Summer 7-8-2022

A Progress Report on Numerical Methods for BGK-Type Kinetic Equations

Evan Habbershaw
University of Tennessee, Knoxville, ehabbers@vols.utk.edu

Steven M. Wise
University of Tennessee, Knoxville, swise1@utk.edu

A Progress Report on Numerical Methods for BGK-Type Kinetic Equations

Evan Habbershaw[†] and Steven M. Wise[*†]

[†]Department of Mathematics, The University of Tennessee, Knoxville, TN, 37996

July 8, 2022

Abstract

In this report we review some preliminary work on the numerical solution of BGK-type kinetic equations of particle transport. Such equations model the motion of fluid particles via a density field when the kinetic theory of rarefied gases must be used in place of the continuum limit Navier-Stokes and Euler equations. The BGK-type equations describe the fluid in terms of phase space variables, and, in three space dimensions, require 6 independent phase-space variables (3 for space and 3 for velocity) for accurate simulation. This requires sophisticated numerical algorithms and efficient code to realize predictions over desired space and time scales. In particular, stable numerical methods must be designed to handle potential discontinuities (shocks) and rarefaction waves in the solutions coming from conservative advection terms and, in addition, numerical stiffness owing to diffusive particle collision terms. Furthermore, the particle interaction terms are non-local in nature, adding yet another layer of complexity, and the interaction length scales of the non-local terms may be orders of magnitude different, when multiple particle species are involved. In this report, we outline strategies for generating efficient and stable numerical algorithms and code, including the use of (i) stable high-order finite volume methods, (ii) fully implicit and implicit-explicit (IMEX) time integration techniques, and (iii) adaptive time-phase-space multi-level methods. The preliminary codes, which will be demonstrated herein, are built in the commercial software package MATLAB for quick

[*]Corresponding author. Email address: swise1@utk.edu

and easy prototyping, but will later be translated into production software using modern open languages.

Key words: Boltzmann equation, BGK approximation, multi-species BGK models, finite volume schemes, MUSCL methods, numerically stiff equations, implicit-explicit time stepping strategies, Runge-Kutta methods.

Contents

1 Introduction

The Navier-Stokes and Euler equations, with which most computational fluid dynamicists are familiar, are used to describe the evolution of a fluid under the assumption that the constituent particles move in, essentially, lock-step motion. In other words, fluid particles with small space separation are assumed to have nearly identical velocity vectors. To be precise, suppose that the mean free path (diffusion length scale) of particles is denoted λ, and the characteristic spatial size of the problem is denoted L. The *Knudsen number* is defined as $K_n = \lambda/L$. When $K_n \ll 1$, the diffusion length scale is too small to resolve accurately, and, in fact, the individual motions and interactions of constituent particles can be coarse-grained (averaged out) without significant loss of fidelity [19]. Indeed, the Navier-Stokes equation, which is applicable in this physical regime, is a highly successful and accurate model.

However, when $K_n = \mathcal{O}(1)$, the Navier-Stokes equation is no longer valid, and particle interactions must be taken into account. Particles with small space separation could move in entirely contrary directions, and, in this regime, the Boltzmann transport equation is an important model of particle evolution [19]. It describes the distribution of particles as a function time, space (3 dimensions), and velocity (3 dimensions). The three dimensions of space and three dimensions of velocity comprise what is known as *phase space*. The Boltzmann equation is complicated not only by the high dimensionality of phase space but also the highly nonlinear, highly nonlocal nature of the collision (particle interaction) operator.

The *Vlasov-Boltzmann equation* for a single species dilute gas is given as follows:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{a} \cdot \nabla_{\boldsymbol{v}} f = Q[f](\boldsymbol{x}, \boldsymbol{v}, t), \tag{1}$$

where f is the density of particles at position $\boldsymbol{x}$ with velocity $\boldsymbol{v}$ at time t; and $\boldsymbol{a}$ is a particle acceleration determined by an external field, for example, an electric or magnetic field. The *Boltzmann transport equation* results from setting $\boldsymbol{a} \equiv \boldsymbol{0}$. The derivation of the equation is a simple exercise using the chain rule. In particular, the total time derivative of the distribution f can be realized as

$$\begin{aligned}\frac{d}{dt} f(\boldsymbol{x}, \boldsymbol{v}, t) &= \sum_{i=1}^{d} \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} + \sum_{i=1}^{d} \frac{\partial f}{\partial v_i} \frac{dv_i}{dt} + \frac{\partial f}{\partial t} \\ &= \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{a} \cdot \nabla_{\boldsymbol{v}} f + \frac{\partial f}{\partial t}.\end{aligned}$$

Thus, the Vlasov-Boltzmann equation may be realized through the following law: the total

time derivative of the distribution is equal to the collision operator. In other words,

$$\frac{d}{dt}f(\boldsymbol{x},\boldsymbol{v},t) = Q[f](\boldsymbol{x},\boldsymbol{v},t).$$

The *Bolztmann collision operator*, $Q[f]$, requires much more physical insight for a clear derivation [19], involving conservation principles, in particular. Therefore, we will content ourselves by only stating its generic form:

$$Q[f](\boldsymbol{x},\boldsymbol{v},t) = \int_{\mathbb{R}^3\times\mathbb{S}^2} \left[f\left(\boldsymbol{x},\boldsymbol{v}',t\right) f\left(\boldsymbol{x},\boldsymbol{v}'_\star,t\right) - f\left(\boldsymbol{x},\boldsymbol{v},t\right) f\left(\boldsymbol{x},\boldsymbol{v}_\star,t\right)\right] B(|\boldsymbol{v}-\boldsymbol{v}_\star|,\boldsymbol{\sigma}) d\boldsymbol{\sigma} d\boldsymbol{v}_\star,$$

where B is the collision kernel describing interactions between particles and $\boldsymbol{\sigma}$ is the unit vector in the scattering direction $\boldsymbol{v}-\boldsymbol{v}_\star$. The velocities of the two interacting particles before collision, $\boldsymbol{v}'$ and $\boldsymbol{v}'_\star$, can be expressed in terms of the velocities of the particles after collision, $\boldsymbol{v}$ and $\boldsymbol{v}_\star$, via the expressions

$$\boldsymbol{v}' = \frac{\boldsymbol{v}+\boldsymbol{v}_\star}{2} + \frac{|\boldsymbol{v}-\boldsymbol{v}_\star|}{2}\boldsymbol{\sigma}, \quad \boldsymbol{v}'_\star = \frac{\boldsymbol{v}+\boldsymbol{v}_\star}{2} - \frac{|\boldsymbol{v}-\boldsymbol{v}_\star|}{2}\boldsymbol{\sigma}.$$

The computation of the Boltzmann collision operator is very expensive. For the 3D case, the (5 dimensional) integral must be computed at every value of $(\boldsymbol{x},\boldsymbol{v})$ in phase space. Additionally, the integral cannot be evaluated analytically, except for the simplest of cases (e.g., Maxwell molecules, with carefully prepared initial conditions). Thus, computation of the collision operator is usually the most expensive part of computing numerical solutions of the Boltzmann equation. For this reason, Monte Carlo methods are generally preferred for numerical simulation [15]. Unfortunately, these are slow to converge, as is well known. What is worse, for the N_s-species case, there are N_s^2 collision operators that must be considered. Thus, a simpler, less costly approximation is typically desired and implemented.

The *Vlasov-BGK equation* — BGK stands for the names Bhatnagar, Gross, and Krook, who introduced it 1954 [1] — is a model derived by approximating the Boltzmann collision operator with a simpler nonlinear, nonlocal relaxation operator of the form

$$Q^{BGK}[f](\boldsymbol{x},\boldsymbol{v},t) = \lambda(M_f(\boldsymbol{x},\boldsymbol{v},t) - f(\boldsymbol{x},\boldsymbol{v},t)), \tag{2}$$

where $\lambda = \frac{1}{\tau}$ is the collision frequency between particles, $\tau > 0$ is a characteristic time, and

M_f is the *Maxwellian*, which is defined by

$$M_f(\boldsymbol{x}, \boldsymbol{v}, t) := n\left(\frac{m}{2\pi\theta}\right)^{\frac{d}{2}} \exp\left(-\frac{m|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}\right), \tag{3}$$

$$n(\boldsymbol{x}, t) := \int_{\mathbb{R}^d} f \, d\boldsymbol{v}, \tag{4}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) := \frac{1}{n}\int_{\mathbb{R}^d} f\boldsymbol{v} \, d\boldsymbol{v}, \tag{5}$$

$$E(\boldsymbol{x}, t) := \int_{\mathbb{R}^d} \frac{1}{2}|\boldsymbol{v}|^2 f \, d\boldsymbol{v}, \tag{6}$$

$$\frac{d}{2}n(\boldsymbol{x}, t)\theta(\mathbf{x},t) := \int_{\mathbb{R}^d} \frac{m}{2}|\boldsymbol{v}-\boldsymbol{u}|^2 f \, d\boldsymbol{v}. \tag{7}$$

Here m stands for the mass of the particles, and the space-time fields n, $\boldsymbol{u}$, E, and θ, are the macroscopic (coarse grained) density, velocity, energy and temperature, respectively. Using this approximation drastically reduces the computational cost of the simulation of dilute gases, and recovers both equilibrium and streaming behavior of the Boltzmann equation in collision-dominated, and collision-free limits. The BGK collision operator also satisfies conservation and entropy properties of the Boltzmann operator, as we show below. Of course, the basic design and principle of the BGK approximation is that the density f should relax over time toward the Maxwellian, and, clearly, at this state, the collision operator gives a zero contribution.

However, the BGK collision operator does not capture some important properties. First, it fails to capture the correct Prandtl number (essentially the ratio of viscosity to thermal conductivity), largely because the collision rate is velocity independent. As a result, the model may not agree with the compressible Navier-Stokes Equations that are derived from the Boltzmann equation in high collision regimes. A number of generalizations of the BGK model have been proposed to deal with this shortcoming. These include the ES-BGK [6] and Shakov [18] models, which incorporate extra degrees of freedom. The model of Mieussens and Struchtrup [11] incorporates a velocity dependent collision rate, which improves the capture of the correct Prandtl number. These models give a more physically realistic model, but cause a substantial increase to computational costs.

To summarize, the single-species Vlasov-BGK equation has the form

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f + \boldsymbol{a}\cdot\nabla_{\boldsymbol{v}} f = \lambda(M_f(\boldsymbol{x}, \boldsymbol{v}, t) - f(\boldsymbol{x}, \boldsymbol{v}, t)), \tag{8}$$

and, in analogy to the previous setting, the *single-species BGK equation* results from setting

$\boldsymbol{a} \equiv \mathbf{0}$. When, in equation 8, the acceleration is determined by an electric field, according to the model

$$\boldsymbol{a}(\boldsymbol{x}, t) = -\chi \nabla_x \Phi(\boldsymbol{x}, t),$$

where $\chi > 0$ is a constant, and

$$-\Delta\Phi(\boldsymbol{x}, t) = n(\boldsymbol{x}, t),$$

the resulting model is known as the *Vlasov-Poisson-BGK equation*.

Numerical methods for the BGK family of equations have been proposed and analyzed extensively. The discretization of phase space is costly, and there are challenges that occur in high-collision regimes. First, the BGK operator becomes stiff as the collision frequency becomes large, so an implicit approach is desired for this term, to avoid unacceptably small time steps. In a seminal paper[2], it was shown that a Backward Euler step could be applied in an explicit manner, allowing stable time stepping for a wide range of collisional regimes. Taking advantage of this, the paper [15] introduced an implicit-explicit (IMEX) Runge-Kutta scheme, treating the convection term explicitly and the collision term implicitly. Second, stable and accurate high-order conservation schemes are required for the convection terms. For example, Pieraccini and Puppo [15] use a high-order weighted essentially non-oscillatory (WENO) finite volume scheme, though high-order flux-limited schemes perform well, as we show, and are easily scaled to higher dimensions.

Most of the numerical schemes for the BGK equation, including the one that we focus on here from [15], lose the exact conservation of mass, momentum and energy at the discrete level, and, additionally, the entropy dissipation at the discrete level. This shortcoming is overcome in the work by Mieussens [9, 10]. Exact conservation is obtained by computing a discrete equilibrium function, which requires the solution of a nonlinear system of 5 equations for the BGK model and a nonlinear system of 10 equations for the ES-BGK model at each grid point in space. Another challenge is to recover consistent numerical solutions of the Euler and Navier-Stokes Equations for compressible flows, which can be derived from the BGK model using a Chapman-Enskog expansion [19].

In some situations, an explicit treatment of the advection terms in the BGK equation — even while giving the collision operator an implicit treatment — can lead to a method that requires excessively small time steps for stability and accuracy [3]. This can happen when there are long time scales that lead to incompressible equations in the high collision limit, or in problems for which the maximum velocity in the computational domain is significantly

larger than the fluid speed of sound. A fully implicit approach can be taken to address these issues. This is common for time dependent kinetic equations in radiation transport contexts, and has been considered for electron transport problems [3]. Fully implicit methods for dilute gases and collisionless plasmas have been proposed in [3]. These approaches use sophisticated iterative methods to manage the cost and memory requirements of the implicit update.

In this progress report, some background for BGK-type models is presented. We do not conduct an extensive review but give the reader (especially those unfamiliar with kinetic equations) a gentle, albeit brief, introduction. In Section 2 we introduce the single species BGK Equation. We describe some basic theory (Sections 2.1 – 2.3), deriving conservation and (mathematical) entropy dissipation properties. Finite volume implicit-explicit (IMEX) Runge-Kutta (RK) numerical methods are presented in Section 2.4. We give some preliminary simulation results in Section 2.4, for problems including the Sod shock tube benchmark and two-stream instability. In Section 3, we give a very brief introduction to multispecies BGK models in order to explain some numerical challenges for such equations. We conclude the report with a brief summary of preliminary work and near future work in Section 4. A prototype $1x1v$ MATLAB code for the Sod shock tube problem is contained in Appendix B.

2 Single Species BGK Kinetic Models

In this section, we describe some basic theory for the single species BGK equation [1, 19], namely,

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f = \lambda \left(M_f - f \right). \tag{9}$$

The existence of nonnegative solutions to the (single species) BGK equation was proved for $\boldsymbol{x} \in \mathbb{R}^d$ by Perthame (1989) [14], and on bounded domains by Ringeissen (1991) [16]. Uniqueness of mild solutions for the (single species) periodic (in $\boldsymbol{x}$) case was shown by Perthame and Pulvirenti (1993) [13], and extended to the full space, $\boldsymbol{x} \in \mathbb{R}^d$, by Mischler (1996) [12].

Beyond questions about existence and uniqueness of solutions, it is important to have a firm grasp of the properties of solutions to BGK-type equations, since it is vital to build numerical approximation schemes that respect analogous features at the fully discrete level. Herein, we review conservation and entropy dissipation solution properties, and, later, we discuss how these are used in the design of numerical schemes.

2.1 Collision Invariants

Let us begin with an important property for the well-definedness of the model. In particular, the temperature is non-negative, since f is nonnegative. To see this, let us show that

Proposition 2.1. *If $f : \Omega \times \mathbb{R}^d \times [0, \infty) \to \mathbb{R}$ is non-negative, then n is nonnegative and*

$$\frac{d}{2} n(\boldsymbol{x}, t)\theta(\boldsymbol{x}, t) = \frac{1}{2} \int_{\mathbb{R}^d} |\boldsymbol{v} - \boldsymbol{u}(\boldsymbol{x}, t)|^2 f(\boldsymbol{x}, \boldsymbol{v}, t)\, d\boldsymbol{v} \geq 0.$$

Thus, the temperature is nonnegative.

Proof. Expanding the right-hand side of the last equation, we have

$$\begin{aligned}
\frac{1}{2} \int_{\mathbb{R}^d} |\boldsymbol{v} - \boldsymbol{u}(\boldsymbol{x}, t)|^2 f(\boldsymbol{x}, \boldsymbol{v}, t)\, d\boldsymbol{v} &= \frac{1}{2} \int_{\mathbb{R}^d} |\boldsymbol{v}|^2 f(\boldsymbol{x}, \boldsymbol{v}, t)\, d\boldsymbol{v} - \boldsymbol{u}(\boldsymbol{x}, t) \cdot \int_{\mathbb{R}^d} \boldsymbol{v} f(\boldsymbol{x}, \boldsymbol{v}, t)\, d\boldsymbol{v} \\
&\quad + \frac{|\boldsymbol{u}(\boldsymbol{x}, t)|^2}{2} \int_{\mathbb{R}^d} f(\boldsymbol{x}, \boldsymbol{v}, t)\, d\boldsymbol{v} \\
&= E(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{x}, t) \cdot \boldsymbol{u}(\boldsymbol{x}, t) n(\boldsymbol{x}, t) + \frac{1}{2} n(\boldsymbol{x}, t) |\boldsymbol{u}(\boldsymbol{x}, t)|^2 \\
&= E(\boldsymbol{x}, t) - \frac{1}{2} n(\boldsymbol{x}, t) |\boldsymbol{u}(\boldsymbol{x}, t)|^2 \\
&= \frac{d}{2} n(\boldsymbol{x}, t)\theta(\boldsymbol{x}, t).
\end{aligned}$$

The proof is complete. ■

For convenience, we define $\langle \cdot \rangle \colon C(\mathbb{R}^d; \mathbb{R}^p) \to \mathbb{R}^p$ by

$$\langle \boldsymbol{g} \rangle = \int_{\mathbb{R}^d} \boldsymbol{g}(\boldsymbol{v})\, d\boldsymbol{v},$$

for all $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^p$. Therefore,

$$\begin{bmatrix} n(\boldsymbol{x}, t) \\ n(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) \\ E(\boldsymbol{x}, t) \end{bmatrix} = \left\langle f(\boldsymbol{x}, \boldsymbol{v}, t) \begin{bmatrix} 1 \\ \boldsymbol{v} \\ \frac{1}{2}|\boldsymbol{v}|^2 \end{bmatrix} \right\rangle.$$

Thus, the macroscopic density, n, momentum, $\boldsymbol{p} \coloneqq n\boldsymbol{u}$, and energy, E, may be viewed as the first three moments of the particle density function f. The first three moments of the collision operator are equal to zero. This gives us some conservation properties, as we show below.

Lemma 2.2. *The following equality holds:*

$$\left\langle \begin{bmatrix} 1 \\ \boldsymbol{v} \\ \frac{|\boldsymbol{v}|^2}{2} \end{bmatrix} \lambda(M_f - f) \right\rangle = \mathbf{0}. \tag{10}$$

Proof. 1. First we show that

$$\int M_f \, d\boldsymbol{v} = \int f \, d\boldsymbol{v}. \tag{11}$$

By definition,

$$\begin{aligned} \int_{\mathbb{R}^d} M_f \, d\boldsymbol{v} &= \int_{\mathbb{R}^d} \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \exp\left(-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}\right) d\boldsymbol{v} \\ &= \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \exp\left(-\left|\frac{\boldsymbol{v}-\boldsymbol{u}}{(2\theta)^{\frac{1}{2}}}\right|^2\right) d\boldsymbol{v}. \end{aligned} \tag{12}$$

Consider the substitution

$$\boldsymbol{s} = \frac{\boldsymbol{v}-\boldsymbol{u}}{(2\theta)^{\frac{1}{2}}} \implies d\boldsymbol{s} = \left(\frac{1}{(2\theta)^{\frac{1}{2}}}\right)^d d\boldsymbol{v} \implies d\boldsymbol{v} = (2\theta)^{\frac{d}{2}} d\boldsymbol{s}. \tag{13}$$

The integral becomes

$$\begin{aligned} \int_{\mathbb{R}^d} M_f \, d\boldsymbol{v} &= \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \exp(-|\boldsymbol{s}|^2)(2\theta)^{\frac{d}{2}} d\boldsymbol{s} \\ &= \frac{n(2\theta)^{\frac{d}{2}}}{\pi^{\frac{d}{2}}(2\theta)^{\frac{d}{2}}} \int_{\mathbb{R}^d} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \\ &= \frac{n}{\pi^{\frac{d}{2}}} \cdot \pi^{\frac{d}{2}} \\ &= \int_{\mathbb{R}^d} f \, d\boldsymbol{v}, \end{aligned} \tag{14}$$

as desired. Note that we have used the definition $n = \int_{\mathbb{R}^d} f \, d\boldsymbol{v}$, and the fact that in d dimensions, the integral $\int_{\mathbb{R}^d} e^{-|\boldsymbol{x}|^2} d\boldsymbol{x} = \pi^{\frac{d}{2}}$. We will use this frequently.

