MODELING STRONGLY-CORRELATED PLASMAS WITH HYDRODYNAMIC DENSITY FUNCTIONAL THEORY

A Thesis

Presented to

The Faculty of the Department of Mathematics and Statistics

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Chris M. Gerlach

May 2023

The Designated Thesis Committee Approves the Thesis Titled

MODELING STRONGLY-CORRELATED PLASMAS WITH HYDRODYNAMIC DENSITY FUNCTIONAL THEORY

by

Chris M. Gerlach

APPROVED FOR THE DEPARTMENT OF MATHEMATICS AND STATISTICS

SAN JOSÉ STATE UNIVERSITY

Dr. Liam Stanton	Department of Mathematics & Statistics
Dr. Daniel Brinkman	Department of Mathematics & Statistics
Dr. Michael Kaufman	Department of Physics & Astronomy

May 2023

ABSTRACT

MODELING STRONGLY-CORRELATED PLASMAS WITH HYDRODYNAMIC DENSITY FUNCTIONAL THEORY

by Chris M. Gerlach

Strongly-coupled plasmas, such as ultracold neutral plasmas, dusty plasmas and warm dense matter, can be difficult model, as a complete understanding of the physics relies on both the dynamics and the underlying particle correlations. Density functional theory (DFT) is a natural formalism for describing such correlations but is limited to equilibrium systems. For non-equilibrium systems, hydrodynamic DFT (HDFT) provides a dynamic generalization of DFT that has recently been applied to plasmas and other fluids [1, 2]. One of the primary advantages of HDFT is that it establishes a direct connection to atomic-scale correlations self-consistently and without the need for an ad hoc equation of state. In this thesis, we extend the HDFT model to include the dynamics of a temperature field, which can be highly relevant to the description of plasmas, as well as examine various choices of correlation functionals in the HDFT model. After deriving the model, we then solved the governing equations numerically using a finite volume method. Furthermore, we address some of the computational challenges that arise from the nonlocal correlation effects as well as the theoretical challenges that arise from heterogeneous and strongly-coupled systems. Finally, we explore the role that correlations play in plasma waves.

DEDICATION

To Mom, Dad and Peter

TABLE OF CONTENTS

Modeling Strongly-Correlated Plasmas With Hydrodynamic Density Functional Theory

APPENDIX

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION TO PLASMAS

Plasmas, the fourth state of matter after solids, liquids, and gases, are complex and unique forms of matter composed of electrically charged particles, including ions, electrons, and neutral atoms. With electrons usually unbound from the nuclei, plasmas exhibit high conductivity and are capable of generating and responding to electric and magnetic fields over large length scales. They are found in diverse natural and artificial environments, ranging from stars, auroras and lighting to neon lighting and fusion reactors, and are critical in numerous technological applications such as semiconductor manufacturing, plasma cutting, and space propulsion using plasma thrusters.

One of the most significant areas of plasma research is in fusion energy, where scientists aim to replicate the process that powers the stars by merging atomic nuclei to release vast amounts of energy for clean and abundant power on Earth. A prominent research focus within plasma physics is strongly-coupled plasmas, characterized by high densities and strong particle interactions. In such plasmas, particles are densely packed, deviating from ideal gas behavior commonly assumed in traditional plasma models. Strongly-coupled plasmas arise in both of the primary avenues for fusion power: (1) inertial confinement fusion, where plasmas are compressed to densities many orders of magnitude higher than a solid, and (2) magnetic confinement fusion, where plasmas become strongly-coupled at the plasma-wall interfaces within tokamaks.

1.1 Plasma Modeling

Strongly-coupled plasmas exist in a regime near the transition between liquids and plasmas (sometimes referred to the warm dense matter regime). These plasmas exhibit the nonlocal effects typically associated with solids and the dynamic effects of a gas, making them difficult to model. In plasma modeling, molecular dynamics, kinetic theory, and continuum mechanics are powerful tools, each with distinct approaches to describing plasma behavior.

Molecular dynamics is a simulation technique that models individual particle behavior in plasmas, considering their interactions with each other. This approach provides insights into plasma behavior at the atomic level, including the behavior of ions and electrons. Molecular dynamics models are useful for studying phenomena such as plasma fusion, plasma turbulence, and plasma surface interactions, although they can be computationally intensive.

In contrast, kinetic theory describe plasma behavior at a microscopic level, accounting for the individual particles' behavior. Kinetic models provide insights into particle distribution, energies, and velocities, making them suitable for studying low-density plasmas and short time scales. They are commonly employed in studying phenomena such as plasma heating, wave-particle interactions, and collisions between particles.

Continuum mechanics describes plasmas as a continuous field, describing their behavior in terms of bulk properties like density, velocity, and pressure. This approach is suitable for macroscopic and long-term studies of plasma behavior, such as plasma instabilities, shock waves, and turbulence. However, hydrodynamic models may be less effective for low-density plasmas and short time scales that require detailed descriptions of individual particle behavior.

1.2 Density Functional Theory

Initially, Density Functional Theory (DFT) was created for quantum mechanics, with models such as Thomas-Fermi and Hartree-Fock being among the first to apply DFT to many-electron quantum systems [4, 5, 6]. However, a classical version of DFT also exists that accounts for particle motion in non-quantized systems. Ebner, Saam, and Stroud developed a similar approach to the Hohenberg-Kohn model [7], which predicts the behavior of fluids in equilibrium through a one-body density function [8]. This classical model accurately describes fluid behavior in equilibrium and has made DFT an essential tool for understanding properties of classical fluids derived from classical mechanics.

A variational approach to DFT was then introduced by Evans [9], which separated different fluid behaviors in inhomogeneous fluids using the Helmholtz Free Energy. This energy is a potential that describes a system's current state, making it a valuable quantity for calculating equilibrium properties [10]. Minimizing the Helmholtz Free Energy determines a fluid's equilibrium state, so much of the development of DFT as a field has been focused on minimizing this potential.

The Helmholtz Free Energy is usually divided into an ideal, external, and excess term, with the excess term containing additional correlations that correspond to fluid inhomogeneities [9]. To match classical hydrodynamic models, several approximation methods have been created, including Local and Weighted Density Approximations [11], Mean Field and Mayer Expansions [12], Linear Response [13], and Gradient Expansions.

Dynamic Density Functional Theory (DDFT) extends DFT to time-dependent fluid models. Proposed by Marconi and Tarazona [14, 15], DDFT describes a fluid's time evolution by assuming the adiabatic approximation, where the system is assumed to be in local equilibrium at any given time. The free energy functional used to describe the fluid's equilibrium state is given time dependence, where the density field becomes the primary variable. DDFT has successfully studied a range of dynamical phenomena in fluids, including hydrodynamic instabilities, phase separation dynamics, and transport properties [16, 17, 18].

Hydrodynamics is the study of fluids in motion, including liquids and gases, and the forces acting upon them. It seeks to quantify macroscopic fields such as fluid velocity, pressure, density, and the forces acting on fluids, such as friction and buoyancy. This field is incredibly valuable because fluid behavior is ubiquitous across all areas of science and engineering.

Historically, scientists used empirical methods to measure changes in these macroscopic variables. For example, the study of gaseous fluids led to the discovery of

Boyle's, Charles', and Avogadro's Law, which relate to the properties mentioned above along with the Ideal Gas Law [19].

As science became more rigorous in its pursuit of understanding, more intense mathematical modeling was used to capture the nature of fluids. The principles of hydrodynamics were derived from fundamental laws of physics such as the conservation of mass, momentum, and energy. Leonhard Euler developed the famous Euler equations that gave new precision to how fluids move [20]. More complicated models followed, including the Navier-Stokes equations [21, 22], which still measure visible macroscopic variables.

Hydrodynamic Density Functional Theory (HDFT) is a new field that attempts to combine the traditional macroscopic study of hydrodynamics with the additional microscopic accuracy of DDFT [1, 23]. The development of DDFT offers a more systematic approach to understanding how all interactions on a microscopic scale create and influence the measurable properties of a fluid. Recently, HDFT has attempted to merge these two disparate fields into one [1, 2], resulting in a more comprehensive understanding of fluid dynamics.

CHAPTER 2

MODELING PLASMAS WITH FLUID EQUATIONS

In this chapter, we will use Hydrodynamic Density Functional Theory [1] to derive a system of fluid equations to describe strongly-coupled plasmas. We begin by considering a system of N identical particles with mass m, position $\mathbf{r}_i$, and momentum $\mathbf{p}_i$ for each particle $i \in 1, 2, \ldots, N$. Let $\mathbf{r}^N = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$ be the set of position coordinates and $\mathbf{p}^N = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ be the set of momentum coordinates. Newton's second law of motion states that:

$$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{p}_i}{m}, \quad \frac{d\mathbf{p}_i}{dt} = \mathbf{F}_i(\mathbf{r}_i), \tag{2.1}$$

where $\mathbf{F}_i$ is the sum of forces on particle i given by $\mathbf{F}_i = -\nabla_{\mathbf{r}_i} V(\mathbf{r}^N, t)$, and V is the potential energy. We assume here that the potential energy can be decomposed into one-body, two-body, three-body interactions, and so on.

$$\begin{aligned} V(\mathbf{r}^N, t) = \sum_{i=1}^{N} V_{\text{ext}}(\mathbf{r}_i, t) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} V_2(\mathbf{r}_i(t), \mathbf{r}_j(t)) \\ + \frac{1}{6}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} V_3(\mathbf{r}_i(t), \mathbf{r}_j(t), \mathbf{r}_k(t)) + \ldots \end{aligned} \tag{2.2}$$

Assuming a constant mass for each particle, we obtain the more familiar form of Newton's second law:

$$m\frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{F}_i(\mathbf{r}_i). \tag{2.3}$$

We can define the classical Hamiltonian of the N-particle system, denoted as H_N, as the sum of the kinetic energy K and potential energy V

$$\begin{aligned} H_N(\mathbf{r}^N, \mathbf{p}^N) &= K(\mathbf{p}^N, t) + V(\mathbf{r}^N, t) \\ &= \sum_{i=1}^{N} \frac{|\mathbf{p}_i|^2}{2m} + \sum_{i=1}^{N} V_{\text{ext}}(\mathbf{r}_i, t) + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_2(\mathbf{r}_i, \mathbf{r}_j, t) + \ldots . \end{aligned} \tag{2.4}$$

The Hamiltonian can be interpreted as the total energy of the system and can be used to derive Hamilton's equations as an alternate formulation to Newtonian mechanics

$$\frac{d\mathbf{r}_i}{dt} = \frac{\partial H_N}{\partial \mathbf{p}_i}, \quad \frac{d\mathbf{p}_i}{dt} = -\frac{\partial H_N}{\partial \mathbf{r}_i}. \tag{2.5}$$

