, 2024.

[113] Peter Silvester. Finite-element solution of homogeneous waveguide problems. 38: 313–317.

[114] C. R. Skolar, K. Bradshaw, J. Juno, and B. Srinivasan. Continuum kinetic investigation of the impact of bias potentials in the current saturation regime on sheath formation. *Physics of Plasmas*, 30(1):012504, 01 2023. ISSN 1070-664X. doi: 10.1063/5.0134656. URL https://doi.org/10.1063/5.0134656.

[115] VA Smalyuk, SX Hu, JD Hager, JA Delettrez, DD Meyerhofer, TC Sangster, and D Shvarts. Rayleigh-Taylor growth measurements in the acceleration phase of spherical implosions on OMEGA. *Physical review letters*, 103(10):105001, 2009.

[116] Y. Song. *Unstructured Nodal Discontinuous Galerkin Method for Convection-Diffusion Equations Applied to Neutral Fluids and Plasmas*. PhD thesis, Virginia Polytechic Institute and State University, 2020.

[117] Yang Song and Bhuvana Srinivasan. A survey of the effects of magnetic fields, resistivity, viscosity and thermal conduction on the Rayleigh-Taylor instability. *Radiation Effects and Defects in Solids: Incorporating Plasma Techniques and Plasma Phenomena*, 175(11/12):1009 – 1014, 2020. ISSN 1042-0150.

[118] Yang Song and Bhuvana Srinivasan. An efficient reconstruction algorithm for diffusion on triangular grids using the nodal discontinuous Galerkin method. *Computer Physics Communications*, 264:107873, 2021. ISSN 0010-4655. doi: 10.1016/j.cpc.2021.107873. URL https://www.sciencedirect.com/science/article/pii/S0010465521000308.

[119] Bhuvana Srinivasan and Ammar Hakim. Role of electron inertia and electron/ion finite Larmor radius effects in low-beta, magneto-Rayleigh-Taylor instability. *Physics of Plasmas*, 25(9):092108, 2018.

[120] Bhuvana Srinivasan and Xian-Zhu Tang. Mechanism for magnetic field generation and growth in Rayleigh-Taylor unstable inertial confinement fusion plasmas. *Physics of Plasmas*, 19(8):082703, 2012.

[121] Bhuvana Srinivasan and Xian-Zhu Tang. Mitigating hydrodynamic mix at the gas-ice interface with a combination of magnetic, ablative, and viscous stabilization. *EPL (Europhysics Letters)*, 107(6):65001, sep 2014. doi: 10.1209/0295-5075/107/65001.

[122] Bhuvana Srinivasan, Guy Dimonte, and Xian-Zhu Tang. Magnetic field generation in Rayleigh-Taylor unstable inertial confinement fusion plasmas. *Physical review letters*, 108(16):165002, 2012.

[123] James M Stone and Thomas Gardiner. The magnetic Rayleigh-Taylor instability in three dimensions. *The Astrophysical Journal*, 671(2):1726, 2007.

[124] W.T. Taitano, L. Chacón, A.N. Simakov, and K. Molvig. A mass, momentum, and energy conserving, fully implicit, scalable algorithm for the multi-dimensional, multi-species Rosenbluth–Fokker–Planck equation. 297:357–380. ISSN 0021-9991. doi: 10.1016/j.jcp.2015.05.025. URL https://www.sciencedirect.com/science/article/pii/S0021999115003599.

[125] W.T. Taitano, L. Chacón, and A.N. Simakov. An adaptive, conservative 0D-2V multi-species Rosenbluth–Fokker–Planck solver for arbitrarily disparate mass and temperature regimes. *Journal of Computational Physics*, 318:391–420, 2016. ISSN 0021-9991. doi: 10.1016/j.jcp.2016.03.071. URL https://www.sciencedirect.com/science/article/pii/S0021999116300729.

[126] W.T. Taitano, L. Chacón, and A.N. Simakov. An adaptive, implicit, conservative, 1D-2V multi-species Vlasov–Fokker–Planck multi-scale solver in planar geometry. *Journal of Computational Physics*, 365:173–205, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.03.007. URL https://www.sciencedirect.com/science/article/pii/S0021999118301591.

[127] Tomonor Takizuka and Hirotada Abe. A binary collision model for plasma simulation with a particle code. *Journal of Computational Physics*, 25(3):205–219, 1977. ISSN 0021-9991. doi: 10.1016/0021-9991(77)90099-7. URL https://www.sciencedirect.com/science/article/pii/0021999177900997.