2. Next we show that

$$\int_{\mathbb{R}^d} \boldsymbol{v} M_f \, d\boldsymbol{v} = \int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v}. \tag{15}$$

Following a similar line of work, we have

$$\begin{aligned}\int_{\mathbb{R}^d} \boldsymbol{v} M_f \, d\boldsymbol{v} &= \int_{\mathbb{R}^d} \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \boldsymbol{v} \exp\left(-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}\right) d\boldsymbol{v} \\ &= \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \boldsymbol{v} \exp\left(-\left|\frac{\boldsymbol{v}-\boldsymbol{u}}{(2\theta)^{\frac{1}{2}}}\right|^2\right) d\boldsymbol{v}. \end{aligned} \tag{16}$$

Using the same substitution as above, the integral becomes

$$\begin{aligned}\int_{\mathbb{R}^d} \boldsymbol{v} M_f \, d\boldsymbol{v} &= \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \left((2\theta)^{\frac{1}{2}}\boldsymbol{s} + \boldsymbol{u}\right) \exp(-|\boldsymbol{s}|^2) \cdot (2\theta)^{\frac{d}{2}} d\boldsymbol{s} \\ &= \frac{n(2\theta)^{\frac{d}{2}}}{\pi^{\frac{d}{2}}(2\theta)^{\frac{d}{2}}} \left[(2\theta)^{\frac{1}{2}} \int_{\mathbb{R}^d} \boldsymbol{s} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} + \boldsymbol{u} \int_{\mathbb{R}^d} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s}\right] \\ &= \frac{n}{\pi^{\frac{d}{2}}} \left[(2\theta)^{\frac{1}{2}} \int_{\mathbb{R}^d} \boldsymbol{s} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} + \boldsymbol{u} \int_{\mathbb{R}^d} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s}\right]. \end{aligned} \tag{17}$$

Let us deal with the first term in (17). One can easily show that

$$\int_{-\infty}^{\infty} x e^{-x^2} \, dx = 0 \implies \int_{\mathbb{R}^d} \boldsymbol{s} e^{-|\boldsymbol{s}|^2} \, d\boldsymbol{s} = \boldsymbol{0}.$$

Therefore, we have

$$\int_{\mathbb{R}^d} \boldsymbol{v} M_f \, d\boldsymbol{v} = \frac{n}{\pi^{\frac{d}{2}}} \cdot \boldsymbol{u} \int_{\mathbb{R}^d} e^{-|\boldsymbol{s}|^2} \, d\boldsymbol{s} = \frac{n\boldsymbol{u}}{\pi^{\frac{d}{2}}} \cdot \pi^{\frac{d}{2}} = n\boldsymbol{u} = \int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v}.$$

3. Finally, we show that

$$\int \frac{|\boldsymbol{v}|^2}{2} M_f \, d\boldsymbol{v} = \int \frac{|\boldsymbol{v}|^2}{2} f \, d\boldsymbol{v}. \tag{18}$$

First, observe that

$$\int_{\mathbb{R}^d} |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} = \frac{d}{2}\pi^{\frac{d}{2}}. \tag{19}$$

(To see how to prove such results, see Appendix A). Now, consider the LHS of (18):

$$\begin{aligned}\int \frac{|\boldsymbol{v}|^2}{2} M_f \, d\boldsymbol{v} &= \int \frac{|\boldsymbol{v}|^2}{2} \frac{n}{(2\pi\theta)^{\frac{d}{2}}} \exp\left(-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}\right) d\boldsymbol{v} \\ &= \frac{1}{2}\frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int |\boldsymbol{v}|^2 e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \\ &= \frac{1}{2}\frac{n}{(2\pi\theta)^{\frac{d}{2}}} \int |\boldsymbol{u} + (\boldsymbol{v}-\boldsymbol{u})|^2 e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\frac{n}{(2\pi\theta)^{\frac{d}{2}}}\int \left(|\boldsymbol{u}|^2 + 2\boldsymbol{u}^T(\boldsymbol{v}-\boldsymbol{u}) + |\boldsymbol{v}-\boldsymbol{u}|^2\right) e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \\
&= \frac{1}{2}\frac{n}{(2\pi\theta)^{\frac{d}{2}}}\int \left(|\boldsymbol{u}|^2 + 2\boldsymbol{u}^T\boldsymbol{v} - 2\boldsymbol{u}^T\boldsymbol{u} + |\boldsymbol{v}-\boldsymbol{u}|^2\right) e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \\
&= \frac{n}{2(2\pi\theta)^{\frac{d}{2}}}\int \left(-|\boldsymbol{u}|^2 + 2\boldsymbol{u}^T\boldsymbol{v} + |\boldsymbol{v}-\boldsymbol{u}|^2\right) e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \\
&= \frac{-n|\boldsymbol{u}|^2}{2(2\pi\theta)^{\frac{d}{2}}}\int e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} + \frac{2n}{2(2\pi\theta)^{\frac{d}{2}}}\boldsymbol{u}^T\int \boldsymbol{v} e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v} \\
&\quad + \frac{n}{2(2\pi\theta)^{\frac{d}{2}}}\int |\boldsymbol{v}-\boldsymbol{u}|^2 e^{-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta}} d\boldsymbol{v}. \qquad (20)
\end{aligned}
$$

Using the same substitution as above,

$$\boldsymbol{s} = \frac{\boldsymbol{v}-\boldsymbol{u}}{(2\theta)^{\frac{1}{2}}} \implies \boldsymbol{v}-\boldsymbol{u} = (2\theta)^{\frac{1}{2}}\boldsymbol{s} \implies \boldsymbol{v} = (2\theta)^{\frac{1}{2}}\boldsymbol{s} + \boldsymbol{u} \qquad (21)$$

$$d\boldsymbol{s} = \frac{1}{(2\theta)^{\frac{d}{2}}} d\boldsymbol{v} \implies d\boldsymbol{v} = (2\theta)^{\frac{d}{2}} d\boldsymbol{s}, \qquad (22)$$

the definition of the moment E, namely,

$$\int \frac{|\boldsymbol{v}|^2}{2} f\, d\boldsymbol{v} = E = \frac{1}{2}n|\boldsymbol{u}|^2 + \frac{d}{2}n\theta, \qquad (23)$$

and (19), we have

$$
\begin{aligned}
\int \frac{|\boldsymbol{v}|^2}{2} M_f\, d\boldsymbol{v} &= \frac{-n|\boldsymbol{u}|^2}{2(2\pi\theta)^{\frac{d}{2}}}\int e^{-|\boldsymbol{s}|^2}(2\theta)^{\frac{d}{2}} d\boldsymbol{s} + \frac{n}{(2\pi\theta)^{\frac{d}{2}}}\boldsymbol{u}^T\int \left[(2\theta)^{\frac{1}{2}}\boldsymbol{s} + \boldsymbol{u}\right] e^{-|\boldsymbol{s}|^2}(2\theta)^{\frac{d}{2}} d\boldsymbol{s} \\
&\quad + \frac{n}{2(2\pi\theta)^{\frac{d}{2}}}\int \left|(2\theta)^{\frac{1}{2}}\boldsymbol{s}\right|^2 e^{-|\boldsymbol{s}|^2}(2\theta)^{\frac{d}{2}} d\boldsymbol{s} \\
&= \frac{-n|\boldsymbol{u}|^2}{2(2\pi\theta)^{\frac{d}{2}}}(2\theta)^{\frac{d}{2}}\pi^{\frac{d}{2}} + \frac{n(2\theta)^{\frac{d}{2}+\frac{1}{2}}}{(2\pi\theta)^{\frac{d}{2}}}\boldsymbol{u}^T\int \boldsymbol{s} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} + \frac{n(2\theta)^{\frac{d}{2}}}{(2\pi\theta)^{\frac{d}{2}}}\boldsymbol{u}^T\boldsymbol{u}\int e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \\
&\quad + \frac{n(2\theta)(2\theta)^{\frac{d}{2}}}{2(2\pi\theta)^{\frac{d}{2}}}\int |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \\
&= \frac{-n|\boldsymbol{u}|^2}{2} + 0 + \frac{n|\boldsymbol{u}|^2}{\pi^{\frac{d}{2}}}\cdot\pi^{\frac{d}{2}} + \frac{n\theta}{\pi^{\frac{d}{2}}}\int |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \\
&= \frac{1}{2}n|\boldsymbol{u}|^2 + \frac{n\theta}{\pi^{\frac{d}{2}}}\cdot\frac{d}{2}\pi^{\frac{d}{2}} \\
&= \frac{1}{2}n|\boldsymbol{u}|^2 + \frac{d}{2}n\theta \\
&= E
\end{aligned}
$$

$$= \int \frac{|\boldsymbol{v}|^2}{2} f \, d\boldsymbol{v}. \tag{24}$$

Therefore,

$$\int \frac{|\boldsymbol{v}|^2}{2} M_f \, d\boldsymbol{v} = \int \frac{|\boldsymbol{v}|^2}{2} f \, d\boldsymbol{v},$$

as desired.

■

Definition 2.3. *We say that a quantity $g(\boldsymbol{v})$ is* **collision invariant** *iff*

$$\int_{\mathbb{R}^d} g(\boldsymbol{v}) \left(M_f - f\right) d\boldsymbol{v} = 0.$$

From the last Lemma, we observe that 1, $\boldsymbol{v}$ and $|\boldsymbol{v}|^2$ are collision invariant. Of course, any linear combination of these functions will also be collision invariant.

Multiplying the BGK Equation by the vector $(1, \boldsymbol{v}, |\boldsymbol{v}|^2)^T$ and integrating, we get expressions for the conservation of mass, momentum, and energy, respectively.

Lemma 2.4. *Suppose that f solves the BGK equation, under the assumption that $\boldsymbol{a} \equiv \boldsymbol{0}$. The following conservation equations hold:*

$$\frac{\partial}{\partial t} n + \nabla_x \cdot (n\boldsymbol{u}) = 0, \tag{25}$$

$$\frac{\partial}{\partial t} (n\boldsymbol{u}) + \nabla_x \cdot (n\boldsymbol{u} \otimes \boldsymbol{u} + \mathsf{P}) = 0, \tag{26}$$

$$\frac{\partial}{\partial t} E + \nabla_x \cdot (E\boldsymbol{u} + \mathsf{P}\boldsymbol{u} + \boldsymbol{q}) = 0, \tag{27}$$

where

$$\mathsf{P} \coloneqq \int_{\mathbb{R}^d} (\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u}) f \, d\boldsymbol{v}$$

is the pressure tensor and

$$\boldsymbol{q} \coloneqq \frac{1}{2} \int_{\mathbb{R}^d} |\mathbf{v} - \mathbf{u}|^2 (\mathbf{v} - \mathbf{u}) f \, d\boldsymbol{v}$$

is the heat flux.

Proof. 1. Multiplying the BGK equation by 1 and integrating, we have

$$\int_{\mathbb{R}^d} \left[\frac{\partial f}{\partial t} + \nabla_x \cdot (\boldsymbol{v} f) \right] d\boldsymbol{v} = \int_{\mathbb{R}^d} \lambda (M_f - f) d\boldsymbol{v} \tag{28}$$

$$\iff \frac{\partial}{\partial t} \left(\int_{\mathbb{R}^d} f \, d\boldsymbol{v} \right) + \nabla_x \cdot \left(\int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v} \right) = 0 \tag{29}$$

$$\iff \frac{\partial}{\partial t} n + \nabla_{\boldsymbol{x}} \cdot (n\boldsymbol{u}) = 0. \tag{30}$$

This verifies the first equation, which is an expression for the conservation of mass.

2. Next, multiplying by $\boldsymbol{v}$ and integrating, we have

$$\int_{\mathbb{R}^d} \left[\boldsymbol{v} \frac{\partial f}{\partial t} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} \otimes \boldsymbol{v} f) \right] d\boldsymbol{v} = \int_{\mathbb{R}^d} \boldsymbol{v} \lambda (M_f - f) d\boldsymbol{v} \tag{31}$$

$$\iff \frac{\partial}{\partial t} \left(\int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v} \right) + \nabla_{\boldsymbol{x}} \cdot \left(\int_{\mathbb{R}^d} \boldsymbol{v} \otimes \boldsymbol{v} f \, d\boldsymbol{v} \right) = \boldsymbol{0}. \tag{32}$$

The first term is equal to $\partial_t (n\boldsymbol{u})$, as desired. It remains to show that

$$\int_{\mathbb{R}^d} \boldsymbol{v} \otimes \boldsymbol{v} f \, d\boldsymbol{v} = n\boldsymbol{u} \otimes \boldsymbol{u} + \mathsf{P}.$$

Writing $\boldsymbol{v} = \boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})$, we have

$$\begin{aligned}
\int_{\mathbb{R}^d} \boldsymbol{v} \otimes \boldsymbol{v} f \, d\boldsymbol{v} &= \int_{\mathbb{R}^d} [\boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})] \otimes [\boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})] \, f \, d\boldsymbol{v} \\
&= \int_{\mathbb{R}^d} [\boldsymbol{u} \otimes \boldsymbol{u} + \boldsymbol{u} \otimes (\boldsymbol{v} - \boldsymbol{u}) + (\boldsymbol{v} - \boldsymbol{u}) \otimes \boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u})] \, f \, d\boldsymbol{v} \\
&= \boldsymbol{u} \otimes \boldsymbol{u} \int_{\mathbb{R}^d} f \, d\boldsymbol{v} + 2 \int_{\mathbb{R}^d} \boldsymbol{u} \otimes (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} + \int_{\mathbb{R}^d} (\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= n\boldsymbol{u} \otimes \boldsymbol{u} + \mathsf{P} + 2 \int_{\mathbb{R}^d} \boldsymbol{u} \otimes (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v}.
\end{aligned} \tag{33}$$

The final term in the above expression is equal to zero:

$$\boldsymbol{u} \otimes \int_{\mathbb{R}^d} (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} = \boldsymbol{u} \otimes \left[\int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v} - \boldsymbol{u} \int_{\mathbb{R}^d} f \, d\boldsymbol{v} \right] = \boldsymbol{u} \otimes [n\boldsymbol{u} - \boldsymbol{u} n] = \boldsymbol{0}.$$

Putting it all together, we have the second equation, an expression for the conservation of momentum.

3. Finally, multiplying by $\frac{1}{2}|\boldsymbol{v}|^2$ and integrating, we have

$$\int_{\mathbb{R}^d} \frac{1}{2} |\boldsymbol{v}|^2 \left(\frac{\partial f}{\partial t} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f) \right) d\boldsymbol{v} = \int_{\mathbb{R}^d} \frac{1}{2} |\boldsymbol{v}|^2 \lambda (M_f - f) d\boldsymbol{v} \tag{34}$$

$$\iff \frac{\partial}{\partial t} \left(\int_{\mathbb{R}^d} \frac{1}{2} |\boldsymbol{v}|^2 f \, d\boldsymbol{v} \right) + \nabla_{\boldsymbol{x}} \cdot \left(\int_{\mathbb{R}^d} \frac{1}{2} |\boldsymbol{v}|^2 \boldsymbol{v} f \, d\boldsymbol{v} \right) = \boldsymbol{0}. \tag{35}$$

The first term is equal to $\partial_t(E)$, as desired. It remains to show that

$$\frac{1}{2}|\mathbf{v}|^2 \boldsymbol{v} f \, d\boldsymbol{v} = E\boldsymbol{u} + \mathrm{P}\boldsymbol{u} + \boldsymbol{q}.$$

Writing $\boldsymbol{v} = \boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})$, we have

$$\begin{aligned}
\int_{\mathbb{R}^d} \frac{1}{2}|\boldsymbol{v}|^2 \boldsymbol{v} f \, d\boldsymbol{v} &= \frac{1}{2}\int_{\mathbb{R}^d} |\boldsymbol{v}|^2 \left[\boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})\right] f d\boldsymbol{v} \\
&= \boldsymbol{u}\int_{\mathbb{R}^d} \frac{1}{2}|\boldsymbol{v}|^2 f d\boldsymbol{v} + \frac{1}{2}\int_{\mathbb{R}^d} |\boldsymbol{v}|^2 (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= E\boldsymbol{u} + \frac{1}{2}\int_{\mathbb{R}^d} \left[\boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})\right]^T \left[\boldsymbol{u} + (\boldsymbol{v} - \boldsymbol{u})\right] (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= E\boldsymbol{u} + \frac{1}{2}\int_{\mathbb{R}^d} \left[|\boldsymbol{u}|^2 + 2\boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u}) + |\boldsymbol{v} - \boldsymbol{u}|^2\right] (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= E\boldsymbol{u} + \frac{1}{2}|\boldsymbol{u}|^2 \int_{\mathbb{R}^d} (\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} + \int_{\mathbb{R}^d} \boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&\quad + \frac{1}{2}\int_{\mathbb{R}^d} |\boldsymbol{v} - \boldsymbol{u}|^2 (\mathbf{v} - \mathbf{u}) f d\boldsymbol{v} \\
&= E\boldsymbol{u} + \boldsymbol{q} + \frac{1}{2}|\boldsymbol{u}|^2 \left(\int_{\mathbb{R}^d} \boldsymbol{v} f d\boldsymbol{v} - \boldsymbol{u}\int_{\mathbb{R}^d} f \, d\boldsymbol{v}\right) \\
&\quad + \int_{\mathbb{R}^d} \boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= E\boldsymbol{u} + \boldsymbol{q} + \frac{1}{2}|\boldsymbol{u}|^2 (n\boldsymbol{u} - \boldsymbol{u}n) + \int_{\mathbb{R}^d} \boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} \\
&= E\boldsymbol{u} + \boldsymbol{q} + \int_{\mathbb{R}^d} \boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v}. \qquad (36)
\end{aligned}$$

It can be shown that

$$\boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) = \left[(\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u})\right] \boldsymbol{u}.$$

Thus, the final term in the above Equation is

$$\begin{aligned}
\int_{\mathbb{R}^d} \boldsymbol{u}^T(\boldsymbol{v} - \boldsymbol{u})(\boldsymbol{v} - \boldsymbol{u}) f \, d\boldsymbol{v} &= \int_{\mathbb{R}^d} \left[(\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u})\right] \boldsymbol{u} f \, d\boldsymbol{v} \\
&= \left(\int_{\mathbb{R}^d} (\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u}) f d\boldsymbol{v}\right) \boldsymbol{u} \\
&= \mathrm{P}\boldsymbol{u}. \qquad (37)
\end{aligned}$$

Putting it all together, we have the third equation, which gives an expression for the

conservation of energy.

■

The conservation laws/properties above are reminiscent of those involved with the derivations of the Navier-Stokes and Euler equations. In fact, using the Chapman-Enskog expansion method one can show that in the limit as $\lambda \to \infty$, or $\tau \to 0$, one recovers macroscopic Navier-Stokes and/or Euler equations as formal limits, under certain assumptions [17, 22, 21, 19]. These limits can even guide in the design of stable numerical methods for the macroscopic models [22].

2.2 Space-Homogeneous Problem

Suppose that the density function is spatially homogeneous and/or particle advection may be neglected in the system. In this case, the disribution function f satisfies the space-homogeneous problem

$$\frac{\partial f}{\partial t} = \lambda(M_f - f). \tag{38}$$

This is a first order integro-differential equation (IDE), which can be solved using the integrating factor method.

Lemma 2.5. *Suppose that f is a solution to the space homogeneous problem* (38)*. Then*

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = e^{-\lambda t} f(\boldsymbol{x}, \boldsymbol{v}, 0) + \left(1 - e^{-\lambda t}\right) M_f(\boldsymbol{x}, \boldsymbol{v}, 0). \tag{39}$$

Proof. We begin by proving that $\frac{\partial n}{\partial t} = 0 = \frac{\partial \theta}{\partial t}$, $\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{0}$, and $\frac{\partial M_f}{\partial t} = 0$.

1. First we show that $\frac{\partial n}{\partial t} = 0$. Utilizing the IDE, we have

$$\frac{\partial}{\partial t} n = \frac{\partial}{\partial t} \int_{\mathbb{R}^d} f \, d\boldsymbol{v} = \int_{\mathbb{R}^d} \frac{\partial f}{\partial t} d\boldsymbol{v} = \int_{\mathbb{R}^d} \lambda(M_f - f) d\boldsymbol{v} = 0, \tag{40}$$

   where the last equality follows from Lemma 2.2. Therefore, $\frac{\partial n}{\partial t} = 0$, as desired.

2. Next we show that $\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{0}$. Recall the definition of $\boldsymbol{u} = \frac{\int_{\mathbb{R}^d} \boldsymbol{v} f d\boldsymbol{v}}{\int_{\mathbb{R}^d} f d\boldsymbol{v}} \iff n\boldsymbol{u} = \int_{\mathbb{R}^d} \boldsymbol{v} f d\boldsymbol{v}$. Applying the quotient rule to $\boldsymbol{u} = \frac{n\boldsymbol{u}}{n}$, and using the fact that $\frac{\partial n}{\partial t} = 0$, we have

$$\begin{aligned} \frac{\partial}{\partial t} \boldsymbol{u} &= \frac{\partial}{\partial t} \frac{n\boldsymbol{u}}{n} \\ &= \frac{\frac{\partial}{\partial t}(n\boldsymbol{u}) \cdot n - (n\boldsymbol{u}) \cdot \frac{\partial n}{\partial t}}{n^2} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\frac{\partial}{\partial t}\int_{\mathbb{R}^d} \boldsymbol{v} f \, d\boldsymbol{v} \\
&= \frac{1}{n}\int_{\mathbb{R}^d} \boldsymbol{v}\frac{\partial f}{\partial t} d\boldsymbol{v} \\
&= \frac{\lambda}{n}\int_{\mathbb{R}^d} \boldsymbol{v}(M_f - f) \\
&= \boldsymbol{0}, \qquad (41)
\end{aligned}
$$

where, again, the last equality follows from Lemma 2.2. Therefore, $\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{0}$, as desired.

3. Next, we show that $\frac{\partial \theta}{\partial t} = 0$. Recall that $\frac{d}{2} n\theta = E - \frac{1}{2} n|\boldsymbol{u}|^2 \iff \theta = \frac{2E}{dn} - \frac{1}{d}|\boldsymbol{u}|^2$, where d is the dimension of the space. Using the quotient rule, we have

$$
\begin{aligned}
\frac{\partial \theta}{\partial t} &= \frac{\partial}{\partial t}\left(\frac{2E}{dn}\right) - \frac{1}{d}\frac{\partial}{\partial t}|\boldsymbol{u}|^2 \\
&= \frac{2}{d}\frac{\left(\frac{\partial E}{\partial t}\right)(n) - (E)\left(\frac{\partial n}{\partial t}\right)}{n^2} - \frac{1}{d}\left(2\boldsymbol{u}\cdot\frac{\partial \boldsymbol{u}}{\partial t}\right) \\
&= \frac{2}{dn}\frac{\partial E}{\partial t} \\
&= \frac{2}{dn}\frac{\partial}{\partial t}\int_{\mathbb{R}^d} \frac{|\boldsymbol{v}|^2}{2} f \, d\boldsymbol{v} \\
&= \frac{2}{dn}\int_{\mathbb{R}^d} \frac{|\boldsymbol{v}|^2}{2}\frac{\partial f}{\partial t} d\boldsymbol{v} \\
&= \frac{2}{dn}\int_{\mathbb{R}^d} \frac{|\boldsymbol{v}|^2}{2}\lambda(M_f - f) d\boldsymbol{v} \\
&= 0, \qquad (42)
\end{aligned}
$$

where the last equality follows from Lemma 2.2.