2.1 Liouville's Equation and the BBGKY Hierarchy

We now consider the probability distribution function $f_N(\mathbf{r}^N, \mathbf{p}^N, t)$, which describes the probability of the system being in a specific state $(\mathbf{r}^N, \mathbf{p}^N)$ at time t. Our goal is to understand how f_N evolves over time which we can do using the Liouville Theorem that states that the volume of any region within the phase space of an N-body Hamiltonian system is conserved. Using this theorem, we set the total derivative with respect to time to zero to obtain the Liouville equation

$$\frac{df_N}{dt} = \frac{\partial f_N}{\partial t} + \frac{\partial f_N}{\partial \mathbf{r}^N} \cdot \frac{d\mathbf{r}^N}{dt} + \frac{\partial f_N}{\partial \mathbf{v}^N} \cdot \frac{d\mathbf{v}^N}{dt} = 0 \tag{2.6}$$

For brevity, we now translate from momentum coordinates to velocity coordinates, i.e., $\mathbf{p}^N \to \mathbf{v}^N$ using the classical relation $\mathbf{p}_i = m\mathbf{v}_i$ for all $i = 1, 2, \ldots, N$. Substituting the velocity coordinates and using Equation (2.1), we obtain the more explicit form

$$\frac{\partial f_N}{\partial t} = -\sum_{i=1}^{N} \nabla_{\mathbf{r}_i} f_N \cdot \mathbf{v}_i - \sum_{i=1}^{N} \nabla_{\mathbf{v}_i} f_N \cdot \mathbf{F}_i(\mathbf{r}_i), \tag{2.7}$$

where we have introduced the gradient in terms of a particular particle's position $\nabla_{\mathbf{r}_i} f_N = \frac{\partial f_N}{\partial \mathbf{r}_i}$ and velocity $\nabla_{\mathbf{v}_i} f_N = \frac{1}{m}\frac{\partial f_N}{\partial \mathbf{p}_i}$. Using Equation (2.5), we can write the Liouville Equation in a more succinct form:

$$\frac{\partial f_N}{\partial t} = -\sum_{i=1}^{N} \frac{\partial f_N}{\partial \mathbf{r}_i} \cdot \frac{\partial H_N}{\partial \mathbf{p}_i} - \sum_{i=1}^{N} \frac{\partial f_N}{\partial \mathbf{v}_i} \cdot \frac{\partial H_N}{\partial \mathbf{r}_i} = \{f_N, H_N\}_N\,, \tag{2.8}$$

where $\{A, B\}_M$ are the Poisson brackets defined by

$$\{A, B\}_M \equiv \sum_{i=1}^{M} \left(\frac{\partial A}{\partial \mathbf{r}_i} \cdot \frac{\partial B}{\partial \mathbf{p}_i} - \frac{\partial A}{\partial \mathbf{p}_i} \cdot \frac{\partial B}{\partial \mathbf{r}_i} \right). \tag{2.9}$$

Next, we consider the marginal probability density function f_{N-1}, which describes the phase space of the first $N-1$ particles. We can obtain f_{N-1} by integrating over $\mathbf{r}_N$ and $\mathbf{v}_N$ as

$$f_{N-1}(\mathbf{r}^{N-1}, \mathbf{v}^{N-1}, t) = N! \iint f_N \, d\mathbf{r}_N \, d\mathbf{v}_N. \tag{2.10}$$

For brevity, integrals without explicitly written boundaries are over $\mathbb{R}^3$. Now, using the approximation for the potential energy up to two-body interactions in Equation (2.2), we can rewrite the force on the i^{th} particle as:

$$\mathbf{F}(\mathbf{r}_i) = \mathbf{F}_{\text{ext}}(\mathbf{r}_i) + \sum_{j \neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j), \tag{2.11}$$

where $\mathbf{F}_{\text{ext}}(\mathbf{r}_i)$ represents the sum of all external forces acting on particle i, and $\mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)$ represents the force acting upon particle i by particle j. We can now consider the time evolution of f_{N-1} due to the one-body and two-body interactions.

Differentiating Equation (2.10) and substituting Equation (2.11), we obtain:

$$\begin{aligned}\frac{\partial}{\partial t} f_{N-1} = &- \iint \sum_{i=1}^{N} \nabla_{\mathbf{r}_i} f_N \cdot \mathbf{v}_i \, d\mathbf{r}_N \, d\mathbf{v}_N \\ &+ \iint \sum_{i=1}^{N} \nabla_{\mathbf{v}_i} f_N \cdot \left(\mathbf{F}_{\text{ext}}(\mathbf{r}_i) + \sum_{j \neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j) \right) d\mathbf{r}_N \, d\mathbf{v}_N . \end{aligned} \tag{2.12}$$

The integrals over $\mathbf{r}_N$ and $\mathbf{v}_N$ can now be evaluated term by term in Equation (2.12) by separating the contributions of the N^{th} particle from the first $N-1$. For example, the first term can be simplified as

$$\iint \sum_{i=1}^{N} \nabla_{\mathbf{r}_i} f_N \cdot \mathbf{v}_i \, d\mathbf{r}_N \, d\mathbf{v}_N$$

$$= \iint \sum_{i=1}^{N-1} \nabla_{\mathbf{r}_i} f_N \cdot \mathbf{v}_i \, d\mathbf{r}_N \, d\mathbf{v}_N + \iint \nabla_{\mathbf{r}_N} f_N \cdot \mathbf{v}_N \, d\mathbf{r}_N \, d\mathbf{v}_N \tag{2.13}$$

$$= \sum_{i=1}^{N-1} \nabla_{\mathbf{r}_i} \left(\iint f_N \, d\mathbf{r}_N \, d\mathbf{v}_N \right) \cdot \mathbf{v}_i \tag{2.14}$$

$$= \frac{1}{N!} \sum_{i=1}^{N-1} \nabla_{\mathbf{r}_i} f_{N-1} \cdot \mathbf{v}_i , \tag{2.15}$$

where we have used the fact that the integral over the N^{th} particle is zero along with the definition of f_{N-1}. The contributions from the external force can be separated in a similar fashion.

$$\iint \sum_{i=1}^{N} \nabla_{\mathbf{v}_i} f_N \cdot \mathbf{F}_{\text{ext}}(\mathbf{r}_i) \, d\mathbf{r}_N \, d\mathbf{v}_N$$

$$= \iint \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} f_N \cdot \mathbf{F}_{\text{ext}}(\mathbf{r}_i) \, d\mathbf{r}_N \, d\mathbf{v}_N + \iint \nabla_{\mathbf{v}_N} f_N \cdot \mathbf{F}_{\text{ext}}(\mathbf{r}_N) \, d\mathbf{r}_N \, d\mathbf{v}_N \tag{2.16}$$

$$= \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} \left(\iint f_N \, d\mathbf{r}_N \, d\mathbf{v}_N \right) \cdot \mathbf{F}_{\text{ext}}(\mathbf{r}_i) \tag{2.17}$$

$$= \frac{1}{N!} \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} f_{N-1} \cdot \mathbf{F}_{\text{ext}}(\mathbf{r}_i) , \tag{2.18}$$

where again, the integral over the N^{th} particle vanishes, and only the interactions between the first $N-1$ particles are left over. And one last time, the N^{th} particle contributions can be separated as

$$\begin{aligned}
\iint \sum_{i=1}^{N} \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N \\
&= \iint \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N \\
&\quad + \iint \nabla_{\mathbf{v}_N} f_N \cdot \sum_{j\neq N}^{N} \mathbf{F}_{Nj}(\mathbf{r}_N - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N && (2.19) \\
&= \sum_{i=1}^{N-1} \iint \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N && (2.20) \\
&= \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j) && (2.21)
\end{aligned}$$

where the integrals over the N^{th} particle vanishes, and only the interactions between the first $N-1$ particles are left over. Combining Equation (2.13), Equation (2.16) and Equation (2.19), we get:

$$\begin{aligned}
\frac{\partial}{\partial t} f_{N-1} + \sum_{i=1}^{N-1} \nabla_{\mathbf{r}_i} f_{N-1} \cdot \mathbf{v}_i + \sum_{i=1}^{N-1} \nabla_{\mathbf{v}_i} f_{N-1} \cdot \mathbf{F}_i(\mathbf{r}_i) && (2.22) \\
= -N! \sum_{i=1}^{N-1} \iint \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N. && (2.23)
\end{aligned}$$

The equation can be reduced using Poisson brackets to

$$\frac{\partial}{\partial t} f_{N-1} + \{H_{N-1}, f_{N-1}\} = -N! \sum_{i=1}^{N-1} \iint \nabla_{\mathbf{v}_i} f_N \cdot \sum_{j\neq i}^{N} \mathbf{F}_{ij}(\mathbf{r}_i - \mathbf{r}_j)\, d\mathbf{r}_N\, d\mathbf{v}_N. \quad (2.24)$$

The integrals on the right hand side cannot be simplified without further knowledge of f_N.

Nevertheless, we can continue integrating over more particle positions and momenta to calculate further marginal distributions. For an arbitrary particle, s, the marginal probability density f_s can be found by

$$f_s(\mathbf{r}^s, \mathbf{v}^s, t) = \frac{N!}{(N-s)!} \int \dots \int f_N(\mathbf{r}^N, \mathbf{v}^N, t)\, d\mathbf{r}_N\, d\mathbf{v}_N \dots d\mathbf{r}_{s+1}\, d\mathbf{v}_{s+1}. \tag{2.25}$$

Again, differentiating with respect to time, substituting in Equation (2.7) and integrating through yields

$$\frac{\partial}{\partial t} f_s + \{H_s, f_s\} = -\frac{(N-s)!}{N!} \sum_{i=1}^{s} \iint \mathbf{F}_{i,s+1} \cdot \nabla_{\mathbf{v}_i} f_{s+1}\, d\mathbf{r}_{s+1}\, d\mathbf{v}_{s+1}, \tag{2.26}$$

where the evolution of f_s is dependent on f_{s+1}. The process can be repeated down to the evolution of the single particle probability density function, f_1, where

$$f_1(\mathbf{r}_1, \mathbf{v}_1, t) = \int \dots \int f_N(\mathbf{r}^N, \mathbf{v}^N, t)\, d\mathbf{r}_2\, d\mathbf{v}_2 \dots d\mathbf{r}_N\, d\mathbf{v}_N. \tag{2.27}$$

The evolution for Equation (2.27) is described by

$$\frac{\partial}{\partial t} f_1 + \{H_1, f_1\} = -\frac{1}{m} \int d\mathbf{r}_2 \int d\mathbf{v}_2 \mathbf{F}_{1,2} \cdot \nabla_{\mathbf{v}_1} f_2, \tag{2.28}$$

where expanding the Poisson Brackets yields,

$$\frac{\partial f_1}{\partial t} + \mathbf{v}_1 \cdot \nabla_{\mathbf{r}_1} f_1 + \frac{1}{m} \mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}_1} f_1 = -\frac{1}{m} \iint \mathbf{F}_{1,2} \nabla_{\mathbf{v}_1} f_2\, d\mathbf{v}_2\, d\mathbf{r}_2. \tag{2.29}$$

The final evolution equation, Equation (2.29), is known as BY-1 [24]. The process of integrating Liouville's Equation, Equation (2.7), over the positions and velocities of various particles is known as the Bogolyubov-Born-Green-Kirkwood-Yvon (BBGKY) Hierarchy [25, 26, 27, 28, 29]. Unfortunately, since the time evolution of any s-body distribution function is dependent on the $(s+1)$-body distribution function, each of the $(N-1)$ partial differential equations must be solved to inform the evolution of f_1. We should expect this, as the hierarchy is formally exact, and thus equivalent to solving the original Liouville equation. Therefore, any hope of analytically or computationally solving BY-1 will require some approximation to truncate this hierarchy.