[128] Geoffrey Ingram Taylor. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I. *Proc. R. Soc. Lond. A*, 201(1065):192–196, 1950. doi: 10.1098/rspa.1950.0052.

[129] A.G.R. Thomas, M. Tzoufras, A.P.L. Robinson, R.J. Kingham, C.P. Ridgers, M. Sherlock, and A.R. Bell. A review of Vlasov–Fokker–Planck numerical modeling of inertial confinement fusion plasma. 231(3):1051–1079. ISSN 0021-9991. doi: 10.1016/j.jcp.2011.09.028. URL https://www.sciencedirect.com/science/article/pii/S0021999111005717. Special Issue: Computational Plasma Physics.

[130] M. Tzoufras, A.R. Bell, P.A. Norreys, and F.S. Tsung. A Vlasov-Fokker-Planck code for high energy density physics. 230(17):6475–6494. ISSN 0021-9991. doi: 10.1016/

j.jcp.2011.04.034. URL https://www.sciencedirect.com/science/article/pii/S0021999111002828.

[131] M. Tzoufras, A. Tableman, F. S. Tsung, W. B. Mori, and A. R. Bell. A multi-dimensional Vlasov-Fokker-Planck code for arbitrarily anisotropic high-energy-density plasmas). *Physics of Plasmas*, 20(5):056303, 04 2013. ISSN 1070-664X. doi: 10.1063/1.4801750.

[132] T. Umeda and Y. Wada. Secondary instabilities in the collisionless Rayleigh-Taylor instability: Full kinetic simulation. *Physics of Plasmas*, 23(11):112117, 2016. doi: 10.1063/1.4967859.

[133] Takayuki Umeda and Yasutaka Wada. Non-MHD effects in the nonlinear development of the MHD-scale Rayleigh-Taylor instability. *Physics of Plasmas*, 24(7):072307, 07 2017. ISSN 1070-664X. doi: 10.1063/1.4991409.

[134] Bram van Leer and S. Nomura. Discontinuous Galerkin for Diffusion. In *17th AIAA Computational Fluid Dynamics*, pages 2005–5108, 06 2005. doi: 10.2514/6.2005-5108.

[135] Bram van Leer, Marcus Lo, and Marc Raalte. A Discontinuous Galerkin Method for Diffusion Based on Recovery. In *17th AIAA Computational Fluid Dynamics*, 06 2007. doi: 10.2514/6.2007-4083.

[136] Liang Wang, Ammar H. Hakim, Jonathan Ng, Chuanfei Dong, and Kai Germaschewski. Exact and locally implicit source term solvers for multifluid-Maxwell systems. 415:109510. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.109510. URL https://www.sciencedirect.com/science/article/pii/S0021999120302849.

[137] Liang Wang, Ammar H. Hakim, A. Bhattacharjee, and K. Germaschewski. Comparison of multi-fluid moment models with particle-in-cell simulations of collisionless magnetic

reconnection. *Physics of Plasmas*, 22(1):012108, 01 2015. ISSN 1070-664X. doi: 10.1063/1.4906063.

[138] Tie Wei and Daniel Livescu. Late-time quadratic growth in single-mode Rayleigh-Taylor instability. *Phys. Rev. E*, 86:046405, Oct 2012. doi: 10.1103/PhysRevE.86.046405.

[139] Andrew R. Winters, Rodrigo C. Moura, Gianmarco Mengaldo, Gregor J. Gassner, Stefanie Walch, Joaquim Peiro, and Spencer J. Sherwin. A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations. *Journal of Computational Physics*, 372:1–21, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.06.016. URL https://www.sciencedirect.com/science/article/pii/S0021999118303942.

[140] I. Zhuravleva, E. Churazov, A. A. Schekochihin, S. W. Allen, A. Vikhlinin, and N. Werner. Suppressed effective viscosity in the bulk intergalactic plasma. *Nature Astronomy*, 3(9):832–837, Sep 2019. ISSN 2397-3366. doi: 10.1038/s41550-019-0794-z.