4. Next, we prove the $\frac{\partial M_f}{\partial t} = 0$.

$$
\begin{aligned}
\frac{\partial}{\partial t} M_f(\boldsymbol{x}, v, t) &= \frac{\partial}{\partial t}\left[\frac{n(\boldsymbol{x}, t)}{(2\pi\theta(\boldsymbol{x}, t))^{\frac{d}{2}}} \exp\left(-\frac{|\boldsymbol{v} - \boldsymbol{u}(x, t)|^2}{2\theta(\boldsymbol{x}, t)}\right)\right] \\
&= (2\pi)^{-\frac{d}{2}}\frac{\partial}{\partial t}\left[n(\boldsymbol{x}, t)\theta(\boldsymbol{x}, t)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}|\boldsymbol{v} - \boldsymbol{u}(x, t)|^2\theta(\boldsymbol{x}, t)^{-1}\right)\right] \\
&= (2\pi)^{-\frac{d}{2}}\left[\frac{\partial n}{\partial t}\theta^{-\frac{d}{2}} \exp\left(-\frac{1}{2}|\boldsymbol{v} - \boldsymbol{u}|^2\theta^{-1}\right)\right. \\
&\quad \left. + n(\boldsymbol{x}, t)\left(-\frac{d}{2}\right)\theta^{-\frac{d+2}{2}}\frac{\partial \theta}{\partial t} \exp\left(-\frac{1}{2}|\boldsymbol{v} - \boldsymbol{u}|^2\theta^{-1}\right)\right.
\end{aligned}
$$

$$
\begin{aligned}
&+ n\theta^{-\frac{d}{2}} \exp\left(-\frac{1}{2}|\boldsymbol{v}-\boldsymbol{u}|^2\theta^{-1}\right)\frac{\partial}{\partial t}\left[-\frac{1}{2}|\boldsymbol{v}-\boldsymbol{u}|^2\theta^{-1}\right]\Bigg] \\
&= -\frac{1}{2}(2\pi)^{-\frac{d}{2}} n\theta^{-\frac{d}{2}} \exp\left(-\frac{1}{2}|\boldsymbol{v}-\boldsymbol{u}|^2\theta^{-1}\right)\frac{\partial}{\partial t}\left[|\boldsymbol{v}-\boldsymbol{u}|^2\theta^{-1}\right] \\
&= -\frac{1}{2}(2\pi)^{-\frac{d}{2}} n\theta^{-\frac{d}{2}} \exp\left(-\frac{1}{2}|\boldsymbol{v}-\boldsymbol{u}|^2\theta^{-1}\right)[0] \\
&= 0. \qquad (43)
\end{aligned}
$$

5. So, we have $\frac{\partial n}{\partial t} = 0 = \frac{\partial \theta}{\partial t}$, $\frac{\partial \boldsymbol{u}}{\partial t} = \mathbf{0}$, and $\frac{\partial M_f}{\partial t} = 0$. To finish, we use these properties to solve the IDE. Rearranging Equation (38) we have

$$\frac{df}{dt} + \lambda f = \lambda M.$$

Multiplying by the integrating factor, $e^{\lambda t}$, and applying the product rule in reverse, we have

$$\frac{d}{dt}\left(e^{\lambda t} f\right) = \lambda M_f e^{\lambda t}.$$

Integrating with respect to t, we have

$$e^{\lambda t} f = C + \lambda \int e^{\lambda t} M_f(\boldsymbol{x}, \boldsymbol{v}, t)\, dt,$$

Now, since $M_f = M_f(\boldsymbol{x}, v, t)$ is constant with respect to t, $M_f(\boldsymbol{x}, v, t) = M_f(\boldsymbol{x}, v, 0)$. Hence, it can be pulled through the integral sign:

$$
\begin{aligned}
e^{\lambda t} f &= C + \lambda M_f(\boldsymbol{x}, \boldsymbol{v}, 0) \int e^{\lambda t}\, dt \\
&= C + M_f(\boldsymbol{x}, \boldsymbol{v}, 0) e^{\lambda t}. \qquad (44)
\end{aligned}
$$

Plugging in $t = 0$, we can get an expression for the constant term:

$$
\begin{aligned}
e^0 f(\boldsymbol{x}, \boldsymbol{v}, 0) &= C + e^0 M(\boldsymbol{x}, \boldsymbol{v}, 0) \qquad (45) \\
\iff C &= f(\boldsymbol{x}, \boldsymbol{v}, 0) - M(\boldsymbol{x}, \boldsymbol{v}, 0). \qquad (46)
\end{aligned}
$$

Putting everything together, and multiplying by $e^{-\lambda t}$, we have the following solution to the space homogeneous problem:

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = e^{-\lambda t}\left(f(x, \boldsymbol{v}, 0) - M_f(\boldsymbol{x}, \boldsymbol{v}, 0)\right) + M_f(\boldsymbol{x}, \boldsymbol{v}, 0)$$

$$= e^{-\lambda t} f(\boldsymbol{x}, \boldsymbol{v}, 0) + \left(1 - e^{-\lambda t}\right) M_f(\boldsymbol{x}, \boldsymbol{v}, 0). \tag{47}$$

■

Since we have the true solution to this space homogeneous problem, this allows us to test the accuracy of the code on the right hand side source term. Initial tests are performed in the Section 2.5.1.

2.3 The H-Theorem

In this section, we discuss a very important solution property for the BGK equation, namely the entropy dissipation property. This is a key stability concept that should, in some way, be preserved in numerical approximations. We start off this section with a definition.

Definition 2.6. *Suppose that* $f : \Omega \times \mathbb{R}^d \times [0, \infty) \to [0, \infty)$ *is a particle density function. The object*

$$H[f](\boldsymbol{x}, t) \coloneqq \int_{\mathbb{R}^d} f(\boldsymbol{x}, \boldsymbol{v}, t) \ln(f(\boldsymbol{x}, \boldsymbol{v}, t))\, d\boldsymbol{v}$$

is called the H **functional**.

We will need the following technical lemma.

Lemma 2.7. *For any* $x, y \in (0, \infty)$

$$(\ln(x) - \ln(y))\,(x - y) \geq 0.$$

Proof. Observe that the function $q(x) = x \ln(x)$ is strictly convex on $[0, \infty)$. In fact, for any $x \in (0, 1)$,

$$q''(x) = \frac{1}{x} > 0.$$

By Taylor's theorem, for any $x, y \in (0, \infty)$,

$$x \ln(x) = y \ln(y) + (\ln(y) + 1)(x - y) + \frac{1}{2}\frac{1}{\xi}(x - y)^2 \geq y \ln(y) + (\ln(y) + 1)(x - y),$$

for some ξ between x and y. The inequality above can be rewritten as

$$x \ln(x) - x \ln(y) \geq x - y.$$

Reversing the roles of x and y, we have

$$y\ln(y) - y\ln(x) \geq y - x.$$

Adding the inequalities, we have

$$(x-y)(\ln(x) - \ln(y)) \geq 0,$$

which is the desired result. ■

Lemma 2.8. *The function* $\ln(M_f)$ *is a collision invariant, that is*

$$\int_{\mathbb{R}^d} \ln(M_f)\,(M_f - f)\,d\boldsymbol{v} = 0.$$

Proof. We already know that 1, $\boldsymbol{v}$, and $|\boldsymbol{v}|^2$ are collision invariants, as are any linear combinations of these functions. Since

$$\ln(M_f) = \ln\left(\frac{n}{(2\pi\theta)^{d/2}}\right) + \frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2\theta},$$

it follows that $\ln(M_f)$ is also a collision invariant. ■

Theorem 2.9. *Suppose that* $f : \Omega \times \mathbb{R}^d \times [0,\infty) \to [0,\infty)$ *is a solution to the spatially homogeneous BGK problem, that is,*

$$\frac{\partial f}{\partial t} = \frac{1}{\tau}\left(M_f - f\right).$$

In particular, let us assume that f *has no variation with respect to* $\boldsymbol{x}$*, i.e.,* $\nabla_{\boldsymbol{x}} f = \boldsymbol{0}$*. Then*

$$\frac{d}{dt}H[f] \leq 0.$$

Proof. Observe that

$$\begin{aligned}
\frac{d}{dt}H[f](t) &= \int_{\mathbb{R}^d} \frac{\partial f}{\partial t}\left(\ln(f)+1\right)d\boldsymbol{v} \\
&= \int_{\mathbb{R}^d} \left(M_f - f\right)\left(\ln(f)+1\right)d\boldsymbol{v} \\
&= \int_{\mathbb{R}^d} \ln(f)\left(M_f - f\right)d\boldsymbol{v}
\end{aligned}$$

To finish the proof, we use that fact that $\ln(M_f)$ is a collision invariant, that is

$$\int_{\mathbb{R}^d} \ln(M_f)\,(M_f - f)\,d\boldsymbol{v} = 0.$$

Thus

$$\begin{aligned}\frac{d}{dt}H[f](t) &= \int_{\mathbb{R}^d} (\ln(f) - \ln(M_f))\,(M_f - f)\,d\boldsymbol{v} \\ &= -\int_{\mathbb{R}^d} (\ln(M_f) - \ln(f))\,(M_f - f)\,d\boldsymbol{v} \leq 0,\end{aligned}$$

where in the last step we used the fact that

$$(\ln(x) - \ln(y))\,(x - y) \geq 0, \quad \forall\, x, y \in (0, \infty).$$

■

More generally, we have

Theorem 2.10. *Suppose that $f : \Omega \times \mathbb{R}^d \times [0,\infty) \to [0,\infty)$ is an Ω-periodic (spatially periodic) solution to the BGK equation, that is,*

$$\partial_t f + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f = \frac{1}{\tau}\,(M_f - f)\,.$$

Then

$$\frac{d}{dt}\int_{\mathbb{R}^d} H[f]\,d\boldsymbol{x} \leq 0.$$

Proof. Using the same ideas as in the last proof, we can show that

$$\partial_t \int_{\mathbb{R}^d} f\ln(f)\,d\boldsymbol{v} + \nabla_{\boldsymbol{x}}\cdot\int_{\mathbb{R}^d} \boldsymbol{v} f\ln(f)\,d\boldsymbol{v} = -\int_{\mathbb{R}^d} (f - M_f)(\ln(f) - \ln(M_f))d\boldsymbol{v}.$$

Using the Ω-periodicity and integrating over Ω, we have

$$d_t \int_\Omega \int_{\mathbb{R}^d} f\ln(f)\,d\boldsymbol{v}d\boldsymbol{x} = -\int_\Omega \int_{\mathbb{R}^d} (f - M_f)(\ln(f) - \ln(M_f))d\boldsymbol{v}d\boldsymbol{x} \leq 0,$$

which proves the result. ■

We should point out that physicists generally prefer a definition of entropy that sees the entropy increasing as a function of time. But, for historical reasons, in the mathematical

and numerical theory of the Boltzmann and the BGK equations, the prevailing definition of entropy is such that it is non-increasing in time. In any case, this is simply a matter of a sign difference, and the mathematical dissipation property is an important marker for the design of numerical methods. In particular, a numerical approximation scheme should satisfy, if possible, some discrete form of entropy dissipation. However, designing fully discrete approximation schemes that *theoretically* satisfy discrete dissipation (as determined by a rigorous proof) is a challenging task. The papers [9, 10] address this issue for the single species BGK equation, but, this dissipation property comes at a rather high computational cost. Thus, it is not clear whether it is practical to pursue this property from the theoretical point of view. On the other hand, checking the dissipation property numerically for benchmark simulations is certainly a worthwhile endeavor.

2.4 Numerical Approximation

In this section, we consider the $d = 1$ case. Therefore, phase space is two-dimensional, with one dimension for physical space and one for velocity space. It is not necessary to equate the dimensions of velocity and physical space, but this is a common practice, and, for the purpose of describing the numerical methods, one does not lose much generality using such simplifying assumptions. We call the present case the $1x1v$ case. The methods that we describe in this section are scalable, meaning that, as more phase-space dimensions are added, the methods themselves change only little. The biggest obstacle for high-dimensional numerical simulation is the added number of degrees of freedom that accompany an increase in phase-space dimensions. Of course, this increase can be significant, since real-world phase space has 6 dimensions in the model.

2.4.1 Finite Volume Space and Velocity Discretization

Since the Vlasov-BGK equation is a nonlinear conservation-like law, shocks (discontinuities) and rarefaction waves can form and propagate in the solution. Thus, the integro-differential equation does not necessarily hold in the classical (strong) sense. In this case, finite volume methods, which are based on the integral form of the differential equation, are typically more appropriate and simpler to use.

First, we truncate the velocity space so that it is finite in size: $V = [-v_{\max}, v_{\max}]$, where $v_{\max} > 0$. This limits the range of velocities that may be approximated, but, as we will see, does not generally affect the accuracy of approximation as long as $v_{\max}$ is chosen sufficiently

large. We note that it is not necessary to make a symmetric truncation about zero velocity, and sometimes it is not advantageous to do so. This is only done for simplicity of presentation. In similar fashion, let us assume that $\Omega = [-L, L]$, with $L > 0$.

Let us define

$$h_x \coloneqq \frac{2L}{N_x} \quad \text{and} \quad h_v \coloneqq \frac{2v_{\max}}{N_v},$$

and then set

$$x_\ell \coloneqq -L + (\ell - 1/2)h_x \quad \text{and} \quad v_\ell \coloneqq -v_{\max} + (\ell - 1/2)h_v,$$

where ℓ can take integer and half-integer values. To discretize phase space, we break the rectangular $1x1v$ domain, $\Omega \times V$, into a two dimensional grid of cells with finite volume, and approximate the cell average of the function. $\Omega \times V \subset \mathbb{R}^2$ can be written as the union of the cells:

$$\Omega \times V = \bigcup_{i,j=1}^{N^x,N^v} C_{i,j} = \bigcup_{i,j=1}^{N^x,N^v} C_i^x \times C_j^v = \bigcup_{i,j=1}^{N^x,N^v} \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[v_{j-\frac{1}{2}}, v_{j+\frac{1}{2}}\right], \tag{48}$$

with the point (x_i, v_j) at the center of cell $C_{i,j} = C_i^x \times C_j^v$. Next, we define

$$\overline{f}_{i,j}(t) \coloneqq \overline{f}(x_i, v_j, t) \coloneqq \frac{1}{|C_i^x \times C_j^v|} \int_{C_i^x \times C_j^v} f(x, v, t)\, dxdv. \tag{49}$$

The integral form of the Vlasov-BGK equation is

$$\partial_t \overline{f}_{i,j}(t) + \frac{1}{|C_{i,j}|} \int_{C_{i,j}} (\partial_x(vf) + \partial_v(a(x,t)f))\, dxdv = \frac{\lambda}{|C_{i,j}|} \int_{C_{i,j}} M_f(x, v, t)\, dxdv - \lambda \overline{f}_{i,j}(t), \tag{50}$$

where we have assumed that the acceleration of particles due to the external field, a, is independent of velocity. Let us define the flux function, $\boldsymbol{F}$, via

$$\boldsymbol{F}(x, v, t) \coloneqq [vf(x, v, t), a(x, t)f(x, v, t)]^T \coloneqq [F(x, v, t), G(x, v, t)]^T,$$

where

$$F(x, v, t) \coloneqq vf(x, v, t), \quad G(x, v, t) \coloneqq a(x, t)f(x, v, t).$$

Define

$$\overline{M}_{i,j}(t) \coloneqq \frac{1}{|C_{i,j}|} \int_{C_{i,j}} M_f(x, v, t)\, dxdv.$$

Applying the Divergence Theorem,

$$\partial_t \overline{f}_{i,j}(t) + \lambda \overline{f}_{i,j}(t) - \lambda \overline{M}_{i,j}(t) = -\frac{1}{h_x h_v} \int_{C_j^v} \left(F(x_{i+1/2}, v, t) - F(x_{i-1/2}, v, t) \right) dv - \frac{1}{h_x h_v} \int_{C_i^x} \left(G(x, v_{j+1/2}, t) - G(x, v_{j-1/2}, t) \right) dx. \quad (51)$$

Now, (51) is exact. To gain a practical numerical method, we must make some approximations. Let us first approximate the flux integrals:

$$\frac{1}{h_x h_v} \int_{C_j^v} \left(F(x_{i+1/2}, v, t) - F(x_{i-1/2}, v, t) \right) dv \approx \frac{F(x_{i+1/2}, v_j, t) - F(x_{i-1/2}, v_j, t)}{h_x}, \quad (52)$$

$$\frac{1}{h_x h_v} \int_{C_i^x} \left(G(x, v_{j+1/2}, t) - G(x, v_{j-1/2}, t) \right) dx \approx \frac{G(x_i, v_{j+1/2}, t) - G(x_i, v_{j-1/2}, t)}{h_v}. \quad (53)$$

To approximate the density function on the cell $C_{i,j}$, we use piecewise linear reconstructions:

$$p_{i,j}(x, v, t) = \overline{f}_{i,j}(t) + \sigma_{i,j}^x (x - x_i) + \sigma_{i,j}^v (v - v_j),$$

with the θ-minmod slope limiter,

$$\sigma_{i,j}^x = \text{minmod}\left(\left(\frac{\overline{f}_{i+1,j} - \overline{f}_{i-1,j}}{2h_x} \right), \theta^x \left(\frac{\overline{f}_{i,j} - \overline{f}_{i-1,j}}{h_x} \right), \theta^x \left(\frac{\overline{f}_{i+1,j} - \overline{f}_{i,j}}{h_x} \right) \right), \quad (54)$$

$$\sigma_{i,j}^v = \text{minmod}\left(\left(\frac{\overline{f}_{i,j+1} - \overline{f}_{i,j-1}}{2h_v} \right), \theta^v \left(\frac{\overline{f}_{i,j} - \overline{f}_{i,j-1}}{h_v} \right), \theta^v \left(\frac{\overline{f}_{i,j+1} - \overline{f}_{i,j}}{h_v} \right) \right), \quad (55)$$

where $\theta^x, \theta^v \in [1, 2]$. Note that for three *real* arguments, the minmod function is defined by

$$\text{minmod}(x, y, z) := \begin{cases} \min\{x, y, z\} & \text{if} \quad x, y, z \geq 0 \\ \max\{x, y, z\} & \text{if} \quad x, y, z \leq 0 \\ 0 & \text{otherwise} \end{cases}.$$

The reconstructions make the density approximation multi-valued at the cell edges, a fact that we use to our advantage in the numerical approximation. Using the reconstructions, we define the density at the midpoints of each cell edge as follows:

$$f_{i+1/2,j}^-(t) := p_{i,j}(x_{i+1/2}, v_j, t), \quad f_{i+1/2,j}^+(t) := p_{i+1,j}(x_{i+1/2}, v_j, t),$$

and, likewise,

$$f^-_{i,j+1/2}(t) \coloneqq p_{i,j}(x_i, v_{j+1/2}, t), \quad f^+_{i,j+1/2}(t) \coloneqq p_{i,j+1}(x_i, v_{j+1/2}, t).$$

Next, the exact fluxes are replaced by numerical fluxes of the form

$$F(x_{i+1/2}, v_j, t) \approx \tilde{F}(f^+_{i+1/2,j}(t), f^-_{i+1/2,j}(t)) =: \hat{F}_{i+1/2,j}(t)$$

and

$$G(x_i, v_{j+1/2}, t) \approx \tilde{G}(f^+_{i,j+1/2}(t), f^-_{i,j+1/2}(t)) =: \hat{G}_{i,j+1/2}(t).$$

We use a simple upwind strategy to construct the numerical fluxes:

$$\hat{F}_{i+1/2,j}(t) = \tilde{F}(f^+_{i+1/2,j}(t), f^-_{i+1/2,j}(t)) = \begin{cases} v_j f^-_{i+1/2,j}(t) & \text{if} \quad v_j \geq 0 \\ v_j f^+_{i+1/2,j}(t) & \text{if} \quad v_j < 0 \end{cases}, \tag{56}$$

$$\hat{G}_{i,j+1/2}(t) = \tilde{G}(f^+_{i,j+1/2}(t), f^-_{i,j+1/2}(t)) = \begin{cases} a(x_i, t) f^-_{i,j+1/2}(t) & \text{if} \quad a(x_i, t) \geq 0 \\ a(x_i, t) f^+_{i,j+1/2}(t) & \text{if} \quad a(x_i, t) < 0 \end{cases}. \tag{57}$$

We can write the approximation scheme to this point as follows:

$$\begin{aligned} \partial_t \overline{f}_{i,j}(t) + \lambda \overline{f}_{i,j}(t) - \lambda \overline{M}_{i,j}(t) = &- \frac{\hat{F}_{i+1/2,j}(t) - \hat{F}_{i-1/2,j}(t)}{h_x} \\ &- \frac{\hat{G}_{i,j+1/2}(t) - \hat{G}_{i,j-1/2}(t)}{h_v} + \tilde{E}_{i,j}(t), \end{aligned} \tag{58}$$

where $\tilde{E}_{i,j}(t)$ is a local truncation (approximation) error.

Finally, to complete the spatial discretization of the Vlasov-BGK equation, we need to approximate the Maxwellian. For this we use the following:

$$n_i(t) = h_v \sum_{j=1}^{N^v} \overline{f}_{i,j}(t), \tag{59}$$

$$u_i(t) = \frac{h_v}{n_i(t)} \sum_{j=1}^{N^v} v_j \overline{f}_{i,j}(t), \tag{60}$$

$$\theta_i(t) = \frac{h_v}{n_i(t)} \sum_{j=1}^{N^v} |v_j|^2 \overline{f}_{i,j}(t) - |u_i(t)|^2, \tag{61}$$

$$\overline{M}_{i,j}(t) \approx \hat{M}_{i,j}(t) = \frac{n_i(t)}{(2\pi\theta_i(t))^{1/2}} \exp\left(-\frac{(v_j - u_i(t))^2}{2\theta_i(t)}\right). \tag{62}$$

Since we are replacing integrations by midpoint quadratures, we lose the collision invariances that we enjoyed at the continuum level. But, for the purpose of approximation, we will continue to assume that these invariances still exist at the discrete level.

We conclude this section by stating the spatially discrete approximation to the Vlasov-BGK equation:

$$\begin{aligned} d_t\overline{f}_{i,j}(t) + \lambda\overline{f}_{i,j}(t) - \lambda\hat{M}_{i,j}(t) = &-\frac{\hat{F}_{i+1/2,j}(t) - \hat{F}_{i-1/2,j}(t)}{h_x} \\ &- \frac{\hat{G}_{i,j+1/2}(t) - \hat{G}_{i,j-1/2}(t)}{h_v} + \hat{E}_{i,j}(t), \end{aligned} \tag{63}$$

where $\hat{E}_{i,j}(t)$ is a local truncation (approximation) error.