2.1.1 Kinetic Equations

The primary method for truncating a hierarchy is through the use of a closure, in which the behavior of some f_{s+1} is approximated in terms of f_s. Given the explosion of dimensions at each stage of the BBGKY hierarchy, closures are typically used to express f_2 in terms of f_1. For example, the Vlasov Equation [30, 31] can be derived by the closure

$$f_2(\mathbf{r}_1, \mathbf{r}_2, \mathbf{v}_1, \mathbf{v}_2, t) = f_1(\mathbf{r}_1, \mathbf{v}_1, t) f_1(\mathbf{r}_2, \mathbf{v}_2, t), \tag{2.30}$$

where the one-body distribution functions are assumed to be independent. Using this closure, Equation (2.29) simplifies to

$$\frac{\partial f_1}{\partial t} + \mathbf{v}_1 \cdot \nabla_{\mathbf{r}_1} f_1 + \frac{1}{m}\mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}_1} f_1 = -\frac{1}{m} \iint \mathbf{F}_{1,2} \cdot \nabla_{\mathbf{v}_1} \left(f_1(\mathbf{r}_1, \mathbf{v}_1, t) f_1(\mathbf{r}_2, \mathbf{v}_2, t) \right) \, d\mathbf{r}_2 \, d\mathbf{v}_2. \tag{2.31}$$

The right hand side can then be integrated by parts to zero to obtain the more common form of the Vlasov equation

$$\frac{\partial f_1}{\partial t} + \mathbf{v}_1 \cdot \nabla_{\mathbf{r}_1} f_1 + \frac{1}{m}\mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}_1} f_1 = 0. \tag{2.32}$$

Another well-known closure results in the Boltzmann equation, which introduces the notion of collisions between particles. The Boltzmann equation can be derived by using conservation of momentum between velocity coordinates before and after collisions by

$$\mathbf{v}_1 + \mathbf{v}_2 = \mathbf{v}_1' + \mathbf{v}_2', \tag{2.33}$$

where $\mathbf{v}_1'$ and $\mathbf{v}_2'$ are the velocities of particles 1 and 2 after collision. After some significant calculations and the so-called *"Stosszahlansatz"* approximation (*i.e.*, molecular chaos), we get

$$\frac{\partial f_1}{\partial t} + \mathbf{v}_1 \cdot \nabla_{\mathbf{r}_1} f_1 + \frac{1}{m}\mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}_1} f_1 = J[f,f], \tag{2.34}$$

$$J[f,f] = \iint \sigma(\Omega)[f(\mathbf{r}_1, \mathbf{v}_1', t) f(\mathbf{r}_1, \mathbf{v}_2', t) - f(\mathbf{r}_1, \mathbf{v}_1, t) f(\mathbf{r}_1, \mathbf{v}_2, t)]\, d\Omega\, d\mathbf{v}_2. \tag{2.35}$$

Here, the collision operator $J[f,f]$ accounts for the collisions between particles, where Ω is the solid angle of scattering, and $\sigma(\Omega)$ is the differential cross section for the collision. For further details on the derivation and underlying assumptions of the Boltzmann equation, see [24].

Other well-known closures lead to the Fokker-Planck [32, 33], Lenard-Balescu [34, 35] and Bhatnager-Gross-Krook [36] equations.

2.2 Moment Expansion

Since we will only consider the one or two body probability density functions, we let

$$f_1 \to f, \qquad \mathbf{r}_1 \to \mathbf{r}, \qquad \mathbf{v}_1 \to \mathbf{v} \tag{2.36}$$

to simplify notation, so Equation (2.31) then becomes

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f + \frac{1}{m} \mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}} f = -\frac{1}{m} \iint \mathbf{F}_{1,2} \cdot \nabla_{\mathbf{v}} f_2 \, d\mathbf{v}_2 \, d\mathbf{r}_2. \tag{2.37}$$

Even with an accurate closure, Equation (2.37) is still a $6 + 1$ dimensional partial differential equation (3 space, 3 velocity and time). This high dimensionality results in numerical calculations that are computationally intractable for any realistic system of interest. Instead, we will assume our system is fairly collisional (which strongly-coupled plasmas are) in terms of the Knudsen number (Kn), the mean free path a particle travels before collision. We expect Kn to be small enough such that we expect statistical moments in the velocity distribution to converge rapidly. We can use our kinetic equation to systematically generate evolution equations for each velocity moment. In doing so, each moment will evolve according to the next higher moment, creating yet another hierarchy.

We can derive this hierarchy of moment equations by first defining the following quantities:

$$n(\mathbf{r},t) = \int f\,d\mathbf{v}, \tag{2.38}$$

$$n\mathbf{u}(\mathbf{r},t) = \int \mathbf{v}f\,d\mathbf{v}, \tag{2.39}$$

$$\mathbf{P}(\mathbf{r},t) = m\int (\mathbf{v}-\mathbf{u})\otimes(\mathbf{v}-\mathbf{u})f\,d\mathbf{v}, \tag{2.40}$$

$$\mathbf{Q}(\mathbf{r},t) = m\int (\mathbf{v}-\mathbf{u})\otimes(\mathbf{v}-\mathbf{u})\otimes(\mathbf{v}-\mathbf{u})f\,d\mathbf{v}. \tag{2.41}$$

Here, the terms are the zeroth-order, first-order, second-order and third-order centralized velocity moments of f, respectively, where we can interpret $n(\mathbf{r},t)$ as the number density of the fluid, $\mathbf{u}(\mathbf{r},t)$ as the average fluid velocity, $\mathbf{P}(\mathbf{r},t)$ as the stress tensor and $\mathbf{Q}(\mathbf{r},t)$ the stress tensor flux. For brevity, we let $\mathbf{ab} \equiv \mathbf{a}\otimes\mathbf{b}$ (*i.e.*, assume the default vector multiplication is an outer product unless otherwise specified) and introduce the reference velocity $\mathbf{w} = \mathbf{v} - \mathbf{u}$. Then, Equations (2.38)-(2.41) simplify to

$$n(\mathbf{r},t) = \int f\,d\mathbf{v}, \tag{2.42}$$

$$n\mathbf{u}(\mathbf{r},t) = \int \mathbf{v}f\,d\mathbf{v}, \tag{2.43}$$

$$\mathbf{P}(\mathbf{r},t) = m\int \mathbf{w}^2 f\,d\mathbf{v}, \tag{2.44}$$

$$\mathbf{Q}(\mathbf{r},t) = m\int \mathbf{w}^3 f\,d\mathbf{v}. \tag{2.45}$$

The notation can be simplified further by introducing the average over the velocity space as

$$\langle\,\circ\,\rangle \equiv \int (\,\circ\,)\,f(\mathbf{r},\mathbf{v},t)\,d\mathbf{v}, \tag{2.46}$$

which then results in

$$n(\mathbf{r},t) = \langle \mathbf{v}^0 \rangle, \quad n\mathbf{u}(\mathbf{r},t) = \langle \mathbf{v} \rangle, \quad \mathbf{P}(\mathbf{r},t) = \langle m\mathbf{w}^2 \rangle, \quad \mathbf{Q}(\mathbf{r},t) = \langle m\mathbf{w}^3 \rangle. \tag{2.47}$$

In the next section, evolution equations can be found for each of these quantities by integrating over the moments of Equation (2.29).

2.2.1 Zeroth-Order Moment

The evolution of the zeroth-order moment can be found by integrating Equation (2.29) with respect to $\mathbf{v}$ as follows

$$\int \left[\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f + \frac{1}{m} \mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}} f \right] d\mathbf{v} = -\frac{1}{m} \iiint \mathbf{F}_{1,2} \cdot \nabla_{\mathbf{v}} f_2 \, d\mathbf{v}_2 \, d\mathbf{r}_2 \, d\mathbf{v}. \tag{2.48}$$

The time derivative can be pulled out of the first term, and the spatial derivative out of second. The external forcing term, and the right hand side can be immediately evaluated to 0. So, Equation (2.48) becomes

$$\frac{\partial}{\partial t} \int f \, d\mathbf{v} + \int \nabla_{\mathbf{r}} \cdot (f\mathbf{v}) \, d\mathbf{v} = 0. \tag{2.49}$$

The spatial divergence can be pulled out of the integral and the macroscopic variables can be substituted in to get

$$\boxed{\frac{\partial}{\partial t} n + \nabla \cdot (n\mathbf{u}) = 0.} \tag{2.50}$$

The evolution of the zeroth-order velocity moment, Equation (2.50), is known as the continuity equation. It is a nonlinear transport equation that represents the conservation of mass (or particle number). However, finding a solution for the zeroth-order moment $n(\mathbf{r},t)$ requires knowledge of the first-order moment $\mathbf{u}(\mathbf{r},t)$.