Appendix A

Example Rosenbluth Potentials and Coefficients

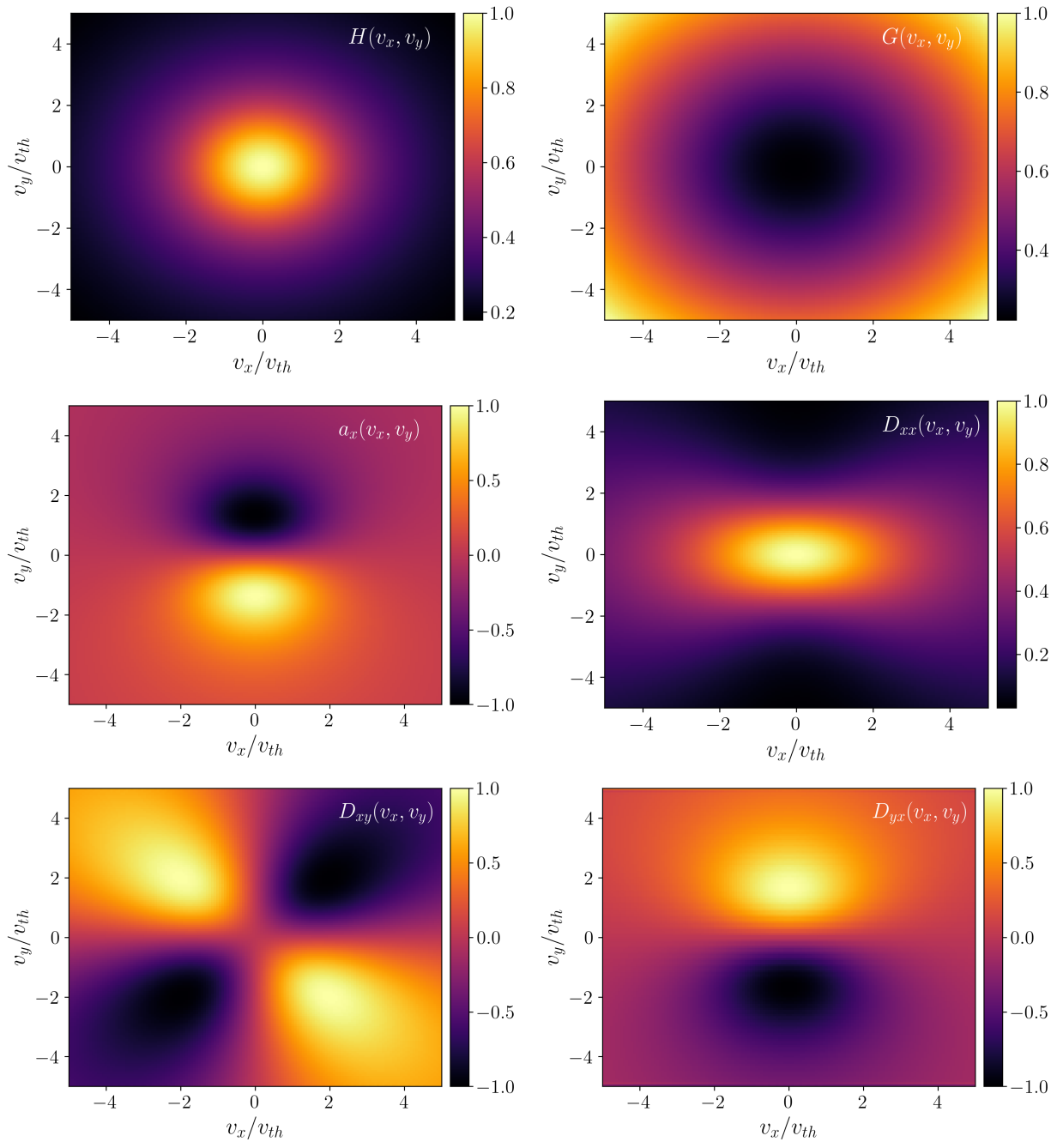

Figure A.1: Normalized examples of the Rosenbluth potentials, H (top left) and G (top right), the drag coefficient, $\mathbf{a}$ (a_x, center left), and elements of the diffusion tensor $\mathbf{D}$ (D_{xx} center right, D_{xy} bottom left, and D_{xz} bottom right) computed in the FPO update. Slices are evaluated at $v_z = 0.0$. These computations are performed with a $p = 2$ Serendipity basis and 48 cells in each velocity space direction, corresponding to an effective resolution of $48 * (p + 1) = 144$ cells.