2.4.2 Implicit-Explicit Runge Kutta Time Stepping

In this section we introduce an implicit-explicit (IMEX) Runge Kutta (RK) method for integration in time that was proposed in [15]. The convection part is treated using an explicit method, while the collision part is solved using a diagonally implicit method. We will work with the phase-space continuous problem first, in order to take advantage of the collision invariants, before moving to the phase-space discrete problem. Let us rewrite the original Vlasov-BGK equation as

$$\frac{\partial}{\partial t} f(x, v, t) = T[f](x, v, t) + Q[f](x, v, t)$$

where T represents the transport term and Q represents the collision term:

$$T[f](x, v, t) \coloneqq -v\partial_x f(x, v, t) - a(x, t)\partial_v f(x, v, t), \tag{64}$$

$$Q[f](x, v, t) \coloneqq \lambda(M_f(x, v, t) - f(x, v, t)). \tag{65}$$

The general s-stage IMEX Runge Kutta scheme (diagonally implicit) is one of the form

$$y^{(r)} = y^k + \Delta t \sum_{\ell=1}^{r-1} \tilde{a}_{r\ell} T[y^{(\ell)}] + \Delta t \sum_{\ell=1}^{r} a_{r\ell} Q[y^{(\ell)}], \quad r \in \{1, \cdots, s\}, \tag{66}$$

$$y^{k+1} = y^k + \Delta t \sum_{r=1}^{s} \tilde{b}_r T[y^{(r)}] + \Delta t \sum_{r=1}^{s} b_r Q[y^{(r)}], \tag{67}$$

where $\tilde{a}_{r\ell}$, $\tilde{b}_r$, $a_{r\ell}$, b_r are taken from the following Butcher tables, respectively:

$$\begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\mathsf{A}} \\ \hline & \tilde{\boldsymbol{b}}^T \end{array} \qquad\qquad \begin{array}{c|c} \boldsymbol{c} & \mathsf{A} \\ \hline & \boldsymbol{b}^T \end{array}. \tag{68}$$

The matrix $\tilde{\mathsf{A}}$, for the explicit part, is strictly lower triangular, and A is lower triangular.

To run the algorithm, for each stage, we first calculate

$$\hat{y}^{(r-1)} := y^k + \Delta t \sum_{\ell=1}^{r-1} \tilde{a}_{r\ell} T[y^{(\ell)}] + \Delta t \sum_{\ell=1}^{r-1} a_{r\ell} Q[y^{(\ell)}]. \tag{69}$$

We then rearrange the terms in the final sum to find an expression for $y^{(r)}$:

$$\begin{aligned} y^{(r)} &= y^k + \Delta t \sum_{\ell=1}^{r-1} \tilde{a}_{r\ell} T[y^{(\ell)}] + \Delta t \sum_{\ell=1}^{r} a_{r\ell} Q[y^{(\ell)}] \\ &= \hat{y}^{(r-1)} + \Delta t a_{rr} Q[y^{(r)}] \\ &= \hat{y}^{(r-1)} + \Delta t a_{rr} \lambda (M_{y^{(r)}} - y^{(r)}) \\ &= \hat{y}^{(r-1)} + \Delta t a_{rr} \lambda M_{y^{(r)}} - \Delta t a_{rr} \lambda y^{(r)} \qquad (70) \\ \iff (1 + \Delta t a_{rr} \lambda) y^{(r)} &= \hat{y}^{(r-1)} + \Delta t a_{rr} \lambda M_{y^{(r)}} \qquad (71) \\ \iff y^{(r)} &= \frac{1}{1 + \Delta t a_{rr} \lambda} \hat{y}^{(r-1)} + \frac{\Delta t a_{rr} \lambda}{1 + \Delta t a_{rr} \lambda} M_{y^{(r)}}. \qquad (72) \end{aligned}$$

Note that we need to compute the Maxwellian $M_{y^{(r)}}$, which involves the current stage. This seems to present an issue. However, using collision invariances, we can circumvent this. In particular,

$$\begin{aligned} \int_{\mathbb{R}^d} \begin{pmatrix} 1 \\ \boldsymbol{v} \\ |\boldsymbol{v}|^2 \end{pmatrix} y^{(r)} d\boldsymbol{v} &= \int_{\mathbb{R}^d} \begin{pmatrix} 1 \\ \boldsymbol{v} \\ |\boldsymbol{v}|^2 \end{pmatrix} \hat{y}^{(r-1)} d\boldsymbol{v} + \Delta t a_{rr} \lambda \int_{\mathbb{R}^d} \begin{pmatrix} 1 \\ \boldsymbol{v} \\ |\boldsymbol{v}|^2 \end{pmatrix} (M_{y^{(r)}} - y^{(r)}) d\boldsymbol{v} \\ &= \int_{\mathbb{R}^d} \begin{pmatrix} 1 \\ \boldsymbol{v} \\ |\boldsymbol{v}|^2 \end{pmatrix} \hat{y}^{(r-1)} d\boldsymbol{v}. \end{aligned} \tag{73}$$

Since the first, second, and third moments of $\hat{y}^{(r-1)}$ are equal to those of $y^{(r)}$, the two Maxwellians at these stages are equal:

$$M_{y^{(r)}} = M_{\hat{y}^{(r-1)}}.$$

Therefore, we have the explicit update formula

$$y^{(r)} = \frac{1}{1+\Delta t a_{rr}\lambda}\hat{y}^{(r-1)} + \frac{\Delta t a_{rr}\lambda}{1+\Delta t a_{rr}\lambda}M_{\hat{y}^{(r-1)}}. \tag{74}$$

With the above expression for $y^{(r)}$, a convex combination of $\hat{y}^{(r-1)}$ and $M_{\hat{y}^{(r-1)}}$, the IMEX-RK scheme can be completed without any complicated inversions.

2.4.3 Fully Discrete Scheme

When we discretize velocity space, we lose the collision invariances at the discrete level; this is because velocity integrals are replaced with midpoint rule quadrature, in addition to the fact that the velocity space is truncated. It follows that (74) breaks down. Nevertheless, we will assume that the phase-space discrete analog to (74) holds so that stages may be updated in explicit fashion.

Suppose that $f_{i,j}^k$ is an approximation of the cell average of the density field over cell $C_{i,j}$ at time $t^n := t^{n-1} + \Delta t$. Similarly, we denote by $f_{i,j}^{(\ell)}$ the ℓ^{th} stage of the IMEX-RK scheme with respect to $f_{i,j}^k$. Define

$$T_{i,j}^{\square} := -\frac{\hat{F}_{i+1/2,j}^{\square} - \hat{F}_{i-1/2,j}^{\square}}{h_x} - \frac{\hat{G}_{i,j+1/2}^{\square} - \hat{G}_{i,j-1/2}^{\square}}{h_v}, \tag{75}$$

$$Q_{i,j}^{\square} := \lambda(M_{i,j}^{\square} - f_{i,j}^{\square}), \tag{76}$$

where the fluxes $\hat{F}_{i+1/2,j}^{\square}$ and $\hat{G}_{i,j+1/2}^{\square}$ are computed with respect to the cell-centered approximation $f_{i,j}^{\square}$ and $\square$ is a stage iteration or a time step index. The fully discrete Maxwellian is computed via

$$n_i^{\square} = h_v \sum_{j=1}^{N^v} f_{i,j}^{\square}, \tag{77}$$

$$u_i^{\square} = \frac{h_v}{n_i^{\square}} \sum_{j=1}^{N^v} v_j f_{i,j}^{\square}, \tag{78}$$

$$\theta_i^{\square} = \frac{h_v}{n_i^{\square}} \sum_{j=1}^{N^v} |v_j|^2 f_{i,j}^{\square} - |u_i^{\square}|^2, \tag{79}$$

$$M_{i,j}^{\square} = \frac{n_i^{\square}}{(2\pi\theta_i^{\square})^{1/2}} \exp\left(-\frac{(v_j - u_i^{\square})^2}{2\theta_i^{\square}} \right). \tag{80}$$

Then we compute

$$\hat{f}_{i,j}^{(r-1)} := f_{i,j}^k + \Delta t \sum_{\ell=1}^{r-1} \tilde{a}_{r\ell} T_{i,j}^{(\ell)} + \Delta t \sum_{\ell=1}^{r-1} a_{r\ell} Q_{i,j}^{(\ell)}, \tag{81}$$

$$f_{i,j}^{(r)} = \frac{1}{1 + \Delta t a_{rr} \lambda} \hat{f}_{i,j}^{(r-1)} + \frac{\Delta t a_{rr} \lambda}{1 + \Delta t a_{rr} \lambda} \hat{M}_{i,j}^{(r-1)}, \tag{82}$$

$$f_{i,j}^{k+1} = f_{i,j}^k + \Delta t \sum_{r=1}^{s} \tilde{b}_r T_{i,j}^{(r)} + \Delta t \sum_{r=1}^{s} b_r Q_{i,j}^{(r)}, \tag{83}$$

where $\hat{M}_{i,j}^{(r-1)}$ is the discrete Maxwellian computed with respect to the fully discrete approximation $\hat{f}_{i,j}^{(r-1)}$. This completes the description of the fully discrete scheme.

2.4.4 Poisson Solver for Vlasov-Poisson-BGK Equation

For the Vlasov-Poisson-BGK equation the acceleration a is determined by an electric field as follows:

$$a(x,t) = -\chi \partial_x \Phi,$$

where $\chi > 0$ is a physical constant and Φ is the electric potential determined via

$$-\partial_{xx} \Phi(x,t) = n(x,t),$$

subject to appropriate boundary conditions.

First let us consider *Dirichlet boundary conditions*:

$$\Phi(x_{1/2} = -L) = \alpha, \qquad \Phi(x_{N^x+1/2} = L) = \beta. \tag{84}$$

Using the standard three-point stencil approximation, we have, for $i = 2, \cdots, N^x - 1$,

$$-\frac{1}{h_x^2} \left[\Phi_{i+1} - 2\Phi_i + \Phi_{i-1} \right] = n_i. \tag{85}$$

The boundary conditions are specified at the peripheral edges of the domain, $-L = x_{1/2}$ and

$L = x_{N^x+1/2}$. As is standard, we use the average of the ghost cell and neighboring interior cell. That is, we use the approximations

$$\frac{\Phi_0 + \Phi_1}{2} = \alpha, \quad \implies \Phi_0 = 2\alpha - \Phi_1, \tag{86}$$

and we cancel the term Φ_0 in the stencil approximation. The first equation in the approximation thus becomes

$$-\Phi_2 + 3\Phi_1 = h_x^2 n_1 + 2\alpha. \tag{87}$$

Using a similar procedure for right boundary condition, we have

$$\Phi_{N^x+1} = 2\beta - \Phi_{N^x}, \implies -\Phi_{N^x-1} + 3\Phi_{N^x} = h_x^2 n_{N^x} + 2\beta. \tag{88}$$

Putting this all into a matrix system, we must solve

$$\begin{pmatrix} 3 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 3 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N^x-1} \\ \Phi_{N^x} \end{pmatrix} = \begin{pmatrix} h_x^2 n_1 + 2\alpha \\ h_x^2 n_2 \\ \vdots \\ h_x^2 n_{N^x-1} \\ h_x^2 n_{N^x} + 2\beta \end{pmatrix}. \tag{89}$$

This is an SPD system and always has a unique solution.

Next, we turn our attention to the case of *Neumann boundary conditions*:

$$\frac{d\Phi}{dx}(-L) = \alpha, \qquad \frac{d\Phi}{dx}(L) = \beta. \tag{90}$$

In this case, using a second-order approximation at the boundary, we must solve matrix system

$$\begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N^x-1} \\ \Phi_{N^x} \end{pmatrix} = \begin{pmatrix} h_x^2 n_1 - h_x \alpha \\ h_x^2 n_2 \\ \vdots \\ h_x^2 n_{N^x-1} \\ h_x^2 n_{N^x} + h_x \beta \end{pmatrix}. \tag{91}$$

This is a symmetric positive semi-definite (SPSD) system. It has a unique mean-zero solution,

that is, a solution satisfying

$$h_x \sum_{i=1}^{N^x} \Phi_i = 0,$$

if and only if

$$h_x \sum_{i=1}^{N^x} n_i = \alpha - \beta.$$

This is the discrete analog of the standard continuous compatibility conditions for unique solvability.

Given $f_{i,j}^{\square}$, we compute the accompanying macroscopic density,

$$n_i^{\square} = h_v \sum_{j=1}^{N^v} f_{i,j}^{\square},$$

and then the associated electric potential $\Phi_i^{\square}$, as described above. Once the discrete potential is available, the acceleration may be approximated via

$$a_{i+1/2}^{\square} = -\chi \frac{\Phi_{i+1}^{\square} - \Phi_i^{\square}}{h_x}.$$

2.5 Sample Computations and Accuracy Tests

In this section, we report on several numerical tests showing accuracy of the numerical implementation.

2.5.1 Relaxation Test

This first test is designed to confirm that the BGK operator is calculated correctly for the single species case. We consider, in particular, the space homogeneous case: the IPDE becomes the following IODE:

$$\begin{cases} \frac{df}{dt} = \lambda(M_f - f) & (x, v, t) \in \Omega \times V \times [0, \infty) \\ f(x, v, 0) = f_0(x, v) & (x, v, t) \in \Omega \times V \times \{t = 0\} \end{cases}. \tag{92}$$

Recall that as shown in Section 2.2, the exact solution is

$$f(x, v, t) = e^{-\lambda t} f(x, v, 0) + \left(1 - e^{-\lambda t}\right) M_f(x, v, 0). \tag{93}$$

For the first test, we chose a function of the form $f_0(x,v) = b(v)g(x)$, where $b(v)$ is a compactly supported C^∞ function, with nonzero values on $v \in (-2,2)$. In particular, we use the function $g(x) = e^{-|x|}$. Consider the function

$$f(x,v,0) = \begin{cases} e^{-|x|}\frac{5\left(v^2+4\right)\exp\left(\frac{5v}{v^2-4}\right)}{\left(v^2-4\right)^2\left(1+\exp\left(\frac{5v}{v^2-4}\right)\right)^2} & \text{if } v \in (-2,2) \\ 0 & \text{otherwise} \end{cases}.$$

Observe that

$$n = \int_V f(x,v)dv = g(x)\int_V b(v)dv = g(x) = e^{-|x|}.$$

Note that $b(v)$ is an even function:

$$\begin{aligned} b(-v) &= \frac{5\left((-v)^2+4\right)\exp\left(\frac{5(-v)}{(-v)^2-4}\right)}{\left((-v)^2-4\right)^2\left(1+\exp\left(\frac{5(-v)}{(-v)^2-4}\right)\right)^2} \\ &= \frac{5\left(v^2+4\right)\exp\left(-\frac{5v}{v^2-4}\right)}{\left(v^2-4\right)^2\left(1+\exp\left(-\frac{5v}{v^2-4}\right)\right)^2}\cdot\frac{\exp\left(\frac{10v}{v^2-4}\right)}{\left[\exp\left(\frac{5v}{v^2-4}\right)\right]^2} \\ &= \frac{5\left(v^2+4\right)\exp\left(\frac{5v}{v^2-4}\right)}{\left(v^2-4\right)^2\left(\exp\left(\frac{5v}{v^2-4}+1\right)\right)^2} \\ &= b(v). \end{aligned}$$

Therefore, the function $h(v) = vb(v)$ is odd. Recall that when integrating an odd function over an interval that is symmetric about the origin, one obtains a zero integral. This means that $u = \frac{1}{n}\int_V vf\,dv \equiv 0$.

The energy density moment is not as easy to compute analytically, so we have found a numerical approximation:

$$E = \frac{1}{2}\int_V v^2 f\,dv \approx 0.3713094964845e^{-|x|}.$$

Thus,

$$\theta = \frac{2E}{dn} - \frac{1}{d}u_0^2 = 2(0.3713094964845)e^{-|x|}.$$

The code is run with $\lambda = 1$ up to a final time $T = 1$. The error is calculated by taking the difference of the true solution minus the computed solution. The computational and true

solutions are given in Figures 2 and 3, respectively. The errors are on the order of 10^{-5} when a mesh of size $N_x = 128$, $N_v = 130$ is used. The plot of the error is given in Figure 4.

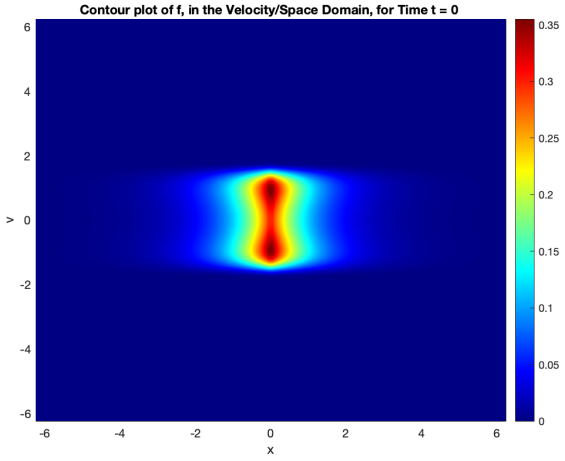


Figure 1: Initial condition function ("bump function").

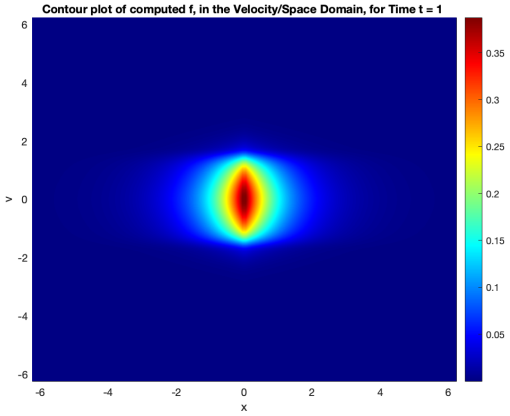


Figure 2: Numerical Solution

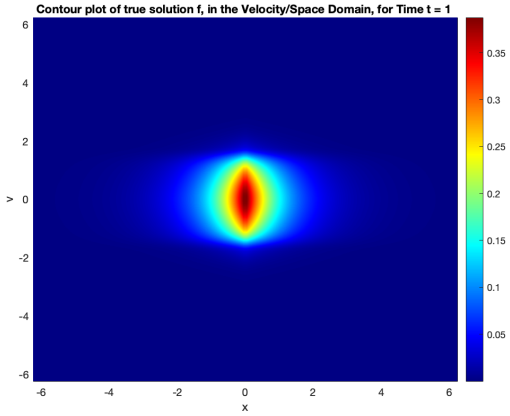


Figure 3: Theoretical Solution

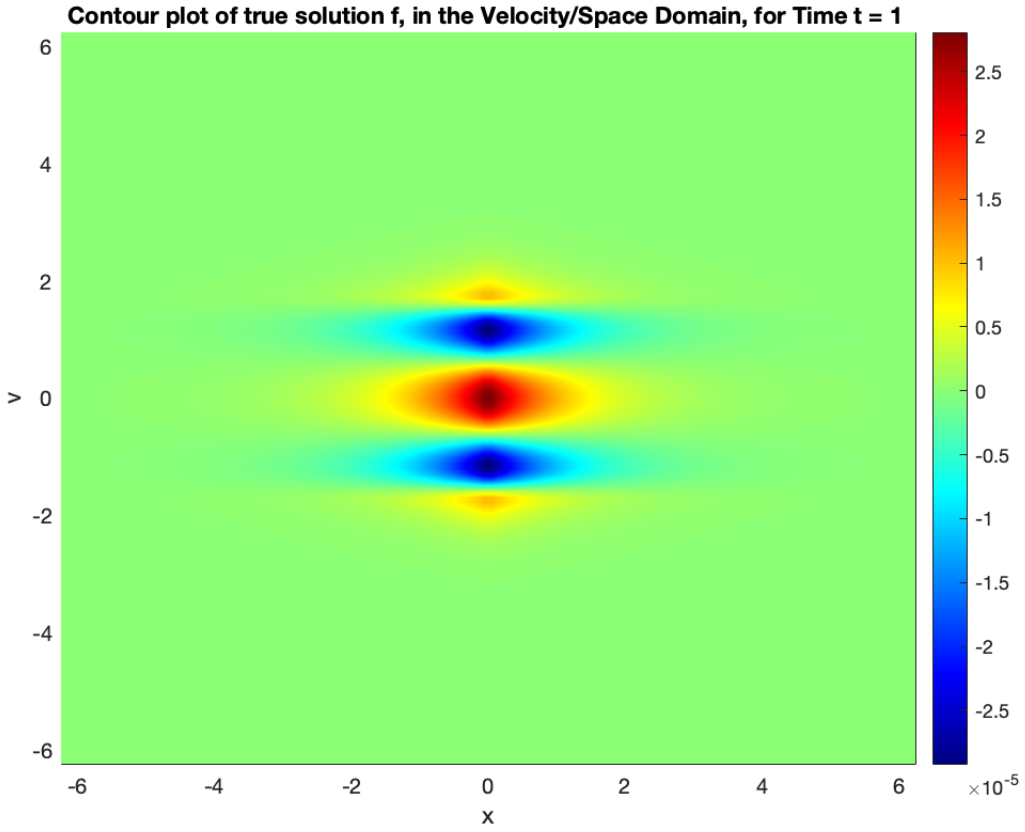


Figure 4: Error between theoretical solution and computed solution. This is the theoretical solution minus the code solution. ($N_x = 128, N_v = 130$.)

For the second test, we set $f_0(x, v)$ to be the sum of four Gaussians, centered at $(x, v) \in \{(\pm 2, \pm 2)\}$. See Figure 5. The trapezoidal rule is used for the initial moments. The code is run with $\lambda = 1$ up to time $T = 1$. The error is calculated by taking the difference of the true solution minus the computed solution. The computed and true solutions are given in Figures 6 and 7 (respectively). The errors are on the order of 10^{-3}, when $N_x = 64$, $N_v = 256$, and are shown in Figure 8.

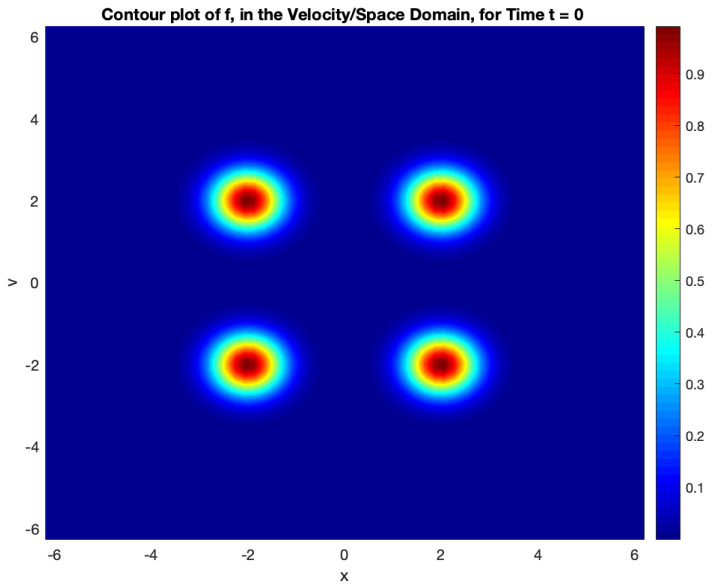


Figure 5: Initial condition function (sum of four Gaussians).