2.2.2 First-Order Moment

We can multiply BY-1, Equation (2.31), by $\mathbf{v}$ and then integrate with respect to $\mathbf{v}$ to derive an evolution equation for $\mathbf{u}(\mathbf{r}, t)$ as follows

$$\int \left[\mathbf{v}\frac{\partial f}{\partial t} + \mathbf{v}\left(\mathbf{v} \cdot \nabla_{\mathbf{r}} f\right) + \mathbf{v}\left(\frac{1}{m}\mathbf{F}_{\text{ext}} \cdot \nabla_{\mathbf{v}} f\right)\right] d\mathbf{v} = -\frac{1}{m}\iiint \mathbf{v}\left(\mathbf{F}_{1,2} \cdot \nabla_{\mathbf{v}} f'\right) d\mathbf{v}_2\, d\mathbf{r}_2\, d\mathbf{v}. \tag{2.51}$$

The time derivative can be pulled out the first term and the spatial divergence in the second term can be rewritten as a spatial divergence of a rank 2 tensor. The external force and right hand side can both be integrated by parts. So, Equation (2.51) becomes

$$\frac{\partial}{\partial t}\int \left(\mathbf{v}f\right) d\mathbf{v} + \int \nabla_{\mathbf{r}} \cdot \left(\mathbf{v}^2 f\right) d\mathbf{v} - \frac{1}{m}\mathbf{F}_{\text{ext}}\int f\, d\mathbf{v} = \frac{1}{m}\iiint \mathbf{F}_{1,2} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2, \tag{2.52}$$

$$\frac{\partial}{\partial t}(n\mathbf{u}) + \nabla \cdot \int \mathbf{v}^2 f\, d\mathbf{v} - \frac{1}{m}n\mathbf{F}_{\text{ext}} = \frac{1}{m}\iiint \mathbf{F}_{1,2} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2. \tag{2.53}$$

where the expectation moments have been plugged in for the first and third terms, and the tensor divergence has been pulled out of the integral. The second term $\nabla \cdot \int \mathbf{v}^2 f\, d\mathbf{v}$ can be expanded into central moments in terms of $\mathbf{w}$ and simplified by

$$\nabla \cdot \int \mathbf{v}^2 f\, d\mathbf{v} = \nabla \cdot \int \left[\mathbf{w}^2 + \mathbf{w}\mathbf{u} + \mathbf{u}\mathbf{w} + \mathbf{u}^2\right] f\, d\mathbf{v} = \nabla \cdot \left[\mathbf{P} + n\mathbf{u}^2\right]. \tag{2.54}$$

A "two-body density", $n_2(\mathbf{r}, \mathbf{r}_2, t)$, can be used to simplify the right hand side, where

$$n_2(\mathbf{r}, \mathbf{r}_2, t) \equiv \iint f_2(\mathbf{r}, \mathbf{v}, \mathbf{r}_2, \mathbf{v}_2, t)\, d\mathbf{v}\, d\mathbf{v}_2. \tag{2.55}$$

Using the two-body density, Equation (2.52) can be reduced to

$$\frac{\partial}{\partial t}(n\mathbf{u}) + \nabla \cdot \left[\mathbf{P} + n\mathbf{u}^2\right] - \frac{1}{m}n\mathbf{F}_{\mathrm{ext}} = \frac{1}{m}\int \mathbf{F}_{1,2}n_2\, d\mathbf{r}_2. \tag{2.56}$$

The equation above is known as the momentum equation and evolves the first-order moment. We intend to simplify this equation by noting the following results:

$$\begin{aligned}\frac{\partial}{\partial t}(n\mathbf{u}) &= \frac{\partial n}{\partial t}\mathbf{u} + n\frac{\partial \mathbf{u}}{\partial t} && (2.57)\\ &= -\nabla \cdot (n\mathbf{u})\,\mathbf{u} + n\frac{\partial \mathbf{u}}{\partial t} && (2.58)\\ &= -\left(\mathbf{u}\cdot\nabla n\right)\mathbf{u} - n\left(\nabla\cdot\mathbf{u}\right)\mathbf{u} + n\frac{\partial \mathbf{u}}{\partial t}, && (2.59)\end{aligned}$$

where the time derivative is expanded and Equation (2.50) is substituted in. Additionally, the tensor dot product can be expanded to

$$\nabla \cdot \left(n\mathbf{u}^2\right) = \mathbf{u}^2 \cdot \nabla n + n\left(\nabla \cdot \mathbf{u}^2\right) = n(\nabla\cdot\mathbf{u})\mathbf{u} + \mathbf{u}\cdot(\nabla n)\mathbf{u} + n(\mathbf{u}\cdot\nabla)\mathbf{u}. \tag{2.60}$$

Combining these results, we see

$$\frac{\partial}{\partial t}(n\mathbf{u}) + \nabla \cdot \left(n\mathbf{u}^2\right) = n\frac{\partial \mathbf{u}}{\partial t} + n(\mathbf{u}\cdot\nabla)\mathbf{u}, \tag{2.61}$$

and Equation (2.56) can be simplified to

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{1}{n}\nabla\cdot\mathbf{P} - \frac{\mathbf{F}_{\mathrm{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2}n_2\, d\mathbf{r}_2. \tag{2.62}$$

The stress tensor can be decomposed into

$$\mathbf{P} = p\mathbf{I} + \mathbf{\Pi}, \tag{2.63}$$

where p is the pressure that accounts for the isotropic components, and $\mathbf{\Pi}$ accounts for the deviatoric (anisotropic) components. Then the momentum equation simplifies to

$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{n}\left(\nabla p + \nabla \cdot \mathbf{\Pi}\right) - \frac{\mathbf{F}_{\text{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2.} \tag{2.64}$$

Again, the equation depends on the next moment in velocity. We can see this leads to yet another hierarchy: a moment expansion of BY-1. Each additional moment will better approximate the true behavior of BY-1. Once again, closures can be used to terminate this hierarchy, where moments are approximated in terms of the lower-order ones.

2.2.3 Second-Order Moment

We will extend the moment expansion once more to find the evolution of $\mathbf{P} = \langle \mathbf{w}^2 f \rangle$ by

$$\begin{aligned}\int w_i w_j \frac{\partial f}{\partial t} \, d\mathbf{v} + \int w_i w_j \left(v_k \partial_{\mathbf{r}_k} f\right) \, d\mathbf{v} + \frac{1}{m} \int w_i w_j \left(F_{\text{ext}k} \partial_{\mathbf{v}_k} f\right) \, d\mathbf{v} \\ = -\frac{1}{m} \iiint w_i w_j \left(F_{1,2_k} \partial_{\mathbf{v}_k} f_2\right) \, d\mathbf{v} \, d\mathbf{v}_2 \, d\mathbf{r}_2,\end{aligned} \tag{2.65}$$

where Einstein notation has been used for both brevity and clarity. Here, $\partial_{\mathbf{r}_k}$ denotes the partial derivative with respect to $\mathbf{r}$, and similarly $\partial_{\mathbf{v}_k}$ with respect to $\mathbf{v}$, over the kth index. From left to right, each term can be evaluated. For the first term, the time derivative can be expanded using the product rule,

$$\int w_i w_j \frac{\partial f}{\partial t}\, d\mathbf{v} = \frac{\partial}{\partial t}\int w_i w_j f\, d\mathbf{v} - \int \left[w_i \frac{\partial w_j}{\partial t} + \frac{\partial w_i}{\partial t} w_j \right] f\, d\mathbf{v} \tag{2.66}$$

$$= \frac{\partial}{\partial t} P_{ij} + \left[\int w_i f\, d\mathbf{v} \right] \frac{\partial u_j}{\partial t} + \frac{\partial u_i}{\partial t} \left[\int w_j f\, d\mathbf{v} \right] \tag{2.67}$$

$$= \frac{\partial}{\partial t} P_{ij}. \tag{2.68}$$

The second term can be rewritten in terms of **w** and **u**

$$\int w_i w_j \left(v_k \partial_{\mathbf{r}_k} f \right)\, d\mathbf{v} = \int w_i w_j \partial_{\mathbf{r}_k} \left(v_k f \right)\, d\mathbf{v} \tag{2.69}$$

$$= \int w_i w_j \partial_{\mathbf{r}_k} \left(w_k f \right)\, d\mathbf{v} + \int w_i w_j \partial_{\mathbf{r}_k} \left(u_k f \right)\, d\mathbf{v}. \tag{2.70}$$

Now the product rule can be used to expand and simplify the first term to

$$\int w_i w_j \partial_{\mathbf{r}_k} \left(w_k f \right)\, d\mathbf{v}$$

$$= \partial_k \int w_i w_j w_k f\, d\mathbf{v} - \int \partial_{\mathbf{r}_k} \left(w_k w_i \right) w_j f\, d\mathbf{v} - \int \partial_{\mathbf{r}_k} \left(w_i w_k \right) w_j f\, d\mathbf{v} \tag{2.71}$$

$$= \partial_k \int w_i w_j w_k f\, d\mathbf{v} + \partial_k \int u_i w_j w_k f\, d\mathbf{v} + \partial_k \int w_i u_j w_k f\, d\mathbf{v} \tag{2.72}$$

$$= \partial_k Q_{ijk} + \partial_k P_{ik} u_j + \partial_k P_{kj} u_i. \tag{2.73}$$

We continue with the second term:

$$\int w_i w_j \partial_{\mathbf{r}_k} \left(u_k f \right)\, d\mathbf{v}$$

$$= \partial_k \int w_i w_j u_k f\, d\mathbf{v} - \int \partial_{\mathbf{r}_k} \left(w_i w_k \right) u_j f\, d\mathbf{v} - \int \partial_{\mathbf{r}_k} \left(w_k w_j \right) u_i f\, d\mathbf{v} \tag{2.74}$$

$$= \partial_k P_{ij} u_k + \partial_k \int u_i w_j u_k f\, d\mathbf{v} + \partial_k \int u_i w_j u_k f\, d\mathbf{v} \tag{2.75}$$

$$= \partial_k P_{ij} u_k. \tag{2.76}$$

Combining the results of Equation (2.71) and Equation (2.74), we have

$$\int w_i w_j \left(v_k \partial_{\mathbf{r}_k} f\right)\, d\mathbf{v} = \partial_k Q_{ijk} + \partial_k P_{jk} u_i + \partial_k P_{ik} u_j + \partial_k P_{ij} u_k. \tag{2.77}$$

The external force term (term 3), can be simplified by first integrating by parts

$$\begin{aligned}
\int w_i w_j \left(\frac{1}{m} F_{\text{ext}k} \partial_{\mathbf{v}_k} f\right) d\mathbf{v} &= \frac{1}{m}\int w_i w_j \partial_{\mathbf{v}_k}\left(F_{\text{ext}_k} f\right)\, d\mathbf{v} && (2.78)\\
&= -\frac{1}{m}\left[\int \left(\partial_{\mathbf{v}_k} w_k w_i + w_i \partial_{\mathbf{v}_k} w_k\right) f\, d\mathbf{v}\right] F_{\text{ext}_j} && (2.79)\\
&= \frac{2}{m}\left[\int w_i f\, d\mathbf{v}\right] F_{\text{ext}_j} && (2.80)\\
&= 0. && (2.81)
\end{aligned}$$