Appendix B

Code Samples

Listed in this Appendix are example scripts that perform the recovery operations described in Chapters 2 and 4. These scripts are written for use in Maxima CAS, but the methodology can be applied in other symbolic computer algebra software. Additionally, these scripts utilize functions defined in the `Gkeyll` framework's kernel generation script repository, https://github.com/ammarhakim/gkylcas. For more details on the use of Maxima CAS in `Gkeyll`, see the `Gkeyll` documentation, https://gkeyll.readthedocs.io.

B.1 Recovery Scripts in Maxima

B.1.1 2-cell Recovery

```
/* Load Gkeyll functions */
load("modal-basis")$
load("recovery")$
basisType : "ser"$
cdim : 1$
vdim : 0$
polyOrder : 1$

/* Initialize the lists of variables and basis functions in configuration and phase space. */
[varsC, bC, varsP, bP] : loadPhaseBasis(basisType,cdim,vdim,polyOrder)$

/* Example analytic function to project. */
f : cos(x+0.5)$

/*  Compute expansion coefficients in left and right cells, [-2,0] and [0,2]. */
fL : shiftFunc(varsC, f, varsC, [-1], [2])$
fhL : calcInnerProdList(varsC, 1, bC, fL)$

fR : shiftFunc(varsC, f, varsC, [1], [2])$
fhR : calcInnerProdList(varsC, 1, bC, fR)$

/*  Manually construct system for 2-cell recovery. */
/* Define recovery polynomial and list of coefficients. */
dof : 2*polyOrder+2$
f_rec_p : sum(x^i*f_rec[i], i, 0, dof-1 )$
```

```
f_rec_lst : makelist(f_rec[i], i, 0, dof-1)$

/*  Shift basis functions into left and right cells and
    expand solutions in those cells. */
bL : shiftBasis(varsC, bC, varsC, [-1], [2])$
bR : shiftBasis(varsC, bC, varsC, [1], [2])$
fhL_e : doExpand(fhL, bL)$
fhR_e : doExpand(fhR, bR)$

/*
    Form equation system from  weak equality constraints in each cell.
    Boundary conditions would be included here as substitutions for the
    weak equality constraints on one side.
*/
eq_lst : []$
eq_lst : append(eq_lst, calcInnerProdListGen([x], [[-2, 0]], 1, bL, ↩
    ↪ f_rec_p-fhL_e))$
eq_lst : append(eq_lst, calcInnerProdListGen([x], [[0, 2]], 1, bR, ↩
    ↪ f_rec_p-fhR_e))$

/* Solve linear system for coefficients of recovery polynomial. */
sol : linsolve(eq_lst, f_rec_lst)$
f_rec2 : subst(sol, f_rec_p)$

/* This process is performed in Gkeyll using this 2-cell recovery function,
    which can also handle boundary condition inputs. */
f_rec2 : calcRecov2CellGen(basisType, x, varsC, 1, dg(fhL), dg(fhR))$
```

B.1.2 1-cell Recovery

```
load("modal-basis")$
load("recovery")$
basisType : "ser"$
cdim : 1$
vdim : 0$
polyOrder : 1$

/* Initialize the lists of variables and basis functions in configuration ↩
    ↪ and phase space. */
[varsC, bC, varsP, bP] : loadPhaseBasis(basisType,cdim,vdim,polyOrder)$
NC : length(bC)$

/*
    Example 3-cell stencil for 1-cell recovery of q in cell C in x direction.
        +----+----+----+
        | qL | qC | qR |
        +----+----+----+

*/

/* Construct lists of expansion coefficients in each cell. */
qL_lst : makelist(qL[i], i, 0, NC)$
qC_lst : makelist(qC[i], i, 0, NC)$
qR_lst : makelist(qR[i], i, 0, NC)$

/* Use Gkeyll 1-cell recovery function to compute recovery polynomial. */
q_rec1 : calcRecov1CellGen(basisType, x, varsC, polyOrder, 1,
    dg(qL_lst), dg(qC_lst), dg(qR_lst))$

```

```
/*
    If we are instead at a domain boundary, for example on the left edge,
    the qL input can be replaced by a BC.
        +----+----+----+
        | ++ | qC | qR |
        +----+----+----+

    In this case, we set the value to 0 and first derivative to 1.
*/
q_rec1 : calcRecov1CellGen(basisType, x, varsC, polyOrder, 1,
    bcs([der=0, val=0.0],[der=1, val=1.0]), dg(qC_lst), dg(qR_lst))$
```