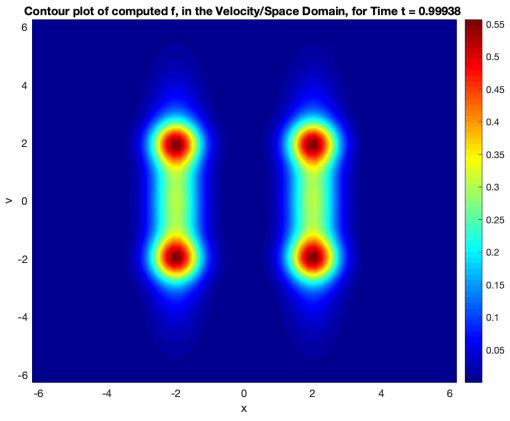


Figure 6: Numerical Solution

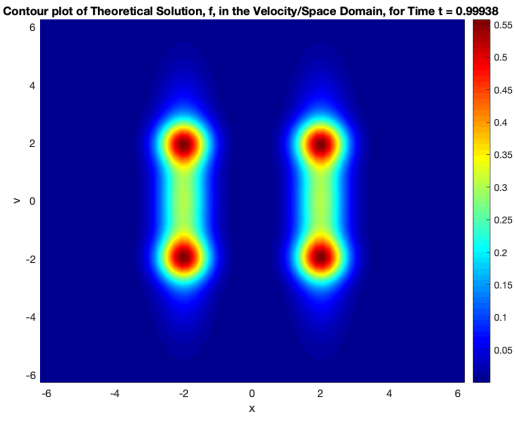


Figure 7: Theoretical Solution

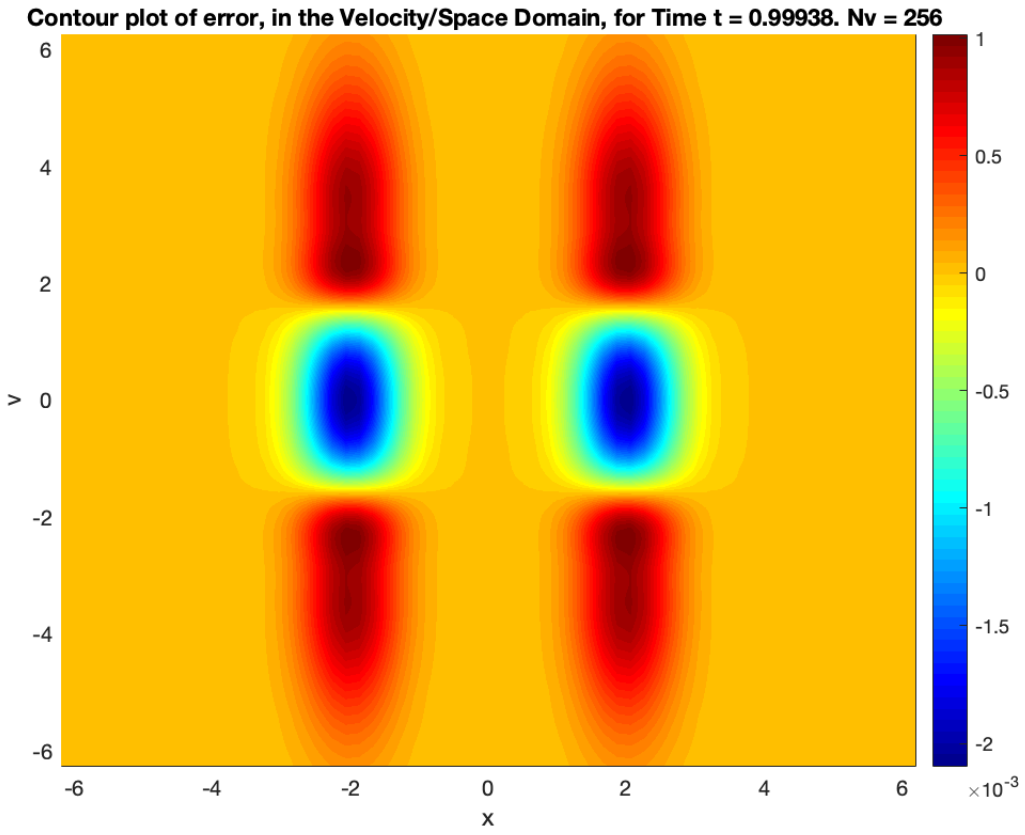


Figure 8: Error between theoretical solution and computed solution. This is the theoretical solution minus the code solution.

2.5.2 Sod Shock Tube (Euler Equation Limit of BGK)

The Sod shock tube test is a standard test. Formally, in the collision limit as $\lambda \to \infty$, the BGK equation is asymptotically equivalent to the Euler Equations. (That is, f converges to the Maxwellian M_f, and the moments $n, \boldsymbol{u}, T$ follow the Euler equations.) Thus, we may test the ability of the code to solve the Sod shock tube problem, by letting λ be large (or equivalently, letting $\tau = \lambda^{-1}$ be small), and setting the initial conditions appropriately. For the test, we used two different values, setting $\lambda = 10^4$ and $\lambda = 10^5$ ($\tau = 10^{-4}, 10^{-5}$).

The BGK equation in 1 dimension is

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = \lambda(M_f - f). \tag{94}$$

To test our IMEX code, we compared the profiles of the number density ($n = \int f dv$), bulk velocity ($u = (\int v f dv)/n$), pressure ($P = n\theta$), and internal energy ($E = \frac{1}{2}\theta$) to the theoretical solution, worked out using the book by Toro [20]. We set the phase-space domain equal to $\Omega \times V = [-0.75, 0.75] \times [-10, 10]$. Since the moments are calculated over $\mathbb{R}^d$, then we must have a function that integrates to approximately the same value, when restricting $\mathbb{R}^d$ to the computational velocity domain (in this case, $[-10, 10] \subset \mathbb{R}$). That is, we must ensure that

$$\int_{\mathbb{R}^d} f(x, v, t) dv \approx \int_V f(x, v, t) dv,$$

where V is the truncated velocity domain. Using $[-10, 10]$ gives a reasonable approximation, as we show.

The setup for the Sod problem is a contact discontinuity separating gases of differing density and temperature, and zero velocity. Thus, for the Sod problem, the initial condition for the particle density is a piecewise Maxwellian with the following values:

$$\begin{pmatrix} n_L \\ u_L \\ \theta_L \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \\ 1.0 \end{pmatrix}, \quad x \in (-0.75, 0), \qquad \begin{pmatrix} n_R \\ u_R \\ \theta_R \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.0 \\ 0.8 \end{pmatrix}, \quad x \in (0, 0.75). \tag{95}$$

That is, the initial condition function (contour plot shown in Figure 9) is given by

$$f_0(x, v) = \begin{cases} \frac{1.0}{\sqrt{2\pi(1.0)}} \exp\left(-\frac{|v-0.0|^2}{2(1.0)}\right) & x \in [-0.75, 0], v \in [-10, 10] \\ \frac{0.125}{\sqrt{2\pi(0.8)}} \exp\left(-\frac{|v-0.0|^2}{2(0.8)}\right) & x \in (0, 0.75], v \in [-10, 10] \end{cases} \tag{96}$$

Figure 10 shows the moments of the numerical/computed solution at time $t = 0.20$. The profiles of the solution seem to follow the correct values (as computed, using the Toro book, [20], for reference).

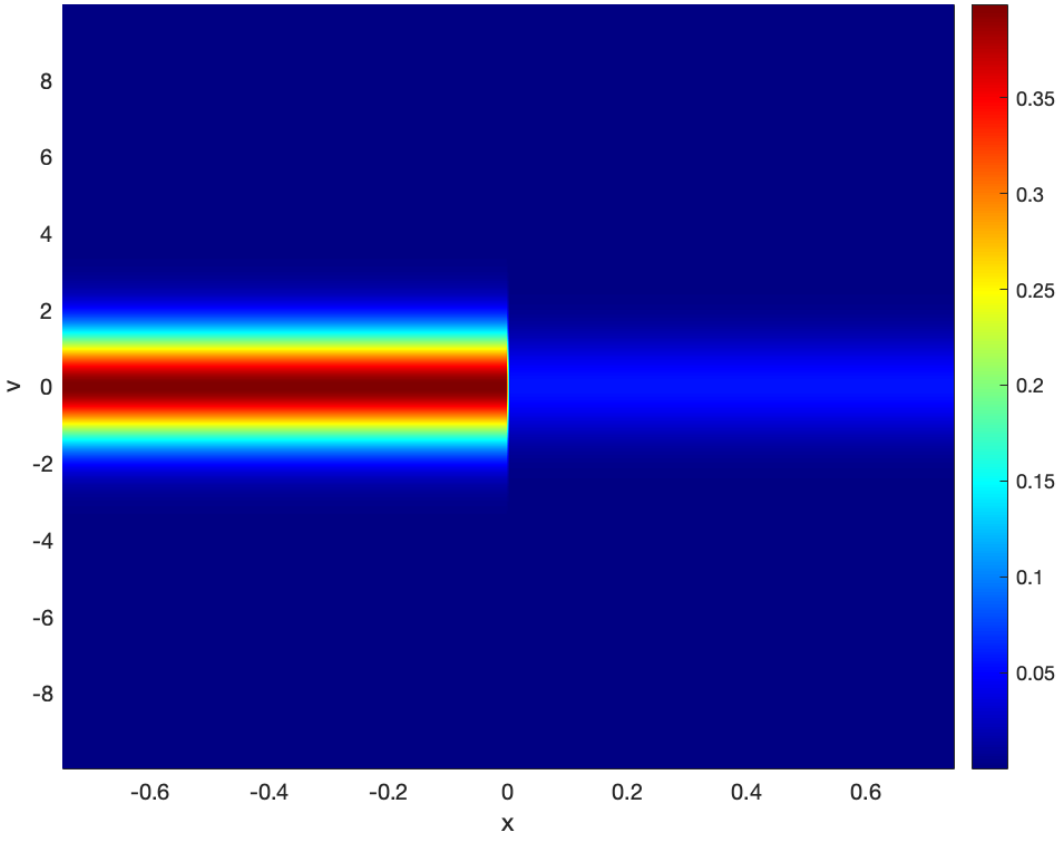


Figure 9: Contour plot of Sod initial condition function.

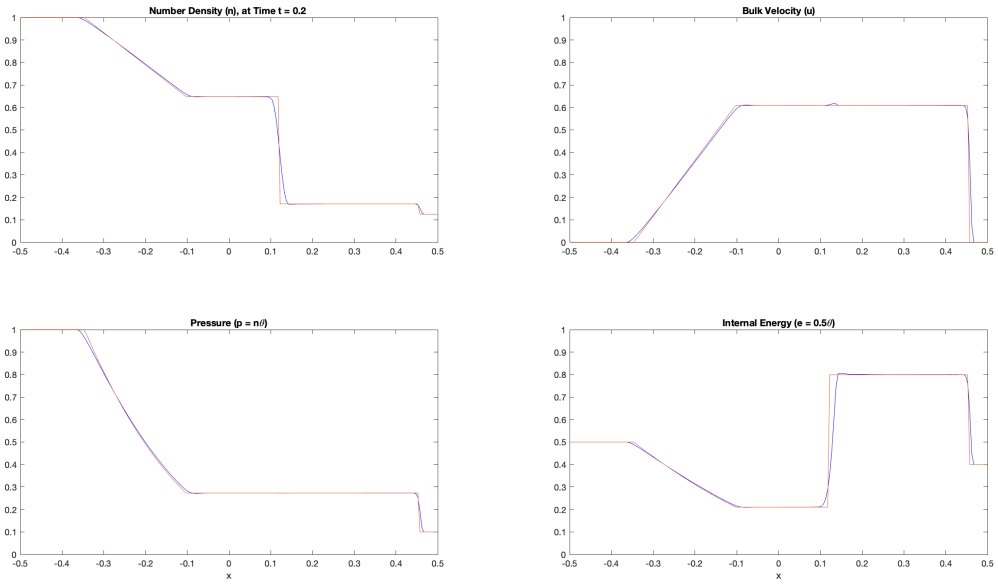


Figure 10: Numerical (Blue, BGK, $\lambda = 10^4$) and Theoretical (Orange, Euler) Solution for Sod Shock tube problem. ($\gamma = 3$) at final time $T = 0.20$.

2.5.3 Square Pulse Rotation

Consider the Vlasov Equation in 1 dimension:

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + a\frac{\partial f}{\partial v} = 0 \tag{97}$$

Setting the acceleration term to $a = -x$ results in a system where an initial distribution is advected counterclockwise, along circular characteristics around the origin. For constructing the characteristics curves, we have the system of ODEs:

$$\frac{dt}{ds} = 1 \tag{98}$$
$$\frac{dx}{ds} = v \tag{99}$$
$$\frac{dv}{ds} = -x. \tag{100}$$

The first equation gives $t = s$, and the remaining two form a system of two coupled ODEs:

$$\dot{x} = v \tag{101}$$
$$\dot{v} = -x. \tag{102}$$

Taking the derivative of the first equation, and plugging in the second equation, we have

$$\ddot{x} = \dot{v} = -x \iff \ddot{x} + x = 0. \tag{103}$$

The solution of this equation is

$$x(t) = c_1\cos(t) + c_2\sin(t). \tag{104}$$

Thus,

$$\dot{v} = -x = -c_1\cos(t) - c_2\sin(t). \tag{105}$$

Integrating gives

$$v(t) = -c_1\sin(t) + c_2\cos(t). \tag{106}$$

Putting these together, we have

$$x(t) = c_1\cos(t) + c_2\sin(t) \tag{107}$$

$$v(t) = -c_1 \sin(t) + c_2 \cos(t). \tag{108}$$

This is a circle, traversed clockwise around the origin in the (x, v)-plane, whose radius depends on the values of c_1, and c_2. Figure 11 shows an image of four square pulses rotated counterclockwise, until the final time $T = \frac{\pi}{2}$ is reached, representing a quarter rotation.

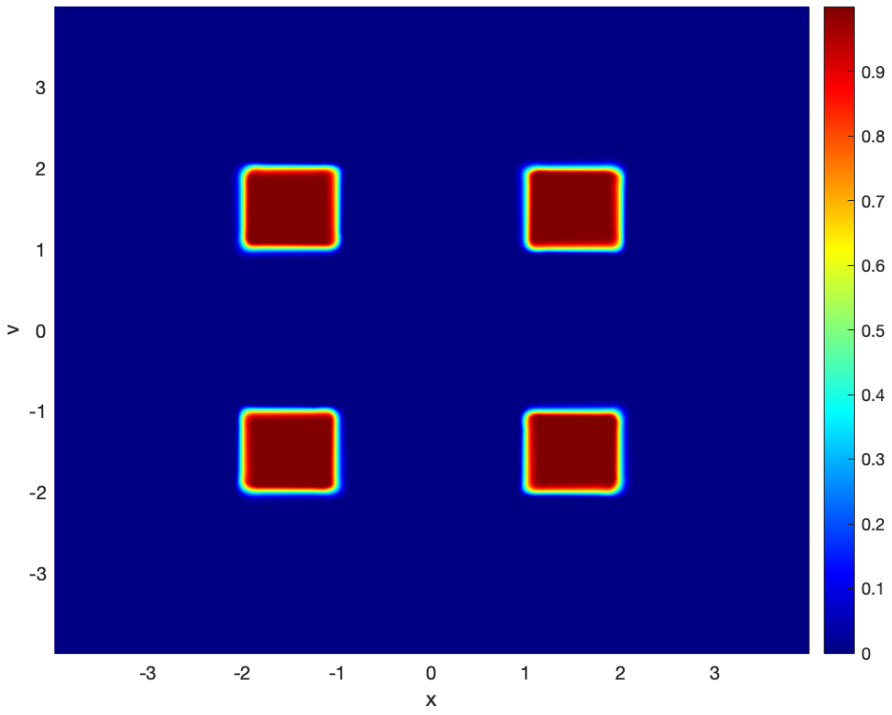


Figure 11: Rotation Problem: $\partial_t f + v\partial_x f - x\partial_v f = 0$. Grid $= 400^2$

2.5.4 Two Stream Instability (Vlasov-Poisson)

The next test that we performed is known as the *Two-Stream Instability Test*, and the present version is taken from Section 5.1.2 of [3]. In this test, two streams of electrons interact and create a highly filamented vortex. The test is designed to assess the code's ability to capture fine structure and examines only the advection piece of our equation (the Vlasov-Poisson equation):

$$\partial_t f + v\partial_x f + E\partial_v f = 0\ , \quad (x, v, t) \in [-2\pi, 2\pi] \times [-2\pi, 2\pi] \times [0, \infty) \tag{109}$$

where

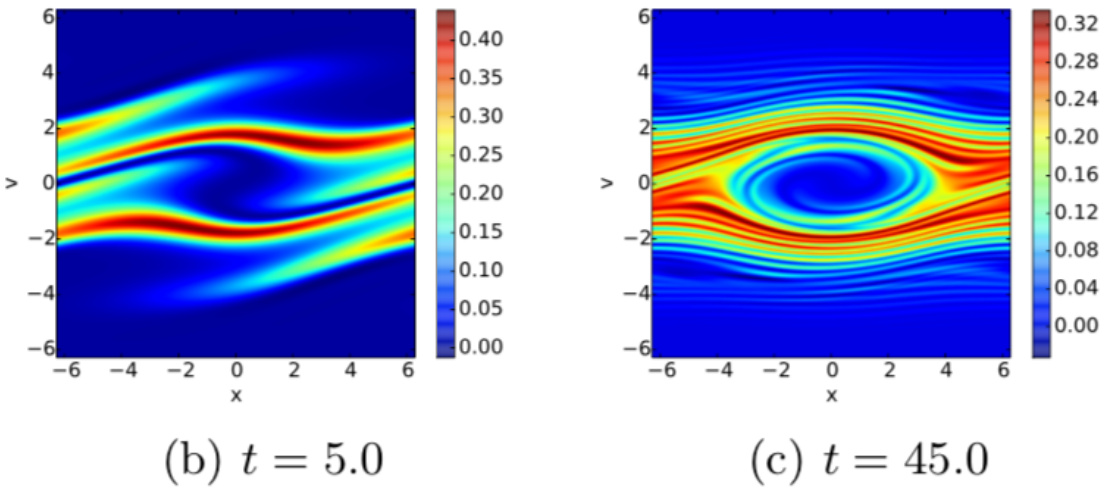
$$-\partial_{xx}\Phi = \frac{e}{\epsilon_0}(n - \overline{n}), \quad E = -\partial_x\Phi, \quad n = \int f\,dv, \quad \overline{n} = \frac{1}{|V|}\int_V f\,dv. \tag{110}$$

For our test runs, we used $\frac{e}{\epsilon_0} = 1$. The initial condition function for the test is

$$f(x, v, 0) = \frac{v^2}{\sqrt{8\pi}}\left(2 - \cos\left(\frac{x}{2}\right)\right)e^{-\frac{v^2}{2}}. \tag{111}$$

The test uses periodic boundary conditions in x, and zero flow boundary conditions in v.

Figure 12, taken from the aforementioned paper, shows the plots of the solution at $t = 5$ and $t = 45$. Figures 13 and 14 show the results of running the code until $t = 5$ and $t = 45$, respectively. For these plots, we used 800^2 grid points; this is twice the number of grid points as in the paper (400^2). We find good agreement with the computed solutions.



(b) $t = 5.0$ (c) $t = 45.0$

Figure 12: Two Stream Instability test case. Figure taken from Garrett & Hauck paper.

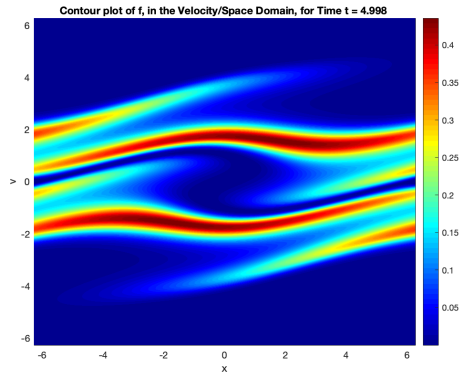


Figure 13: Code generated solution at $t = 5$.

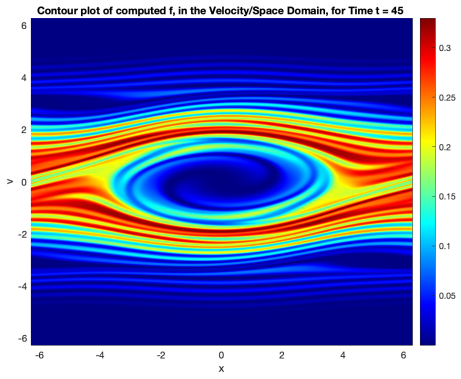


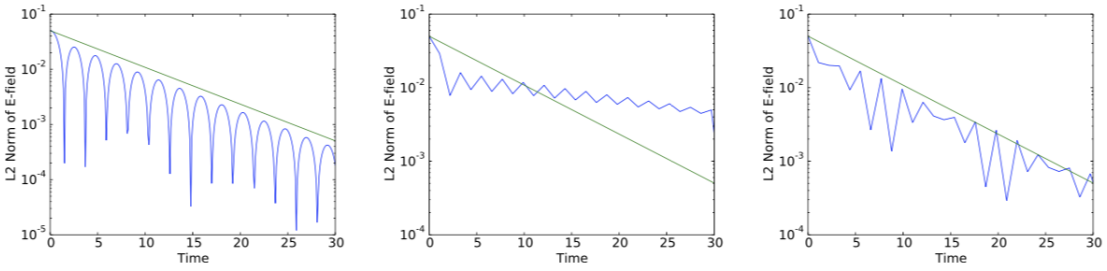
Figure 14: Code generated solution at $t = 45$.

2.5.5 Landau Damping (Vlasov-Poisson)

This test is from Section 5.1.3 of [3]. To perform the test, the Vlasov-Poisson equation is solved and the L^2 norm of the electric field is computed at each time step. The initial condition function is

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}} \left(1 + \alpha \cos(kx)\right) e^{-\frac{v^2}{2}}, \tag{112}$$

where $\alpha = 0.01$, $k = 0.5$. According to the paper, with these parameters, the L^2 norm of the electric field should decay exponentially at a rate of -0.1533. Figure 15 shows the plots taken from the paper. Figure 16 shows the result from our code and indicates qualitative agreement.