The right hand side of Equation (2.65) can also be integrated by parts to yield

$$\begin{aligned}
-\frac{1}{m}&\iiint w_i w_j F_{1,2_k} \partial_{\mathbf{v}_k} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2\\
&= \frac{1}{m}\iiint \left(\partial_{\mathbf{v}_k} w_k w_i + w_i \partial_{\mathbf{v}_k} w_k\right) F_{1,2_j} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2 && (2.82)\\
&= \frac{2}{m}\iiint w_i F_{1,2_j} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2. && (2.83)
\end{aligned}$$

Without further knowledge of $f_2(\mathbf{r}, \mathbf{v}, \mathbf{r}_2, \mathbf{v}_2, t)$, the right hand side cannot be simplified further. Combining Equation (2.66), Equation (2.77), Equation (2.78) and Equation (2.82), we get

$$\boxed{\frac{\partial}{\partial t} P_{ij} + \partial_k Q_{ijk} + \partial_k P_{jk} u_i + \partial_k P_{ik} u_j + \partial_k P_{ij} u_k = \frac{1}{m}\iiint \left(w_j + w_i\right) F_{1,2_k} f_2\, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2.} \tag{2.84}$$

Thus, we have the following system of equations for evolving the moments of

BY-1:

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{u}) = 0 \tag{2.85}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} + \frac{1}{n}\nabla \cdot \mathbf{P} - \frac{\mathbf{F}_{\text{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2 \tag{2.86}$$

$$\frac{\partial}{\partial t}P_{ij} + \partial_k Q_{ijk} + \partial_k P_{jk} u_i + \partial_k P_{ik} u_j + \partial_k P_{ij} u_k = \frac{1}{m}\iiint (w_j + w_i)\, F_{1,2_k} f_2 \, d\mathbf{v}\, d\mathbf{v}_2\, d\mathbf{r}_2 \tag{2.87}$$

More moments can be taken, and more evolution equations can be found; however, we choose to stop here, as these first three moments are the most physically relevant, because they pertain to conservation laws.

2.3 Moment Expansion Closure

The evolution equation for $\mathbf{P}$ remains unclosed without knowledge of $\mathbf{Q}$, and both evolution equations for $\mathbf{u}$ and $\mathbf{P}$ require knowledge f_2. This section describes the various model closures including the STLS Approximation, utilization of Dynamic Density Functional Theory and the Conservation of Energy.

2.3.1 STLS-Approximation

The Singwi-Tosi-Land-Sjolander (STLS) approximation is a widely used theoretical method in statistical mechanics that provides a description of the structural and thermodynamic properties of liquids and dense fluids [37]. The approximation is based on a mean-field approach that treats the effects of particle correlations beyond the simple pair correlations, which are captured by the Radial Distribution Function (RDF), $g(r)$. The RDF is related to the probability density of finding a pair of particles at positions $\mathbf{r}$ and $\mathbf{r}_2$ simultaneously. More specifically, $g(r)$ measures the number of particles that are statistically observed between r and $r+dr$ from a given reference particle relative to the mean density (*i.e.*, $g(r) \to 1$ for large r), where $r = |\mathbf{r} - \mathbf{r}_2|$. The STLS approximation extends the pair correlation function to include three-body and higher-order correlations, allowing for a more accurate description of the interactions among particles in a dense system. The STLS approximation assumes the two-particle distribution function can be written as the product of the individual one body distribution functions times the RDF.

$$f_2(\mathbf{r}, \mathbf{r}_2, \mathbf{v}, \mathbf{v}_2, t) = f_1(\mathbf{r}, \mathbf{v}, t) f_1(\mathbf{r}_2, \mathbf{v}_2, t) g(r) \tag{2.88}$$

The approximation decouples the position and velocity of each individual particle and let's the radial distribution function determine the degree of correlation between them. When plugging the STLS approximation into the right hand side temperature equation, we get

$$\frac{2}{m} \iiint w_i F_{1,2_j} f_2 \, d\mathbf{v} \, d\mathbf{v}_2 \, d\mathbf{r}_2 = \frac{2}{m} \left(nu_i - nu_i \right) \iiint F_{1,2_j} f(\mathbf{r}_2, \mathbf{v}_2, t) g(r) \, d\mathbf{v}_2 \, d\mathbf{r}_2 = 0. \tag{2.89}$$

In particular, the decoupling of the two-body velocity field simplifies the right side of Equation (2.84) to 0. Then $\mathbf{P}$ is governed by the following equation

$$\boxed{\frac{\partial}{\partial t} P_{ij} + \partial_k Q_{ijk} + P_{ij} \partial_k u_k + P_{ik} \partial_k u_j + P_{kj} \partial_k u_i = 0.} \tag{2.90}$$

2.3.2 Density Functional Theory

In order to develop the remaining closures, we must introduce Density Functional Theory, an equilibrium framework that will be used to determine the collisional properties of the particles on a microscopic level [9]. This will be done through a free energy approximation. We start with the Helmholtz free energy, denoted as $\mathscr{F}[n(\mathbf{r})]$, which is a thermodynamic potential describing the state of a system at constant temperature, volume, and density. It contains information about correlations between particles, allowing for the approximation of interactions near equilibrium that are typically unaccounted for in the ideal or mean field limit.

Before being able to connect the free energy to a moment expansion, we must investigate the behavior of a plasma in equilibrium. The variational minimization of the free energy determines the equilibrium density of the system. However, for a system with a fixed number of particles, we must introduce the Grand Potential Ω, which is the free energy minus a constraint with respect to the total number of particles in the system:

$$\Omega[n] = \mathscr{F}[n] - \mu \int n \, d\mathbf{r}. \tag{2.91}$$

Here, μ acts as a Lagrange multiplier for the particle constraint. The minimization of the grand potential can be found by setting the variational derivative to 0 to obtain

$$\frac{\delta \Omega}{\delta n} = 0 \implies \mu = \frac{\delta \mathscr{F}}{\delta n}. \tag{2.92}$$

The quantity μ contains all microscopic interaction information of the plasma in equilibrium and is named the chemical potential. Note that since μ is constant in equilibrium

$$\nabla \mu = \mathbf{0} \implies \nabla \frac{\delta \mathscr{F}}{\delta n} = \mathbf{0}. \tag{2.93}$$

The free energy can be decomposed into ideal, external and excess components [14],

$$\mathscr{F}[n(\mathbf{r})] = \mathscr{F}_{\text{id}} + \mathscr{F}_{\text{ext}} + \mathscr{F}_{\text{ex}}. \tag{2.94}$$

The ideal term tracks the noninteracting behavior and can be written as:

$$\mathscr{F}_{\text{id}}[n(\mathbf{r})] = k_B T \int \Big(n(\mathbf{r}) \log(\Lambda^3 n(\mathbf{r})) - n(\mathbf{r}) \Big) \, d\mathbf{r}. \tag{2.95}$$

where $\Lambda = \frac{\hbar}{\sqrt{m k_B T}}$ is the thermal de Broglie wavelength, $\hbar$ is the reduced Planck constant, k_B is the Boltzmann constant and T is the temperature. The external term accounts for interactions with all external potential fields and can be written as:

$$\mathscr{F}_{\text{ext}}[n(\mathbf{r})] = \int n(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) \, d\mathbf{r}. \tag{2.96}$$

Together, the free energy can be defined as

$$\mathscr{F}[n(\mathbf{r})] = \underbrace{k_B T \int \Big(n \log(\Lambda^3 n) - n \Big) \, d\mathbf{r}}_{\text{Ideal}} + \underbrace{\int n V_{\text{ext}} \, d\mathbf{r}}_{\text{External}} + \mathscr{F}_{\text{ex}}. \tag{2.97}$$

The excess term, denoted as $\mathscr{F}_{\text{ex}}$, represents collision and correlation effects that are not accounted for by the ideal and external terms. A correlation expansion can be used to decompose the excess free energy into two-body, three-body, etc. interactions. This expansion is written as

$$\begin{aligned}
\mathscr{F}_{\text{ex}}[n(\mathbf{r})] = &- \frac{1}{2!} k_B T \iint \Delta n(\mathbf{r}) \Delta n(\mathbf{r}') c^{(2)}(\mathbf{r}, \mathbf{r}') \, d\mathbf{r} \, d\mathbf{r}' \\
&- \frac{1}{3!} k_B T \iiint \Delta n(\mathbf{r}) \Delta n(\mathbf{r}') \Delta n(\mathbf{r}'') c^{(3)}(\mathbf{r}, \mathbf{r}', \mathbf{r}'') \, d\mathbf{r} \, d\mathbf{r}' \, d\mathbf{r}'' \\
&- \frac{1}{4!} k_B T \iiiint \Delta n(\mathbf{r}) \Delta n(\mathbf{r}') \Delta n(\mathbf{r}'') \Delta n(\mathbf{r}''') c^{(4)}(\mathbf{r}, \mathbf{r}', \mathbf{r}'', \mathbf{r}''') \, d\mathbf{r} \, d\mathbf{r}' \, d\mathbf{r}'' \, d\mathbf{r}''' \\
&- \cdots
\end{aligned} \tag{2.98}$$

where $c^{(i)}(\mathbf{r}, \mathbf{r}', \ldots, \mathbf{r}^{(i)})$ is the i^{th}-body direct correlation function (DCF), and

$$\Delta n(\mathbf{r}) = n(\mathbf{r}) - n_0 \tag{2.99}$$

is the deviation of the density the mean density of the system n_0. However, including three or more body correlations in the expansion becomes computationally expensive due to the large computational complexity. Therefore, we will truncate the expansion to only include two-body correlations [38] by

$$\mathscr{F}_{\text{ex}}[n(\mathbf{r})] \approx -\frac{1}{2}k_B T \iint \Delta n(\mathbf{r})\Delta n(\mathbf{r}')c^{(2)}(\mathbf{r}, \mathbf{r}')\, d\mathbf{r}\, d\mathbf{r}', \tag{2.100}$$

which is known as either the Ramakrishnan-Yussouff or hypernetted-chain approximation.

The final form of the Helmholtz free energy is then given by

$$\begin{aligned}\mathscr{F}[n(\mathbf{r})] =& k_B T \int \Big(n(\mathbf{r})\log(\Lambda^3 n(\mathbf{r})) - n(\mathbf{r})\Big)\, d\mathbf{r} + \int n(\mathbf{r})V_{\text{ext}}(\mathbf{r})\, d\mathbf{r} \\ &- \frac{1}{2}k_B T \iint \Delta n(\mathbf{r})\Delta n(\mathbf{r}')c^{(2)}(\mathbf{r}, \mathbf{r}')\, d\mathbf{r}\, d\mathbf{r}'\end{aligned} \tag{2.101}$$

Thus, in equilibrium, we have:

$$\boxed{\nabla\frac{\delta\mathscr{F}}{\delta n} = k_B T\nabla\log(n) - \mathbf{F}_{\text{ext}} - k_B T\nabla C(\mathbf{r}) = 0,} \tag{2.102}$$

where

$$C(\mathbf{r}) = \int \Delta n(\mathbf{r}')c^{(2)}(\mathbf{r}, \mathbf{r}')\, d\mathbf{r}'. \tag{2.103}$$

The equilibrium description through the free energy contains information about the correlations physics of the plasma ions at a microscopic level.