B.1.3 6-cell Recovery

```
load("modal-basis")$
load("recovery")$
basisType : "ser"$
cdim : 2$
vdim : 0$
polyOrder : 1$

/* Initialize the lists of variables and basis functions in configuration ↩
    ↪ and phase space. */
[varsC, bC, varsP, bP] : loadPhaseBasis(basisType,cdim,vdim,polyOrder)$
NC : length(bC)$

/*
    Example stencil for 6-cell recovery of q in cell CC.
    Recovering in x-direction at y-boundary between CC and TC.
        +-----+-----+-----+
        | qTL | qTC | qTR |
        +-----+-----+-----+
        | qCL | qCC | qCR |
        +-----+-----+-----+
*/

/* Construct lists of expansion coefficients in each cell. */
qTL_lst : makelist(qTL[i], i, 0, NC)$
qTC_lst : makelist(qTC[i], i, 0, NC)$
qTR_lst : makelist(qTR[i], i, 0, NC)$
qCL_lst : makelist(qCL[i], i, 0, NC)$
qCC_lst : makelist(qCC[i], i, 0, NC)$
qCR_lst : makelist(qCR[i], i, 0, NC)$
```

```
29
30 /*  Use Gkeyll 1-cell recovery function to compute recovery polynomial.  */
31 recovDir : x$
32 interfaceDir : y$
33 numDer : 0$ /*  Number of derivatives to take of each 2-cell solution. */
34 faceLoc : 0.0$ /* Location of 2-cell recovery evaluation. Generally 0.0. */
35 C : 0$ /* Degree of continuity to enforce in final 1-cell recovery. */
36
37 q_rec6 : calcRecovFaceGen(basisType, [interfaceDir, recovDir],
38     varsC, numDer, faceLoc, polyOrder, C,
39     dg(dg(qCL_lst), dg(qTL_lst)), /* Pairs for 2-cell recovery. */
40     dg(dg(qCC_lst), dg(qTC_lst)),
41     dg(dg(qCR_lst), dg(qTR_lst)))$
42
43 /*
44     As an example, consider a left domain boundary.
45         +-----+-----+-----+
46         | +++ | qTC | qTR |
47         +-----+-----+-----+
48         | +++ | qCC | qCR |
49         +-----+-----+-----+
50     Boundary conditions are input as conditions on the 1-cell recovery.
51     Here, we require the value to be 1.0 and the first x-derivative to be 1.0.
52 */
53
54 q_rec6 : calcRecovFaceGen(basisType, [interfaceDir, recovDir],
55     varsC, numDer, faceLoc, polyOrder, C,
56     bcs([der=0, val=1.0], [der=1, val=0.0]),
57     dg(dg(qCC_lst), dg(qTC_lst)),
58     dg(dg(qCR_lst), dg(qTR_lst)))$
```

B.2 `Gkeyll` FPO Kernel Generation Maxima Examples

B.2.1 Drag Coefficient Kernel Generation

```
/* Prior to this section:
    - Set up dimensionality.
    - Begin iteration over domain stencil locations: stencilLoc.
    - Open kernel file, write header information and function signature.
   Also computing surface projection of drag coefficient in this section.
*/

################################################################################

[varsC, bC, varsP, bP] : loadPhaseBasis(basisFun, cdim, vdim, polyOrder),

 /* Initialize surface basis for surface projections. */
surfVars : delete(pv, varsP),
bSurf : basisFromVars(basisFun, surfVars, polyOrder),

NC : length(bP),
numSurf : length(bSurf),

H_L_lst : makelist(H_L[i], i, 0, NC-1),
H_C_lst : makelist(H_C[i], i, 0, NC-1),
H_R_lst : makelist(H_R[i], i, 0, NC-1),

H_C_e : doExpand1(H_C, bP),

/* Surface projection of H */
dHdv_surf_C_e : doExpand1(dHdv_surf_C, bSurf)/dv1,