(a) Linear $\Delta t = 0.1$ (b) Linear $\Delta t = 1.1$ (c) Nonlinear $\Delta t = 1.1$

Figure 15: Landau Damping test case. Figure taken from Garrett and Hauck [3].

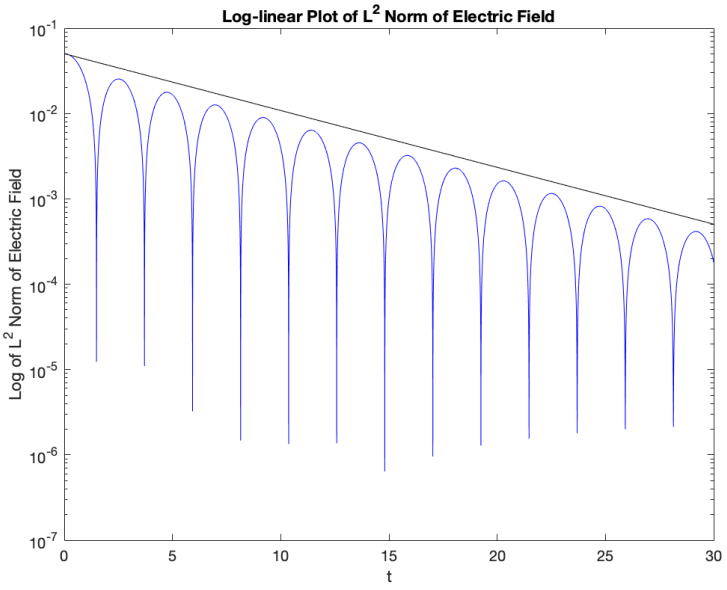


Figure 16: L^2 norm of the electric field.

3 Multispecies BGK Equations

In the present section, we describe a relatively recent BGK-type model for the multi-species setting [4]. This model, which generalizes the one-species case, satisfies conservation of mass, momentum, and total kinetic energy. At the same time, it satisfies a multi-species analog of Boltzmann's H-Theorem [4]. In the multispecies setting, the locally conserved quantities are the species number $m_i\langle f_i\rangle$, the total momentum $\sum_i m_i\langle \boldsymbol{v} f_i\rangle$ and the total energy $\sum_i m_i\langle |\boldsymbol{v}|^2 f_i\rangle$; and the dissipated entropy (Lyapunov functional) is the total (mathematical) entropy $\sum_i \eta(f_i)$, where $\eta(f) \coloneqq f\ln(f) - f$ is the kinetic entropy density. In the zero-relaxation-time limit, hydrodynamic equations can be derived along the lines of the single species BGK equation [4]. Other consistent multi-species models can be found in [5, 8].

3.1 Theory

Suppose that there are $N_s \in \mathbb{N}$ ($N_s \geq 2$) species of particles in a mixture of gases. The *multispecies Vlasov-BGK equation* models the evolution of the distribution fields $f_i(\boldsymbol{x}, v, t)$, $i = 1, \ldots, N_s$, via the system

$$\frac{\partial f_i}{\partial t} + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f_i + \boldsymbol{a}_i\cdot\nabla_{\boldsymbol{v}} f_i = \sum_{j=1}^{N_s} \lambda_{i,j}\left(M_{i,j} - f_i\right), \quad i = 1, \ldots, N_s, \tag{113}$$

where $\boldsymbol{a}_i$ describes the acceleration of particles of species i, $\lambda_{i,j}$ is the collision frequency between species i and j, and

$$M_{i,j} = M_{i,j}[f_i, f_j](\boldsymbol{x}, \boldsymbol{v}, t) = n_i\left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \exp\left(-\frac{m_i|\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right), \tag{114}$$

$$n_i = \int_{\mathbb{R}^d} f_i\, d\boldsymbol{v}, \tag{115}$$

$$\rho_i = m_i n_i = \int_{\mathbb{R}^d} m_i f_i\, d\boldsymbol{v}, \tag{116}$$

$$\boldsymbol{u}_i = \frac{1}{\rho_i}\int_{\mathbb{R}^d} m_i \boldsymbol{v} f_i\, d\boldsymbol{v} = \frac{\int_{\mathbb{R}^d} m_i \boldsymbol{v} f_i\, d\boldsymbol{v}}{\int_{\mathbb{R}^d} m_i f_i\, d\boldsymbol{v}} = \frac{\int_{\mathbb{R}^d} \boldsymbol{v} f_i\, d\boldsymbol{v}}{\int_{\mathbb{R}^d} f_i\, d\boldsymbol{v}}, \tag{117}$$

$$\boldsymbol{u}_{i,j} = \frac{\lambda_{i,j}\rho_i\boldsymbol{u}_i + \lambda_{j,i}\rho_j\boldsymbol{u}_j}{\lambda_{i,j}\rho_i + \lambda_{j,i}\rho_j} = \frac{\lambda_{i,j}\rho_i}{\lambda_{i,j}\rho_i + \lambda_{j,i}\rho_j}\,\boldsymbol{u}_i + \frac{\lambda_{j,i}\rho_j}{\lambda_{i,j}\rho_i + \lambda_{j,i}\rho_j}\,\boldsymbol{u}_j, \tag{118}$$

$$T_i = \frac{2}{dn_i}\int_{\mathbb{R}^d} \frac{m_i}{2}|\boldsymbol{v} - \boldsymbol{u}_i|^2 f_i\, d\boldsymbol{v}, \tag{119}$$

$$T_{i,j} = \frac{\lambda_{i,j} n_i T_i + \lambda_{j,i} n_j T_j}{\lambda_{i,j} n_i + \lambda_{j,i} n_j} + \frac{\lambda_{i,j} \rho_i \left(|\boldsymbol{u}_i|^2 - |\boldsymbol{u}_{i,j}|^2 \right) + \lambda_{j,i} \rho_j \left(|\boldsymbol{u}_j|^2 - |\boldsymbol{u}_{j,i}|^2 \right)}{d(\lambda_{i,j} n_i + \lambda_{j,i} n_j)}. \tag{120}$$

The mixture velocities and temperatures, $\boldsymbol{u}_{i,j}$ and $T_{i,j}$, are chosen so that certain collision invariances hold, as we show momentarily. The existence and uniqueness of nonnegative mild solutions to the multispecies BGK equation was proved by Klingenberg & Pirner (2018) [7], for periodic physical space and under certain restrictions on the collision frequencies.

It is straightforward to check that the proposed system satisfies the usual conservation properties and an entropy dissipation property via an H-Theorem-like result. See [4] for details. In this next computation, we show that the system, as defined above, satisfies certain collision invariances.

Lemma 3.1. *If the mixture velocities $\boldsymbol{u}_{i,j}$ and the mixture temperatures $T_{i,j}$ are given by expressions* (118) *and* (120)*, respectively, then the multispecies BGK collision operators*

$$Q_{i,j}[f_i, f_j] = \lambda_{i,j}(M_{i,j} - f_i) \tag{121}$$

satisfy the following conservation properties, which correspond to the conservation of mass, total momentum, and total energy: for any $i, j \in \{1, \dots, N_s\}$,

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i)\, d\boldsymbol{v} = 0, \tag{122}$$

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i) m_i \boldsymbol{v}\, d\boldsymbol{v} + \int_{\mathbb{R}^d} \lambda_{j,i}(M_{j,i} - f_j) m_j \boldsymbol{v}\, d\boldsymbol{v} = \boldsymbol{0}, \tag{123}$$

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i) m_i |\boldsymbol{v}|^2\, d\boldsymbol{v} + \int_{\mathbb{R}^d} \lambda_{j,i}(M_{j,i} - f_j) m_j |\boldsymbol{v}|^2\, d\boldsymbol{v} = 0. \tag{124}$$

Proof. Recall that the Maxwellians are expressed as

$$M_{i,j} = M_{i,j}[f_i, f_j] = n_i \left(\frac{m_i}{2\pi T_{i,j}} \right)^{\frac{d}{2}} \exp\left(-\frac{m_i |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}} \right),$$

where $\boldsymbol{u}_{i,j}$ and $T_{i,j}$ are functions of $(n_i, \boldsymbol{u}_i, T_i, m_i)$, and $(n_j, \boldsymbol{u}_j, T_j, m_j)$. We will make use of the following facts:

$$\int_{\mathbb{R}^d} \exp(-|\boldsymbol{s}|^2)\, d\boldsymbol{s} = \pi^{\frac{d}{2}}, \tag{125a}$$

$$\int_{\mathbb{R}^d} \boldsymbol{s} \exp(-|\boldsymbol{s}|^2) d\, \boldsymbol{s} = 0, \tag{125b}$$

$$\int_{\mathbb{R}^d} |\boldsymbol{s}|^2 \exp(-|\boldsymbol{s}|^2)\, d\boldsymbol{s} = \frac{d}{2}\pi^{\frac{d}{2}}. \tag{125c}$$

For what follows, the following substitution is used multiple times:

$$\boldsymbol{s} = \frac{\boldsymbol{v} - \boldsymbol{u}_{i,j}}{\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}}} \quad \Longrightarrow \quad \boldsymbol{v} - \boldsymbol{u}_{i,j} = \boldsymbol{s}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}} \quad \Longrightarrow \quad \boldsymbol{v} = \boldsymbol{s}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}} + \boldsymbol{u}_{i,j} \tag{126a}$$

$$d\boldsymbol{s} = \left(\frac{m_i}{2T_{i,j}}\right)^{\frac{d}{2}} d\boldsymbol{v} \implies d\boldsymbol{v} = \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} d\boldsymbol{s}. \tag{126b}$$

1. First, consider

$$\begin{aligned} I_1 &\coloneqq \int \lambda_{i,j}(M_{i,j} - f_i) d\boldsymbol{v} \\ &= \lambda_{i,j} \int \left(n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \exp\left(-\frac{m_i|\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right) - f_i \right) d\boldsymbol{v} \\ &= \lambda_{i,j} n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \int \exp\left(- \left| \frac{\boldsymbol{v} - \boldsymbol{u}_{i,j}}{\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}}} \right|^2 \right) d\boldsymbol{v} - \lambda_{i,j} \int f_i \, d\boldsymbol{v}. \end{aligned} \tag{127}$$

   Using the substitution given in (126) and Equation (125a), we have

$$\begin{aligned} I_1 &= \lambda_{i,j} n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \int e^{-|\boldsymbol{s}|^2} \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} d\boldsymbol{s} - \lambda_{i,j} n_i \\ &= \lambda_{i,j} n_i \pi^{-\frac{d}{2}} \pi^{\frac{d}{2}} - \lambda_{i,j} n_i \\ &= \lambda_{i,j} n_i - \lambda_{i,j} n_i \\ &= 0. \end{aligned} \tag{128}$$

2. Next, consider

$$\begin{aligned} \boldsymbol{I}_2 &\coloneqq \int Q_{i,j}^{BGK} m_i \boldsymbol{v} d\boldsymbol{v} + \int Q_{j,i}^{BGK} m_j \boldsymbol{v} d\boldsymbol{v} \\ &= \int \lambda_{i,j}(M_{i,j} - f_i) m_i \boldsymbol{v} d\boldsymbol{v} + \int \lambda_{j,i}(M_{j,i} - f_j) m_j \boldsymbol{v} d\boldsymbol{v} \\ &= \lambda_{i,j} m_i \int (M_{i,j}\boldsymbol{v} - f_i \boldsymbol{v}) d\boldsymbol{v} + \lambda_{j,i} m_j \int (M_{j,i}\boldsymbol{v} - f_j \boldsymbol{v}) d\boldsymbol{v} \end{aligned}$$

$$
\begin{aligned}
&= \lambda_{i,j} m_i \int \left(n_i \left(\frac{m_i}{2\pi T_{i,j}} \right)^{\frac{d}{2}} \exp\left(-\frac{m_i |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}} \right) \boldsymbol{v} - f_i \boldsymbol{v} \right) d\boldsymbol{v} \\
&\quad + \lambda_{j,i} m_j \int \left(n_{j,i} \left(\frac{m_j}{2\pi T_{j,i}} \right)^{\frac{d}{2}} \exp\left(-\frac{m_j |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{j,i}} \right) \boldsymbol{v} - f_j \boldsymbol{v} \right) d\boldsymbol{v} \\
&= \lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}} \right)^{\frac{d}{2}} \int \exp\left(-\frac{m_i |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}} \right) \boldsymbol{v} d\boldsymbol{v} - \lambda_{i,j} m_i \int f_i \boldsymbol{v} d\boldsymbol{v} \\
&\quad + \lambda_{j,i} m_j n_j \left(\frac{m_j}{2\pi T_{j,i}} \right)^{\frac{d}{2}} \int \exp\left(-\frac{m_j |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{j,i}} \right) \boldsymbol{v} d\boldsymbol{v} - \lambda_{j,i} m_j \int f_j \boldsymbol{v} d\boldsymbol{v}. \qquad (129)
\end{aligned}
$$

Using Equation (125b) and the same substitution as before (Equation (126)), we have

$$
\begin{aligned}
\boldsymbol{I}_2 &= \lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}} \right)^{\frac{d}{2}} \left(\frac{2T_{i,j}}{m_i} \right)^{\frac{d}{2}} \left[\int \left(\frac{2T_{i,j}}{m_i} \right)^{\frac{1}{2}} \boldsymbol{s} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} + \boldsymbol{u}_{i,j} \int e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \right] - \lambda_{i,j} \rho_i \boldsymbol{u}_i \\
&\quad + \lambda_{j,i} m_j n_j \left(\frac{m_j}{2\pi T_{j,i}} \right)^{\frac{d}{2}} \left(\frac{2T_{j,i}}{m_j} \right)^{\frac{d}{2}} \left[\int \left(\frac{2T_{j,i}}{m_j} \right)^{\frac{1}{2}} \boldsymbol{s} e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} + \boldsymbol{u}_{j,i} \int e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} \right] - \lambda_{j,i} \rho_j \boldsymbol{u}_j \\
&= \lambda_{i,j} m_i n_i \pi^{-\frac{d}{2}} \left[0 + \boldsymbol{u}_{i,j} \pi^{\frac{d}{2}} \right] - \lambda_{i,j} \rho_i \boldsymbol{u}_i + \lambda_{j,i} m_j n_j \pi^{-\frac{d}{2}} \left[0 + \boldsymbol{u}_{j,i} \pi^{\frac{d}{2}} \right] - \lambda_{j,i} \rho_j \boldsymbol{u}_j \\
&= \lambda_{i,j} m_i n_i \boldsymbol{u}_{i,j} - \lambda_{i,j} \rho_i \boldsymbol{u}_i + \lambda_{j,i} m_j n_j \boldsymbol{u}_{j,i} - \lambda_{j,i} \rho_j \boldsymbol{u}_j. \qquad (130)
\end{aligned}
$$

Since, $\boldsymbol{u}_{i,j} = \boldsymbol{u}_{j,i}$ we have

$$
\begin{aligned}
\boldsymbol{u}_{i,j} &= \frac{\lambda_{i,j} \rho_i \boldsymbol{u}_i + \lambda_{j,i} \rho_j \boldsymbol{u}_j}{\lambda_{i,j} \rho_i + \lambda_{j,i} \rho_j} \qquad (131) \\
\iff \boldsymbol{u}_{i,j} \left(\lambda_{i,j} m_i n_i + \lambda_{j,i} m_j n_j \right) &= \lambda_{i,j} \rho_i \boldsymbol{u}_i + \lambda_{j,i} \rho_j \boldsymbol{u}_j \qquad (132) \\
\iff \boldsymbol{I}_2 &= \boldsymbol{0}. \qquad (133)
\end{aligned}
$$

3. Finally, consider

$$
\begin{aligned}
I_3 &\coloneqq \int Q_{i,j}^{BGK} m_i \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} + \int Q_{j,i}^{BGK} m_j \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} \\
&= \int \lambda_{i,j} (M_{i,j} - f_i) m_i \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} + \int \lambda_{j,i} (M_{j,i} - f_j) m_j \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} \\
&= \lambda_{i,j} m_i \int n_i \left(\frac{m_i}{2\pi T_{i,j}} \right)^{\frac{d}{2}} \exp\left(-\frac{m_i |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{i,j}} \right) \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} - \lambda_{i,j} m_i \int f_i \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} \\
&\quad + \lambda_{j,i} m_j \int n_j \left(\frac{m_j}{2\pi T_{j,i}} \right)^{\frac{d}{2}} \exp\left(-\frac{m_j |\boldsymbol{v} - \boldsymbol{u}_{i,j}|^2}{2T_{j,i}} \right) \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} - \lambda_{j,i} m_j \int f_j \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \int \exp\left(-\frac{m_i|\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right) \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} - \lambda_{i,j} m_i \int f_i \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} \\
& + \lambda_{j,i} m_j n_j \left(\frac{m_j}{2\pi T_{j,i}}\right)^{\frac{d}{2}} \int \exp\left(-\frac{m_j|\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2}{2T_{j,i}}\right) \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v} - \lambda_{j,i} m_j \int f_j \frac{|\boldsymbol{v}|^2}{2} d\boldsymbol{v}.
\end{aligned} \tag{134}
$$

For simplicity, we examine the first two terms of this equation separately, and infer the final forms of the other two terms by switching the indices $i \leftrightarrow j$.

First consider the integral

$$
I_{3,1} \coloneqq \frac{1}{2}\lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \int \exp\left(-\frac{m_i|\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right) |\boldsymbol{v}|^2 d\boldsymbol{v}. \tag{135}
$$

Using $|\boldsymbol{v}| = |\boldsymbol{u}_{i,j} + (\boldsymbol{v}-\boldsymbol{u}_{i,j})|$, note that

$$
\begin{aligned}
|\boldsymbol{v}|^2 &= |\boldsymbol{u}_{i,j} + (\boldsymbol{v}-\boldsymbol{u}_{i,j})|^2 \\
&= (\boldsymbol{u}_{i,j} + (\boldsymbol{v}-\boldsymbol{u}_{i,j}))^T (\boldsymbol{u}_{i,j} + (\boldsymbol{v}-\boldsymbol{u}_{i,j})) \\
&= \boldsymbol{u}_{i,j}^T \boldsymbol{u}_{i,j} + (\boldsymbol{v}-\boldsymbol{u}_{i,j})^T \boldsymbol{u}_{i,j} + \boldsymbol{u}_{i,j}^T (\boldsymbol{v}-\boldsymbol{u}_{i,j}) + (\boldsymbol{v}-\boldsymbol{u}_{i,j})^T (\boldsymbol{v}-\boldsymbol{u}_{i,j}) \\
&= |\boldsymbol{u}_{i,j}|^2 + \boldsymbol{v}^T \boldsymbol{u}_{i,j} - \boldsymbol{u}_{i,j}^T \boldsymbol{u}_{i,j} + \boldsymbol{u}_{i,j}^T \boldsymbol{v} - \boldsymbol{u}_{i,j}^T \boldsymbol{u}_{i,j} + |\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2 \\
&= -|\boldsymbol{u}_{i,j}|^2 + 2\boldsymbol{u}_{i,j}^T \boldsymbol{v} + |\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2.
\end{aligned} \tag{136}
$$

Next, using the substitution given in (126), the integral becomes

$$
\begin{aligned}
I_{3,1} &= \frac{1}{2}\lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \int \exp\left(-\frac{m_i|\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right) |\boldsymbol{v}|^2 d\boldsymbol{v} \\
&= \frac{1}{2}\lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \Bigg[\left(-|\boldsymbol{u}_{i,j}|^2\right) \int e^{-|\boldsymbol{s}|^2} \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} d\boldsymbol{s} \\
&\quad + 2u_{i,j}^T \int \left(\boldsymbol{s}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}} + \boldsymbol{u}_{i,j}\right) e^{-|\boldsymbol{s}|^2} \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} d\boldsymbol{s} \\
&\quad + \int \left|\boldsymbol{s}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}}\right|^2 e^{-|\boldsymbol{s}|^2} \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} d\boldsymbol{s} \Bigg] \\
&= \frac{1}{2}\lambda_{i,j} m_i n_i \left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}} \Bigg[-|\boldsymbol{u}_{i,j}|^2 \left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}} \pi^{\frac{d}{2}}
\end{aligned}
$$

$$
\begin{aligned}
&+2\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{1}{2}}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\boldsymbol{u}_{i,j}^T\int \boldsymbol{s}e^{-|\boldsymbol{s}|^2}d\boldsymbol{s}\\
&+2\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\boldsymbol{u}_{i,j}^T\boldsymbol{u}_{i,j}\int e^{-|\boldsymbol{s}|^2}d\boldsymbol{s}+\left(\frac{2T_{i,j}}{m_i}\right)\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\int |\boldsymbol{s}|^2e^{-|\boldsymbol{s}|^2}d\boldsymbol{s}\Bigg]\\
=&\frac{1}{2}\lambda_{i,j}m_in_i\left(\frac{m_i}{2T_{i,j}}\right)^{\frac{d}{2}}\pi^{-\frac{d}{2}}\Bigg[-|\boldsymbol{u}_{i,j}|^2\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\pi^{\frac{d}{2}}\ +\ 0\\
&+2|\boldsymbol{u}_{i,j}|^2\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\pi^{\frac{d}{2}}+\left(\frac{2T_{i,j}}{m_i}\right)\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\frac{d}{2}\pi^{\frac{d}{2}}\Bigg],
\end{aligned}
\tag{137}
$$

where we have used the facts in Equations (125a), (125b), and (125c). Thus,

$$
\begin{aligned}
I_{3,1}&=\frac{1}{2}\lambda_{i,j}m_in_i\left(\frac{m_i}{2\pi T_{i,j}}\right)^{\frac{d}{2}}\int \exp\left(-\frac{m_i|\boldsymbol{v}-\boldsymbol{u}_{i,j}|^2}{2T_{i,j}}\right)|\boldsymbol{v}|^2d\boldsymbol{v}\\
&=\frac{1}{2}\lambda_{i,j}m_in_i\left(\frac{m_i}{2T_{i,j}}\right)^{\frac{d}{2}}\pi^{-\frac{d}{2}}\pi^{\frac{d}{2}}\left(\frac{2T_{i,j}}{m_i}\right)^{\frac{d}{2}}\left[|\boldsymbol{u}_{i,j}|^2+\frac{d}{2}\left(\frac{2T_{i,j}}{m_i}\right)\right]\\
&=\frac{1}{2}\lambda_{i,j}m_in_i|\boldsymbol{u}_{i,j}|^2+\frac{d}{2}\lambda_{i,j}m_in_i\left(\frac{T_{i,j}}{m_i}\right)\\
&=\frac{1}{2}\lambda_{i,j}\rho_i|\boldsymbol{u}_{i,j}|^2+\frac{d}{2}\lambda_{i,j}n_iT_{i,j}.
\end{aligned}
\tag{138}
$$

Next, using the expansion $|\boldsymbol{v}|^2=|\boldsymbol{u}_i+(\boldsymbol{v}-\boldsymbol{u}_i)|^2$, as in Equation (136), we have

$$
\begin{aligned}
I_{3,2}&\coloneqq\frac{1}{2}\lambda_{i,j}m_i\int f_i|\boldsymbol{v}|^2d\boldsymbol{v}\\
&=\frac{1}{2}\lambda_{i,j}m_i\int f_i\left(-|\boldsymbol{u}_i|^2+2\boldsymbol{u}_i^T\boldsymbol{v}+|\boldsymbol{v}-\boldsymbol{u}_i|^2\right)d\boldsymbol{v}\\
&=\frac{1}{2}\lambda_{i,j}m_i(-|\boldsymbol{u}_i|^2)\int f_id\boldsymbol{v}+\frac{1}{2}\lambda_{i,j}2\boldsymbol{u}_i^T\int f_i\boldsymbol{v}d\boldsymbol{v}+\lambda_{i,j}\int\frac{m_i}{2}f_i|\boldsymbol{v}-\boldsymbol{u}_i|^2d\boldsymbol{v}\\
&=-\frac{1}{2}\lambda_{i,j}|\boldsymbol{u}_i|^2m_in_i+\lambda_{i,j}\rho_i\boldsymbol{u}_i^T\boldsymbol{u}_i+\lambda_{i,j}\left(\frac{d}{2}\right)n_iT_i\\
&=-\frac{1}{2}\lambda_{i,j}|\boldsymbol{u}_i|^2\rho_i+\lambda_{i,j}\rho_i|\boldsymbol{u}_i|^2+\frac{d}{2}\lambda_{i,j}n_iT_i\\
&=\frac{1}{2}\lambda_{i,j}\rho_i|\boldsymbol{u}_i|^2+\frac{d}{2}\lambda_{i,j}n_iT_i.
\end{aligned}
\tag{139}
$$

Therefore, using Equations (138), (139), and their corresponding equations (found by

switching the indices $i \leftrightarrow j$), we get

$$\begin{aligned} I_3 &= \frac{1}{2}\lambda_{i,j}\rho_i|\boldsymbol{u}_{i,j}|^2 + \frac{d}{2}\lambda_{i,j}n_i T_{i,j} - \frac{1}{2}\lambda_{i,j}\rho_i|\boldsymbol{u}_i|^2 - \frac{d}{2}\lambda_{i,j}n_i T_i \\ &\quad + \frac{1}{2}\lambda_{j,i}\rho_j|\boldsymbol{u}_{j,i}|^2 + \frac{d}{2}\lambda_{j,i}n_j T_{j,i} - \frac{1}{2}\lambda_{j,i}\rho_j|\boldsymbol{u}_j|^2 - \frac{d}{2}\lambda_{j,i}n_j T_j \\ &= 0, \end{aligned} \tag{140}$$

where we have used the definition of $T_{i,j}$ and fact that $T_{i,j} = T_{j,i}$.