2.4 Temperature Closure

We would like to connect the evolution equations with equilibrium fluid behavior described by DFT. However, the temperature, T, is not defined in the context of the moment expansion. Thus, we introduce the Maxwellian distribution by noting that the fluid velocity distribution can take on a drifting Maxwellian when in equilibrium as follows:

$$f_M \sim \left(\frac{m}{2\pi k_b T}\right)^{\frac{3}{2}} n(\mathbf{r}, t) e^{-\frac{mw^2}{2k_B T}}, \tag{2.104}$$

where $w^2 = |\mathbf{w}|^2 = (\mathbf{v} - \mathbf{u}) \cdot (\mathbf{v} - \mathbf{u})$ is the inner product of $\mathbf{w}$ with itself. Note the temperature can be defined in the context of the Maxwellian and used to combine the moment expansion with the free energy. Since we are working between the Grand Canonical and Canonical Ensembles, in which temperature is assumed to be locally constant. Then one can find the temperature in terms of the velocity moments of the drifting Maxwellian. Note the following well-known results for the moments of the Gaussian can be used to simplify computation:

$$\int e^{-ax^2}\, dx = \sqrt{\frac{\pi}{a}}, \quad \int x e^{-ax^2}\, dx = 0, \quad \int x^2 e^{-ax^2}\, dx = \sqrt{\frac{\pi}{2a^3}}. \tag{2.105}$$

The zeroth-order moment is found by integrating over Equation (2.104) as follows

$$\langle \mathbf{v}^0 \rangle = \int \left(\frac{m}{2\pi k_b T}\right)^{\frac{3}{2}} n e^{-\frac{mw^2}{2k_B T}} d\mathbf{v} = \left(\frac{m}{2\pi k_b T}\right)^{\frac{3}{2}} n(\mathbf{r}) \int e^{-\frac{mw^2}{2k_B T}} d\mathbf{w} = n(\mathbf{r}, t) \tag{2.106}$$

and is equal to the density. The first-order moment can be found by

$$\begin{aligned}\langle \mathbf{v}^1 \rangle &= \int \mathbf{v} \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n(\mathbf{r}) e^{-\frac{mw^2}{2k_B T}} d\mathbf{v} &\quad (2.107)\\ &= \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n(\mathbf{r}, t) \left[\int \mathbf{w} e^{-\frac{mw^2}{2k_B T}} d\mathbf{w} + \int \mathbf{u} e^{-\frac{mw^2}{2k_B T}} d\mathbf{v} \right] &\quad (2.108)\\ &= n(\mathbf{r}, t)\mathbf{u}(\mathbf{r}, t). &\quad (2.109)\end{aligned}$$

The centralized second-order moment can be found by

$$\begin{aligned}\langle \mathbf{w}^2 \rangle &= \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n \int e^{-\frac{mw^2}{2k_B T}} \mathbf{w}^2 \, d\mathbf{v} &\quad (2.110)\\ &= \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n \int e^{-\frac{mw^2}{2k_B T}} \mathbf{w}^2 \, d\mathbf{w} &\quad (2.111)\\ &= \delta_{ij} \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n \int e^{-\frac{mw^2}{2k_B T}} w_i w_j \, d\mathbf{w}, &\quad (2.112)\end{aligned}$$

where the off-diagonal integrals are zero by oddness. In fact, the off-diagonals of $\langle \mathbf{w}^2 \rangle$ will be zero for any isotropic system. Let the component-wise representation be $\mathbf{w} = [w_1, w_2, w_3]^T$. Then Equation (2.110) can be split component-wise and evaluated using Equation (2.105) as

$$\begin{aligned}\langle \mathbf{w}^2 \rangle &= \delta_{ij} \left(\frac{m}{2\pi k_b T} \right)^{\frac{3}{2}} n \int w_1^2 e^{-\frac{mw_1^2}{2k_B T}} \, dw_1 \left(\int e^{-\frac{mw_2^2}{2k_B T}} \, dw_2 \right) \left(\int e^{-\frac{mw_3^2}{2k_B T}} \, dw_3 \right) &\quad (2.113)\\ &= \delta_{ij} \left(\frac{m}{2\pi k_B T} \right)^{\frac{3}{2}} n \left(\frac{k_B T}{m} \sqrt{\frac{2\pi k_B T}{m}} \right) \left(\sqrt{\frac{2\pi k_B T}{m}} \right) \left(\sqrt{\frac{2\pi k_B T}{m}} \right) &\quad (2.114)\\ &= \delta_{ij} \frac{k_B T}{m} n &\quad (2.115)\end{aligned}$$

Then the temperature can be defined in terms of the Maxwellian whereby the following relationship can be drawn:

$$n k_B T(\mathbf{r}, t) = \frac{m}{3} \int w^2 f_M \, d\mathbf{v} = \frac{m}{3} \mathrm{tr} \left\{ \int \mathbf{w}^2 f_M \, d\mathbf{v} \right\} = \frac{m}{3} \mathrm{tr} \left\{ \mathbf{P} \big|_{f = f_M} \right\}, \quad (2.116)$$

where the temperature has been promoted into a spatial and time dependent quantity. The temperature evolution can be found by taking the trace of the stress tensor evolution, Equation (2.90), as follows

$$\frac{\partial}{\partial t} P_{ii} + \partial_k Q_{iik} + P_{ii}\partial_k u_k + P_{ik}\partial_k u_i + P_{ki}\partial_k u_i = 0 \tag{2.117}$$

where a tensor contraction of indices i and j has been used to take the trace in Einstein Notation. Then substituting in the temperature to Equation (2.117) yields

$$\frac{\partial}{\partial t}\left(\frac{3}{m} n k_B T\right) + \frac{1}{m}\partial_k \left(nq\right)_k + \left(\frac{3}{m} n k_B T\right)\partial_k u_k + 2\partial_k \left(\frac{3}{m} n k_B T\right) u_k = 0 \tag{2.118}$$

where $n\mathbf{q} = \langle m w^2 \mathbf{w} f\rangle$ is the heat flow. Converting back to vector notation, we get

$$\frac{\partial}{\partial t}\left(n k_B T\right) + \nabla \cdot \left(n\mathbf{q}\right) + n k_B T \nabla \cdot \mathbf{u} + 2\nabla\left(n k_B T\right)\cdot \mathbf{u} = 0 \tag{2.119}$$

To simplify, we note that the continuity equation, Equation (2.50), can be substituted in the first term to get

$$\begin{aligned} \frac{\partial}{\partial t}\left(n k_B T\right) &= n\frac{\partial}{\partial t}\left(k_B T\right) - k_B T \nabla \cdot \left(n\mathbf{u}\right) && (2.120)\\ &= n\frac{\partial}{\partial t}\left(k_B T\right) - k_B T \nabla n \cdot \mathbf{u} - n k_B T \nabla \cdot \mathbf{u}, && (2.121) \end{aligned}$$

while the last term can be expanded with the product rule,

$$2\nabla\left(n k_B T\right)\cdot \mathbf{u} = 2 k_B T \nabla n \cdot \mathbf{u} + 2n\nabla\left(k_B T\right)\cdot \mathbf{u}. \tag{2.122}$$

Combining these terms, Equation (2.119) simplifies to

$$n\frac{\partial}{\partial t}\left(k_B T\right) + \nabla \cdot \left(n\mathbf{q}\right) + k_B T \nabla n \cdot \mathbf{u} + 2n\nabla\left(k_B T\right)\cdot \mathbf{u} = 0. \tag{2.123}$$

Dividing by n, we get the a simplified temperature evolution equation:

$$\frac{\partial}{\partial t}\left(k_B T\right) + \frac{1}{n}\nabla \cdot (n\mathbf{q}) + k_B T \nabla \log(n) \cdot \mathbf{u} + 2\nabla \left(k_B T\right) \cdot \mathbf{u} = 0. \tag{2.124}$$

The temperature evolution is still not closed since $\mathbf{q}$ is unknown. However, Fourier's Law can be used as a physically motivated closure. Fourier's Law dictates that the heat flow is proportional to the negative gradient of the temperature,

$$\mathbf{q} \approx -\lambda \nabla \left(k_B T\right) \tag{2.125}$$

where λ is the thermal conductivity. By noting,

$$\frac{1}{n}\nabla \cdot (\lambda n \nabla T) = \nabla \cdot (\lambda \nabla T) + \lambda \nabla \log(n) \cdot \nabla T, \tag{2.126}$$

the temperature evolution can finally be simplified to

$$\boxed{\frac{\partial T}{\partial t} + \big(T\mathbf{u} - \lambda \nabla T\big) \cdot \nabla \log(n) + 2\nabla T \cdot \mathbf{u} = \nabla \cdot (\lambda \nabla T).} \tag{2.127}$$

Now that the temperature is defined with respect to the one-body particle distribution, and its time evolution is known, the free energy equilibrium description can be connected to the moment expansion.

2.4.1 Matching Equilibrium Behavior

This section will focus on applying closures using the theory in the previous subsections, namely temperature and DDFT. Assuming local equilibrium, f takes on a drifting Maxwellian. Revisiting the stress tensor decomposition, Equation (2.63), the temperature moment can be substituted in using Equation (2.116) to get

$$\mathbf{P} = p\mathbf{I} + \mathbf{\Pi} \approx n k_B T \mathbf{I}, \tag{2.128}$$

where $p \approx nk_BT$ near equilibrium and $\mathbf{\Pi} = \mathbf{0}$ due to isotropy. Then the momentum equation, Equation (2.62), can be simplified to

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{n}\nabla(nk_BT) - \frac{\mathbf{F}_{\text{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2. \tag{2.129}$$

Given a constant temperature, we have

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + k_BT\nabla \log(n) - \frac{\mathbf{F}_{\text{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2. \tag{2.130}$$

We can now work on closing the right hand side. In this regime, the velocity field is constant, so the first two terms of Equation (2.130) are zero:

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{f=f_M} = \mathbf{0}, \qquad \left.(\mathbf{u} \cdot \nabla)\mathbf{u}\right|_{f=f_M} = \mathbf{0}. \tag{2.131}$$

Then the momentum equation, Equation (2.130), can be simplified to

$$k_BT\nabla \log(n) - \frac{\mathbf{F}_{\text{ext}}}{m} = \frac{1}{mn}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2. \tag{2.132}$$

Since $\nabla \frac{\delta \mathscr{F}}{\delta n} = 0$ in equilibrium, we can match terms between Equation (2.102) and Equation (2.116)

$$k_BT\nabla C(\mathbf{r}) = \frac{1}{n}\int \mathbf{F}_{1,2} n_2 \, d\mathbf{r}_2, \tag{2.133}$$

by recognizing the left hand side of Equation (2.132) to be the ideal and external contributions. Now, finally, the free energy can be connected to the momentum equation. Substituting Equation (2.133) in to close the right hand side of Equation (2.130), we get

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{k_BT}{m}\nabla \log(n) - \frac{1}{m}\mathbf{F}_{\text{ext}} = \frac{1}{m}k_BT\nabla C(\mathbf{r}) \tag{2.134}$$

Then the free energy contributions can be collected together:

$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{m}\nabla\frac{\delta \mathscr{F}}{\delta n}.} \tag{2.135}$$

Now the momentum equation is closed, up to the approximation of the Mean Field potential and the two-body direct correlation function.