```

```
28 if stencilLoc[sI] = [0] then (
29     /* Domain interior
30         +-----+-----+-----+
31         | H_L | H_C | H_R |
32         +-----+-----+-----+
33     */
34
35     /* 1-cell recovery for drag coefficient in cell volume */
36     H_rec1 : calcRecov1CellGen(basisFun, pv, varsP, polyOrder, 1,
37         dg(H_L_lst), dg(H_C_lst), dg(H_R_lst)),
38
39     /* 2-cell recovery at lower cell boundary for surface projection */
40     H_rec2 : calcRecov2CellGen(basisFun, pv, varsP, polyOrder,
41         dg(H_L_lst), dg(H_C_lst))
42 )
43 elseif stencilLoc[sI] = [-1] then (
44     /* Left edge
45         +-----+-----+-----+
46         | +++ | H_C | H_R |
47         +-----+-----+-----+
48     */
49
50     /* 1-cell recovery for drag coefficient in cell volume */
51     H_rec1 : calcRecov1CellGen(basisFun, pv, varsP, polyOrder, 1,
52         bcs([der=1, val=dHdv_surf_C_e]), dg(H_C_lst), dg(H_R_lst)),
53
54     /* No surface projection of drag coefficient at lower boundary because ↩
       ↪ zero-flux BCs */
55     H_rec2 : 0
56 )
57
```

```
58 elseif stencilLoc[sI] = [1] then (
59     /* Right edge
60         +-----+-----+-----+
61         | H_L | H_C | +++ |
62         +-----+-----+-----+
63     */
64
65     /* 1-cell recovery for drag coefficient in cell volume */
66     H_rec1 : calcRecov1CellGen(basisFun, pv, varsP, polyOrder, 1,
67         dg(H_L_lst), dg(H_C_lst), bcs([der=1, val=dHdv_surf_C_e])),
68
69     /* 2-cell recovery at lower cell boundary for surface projection */
70     H_rec2 : calcRecov2CellGen(basisFun, pv, varsP, polyOrder,
71         dg(H_L_lst), dg(H_C_lst))
72 ),
73
74 /* Compute drag coefficient by differentiating H */
75 drag_coeff_out : calcInnerProdList(varsP, 1, bP, diff(H_rec1, pv)),
76 drag_coeff_surf_out : calcInnerProdList(surfVars, 1, bSurf,
77     subst(pv=0.0,diff(H_rec2, pv))),
78
79 ########################################################################
80 /*
81     After computing expansion coefficients of the drag coefficient,
82     write them out to the kernel file.
83 */
```

B.2.2 Diffusion Coefficient Kernel Generation

This section shows examples of different recoveries required in computing the off-diagonal elements of the diffusion tensor, Eq. 4.25. Note the 6-cell recovery is performed at a vertical (pv2) cell boundary in the horizontal direction (pv1). Cases shown:

1. Domain interior
2. Left edge
3. Bottom edge
4. Top left corner

In each case, we need two 2-cell recoveries and two 6-cell recoveries of G for the four surface contributions. These are then used to compute surface expansions of G and $\partial G/\partial v_i$, respectively, for the final calculation of the diffusion tensor.

```
1  ##########################################################################
2  /* Components of G for upcoming recovery */
3  GTL_lst : makelist(GTL[i], i, 0, NC-1),
4  GTC_lst : makelist(GTC[i], i, 0, NC-1),
5  GTR_lst : makelist(GTR[i], i, 0, NC-1),
6  GCL_lst : makelist(GCL[i], i, 0, NC-1),
7  GCC_lst : makelist(GCC[i], i, 0, NC-1),
8  GCR_lst : makelist(GCR[i], i, 0, NC-1),
9  GBL_lst : makelist(GBL[i], i, 0, NC-1),
10 GBC_lst : makelist(GBC[i], i, 0, NC-1),
11 GBR_lst : makelist(GBR[i], i, 0, NC-1),
12
13 GCC_e : doExpand1(GCC, bP),
14
15 /* Surface expansions of G_pv2 and (dG/dv2)_pv1 */
16 G_surf_CC_pv2_e : doExpand1(eval_string(sconcat("G_surf_CC_",pv2)), bSurf2),
17 G_surf_CC_pv1_e : doExpand1(eval_string(sconcat("G_surf_CC_",pv1)), bSurf1),
18 dGdpv2_surf_CC_pv1_e : ↩
       ↪ doExpand1(eval_string(sconcat("dGd",pv2,"_surf_CC_",pv1)), bSurf1)/dv2,
19 dGdpv1_surf_CC_pv2_e : ↩
       ↪ doExpand1(eval_string(sconcat("dGd",pv1,"_surf_CC_",pv2)), bSurf2)/dv1,
20 ##########################################################################
```

```
################################################################################
if (stencilLoc[sI] = [0, 0]) then (
    /*
      Center
      +-----+-----+-----+
      | GTL | GTC | GTR |
      +-----+-----+-----+
      | GCL | GCC | GCR |
      +-----+-----+-----+
      | GBL | GBC | GBR |
      +-----+-----+-----+
    */
    /* 2- and 6-cell recoveries of G */
    G_rec2_lo_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCL_lst), dg(GCC_lst)),
    G_rec2_up_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCC_lst), dg(GCR_lst)),

    G_rec6_lo_p : calcRecovFaceGen(basisFun, [pv2, pv1], varsP, 0, 0, ↩
    ↪ polyOrder, 0,
      dg(dg(GBL_lst), dg(GCL_lst)), dg(dg(GBC_lst), dg(GCC_lst)),
      dg(dg(GBR_lst), dg(GCR_lst))),
    G_rec6_up_p : calcRecovFaceGen(basisFun, [pv2, pv1], varsP, 0, 0, ↩
    ↪ polyOrder, 0,
      dg(dg(GCL_lst), dg(GTL_lst)), dg(dg(GCC_lst), dg(GTC_lst)),
      dg(dg(GCR_lst), dg(GTR_lst))),

    /* Surface projections of (G_rec)_v1 and (dG/dv1)_v2 */
    G_rec_lo_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
    ↪ G_rec2_lo_p)),
```

```
26     G_rec_up_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
       ↪ G_rec2_up_p)),
27
28     dG_rec_lo_c : calcInnerProdList(surfVars2, 1, bSurf2,
29       subst(pv2=0, diff(G_rec6_lo_p, pv1))),
30     dG_rec_up_c : calcInnerProdList(surfVars2, 1, bSurf2,
31       subst(pv2=0, diff(G_rec6_up_p, pv1)))
32   )
33 ##############################################################################
```

```
################################################################################
elseif (stencilLoc[sI] = [-1, 0]) then (
    /*
      [-1, 0]
      Left
      +-----+-----+-----+
      | +++ | GTC | GTR |
      +-----+-----+-----+
      | +++ | GCC | GCR |
      +-----+-----+-----+
      | +++ | GBC | GBR |
      +-----+-----+-----+
    */
    /* 2- and 6-cell recoveries of G */
    G_rec2_lo_p : 0,
    G_rec2_up_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCC_lst), dg(GCR_lst)),

    G_rec6_lo_p : calcRecovFaceGen(basisFun, [pv2, pv1], varsP, 0, 0, ↩
    ↪ polyOrder, 0,
      bcs([der=0, val=subst([pv1=-1,pv2=-1], GCC_e)]), dg(dg(GBC_lst), ↩
    ↪ dg(GCC_lst)),
      dg(dg(GBR_lst), dg(GCR_lst))),
    G_rec6_up_p : calcRecovFaceGen(basisFun, [pv2, pv1], varsP, 0, 0, ↩
    ↪ polyOrder, 0,
      bcs([der=0, val=subst([pv1=-1,pv2=1], GCC_e)]), dg(dg(GCC_lst), ↩
    ↪ dg(GTC_lst)),
      dg(dg(GCR_lst), dg(GTR_lst))),

    /* Surface projections of (G_rec)_v1 and (dG/dv1)_v2 */
```

```
    G_rec_lo_c : calcInnerProdList(surfVars1, 1, bSurf1, G_surf_CC_pv1_e),
    G_rec_up_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
    ↪ G_rec2_up_p)),

    dG_rec_lo_c : calcInnerProdList(surfVars2, 1, bSurf2,
      subst(pv2=0, diff(G_rec6_lo_p, pv1))),
    dG_rec_up_c : calcInnerProdList(surfVars2, 1, bSurf2,
      subst(pv2=0, diff(G_rec6_up_p, pv1)))
)
################################################################################
```

```
##############################################################################
elseif (stencilLoc[sI] = [0, -1]) then (
    /*
      [0, -1]
      Bottom
      +-----+-----+-----+
      | GTL | GTC | GTR |
      +-----+-----+-----+
      | GCL | GCC | GCR |
      +-----+-----+-----+
      | +++ | +++ | +++ |
      +-----+-----+-----+
    */

    /* 2- and 6-cell recoveries of G */
    G_rec2_lo_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCL_lst), dg(GCC_lst)),
    G_rec2_up_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCC_lst), dg(GCR_lst)),

    G_rec6_lo_p : 0,
    G_rec6_up_p : calcRecovFaceGen(basisFun, [pv2, pv1], varsP, 0, 0, ↩
    ↪ polyOrder, 0,
      dg(dg(GCL_lst), dg(GTL_lst)), dg(dg(GCC_lst), dg(GTC_lst)),
      dg(dg(GCR_lst), dg(GTR_lst))),

    /* Surface projections of (G_rec)_v1 and (dG/dv1)_v2 */
    G_rec_lo_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
    ↪ G_rec2_lo_p)),
    G_rec_up_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
```

```
    ↪ G_rec2_up_p)),
27
28      dG_rec_lo_c : calcInnerProdList(surfVars2, 1, bSurf2,
29        dGdpv1_surf_CC_pv2_e),
30      dG_rec_up_c : calcInnerProdList(surfVars2, 1, bSurf2,
31        subst(pv2=0, diff(G_rec6_up_p, pv1)))
32  )
33  ##############################################################################
```

```
##############################################################################
elseif (stencilLoc[sI] = [-1, 1]) then (
    /*
      [-1, 1]
      Top Left Corner
      +-----+-----+-----+
      | +++ | +++ | +++ |
      +-----+-----+-----+
      | +++ | GCC | GCR |
      +-----+-----+-----+
      | +++ | GBC | GBR |
      +-----+-----+-----+
    */
    /* 2- and 6-cell recoveries of G */
    G_rec2_lo_p : 0,
    G_rec2_up_p : calcRecov2CellGen(basisFun, pv1, varsP, polyOrder, ↩
    ↪ dg(GCC_lst), dg(GCR_lst)),

    G_rec6_lo_p : calcRecovFaceGen(basisFun, [pv2, pv1],
        varsP, 0, 0, polyOrder, 0,
      bcs([der=0, val=subst([pv1=-1,pv2=-1], GCC_e)]),
      dg(dg(GBC_lst), dg(GCC_lst)),
      dg(dg(GBR_lst), dg(GCR_lst))),
    G_rec6_up_p : 0,

    /* Surface projections of (G_rec)_v1 and (dG/dv1)_v2 */
    G_rec_lo_c : calcInnerProdList(surfVars1, 1, bSurf1, G_surf_CC_pv1_e),
    G_rec_up_c : calcInnerProdList(surfVars1, 1, bSurf1, subst(pv1=0, ↩
    ↪ G_rec2_up_p)),