■

The converse of the last result is also true, as is easy to show. We omit the proof for the sake of brevity.

Lemma 3.2. *Suppose that, for any $i, j \in \{1, \ldots, N_s\}$, the following collision invariances hold:*

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i)\, d\boldsymbol{v} = 0, \tag{141}$$

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i)m_i\boldsymbol{v}\, d\boldsymbol{v} + \int_{\mathbb{R}^d} \lambda_{j,i}(M_{j,i} - f_j)m_j\boldsymbol{v}\, d\boldsymbol{v} = \boldsymbol{0}, \tag{142}$$

$$\int_{\mathbb{R}^d} \lambda_{i,j}(M_{i,j} - f_i)m_i|\boldsymbol{v}|^2\, d\boldsymbol{v} + \int_{\mathbb{R}^d} \lambda_{j,i}(M_{j,i} - f_j)m_j|\boldsymbol{v}|^2\, d\boldsymbol{v} = 0. \tag{143}$$

Then it must be that the mixture velocities and temperatures satisfy

$$\boldsymbol{u}_{i,j} = \frac{\lambda_{i,j}\rho_i\boldsymbol{u}_i + \lambda_{j,i}\rho_j\boldsymbol{u}_j}{\lambda_{i,j}\rho_i + \lambda_{j,i}\rho_j} \tag{144}$$

$$T_{i,j} = \frac{\lambda_{i,j}n_i T_i + \lambda_{j,i}n_j T_j}{\lambda_{i,j}n_i + \lambda_{j,i}n_j} + \frac{\lambda_{i,j}\rho_i\left(|\boldsymbol{u}_i|^2 - |\boldsymbol{u}_{i,j}|^2\right) + \lambda_{j,i}\rho_j\left(|\boldsymbol{u}_j|^2 - |\boldsymbol{u}_{j,i}|^2\right)}{d(\lambda_{i,j}n_i + \lambda_{j,i}n_j)}. \tag{145}$$

3.2 Numerics

To see what difficulties lie ahead for the numerical analysis of the multi-species case, let us consider the space homogeneous problem.

As before, the space homogeneous problem is as follows:

$$\frac{\partial f_i}{\partial t} = \sum_{j=1}^{N_s} \lambda_{i,j}(M_{i,j} - f_i), \qquad \text{for } i \in \{1, \cdots, N_s\}. \tag{146}$$

We discretize using the Backward Euler method:

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \sum_{j=1}^{N_s} \lambda_{i,j}(M_{i,j}^{n+1} - f_i^{n+1}) \tag{147}$$

$$\iff f_i^{n+1} = f_i^n + \Delta t \sum_{j=1}^{N_s} \lambda_{i,j} M_{i,j}^{n+1} - \Delta t f_i^{n+1} \sum_{j=1}^{N_s} \lambda_{i,j} \tag{148}$$

$$\iff f_i^{n+1} \left(1 + \Delta t \sum_{j=1}^{N_s} \lambda_{i,j}\right) = f_i^n + \Delta t \sum_{j=1}^{N_s} \lambda_{i,j} M_{i,j}^{n+1} \tag{149}$$

$$\iff f_i^{n+1} = \frac{f_i^n + \Delta t \sum_{j=1}^{N_s} \lambda_{i,j} M_{i,j}^{n+1}}{1 + \Delta t \sum_{j=1}^{N_s} \lambda_{i,j}}. \tag{150}$$

Note that Equation (150) has the terms $M_{i,j}^{n+1}$. Recall that in the single species case, the moments are constant with respect to time. Thus, the Maxwellian is also constant with respect to time, so that $M^n = M^{n+1}$. This is not the case in the multispecies case, since the moments are nonzero, in general, and it represents a significant numerical challenge. Care must be taken to give a proper implicit update for the collision operator, and this is the subject of current work in the project.

As with fluid equations, spatially adaptive meshes are a requirement for highly efficient simulations of flows with fine-scale structures in phase space. However, kinetic equations also require adaptivity, not only for the resolution of fine scale structures, but also to address the fact that the effective support of the kinetic distribution may vary dramatically in phase space. Typically the size of the domain is based on the temperature of the distribution; for multi-species problems, each species may have its own temperature, which adds a complication not found in the single species setting. In other words, the Maxwellians may require different phase space resolutions. One way that this can be addressed is by defining the Maxwellians on different compatible grids that adequately resolve their individual supports.

4 Summary and Next Steps

This report presents a review of basic background on the theory and numerical solution of BGK approximations for Boltzmann-type kinetic equations. The BGK equations, which simplify their Boltzmann equation counterparts, are highly nonlinear, nonlocal, and high-dimensional models of particle kinetics in rarified gases and plasmas. We discuss both single species models and self-consistent extensions to multi-species BGK models. Preliminary work toward the efficient numerical simulation of BGK-type kinetic equations is presented via several benchmark problems.

Theoretical aspects of single species BGK kinetic models have been presented in order to motivate and describe the numerical methods used. In particular, conservation and entropy dissipation properties were presented, along with an analysis of the space homogeneous (no advection) problem. Numerical methods must be sophisticated enough to respect these conservation and dissipation properties at the fully discrete level. A stable finite volume numerical approximation framework has been outlined to capture potential discontinuities in the approximate solutions. Since the problem can be numerically stiff for some parameter regimes, implicit and semi-implicit time integration schemes are of particular importance for stability. Particular focus is given to an implicit-explicit Runge-Kutta (IMEX-RK) time stepping, as they give a reasonable balance between accuracy, efficiency, and stability. However, in some cases, fully implicit integration strategies are demanded, and the work here is a stepping stone toward developing such algorithms and codes.

Several numerical benchmark problems and tests of the prototype MATLAB codes are presented, including the Sod shock tube and two-stream instability test cases. The tests are presented for $1x1v$ phase spaces, but code is currently under development for more realistic higher-dimensional cases. In particular, $1x3v$ and $2x3v$ codes for "slab geometries" will be designed and benchmarked in the near future. The numerical methods that are presented are inherently scalable, and, thus, the only impediment for efficient and stable numerical simulation is the increased number of degrees of freedom. An example $1x1v$ code, used for numerical solutions of the Sod shock tube problem, is given in Appendix B. This work is done to address the goal of developing practical stable and efficient semi-implicit and fully implicit time integrator strategies for the problem.

A very brief introduction to a particular self-consistent BGK-type model for multi-species particle kinetics is given. The combinatorial complexity of the model grows with the addition of distinct chemical constituents, making such multi-species models even more challenging

for numerical solution. There are several outstanding issues that this work will address in the future. In particular, as in the single species case, an implicit (or, at least, semi-implicit) approach is desired for the computation of the stiff collision operators for stability. However, the procedures used in the single species case do not work directly in the multi-species case. A sophisticated fixed point iterative scheme is currently in development to address this issue in computing the implicit update. The scheme is designed to relax the implicit updates of the velocity and temperature to appropriate values, depending on collisional frequencies that themselves depend on the implicit moments. This allows for the desired implicit update of the BGK collision operators. In the longer term, in addition to semi- and fully-implicit solvers, this work will focus on the design of fast adaptive phase-space methods and block-structured adaptive mesh refinement (AMR) that will efficiently accommodate disparate scales that are inherent in multi-species problems, owing, for example, to disparate particle sizes and temperatures. Incorporating implicit solver technology with AMR, especially in the context of such highly nonlinear and nonlocal models is expected to be challenging. But, the pay-off for efficient, stable simulation of high phase-space problems is vital to accommodate real-world time and space scales.

5 Acknowledgements

This work is supported by a contract from Oak Ridge National Laboratory under subcontract UTB-CW24420, "BGK Kinetic Equations." The authors wish to thank Cory Hauck (ORNL) for several useful discussions regarding numerical methods for BGK-type equations.

6 References

[1] P.L. Bhatnagar, E.P. Gross, and M. Krook. Model for collision processes in gases I: Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511, 1954.

[2] F. Coron and B. Perthame. Numerical passage from kinetic to fluid equations. *SIAM J. Numer. Anal.*, 28:26–42, 1991.

[3] C. Kristopher Garrett and Cory D. Hauck. A fast solver for implicit integration of the vlasov-poisson system in the eulerian framework. *SIAM J. Sci. Comput.*, 40, 2018.

[4] J.R. Haack, C.D. Hauck, and M.S. Murillo. A conservative, entropic multispecies bgk model. *J. Stat Phys*, 168:826–856, 2017.

[5] B.B. Hamel. Kinetic model for binary gas mixtures. *Physics of Fluids*, 8:418–425, 1965.

[6] H. Holway. New statistical models for kinetic theory: Methods of construction. *Phys. Fluids*, 9:1658, 1966.

[7] Christian Klingenberg and Marlies Pirner. Existence, uniqueness and positivity of solutions for bgk models for mixtures. *Journal of Differential Equations*, 264, 09 2017.

[8] Christian Klingenberg, Marlies Pirner, and Gabriella Puppo. A consistent kinetic model for a two-component mixture with an application to plasma. *Kinetic and Related Models*, 10(2):445–465, 2017.

[9] L. Mieussens. Discrete velocity model and implicit scheme for the bgk equation of rarefied gas dynamics. *Math. Models and Methods Appl. Sci.*, 10:1121–1149, 2000.

[10] L. Mieussens. Schemes for boltzmann-bgk equation in plane and axisymmetric geometries. *J. Comput. Phys.*, 162:429–466, 2000.

[11] L. Mieussens and H. Struchtrup. Numerical comparison of Bhatnagar-Gross-Krook models with proper Prandtl number. *Phys. Fluids*, 16:2797–2813, 2004.

[12] Stéphane Mischler. Uniqueness for the BGK-Equation in $\mathbb{R}^N$ and Rate of Convergence for a Semi-Discrete Scheme. *Differential and Integral Equations*, 9(5):1119 – 1138, 1996.

[13] B. Perthame and M. Pulvirenti. Weighted L^∞ bounds and uniqueness for the Boltzmann BGK model. *Archive Rat. Mech. Anal.*, 125(3):289–295, 1993.

[14] Benoît Perthame. Global existence to the bgk model of boltzmann equation. *J. Diff. Eq.*, 82:191–205, 1989.

[15] Sandra Pieraccini and Gabriella Puppo. Implicit-explicit schemes for bgk kinetic equations. *J. Sci. Comput.*, 32:1–28, 2007.

[16] E. Ringeissen. *Thesis*. PhD thesis, University of Paris VII, 1991.

[17] L. Saint-Raymond. From the BGK model to the Navier-Stokes equations. *Ann. Sci. Ecole Norm. Sup.*, 36:271–317, 2003.

[18] E.M. Shakhov. On the generalization of the Krook kinetic equation. *Izv. Russ. Acad. Sci. Fluid Dyn.*, 5:142–145, 1968.

[19] H. Struchtrup. *Macroscopic Transport Equations for Rarefied Gas Flows: Approximation Methods in Kinetic Theory*. Springer Verlag, Berlin, Germany, 2005.

[20] Eleuterio Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, Germany, 2nd edition, 1999.

[21] W.G. Vincenti and C.H. Kruger. *Introduction to Physical Gas Dynamics*. Krieger, Malabar, Florida, 1986.

[22] K. Xu. A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov methods. *J. Comput. Phys.*, 171:289–335, 2001.

A A Technical Lemma

Lemma A.1.

$$\int_{\mathbb{R}^d} |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} = \frac{d}{2}\pi^{\frac{d}{2}}. \tag{151}$$

Proof.

$$\begin{aligned}
\int_{\mathbb{R}^d} |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} &= \int_{\mathbb{R}^d} (s_1^2 + \cdots + s_d^2) e^{-(s_1^2+\cdots+s_d^2)} d\boldsymbol{s} \\
&= \int_{\mathbb{R}^d} \left(\sum_{j=1}^{d} s_j^2\right) \prod_{k=1}^{d} e^{-s_k^2} d\boldsymbol{s} \\
&= \int_{\mathbb{R}^d} s_1^2 \prod_{k=1}^{d} e^{-s_k^2} d\boldsymbol{s} + \cdots + \int_{\mathbb{R}^d} s_d^2 \prod_{k=1}^{d} e^{-s_k^2} d\boldsymbol{s} \\
&= \left(\prod_{\substack{k=1 \\ k\neq 1}}^{d} \int_{\mathbb{R}} e^{-s_k^2} ds_k\right) \int_{\mathbb{R}} s_1^2 e^{-s_1^2} ds_1 + \cdots + \left(\prod_{\substack{k=1 \\ k\neq d}}^{d} \int_{\mathbb{R}} e^{-s_k^2} ds_k\right) \int_{\mathbb{R}} s_d^2 e^{-s_d^2} ds_d \\
&= \left(\pi^{\frac{1}{2}}\right)^{d-1} \int_{\mathbb{R}} s_1^2 e^{-s_1^2} ds_1 + \cdots + \left(\pi^{\frac{1}{2}}\right)^{d-1} \int_{\mathbb{R}} s_d^2 e^{-s_d^2} ds_d \\
&= \pi^{\frac{d-1}{2}} \sum_{i=1}^{d} \int_{\mathbb{R}} s_i^2 e^{-s_i^2} ds_i,
\end{aligned} \tag{152}$$

where we have used many elementary integration techniques, and the fact that $\int_{\mathbb{R}} e^{-x^2} dx = \pi^{\frac{1}{2}}$. It remains to determine the value of the terms of the form $\int_{\mathbb{R}} s^2 e^{-s^2} ds$. To that end, utilizing integration by parts, with

$$u = s_i \implies du = ds_i \tag{153}$$

$$dv = 2s_i e^{-s_i^2} ds_i \implies v = -e^{-s_i^2}, \tag{154}$$

we have

$$\begin{aligned}
\frac{1}{2}\int_{\mathbb{R}} 2s_i^2 e^{-s_i^2} ds_i &= \frac{1}{2}\left[\left[-s_i e^{-s_i^2}\right]_{-\infty}^{\infty} - \int_{\mathbb{R}} -e^{-s_i^2} ds_i\right] \\
&= \frac{1}{2}\int_{\mathbb{R}} e^{-s_i^2} ds_i \\
&= \frac{1}{2}\pi^{\frac{1}{2}}.
\end{aligned} \tag{155}$$

Thus, Equation (152) gives us

$$\int_{\mathbb{R}^d} |\boldsymbol{s}|^2 e^{-|\boldsymbol{s}|^2} d\boldsymbol{s} = \pi^{\frac{d-1}{2}} \sum_{i=1}^{d} \frac{1}{2}\pi^{\frac{1}{2}} = \pi^{\frac{d-1}{2}} \left(\frac{d}{2}\pi^{\frac{1}{2}}\right) = \frac{d}{2}\pi^{\frac{d}{2}}, \tag{156}$$

as desired. ■

B Code

The MATLAB code listed in this appendix is that used to generate the results of Section 2.5.2 for the Sod shock tube benchmark problem. This code is developed for prototyping and demonstration purposes only and is not meant to represent production-quality software.