Compiling the newly closed system from Equation (2.50), Equation (2.127) and Equation (2.135), we get:

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{u}) = 0, \tag{2.136}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{m}\nabla\frac{\delta \mathscr{F}}{\delta n}, \tag{2.137}$$

$$\frac{\partial T}{\partial t} + \big(T\mathbf{u} - \lambda\nabla T\big) \cdot \nabla \log(n) + 2\nabla T \cdot \mathbf{u} = \nabla \cdot (\lambda \nabla T). \tag{2.138}$$

It should be noted that this system of equations is *novel* to the physics community.

CHAPTER 3

MODELING DIRECT CORRELATION FUNCTIONS

Modeling direct correlation functions (DCF) is a challenging task that requires the development of sophisticated methods to capture the complex behavior of correlated plasmas. In this chapter, we will explore several approaches commonly used for modeling DCF, including linear response theory, gradient expansions, local density approximations, weighted density approximations, mean field theory, and Mayer f-function method. We will delve into these different methods for modeling DCF, discussing their underlying principles, assumptions, and limitations. We will explore their applications, strengths, and weaknesses in different plasma systems, and discuss their reliability and accuracy in capturing the correlations and interactions between particles in a plasma.

Linear response theory allows us to study the response of a system to small perturbations. Gradient expansions, on the other hand, involve expanding the direct correlation function in a power series of gradients of the plasma properties, which provides a way to account for spatial variations and gradients in the system.

Local density approximations (LDAs) approximate the direct correlation function at a given point in space by expressing it in terms of the local density of the plasma at that point. Weighted density approximations (WDAs) are an extension of LDAs that take into account the contributions of higher-order density moments, allowing for improved accuracy in modeling the direct correlation function.

Mean field theory approximates the direct correlation function by neglecting higher-order correlations and assuming that the particles in the system interact only through an average or mean field, which simplifies the calculation of the direct correlation function. Finally, the Mayer f-function method approximates the pair term of the direct correlation function using a function called the f-function, which is related to the pair distribution function.

3.0.1 Linear Response

The linear response method is a perturbation-based approach based on the idea that the response of a system to a small perturbation is linearly related to the perturbation itself [9].

$$\mathscr{F}_{\text{ex}}[n] \approx \mathscr{F}_{\text{ex}}[n_0] + \mu_{\text{ex}} \int \Delta n(\mathbf{r})\, d\mathbf{r} \tag{3.1}$$

where μ_{ex} is the excess chemical potential. The linear response method typically involves solving a linear equation or set of linear equations to obtain the response of the system to the perturbation.

3.0.2 Gradient Expansion

The gradient expansion method is a mathematical technique used in the context of modeling plasma correlations through the direct correlation function. It is a perturbation-based approach that involves expanding the direct correlation function in powers of the gradient of the particle distribution function or other relevant variables [39].

The gradient expansion method is often used in plasma physics to approximate the direct correlation function in cases where the system is not amenable to exact calculations or when simplifications are needed to obtain tractable analytical or numerical solutions. The method is based on the idea that the direct correlation function can be expressed as a series expansion in powers of the gradients of relevant variables, such as the density or velocity gradients of the plasma.

The gradient expansion method allows for the systematic inclusion of higher-order gradient terms in the approximation of the direct correlation function, which can capture the effects of spatial variations and gradients in the plasma. This can be particularly useful in cases where the plasma has spatially varying properties or exhibits complex behavior, such as in non-uniform plasmas or plasmas with strong gradients.

The gradient expansion method has been widely used in the study of plasma correlations, collective behavior, and transport properties. It provides a flexible and versatile approach to approximate the direct correlation function in plasma systems, and it has found applications in various areas of plasma physics research, including plasma thermodynamics, kinetic theory, and transport theory. However, it is important to note that the accuracy and applicability of the gradient expansion method depend on the specific system under consideration and the order of the expansion used, and careful validation and assessment of the results are necessary to ensure the reliability of the obtained approximations.

3.0.3 Local Density Approximation

The LDA is based on the idea that the direct correlation function of a plasma system can be approximated by the known direct correlation function of a uniform plasma with the same local density as the system. Mathematically, the LDA can be expressed as:

$$\mathscr{F}_{\text{ex}}[n] \approx \mathscr{F}_{\text{ex}}[n_0] \tag{3.2}$$

This approximation is particularly useful in cases where the plasma exhibits low local density variations, such as in homogeneous plasmas or plasmas with weak density gradients. It is exact in the uniform limit ($\Delta n(r) \to 0$, $\mathscr{F}_{\text{ex}} \to \text{const.}$), but loses accuracy for systems large with large gradients. The LDA method provides a simple and computationally efficient approximation for the DCF in near-homogeneous plasma systems. LDA also has the advantage that highly-accurate models are available for uniform systems, so in this limit, the results are effectively “exact”.

3.0.4 Weighted Density Approximation

The Weighted Density Approximation (WDA) is an extension of the LDA method to non-uniform systems. It involves expressing the direct correlation function as a weighted sum of the DCF of reference uniform plasmas with different densities, where the weights are determined by the local density of the system under consideration.

The WDA can be expressed as:

$$\overline{n}(\mathbf{r}) \approx \int n(\mathbf{r}+\mathbf{r}')W(\mathbf{r}',\overline{n})\,d\mathbf{r}', \tag{3.3}$$

$$\mathscr{F}_{\mathrm{ex}}[n] \approx \int n(\mathbf{r})\Psi(\overline{n}(\mathbf{r})), \tag{3.4}$$

where $\overline{n}(\mathbf{r})$ is the weighted density based off some weight function, W, and the free energy is weighted based off some function, $\Psi(\overline{n}(\mathbf{r}))$.

The WDA allows for the incorporation of the effects of local density variations in the plasma correlations, as the weighted sum takes into account the contributions from different reference uniform plasmas with different densities. This makes the WDA particularly useful in cases where the plasma exhibits significant density variations, such as in inhomogeneous plasmas or plasmas with strong density gradients. The WDA provides a more robust approach than LDA. However, it is important to note that its accuracy and applicability depend on the specific system under consideration and the choice of weight function.

3.0.5 Mean Field Approximation

The Mean Field (MF) approximation is a widely used approach that approximates the collective interactions among particles in a plasma by assuming that each particle interacts with an average or "mean" field generated by the other particles, neglecting the detailed microscopic correlations. Within the MF limit, the direct correlation function $c^{(2)}(r)$ can be approximated as:

$$c^{(2)}(r) \approx -\beta\varphi(r), \tag{3.5}$$

where $\varphi(r)$ is the effective potential between particles, $\beta = \frac{1}{k_B T}$, and $r = |\mathbf{r} - \mathbf{r}'|$. As we will show below, this approximation results in a self-consistent mean field in the governing equations. The mean field approximation is exact in the large-r limit.

The MF approximation is a powerful and widely used method due to its simplicity and computational efficiency, making it well-suited for studying large-scale systems and long-time dynamics. However, it is important to note that the MF approximation neglects the effects of microscopic correlations and fluctuations, and does not accurately capture the behavior of strongly correlated plasma systems. Furthermore, the MF approximation often leads to spurious results (*e.g.*, negative densities and unbounded DCF) when exploring the physics at small r. We discuss using the MF approoximation in more detail in Section (3.1).

3.0.6 Cluster Expansion

The cluster expansion can be viewed as an extension of the MF approximation and is closely related to the virial expansion of thermodynamics. The expansion is based on the idea of representing the energy as an expansion of increasingly complex groupings (or clusters) of interacting particles, where the expansion "parameter" is the so-called Mayer f-function that characterizes 2-body interactions. Mathematically, the direct correlation function $c^{(2)}(r)$ can be expressed in terms of this expansion, where the leading order term will be the f-function itself

$$c^{(2)}(r) \approx f(r) \equiv e^{-\beta\varphi(r)} - 1. \tag{3.6}$$

Note that as r becomes large, the inter-particle potential is expected to go to zero, which then recovers the mean field limit, as $f \approx -\beta\varphi$. Unlike in the MF approximation, the DCF is bounded at small r when using the Mayer f-function. The cluster expansion allows for a systematic and flexible approach to modeling higher-order correlations in a system. Additional terms can be included in the expansion, but they involve increasingly higher-dimensional integrals that quickly become computationally intractable. This method has been used in various applications, including the study of charged particle systems, systems with short-range Van der Waals interactions and high-temperature systems. A summary of these approximations is shown in Figure (3.1).

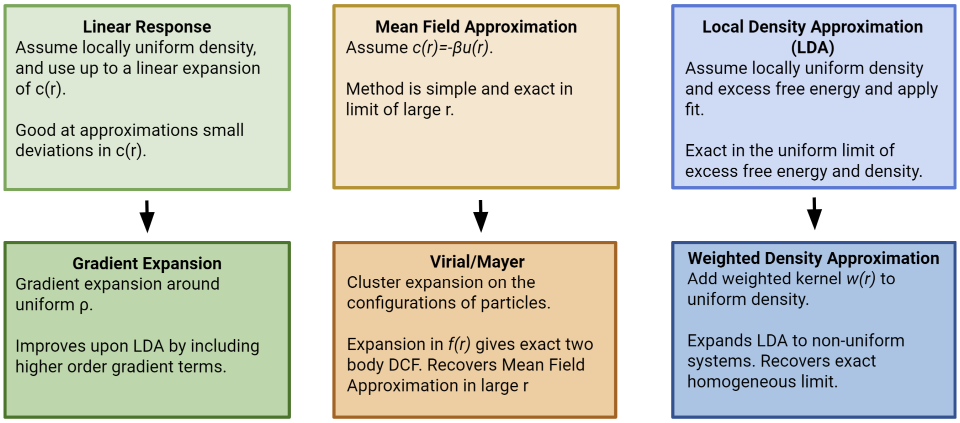

Linear Response
Assume locally uniform density, and use up to a linear expansion of c(r).
Good at approximations small deviations in c(r).
Mean Field Approximation
Assume *c(r)=-βu(r)*.
Method is simple and exact in limit of large r.
Local Density Approximation (LDA)
Assume locally uniform density and excess free energy and apply fit.
Exact in the uniform limit of excess free energy and density.
Gradient Expansion
Gradient expansion around uniform ρ.
Improves upon LDA by including higher order gradient terms.
Virial/Mayer
Cluster expansion on the configurations of particles.
Expansion in *f(r)* gives exact two body DCF. Recovers Mean Field Approximation in large r
Weighted Density Approximation
Add weighted kernel *w(r)* to uniform density.
Expands LDA to non-uniform systems. Recovers exact homogeneous limit.