```

```
    dG_rec_lo_c : calcInnerProdList(surfVars2, 1, bSurf2,
      subst(pv2=0, diff(G_rec6_lo_p, pv1))),
    dG_rec_up_c : calcInnerProdList(surfVars2, 1, bSurf2,
      dGdpv1_surf_CC_pv2_e)
)
################################################################################
```

```
##########################################################################
/* Compute each term of the discrete diffusion coefficient,
    and sum to get final expansion coefficients. */

/* surft1 = < psi- * d/dv1(G_rec) > dS_v2 */
surft1_lo_c : calcInnerProdList(surfVars2, 1, bSurf2, dG_rec_lo_e),
surft1_up_c : calcInnerProdList(surfVars2, 1, bSurf2, dG_rec_up_e),

/* surft2 = -< d/dv2(psi-) G_rec > dS_v1 */
surft2_lo_c : calcInnerProdList(surfVars1, 1, bSurf1, G_rec_lo_e),
surft2_up_c : calcInnerProdList(surfVars1, 1, bSurf1, G_rec_up_e),

/* vol = < G d^2/dv2 dv1(psi) > */
vol_c : calcInnerProdList(varsP, 1, diff(diff(bP, pv2), pv1), GCC_e),

/* Expansions of surface and volume terms */
surft1_lo_e : doExpand1(surft1_lo, bSurf2),
surft1_up_e : doExpand1(surft1_up, bSurf2),
surft2_lo_e : doExpand1(surft2_lo, bSurf1),
surft2_up_e : doExpand1(surft2_up, bSurf1),
vol_lst : makelist(vol[i], i, 0, NC-1),

kernel : calcInnerProdList(surfVars2, 1, subst(pv2=1, bP), surft1_up_e) +
    calcInnerProdList(surfVars2, -1, subst(pv2=-1, bP), surft1_lo_e) +
    calcInnerProdList(surfVars1, -1,
        subst(pv1=1, diff(bP, pv2)), surft2_up_e) +
    calcInnerProdList(surfVars1, 1,
        subst(pv1=-1, diff(bP, pv2)), surft2_lo_e) +
    vol_lst,

```

```
31 ##############################################################################
32 /*
33     After computing expansion coefficients of the drag coefficient,
34     write them out to the kernel file.
35 */
```