B.1 Main Driver: `vlasovPoissonBGKMain.m`

```
% Script to solve Vlasov-Poisson-BGK Equation:
%
% 1X1V
%
% f_t + v * f_x + a * f_v = 1/tau * (M-f)
%
% This set of code is designed to solve the Sod Shock tube problem.
% In the limit as (\tau -> 0) or (\lambda -> \infty), BGK -> Euler.
% So, for the current test, set tau=10^{-N}, set a = 0.
%
% Scheme:
% IMEX RK: Explicit Advection, Implicit Collision.
%
clear;
clc;
tic
%
% Number of ghost cells:
del = 1;
%
% Collision time:
tau = 10^(-4);
%
% Theta values for minmod:
thetaX = 2;
thetaV = 2;
%
% Spatial domain:
xL = -0.5;
xR =  0.5;
```

```
%
% Number of cells:
Nx = 256;
%
% Cell edge points:
x = linspace(xL,xR,Nx+1);
%
% Cell center points:
xC = x(1:end-1) + 0.5 * (x(2:end) - x(1:end-1));
xCenter = [xC(1:del) , xC , xC(end-del+1:end)];
hX = xC(2) - xC(1);
%
% Velocity domain:
vMin = -10;
vMax =  10;
%
% Number of cells:
Nv = 258;
%
% Cell edge points:
v = linspace(vMin,vMax,Nv+1);
%
% Cell center points:
vC = v(1:end-1) + 0.5 * (v(2:end) - v(1:end-1));
vCenter = [vC(1:del) , vC , vC(end-del+1:end)];
hV = vC(2) - vC(1);
%
% Define time levels:
tInit = 0;
tFin  = 0.2;
%
% Vectors for ghost cells
gVx = [1,length(xC)]+del;
gVv = [1,length(vC)]+del;
%
tableNum = 8;
[Ae,be,ce,Ai,bi,ci] = butcherTable(tableNum);
butcher.Ae = Ae; butcher.Ai = Ai;
butcher.be = be; butcher.bi = bi;
butcher.ce = ce; butcher.ci = ci;
```

```
%
% Stiffness matrix for the Poisson solve...zero Dirichlet BC
% (as per the test problems). This is not needed for the Sod problem,
% but I kept it so I didn't have to change all my function
% dependencies.
%
temp = zeros(1,Nx);
temp(1)=2; temp(2)=-1;
A = toeplitz(temp);
A(1,1) = 3; A(end,end) = 3;
%
grid.del = del;
grid.gVx = gVx; grid.Nx = Nx; grid.xC = xC; grid.xCenter = xCenter;
grid.gVv = gVv; grid.Nv = Nv; grid.vC = vC; grid.vCenter = vCenter;
grid.hX = hX;
grid.hV = hV;
grid.xL   = xL  ; grid.xR   = xR  ;
grid.vMin = vMin; grid.vMax = vMax;
grid.tInit = tInit; grid.tFin = tFin;
grid.thetaX = thetaX;
grid.thetaV = thetaV;
%
testNum = 1;
[f] = initialCondition(testNum,grid);
%
[tVec,y,MAX] = vlasovPoissonBGKSolver(f,butcher,grid,tau,A,f);
%
figure(107)
pcolor(xC,vC,y);
title(['Contour plot of computed f, in the Velocity/Space', ...
  ' Domain, for Time t = ', num2str(tFin), ' '])
xlabel('x')
ylabel('v')
shading interp;
colormap(jet);
colorbar;
%
toc
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
% Embedded functions below:
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [f] = initialCondition(testNum,grid)
%
xC = grid.xC; xCenter = grid.xCenter;
vC = grid.vC; vCenter = grid.vCenter;
%
% Initialize solution array:
f = zeros(length(vC),length(xC),1);
%
switch testNum
  case 1 % tFin = 0.250
    nL = 1.00000; uL =  0.00000; thetaL = 0001.000;
    nR = 0.12500; uR =  0.00000; thetaR = 0000.800;
  case 2 % tFin = 0.150
    nL = 1.00000; uL = -2.00000; thetaL = 0000.400;
    nR = 1.00000; uR =  2.00000; thetaR = 0000.400;
  case 3 % tFin = 0.012
    nL = 1.00000; uL =  0.00000; thetaL = 1000.000;
    nR = 1.00000; uR =  0.00000; thetaR = 0000.010;
  case 4 % tFin = 0.035
    nL = 1.00000; uL =  0.00000; thetaL = 0000.010;
    nR = 1.00000; uR =  0.00000; thetaR = 0100.000;
  case 5 % tFin = 0.035
    nL = 5.99924; uL = 19.59750; thetaL = 0460.894;
    nR = 5.99242; uR = -6.19633; thetaR = 046.0950;
end
%
% Initial condition (make a separate function?):
%
for j = 1:length(vC)
  for i = 1:length(xC)
    if xCenter(i) <= 0.0
      f(j,i) = nL / sqrt(2*pi*thetaL) * exp(-(vC(j)-uL)^2 ...
        / (2*thetaL));
    else
      f(j,i) = nR / sqrt(2*pi*thetaR) * exp(-(vC(j)-uR)^2 ...
```

```
        / (2*thetaR));
    end
  end
end
%
% %Plot the initial conditions....
% figure(110)
% % pcolor(x_c,v_c,f);
% surf(x_c,v_c,f); view(-30,50); zlim([0,2.1]);
% % title(['Contour plot of f, in the Velocity/Space Domain,' ...
% ' for Time t = ',num2str(0),' '])
% xlabel('x')
% ylabel('v')
% shading interp;
% colormap(jet);
% colorbar;
%
end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [Ae,be,ce,Ai,bi,ci] = butcherTable(table)
%
% Butcher Tableaux for the IMEX-RK scheme. Tables 2-6 come from
% the Pareschi & Russo paper. Table 7 is Backward Euler. Table 8
% (needs reference, from Cory).
%
switch table
  case 2
    Ae = [0,0 ; 1,0];
    be = [0.5 ; 0.5];
    ce = [0 ; 1];
%
    gam = 1-1/sqrt(2);
    Ai = [gam,0 ; 1-2*gam,gam];
    bi = [0.5 ; 0.5];
    ci = [gam ; 1-gam];
%
  case 3
    Ae = [0,0,0 ; 0,0,0 ; 0,1,0];
```

```
    be = [0 ; 0.5 ; 0.5];
    ce = [0 ; 0 ; 1];
%
    Ai = [0.5,0,0 ; -0.5,0.5,0 ; 0,0.5,0.5];
    bi = [0 ; 0.5 ; 0.5];
    ci = [0.5 ; 0 ; 1];
%
  case 4
    Ae = [0,0,0 ; 0.5,0,0 ; 0.5,0.5,0];
    be = [1/3 ; 1/3 ; 1/3];
    ce = [0 ; 0.5 ; 1];
%
    Ai = [0.25,0,0 ; 0,0.25,0 ; 1/3,1/3,1/3];
    bi = [1/3 ; 1/3 ; 1/3];
    ci = [0.25 ; 0.25 ; 1];
%
  case 5
    Ae = [0,0,0 ; 1,0,0 ; 0.25,0.25,0];
    be = [1/6 ; 1/6 ; 2/3];
    ce = [0 ; 1 ; 0.5];
%
    gam = 1-1/sqrt(2);
    Ai = [gam,0,0 ; 1-2*gam,gam,0 ; 0.5-gam,0,gam];
    bi = [1/6 ; 1/6 ; 2/3];
    ci = [gam ; 1-gam ; 0.5];
%
  case 6
    Ae = [0,0,0,0 ; 0,0,0,0 ; 0,1,0,0 ; 0,0.25,0.25,0];
    be = [0 ; 1/6 ; 1/6 ; 2/3];
    ce = [0 ; 0 ; 1 ; 0.5];
%
    alpha = 0.24169426078821;
    beta  = 0.06042356519705;
    eta   = 0.12915286960590;
    Ai = [alpha,0,0,0; -alpha,alpha,0,0; 0,1-alpha,alpha,0; ...
      beta,eta,0.5-beta-eta-alpha,alpha];
    bi = [0 ; 1/6 ; 1/6 ; 2/3];
    ci = [alpha ; 0 ; 1 ; 0.5];
%
  case 7
```

```
%
% Forward Euler:
    Ae = 0;
    be = 1;
    ce = 0;
%
% Backward Euler
    Ai = 1;
    bi = 1;
    ci = 1;
%
  case 8
    gam = 1 - 1/sqrt(2);
    delt = 1 - 1/(2*gam);
%
    Ae = [0,0,0 ; gam,0,0 ; delt,1-delt,0];
    be = [delt;1-delt;0];
    ce = [0;gam;1];
%
    Ai = [0,0,0 ; 0,gam,0 ; 0,1-gam,gam];
    bi = [0;1-gam;gam];
    ci = [0;gam;1];
end
end
```

Listing 1: Main driver: `vlasovPoissonBGKMain.m`.

B.2 vlasovPoissonBGKSolver.m

```
function [t,Z,M] = vlasovPoissonBGKSolver(y0,butcher,grid,tau,A,f)
%
% y0 = IC grid function (j,i) = (velocity,space)
% Ae, be, ce: Butcher Tableau for explicit solve
% Ai, bi, ci: Butcher tableau for implicit solve
%
Ai = butcher.Ai;
bi = butcher.bi;
ci = butcher.ci;
Ae = butcher.Ae;
be = butcher.be;
ce = butcher.ce;
%
gVx = grid.gVx; Nx = grid.Nx; hX = grid.hX;
gVv = grid.gVv; Nv = grid.Nv; hV = grid.hV;
%
% A is the stiffness matrix for the Poisson solve.
%
% Number of ghost cells
del = grid.del;
%
% Spatial domain:
xL = grid.xL; xR = grid.xR;
%
% Velocity domain:
vMin = grid.vMin; vMax = grid.vMax;
%
% Define CFL(s)
CFLMaxX = min(hX) / max(abs(vMax),abs(vMin));
CFLMaxV = min(hV) / max( abs(xL) , abs(xR) );
%
% Define initial/final time levels:
tCurrent = grid.tInit;
tFinal   = grid.tFin ;
%
% size of grid function array y including ghost cells:
d = size(y0)+2*del;
```

```
%
% Initialize array of solutions:
y = zeros([d,2]);
%
% Input initial conditions into solution vector
y(gVv(1):gVv(2),gVx(1):gVx(2),1) = f;
%
% Store appropriate initial values in ghost cells
for i = 1:del
  y(:,gVx(1)-i,1) = y(:,gVx(1),1);
  y(:,gVx(2)+i,1) = y(:,gVx(2),1);
end
for j = 1:del
  y(gVv(1)-j,:,1) = y(gVv(1),:,1);
  y(gVv(2)+j,:,1) = y(gVv(2),:,1);
end
%
% Number of stages of RK scheme:
s = length(ce);
%
% Initial vectors for RK scheme:
uo  = zeros([d,s]);
fI  = zeros([d,s]);
fE  = zeros([d,s]);
M   = zeros([d,s]);
%
redo = 0;
EMax = 1;
t = 1;
%
while tCurrent < tFinal
%
  dt = min(0.24 * min(CFLMaxX,CFLMaxV),tFinal-tCurrent) ...
    / (1 + (redo > 0));
%
% This flag indicates that the electric field gets bigger than E_max.
% If so, the step needs to recalculate.
  redo = 0;
%
  uo(:,:,:) = 0;
```

```
  uo(:,:,1) = y(:,:,1);
  [~,M(:,:,1)] = BGKCollision(tCurrent,y(:,:,1),grid,tau,tCurrent);
  u(:,:,1) = (tau*uo(:,:,1) + dt*Ai(1,1)*M(:,:,1)) ...
    / (tau + dt*Ai(1,1));
  fI(:,:,1) = BGKCollision(tCurrent+ci(1)*dt,u(:,:,1),grid, ...
    tau,tCurrent);
%
  [fE(:,:,1),re] = divFlux(tCurrent+ce(1)*dt,u(:,:,1),grid,EMax,A);
  redo = redo+re;

  for r = 2:s
    uo(:,:,r) = y(:,:,1) ...
      + dt * reshape(Ae(r,:) * reshape(permute(fE,[3,1,2]),s, ...
      (Nv+2*del)*(Nx+2*del)), Nv+2*del, Nx+2*del) ...
      + dt * reshape(Ai(r,:) * reshape(permute(fI,[3,1,2]),s, ...
      (Nv+2*del)*(Nx+2*del)), Nv+2*del, Nx+2*del);
%
    [~,M(:,:,r)] = BGKCollision(tCurrent+ci(r)*dt,uo(:,:,r), ...
      grid,tau,tCurrent);
%
    u(:,:,r) = (tau*uo(:,:,r) + dt*Ai(r,r)*M(:,:,r)) ...
      / (tau + dt*Ai(r,r));

    [fE(:,:,r),re] = divFlux(tCurrent+ce(r)*dt,u(:,:,r),grid,EMax,A);
    redo = redo+re;

    fI(:,:,r) = BGKCollision(tCurrent+ci(r)*dt,u(:,:,r), ...
      grid,tau,tCurrent);
  end
%
  y(:,:,2) = y(:,:,1) ...
    + (dt * reshape(be' * reshape(permute(fE,[3,1,2]),s, ...
    (Nv+2*del)*(Nx+2*del)), Nv+2*del,Nx+2*del) ...
    +  dt * reshape(bi' * reshape(permute(fI,[3,1,2]),s, ...
    (Nv+2*del)*(Nx+2*del)), Nv+2*del,Nx+2*del) ...
    ) * (1 - (redo > 0));
%
  y(:,:,1) = y(:,:,2);
  tCurrent = tCurrent + dt*(1 - (redo>0));
%
```

```
% This computes the moments and Maxwellian at the final time step.
  if tCurrent == tFinal
    BGKCollision(tCurrent+ci(r)*dt,y(:,:,1),grid,tau,tCurrent);
  end
end
%
% Return solution at final time
Z = y(gVv(1):gVv(2),gVx(1):gVx(2),1);
%
end
```

Listing 2: `vlasovPoissonBGKSolver.m`.

B.3 BGKCollision.m

```
function [z,M] = BGKCollision(t,y,grid,tau,tCurrent)
%
del = grid.del;
hx = grid.hX; gVx = grid.gVx; x = grid.xC;
hv = grid.hV; gVv = grid.gVv; v = grid.vC;
%
% Grid function array must have ghost cells already.
M = zeros(length(x)+2*del,length(v)+2*del);
nDens = zeros(1,length(x));
mDens = zeros(1,length(x));
EDens = zeros(1,length(x));

uDens = zeros(1,length(x));
theta = zeros(1,length(x));
%
% Moment integrals: n, n*u, and E.
nDens(1:length(x)) = hv * sum(y(gVv(1):gVv(2),1+del:length(x)+del));
mDens(1:length(x)) = hv * sum(v(:).*y(gVv(1):gVv(2), ...
  1+del:length(x)+del));
EDens(1:length(x)) = 0.5 * hv * sum(v(:).^2 .* y(gVv(1):gVv(2), ...
  1+del:length(x)+del));

uDens(:) = mDens(:)./nDens(:);

theta(:) = 2*EDens(:)./nDens(:)-uDens(:).^2;

M(gVx(1):gVx(2),gVv(1):gVv(2)) = nDens(:) ./ sqrt(2*pi*theta(:)) ...
  .* exp(-0.5 * abs(v - uDens(:)).^2 ./ theta(:));
%
% Ghost cells: zero flow boundaries.
for i = 1:del
  M(:,gVv(1)-i) = M(:,gVv(1));
  M(:,gVv(2)+i) = M(:,gVv(2));
end
for j = 1:del
  M(gVx(1)-j,:) = M(gVx(1),:);
  M(gVx(2)+j,:) = M(gVx(2),:);
```

```
end
%
if tCurrent == 0.2
%
% initial shock happens here.
  x0 = 0;
  gam = 3;
  uL = [1,0,1]; uR = [0.125,0,0.1];
%
  pDens = nDens .* theta;
%
  [nSod,uSod,pSod,eSod] = sodSoln(x,tCurrent,uL,uR,gam,x0);
%
  figure(1)
  plot(x,nDens,'b','LineWidth',3)
  hold on
  plot(x,nSod,'LineWidth',3)
  xlim([-0.5,0.5])
  ylim([0,1])
  set(gca,'fontsize',24)
  legend('BGK','Euler')
  hold off
%
  figure(2)
  plot(x,uDens,'b','LineWidth',3)
  hold on
  plot(x,uSod,'LineWidth',3)
  xlim([-0.5,0.5])
  ylim([0,1])
  set(gca,'fontsize',24)
  legend('BGK','Euler')
  hold off
%
  figure(3)
  plot(x,theta,'b','LineWidth',3)
  hold on
  plot(x,2*eSod,'LineWidth',3)
  xlim([-0.5,0.5])
  ylim([0,2]) % may need to shift this
  set(gca,'fontsize',24)
```

```
  legend('BGK','Euler')
  hold off
end
%
z = (M'-y)/tau;
%
% M is a [Nv]X[Nx] array after transpose
M = M';
%
end
```

Listing 3: `BGKCollision.m`

B.4 divFlux.m

```
function [z,redo] = divFlux(t,y,grid,EMax,A)
%
% y = grid function of f_{ij}^k values for fixed k
%
del = grid.del;
hx = grid.hX; gVx = grid.gVx; x = grid.xCenter; thetaX = grid.thetaX;
hv = grid.hV; gVv = grid.gVv; v = grid.vCenter; thetaV = grid.thetaV;
%
fP = zeros(size(y));
fM = zeros(size(y));
gP = zeros(size(y));
gM = zeros(size(y));
%
% construct slopes
[sX(:,:),sV(:,:)] = slopeReconstruction(y,grid);
%
% % Spatial fluxes
fP(:,gVx(1):gVx(2)) = (y(:,gVx(1):gVx(2)) ...
  + 0.5*hx*sX(:,gVx(1):gVx(2))) .* v(:).*(v>0)' ...
  +(y(:,gVx(1)+1:gVx(2)+1) - 0.5*hx*sX(:,gVx(1)+1:gVx(2)+1)) ...
  .* v(:).*(v<=0)' ;
fM(:,gVx(1):gVx(2)) = (y(:,gVx(1)-1:gVx(2)-1) ...
  + 0.5*hx*sX(:,gVx(1)-1:gVx(2)-1)) .* v(:).*(v>0)' ...
  +(y(:,gVx(1):gVx(2)) - 0.5*hx*sX(:,gVx(1):gVx(2))) ...
  .* v(:).*(v<=0)' ;
%
% The electric field is unnecessary for this test, so we set it to
% zero here.
E = zeros(1,length(x));
redo = 0;

% % Velocity fluxes
gP(gVv(1):gVv(2),:) = (y(gVv(1):gVv(2),:) ...
  + 0.5*hv*sV(gVv(1):gVv(2),:)) .* E.*(E>0) ...
  + (y(gVv(1)+1:gVv(2)+1,:) - 0.5*hv*sV(gVv(1)+1:gVv(2)+1,:)) ...
  .* E.*(E<=0) ;
gM(gVv(1):gVv(2),:) = (y(gVv(1)-1:gVv(2)-1,:) ...
```

```
  + 0.5*hv*sV(gVv(1)-1:gVv(2)-1,:)) .* E.*(E>0) ...
  + (y(gVv(1):gVv(2),:) - 0.5*hv*sV(gVv(1):gVv(2),:)) ...
  .* E.*(E<=0) ;

z = - (fP - fM)/hx - (gP - gM)/hv;

end
```

Listing 4: `divFlux.m`

B.5 slopeReconstruction.m

```
function [sX,sV] = slopeReconstruction(y,grid)
%
sX = zeros(size(y));
sV = zeros(size(y));
%
del = grid.del;
hx = grid.hX; gVx = grid.gVx; thetaX = grid.thetaX;
hv = grid.hV; gVv = grid.gVv; thetaV = grid.thetaV;
%
SV(:,:,1) = (y(gVv(1)+1:gVv(2)+1,gVx(1):gVx(2)) ...
  - y(gVv(1)-1:gVv(2)-1,gVx(1):gVx(2))) / 2 ;
SV(:,:,2) = thetaV * (y(gVx(1):gVv(2),gVx(1):gVx(2)) ...
  - y(gVv(1)-1:gVv(2)-1,gVx(1):gVx(2))) ;
SV(:,:,3) = thetaV * (y(gVx(1)+1:gVv(2)+1,gVx(1):gVx(2)) ...
  - y(gVv(1):gVv(2),gVx(1):gVx(2))) ;
%
sV(gVv(1):gVv(2),gVx(1):gVx(2)) = minMod(SV)/hv;
%
SX(:,:,1) = (y(gVv(1):gVv(2),gVx(1)+1:gVx(2)+1) ...
  - y(gVv(1):gVv(2),gVx(1)-1:gVx(2)-1)) / 2 ;
SX(:,:,2) = thetaX * (y(gVx(1):gVv(2),gVx(1):gVx(2)) ...
  - y(gVv(1):gVv(2),gVx(1)-1:gVx(2)-1)) ;
SX(:,:,3) = thetaX * (y(gVx(1):gVv(2),gVx(1)+1:gVx(2)+1) ...
  - y(gVv(1):gVv(2),gVx(1):gVx(2))) ;
%
sX(gVv(1):gVv(2),gVx(1):gVx(2)) = minMod(SX)/hx;
%
for i = 1:del
  sX(:,gVx(1)-i) = sX(:,gVx(1));
  sX(:,gVx(2)+i) = sX(:,gVx(2));
end
for j = 1:del
  sV(gVv(1)-j,:) = sV(gVv(2),:);
  sV(gVv(2)+j,:) = sV(gVv(1),:);
end
%
```

```
end
```

Listing 5: `slopeReconstruction.m`

B.6 minMod.m

```
function M = minMod(x)
%
M(:,:) = 0.25*abs(sign(x(:,:,1))+sign(x(:,:,2))) ...
  .*(sign(x(:,:,1))+sign(x(:,:,3))).* min(abs(x),[],3);
%
end
```

Listing 6: minMod.m

B.7 sodSoln.m

```
function [n,u,p,e] = sodSoln(x,tCurrent,phiL,phiR,gam,x0)
%
% Script to compute theoretical sod solution at given time with
% given gamma.
%
%
% phiL and phiR are ordered as (n,u,p)
nL = phiL(1); uL = phiL(2); pL = phiL(3);
nR = phiR(1); uR = phiR(2); pR = phiR(3);
%
% This function does an iterative solve to compute the value of
% pStar.
pStar = pStarSolve(10^(-8));
thL = pL / nL;
thR = pR / nR;

aL = sqrt(3*thL);
aStarL = aL * (pStar/pL)^((gam-1)/(2*gam));
aR = sqrt(3*thR);

nFan = nL * (2/(gam+1) + (gam-1)*(uL-x/tCurrent) ...
  /(gam+1)/aL).^(2/(gam-1));

nStarL = nL * (pStar/pL)^(1/gam);
nStarR = nR * ( (pStar/pR + (gam-1)/(gam+1)) ...
  / ((gam-1)*pStar/(gam+1)/pR + 1) );

uFan = 2/(gam+1) * (aL + (gam-1)*uL/2 + x/tCurrent);

uStar = uL - 2*aL / (gam-1) * ((pStar/pL)^((gam-1)/(2*gam)) - 1);

pFan = pL * (2/(gam+1) + (gam-1)*(uL-x/tCurrent) ...
  / (gam+1) / aL).^(2*gam/(gam-1));

eL = 0.5*pL/nL;
eStarL = 0.5*pStar/nStarL;
eStarR = 0.5*pStar/nStarR;
```

```
eR = 0.5*pR/nR;

sHL = uL - aL;
sTL = uStar - aStarL;
%
% speed of the contact wave.
% lambda_2(U_star_L) = S_2 = lambda_2(U_star_R)
% Toro book: Equation 2.134, and page 96
sC = uStar;sR = uR + aR*( (gam+1)*pStar / (2*gam*pR)  ...
  + 0.5*(gam-1)/gam)^(0.5);

nFunc = @(x) nL .* ((x<x0 + sHL*tCurrent)) ...
  + nFan .* ((x0 + sHL*tCurrent <= x) & (x < x0 + sTL*tCurrent)) ...
  + nStarL .* ((x0 + sTL*tCurrent <= x) & (x < x0 + sC*tCurrent)) ...
  + nStarR .* ((x0 + sC*tCurrent <= x) & (x < x0 + sR*tCurrent)) ...
  + nR .* ((x0 + sR*tCurrent <= x));

uFunc = @(x) uL .* ((x<x0 + sHL*tCurrent)) ...
  + uFan .* ((x0 + sHL*tCurrent <= x) & (x < x0 + sTL*tCurrent)) ...
  + uStar .* ((x0 + sTL*tCurrent <= x) & (x < x0 + sR*tCurrent)) ...
  + uR .* ((x0 + sR*tCurrent <= x));

pFunc = @(x) pL .* ((x<x0 + sHL*tCurrent)) ...
  + pFan .* ((x0 + sHL*tCurrent <= x) & (x < x0 + sTL*tCurrent)) ...
  + pStar .* ((x0 + sTL*tCurrent <= x) & (x < x0 + sR*tCurrent)) ...
  + pR .* ((x0 + sR*tCurrent <= x));

eFunc = @(x) eL .* ((x<x0 + sHL*tCurrent)) ...
  + (0.5*pFan./nFan) .* ((x0 + sHL*tCurrent <= x) ...
  & (x < x0 + sTL*tCurrent)) ...
  + eStarL .* ((x0 + sTL*tCurrent <= x) & (x < x0 + sC*tCurrent)) ...
  + eStarR .* ((x0 + sC*tCurrent <= x) & (x < x0 + sR*tCurrent)) ...
  + eR .* ((x0 + sR*tCurrent <= x));

n = nFunc(x);
u = uFunc(x);
p = pFunc(x);
e = eFunc(x);
```

```
end
```

Listing 7: `sodSoln.m`

B.8 pStarSolve.m

```
function pStar = pStarSolve(TOL)
%
% Newton Raphson to compute the pressure in Sod problem.
%
ERR = 1;
%
% Initial guess.
pOld = 0.9;
k = 0;
%
while ERR > TOL
    pNew = pOld - (sqrt(3)*pOld^(1/3) - sqrt(3) + ...
      2*(pOld-0.1)/sqrt(pOld+0.05)) / (sqrt(3)*pOld^(-2/3)/3 ...
      + 2*(pOld+0.05)^(-0.5) - (pOld-0.1)*(pOld+0.05)^(-1.5));
    ERR = abs(pOld-pNew) / (0.5*(pOld+pNew));
    pOld = pNew;
    k = k+1;
end
%
pStar = pNew;
%
end
```

Listing 8: pStarSolve.m