Figure 3.1: Summaries of the various DCF models are summarized above with their generalizations in the second row.

3.1 Correlations in the Mean Field Limit

While any the above options for DCF models can be used, we choose to explore the approximations associated with the MF limit and the cluster expansion. We make this choice, as plasmas are characterized by long-range interactions, which motivates the use of an underlying mean field. If we ignore the screening effects of electrons, we can assume that interactions between ions in the plasma can be described by the Coulomb potential

$$\varphi(r) = \frac{q^2}{r}, \tag{3.7}$$

where q is the (effective) charge of the ions. The MF approximation of the excess free energy takes the form of

$$\begin{aligned} \mathscr{F}_{\text{ex}}[n(\mathbf{r})] &= -\frac{1}{2}k_B T \iint \Delta n(\mathbf{r})\Delta n(\mathbf{r}')c^{(2)}(\mathbf{r},\mathbf{r}')\, d\mathbf{r}\, d\mathbf{r}' \qquad (3.8)\\ &= \frac{1}{2}\iint \frac{q^2 \Delta n(\mathbf{r})\Delta n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\, d\mathbf{r}\, d\mathbf{r}' \qquad (3.9) \end{aligned}$$

Upon taking the variational derivative, we then obtain

$$\frac{\delta \mathscr{F}_{\text{ex}}}{\delta n} = \int \frac{q^2 \Delta n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' \equiv q\Phi(\mathbf{r}). \tag{3.10}$$

Here, we have introduced the electrostatic potential $\Phi(\mathbf{r})$, which is an auxiliary function that solves the Poisson Equation:

$$\boxed{-\frac{1}{4\pi}\nabla^2\Phi = q\Delta n.} \tag{3.11}$$

It can be seen then that the inter-particle potential $\varphi(r)$ is none other than the Green's function for the Poisson equation. The quantity $q\Phi(\mathbf{r})$ is then the electrostatic *energy*, which is connected to the self-consistent electric field through the relation $\mathbf{E} = -q\nabla\Phi$. The electrostatic energy $q\Phi(\mathbf{r})$ acts as the mean field and can be substituted into the momentum equation, Equation (2.135), to get

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{k_B T}{m} \nabla \log(n) + \frac{1}{m}\mathbf{F}_{\text{ext}} - \frac{q}{m}\nabla\Phi. \tag{3.12}$$

Additionally, the Coulomb potential can be generalized to the Yukawa (or screened Coulomb) potential

$$\varphi(r) = \frac{q^2}{r} e^{-r/\lambda_s}, \tag{3.13}$$

where λ_s is the screening length. In the classical limit, this screening length reduces to the Debye length [40]. The Yukawa potential improves upon the Coulomb potential by introducing electron screening into the calculation of long range interactions. The functional form of the Yukawa potential can be seen as formally exact for large r, as it arises in any derivation involving a long-wave expansion (*i.e.*, an expansion of the response function in Fourier space for small k).

3.2 Correlations Through the Cluster Expansion

Cluster expansions are one way to generalize the mean field approach by accounting for interactions past a system average. To create the cluster expansion, we start with the partition function, $\mathcal{Z}$, which represents the sum (or integral) over all probabilities of finding the system in any of its possible energy states. For continuous, classical systems, the partition function can be expressed as

$$\mathcal{Z} = \frac{1}{N! h^{3N}} \iint \ldots \iint \exp\left\{ -\beta \sum_{i=1}^{N} \left(\frac{p_i^2}{2m} + \sum_{j>i}^{N} \varphi_{ij} \right) \right\} d\mathbf{r}_1\, d\mathbf{p}_1 \ldots d\mathbf{r}_N\, d\mathbf{p}_N, \tag{3.14}$$

where integrals with respect to $d\mathbf{r}_i$ are taken over some volume V, where integrals with respect to $d\mathbf{p}_i$ are taken over $\mathbb{R}^3$, h is the Planck constant, and $\varphi_{ij} = \varphi(|\mathbf{r}_i - \mathbf{r}_j|)$. The volume V is arbitrary and will ultimately be taken to the thermodynamic limit $V \to \infty$ such that $N/V \to n_0$. Note that the kinetic and potential energy contributions can be decomposed into "ideal" and "configurational" components, respectively, to obtain $\mathcal{Z} = \mathcal{Z}_0 \mathcal{Q}$, where

$$\mathcal{Z}_0 = \frac{1}{N! h^{3N}} \int \dots \int \exp\left\{ -\beta \sum_{i=1}^{N} \frac{p_i^2}{2m} \right\} d\mathbf{p}_1 \dots d\mathbf{p}_N, \tag{3.15}$$

$$\mathcal{Q} = \int \dots \int \exp\left\{ -\frac{1}{2}\beta \sum_{i=1}^{N} \sum_{j>i}^{N} \varphi_{ij} \right\} d\mathbf{r}_1 \dots d\mathbf{r}_N. \tag{3.16}$$

First, the ideal component can be reduced to

$$\begin{aligned}
\mathcal{Z}_0 &= \frac{1}{N! h^{3N}} \int \dots \int \exp\left\{ -\beta \sum_{i=1}^{N} \frac{p_i^2}{2m} \right\} d\mathbf{p}_1 \dots d\mathbf{p}_N & (3.17)\\
&= \frac{1}{N! h^{3N}} \int \dots \int \prod_{i=1}^{N} \exp\left\{ -\beta \frac{p_i^2}{2m} \right\} d\mathbf{p}_1 \dots d\mathbf{p}_N & (3.18)\\
&= \frac{1}{N! h^{3N}} \prod_{i=1}^{N} \int_{\mathbb{R}^3} \exp\left\{ -\beta \frac{p_i^2}{2m} \right\} d\mathbf{p}_i & (3.19)\\
&= \frac{1}{N! h^{3N}} \big(2\pi m k_B T\big)^{3N/2} & (3.20)\\
&= \frac{1}{N! \Lambda^{3N}}, & (3.21)
\end{aligned}$$

where Λ is the de Broglie wavelength defined previously. The remaining configurational component can be simplified as well by separating the exponential terms into products as

$$\begin{aligned}
\mathcal{Q} &= \int \dots \int \exp\left\{ -\beta \sum_{i=1}^{N} \sum_{j>i}^{N} \varphi_{ij} \right\} d\mathbf{r}_1 \dots d\mathbf{r}_N & (3.22)\\
&= \int \dots \int \prod_{i=1}^{N} \prod_{j>i}^{N} e^{-\beta \varphi_{ij}} \, d\mathbf{r}_1 \dots d\mathbf{r}_N. & (3.23)
\end{aligned}$$

The quantity $e^{-\beta\varphi_{ij}}$ does not serve as a useful expansion parameter, as it approaches unity for large r. To remedy this, we add and subtract 1 to the integrand to obtain

$$\mathcal{Q} = \int \dots \int \prod_{i=1}^{N} \prod_{j>i}^{N} (1 + f_{ij}) \, d\mathbf{r}_1 \dots d\mathbf{r}_N, \tag{3.24}$$

where the Mayer f-function, which will vanish at infinity, is defined by

$$f_{ij} \equiv e^{-\beta\varphi_{ij}} - 1. \tag{3.25}$$

Expanding out the products yields

$$\mathcal{Q} = \int \dots \int \left(1 + \sum_{i,j>i}^{N} f_{ij} + \sum_{i,j>i,k>j}^{N} f_{ij} f_{ik} + \sum_{i,j>i,\ell>k}^{N} f_{ij} f_{k\ell} + \dots \right) d\mathbf{r}_1 \dots d\mathbf{r}_N, \tag{3.26}$$

Each term in this series (*i.e.*, each sum over a unique combination of Mayer f-functions) represents a cluster of interacting particles. Cluster diagrams are often used to better visualize the different combinatorial possibilities. Truncating this expansion up to the two-body clusters, we obtain

$$\begin{aligned}
\mathcal{Q} &\approx \int \dots \int \left(1 + \sum_{i=1}^{N} \sum_{j>i}^{N} f_{ij} \right) d\mathbf{r}_1 \dots d\mathbf{r}_N && (3.27)\\
&= V^N + V^{N-2} \sum_{i=1}^{N} \sum_{j>i}^{N} \iint f_{ij} \, d\mathbf{r}_i \, d\mathbf{r}_j && (3.28)\\
&= V^N + \frac{1}{2} N(N-1) V^{N-2} \iint f(r) \, d\mathbf{r} \, d\mathbf{r}'. && (3.29)
\end{aligned}$$

Here, we have used the change of variables $\mathbf{r}_i \to \mathbf{r}$, $\mathbf{r}_j \to \mathbf{r}'$ and $r = |\mathbf{r} - \mathbf{r}'|$ in the integrals over f_{ij} to reveal that these integrals are identical. Recombining the ideal and configurational partition functions, we have

$$\mathcal{Z} = \mathcal{Z}_0 \mathcal{Q} \approx \frac{V^N}{N! \Lambda^{3N}} \left(1 + \frac{1}{2V^2} N(N-1) \iint f(r) \, d\mathbf{r} \, d\mathbf{r}' \right).$$

With an approximate representation of the partition function, we can now calculate the Helmholtz free energy as $\mathscr{F} = -k_B T \log(\mathcal{Z})$, and after subtracting the ideal contribution, the two-body DCF can be approximated as

$$c^{(2)}(r) \approx f(r) = e^{-\beta\varphi(r)} - 1. \tag{3.30}$$

This approximation accounts for the mean field behavior as well (as can be seen by Taylor expanding $f(r)$ in terms of $\beta\varphi(r)$ to linear order). Since the mean field contribution can be accounted for by the Poisson equation (3.11), it is more convenient subtract the mean field from the excess free energy and introduce the new term

$$C(\mathbf{r}) \equiv \beta \frac{\delta \mathscr{F}_{\text{ex}}}{\delta n} + \beta q \Phi = \int \Delta n(\mathbf{r}') \left[e^{-\beta\varphi} - 1 + \beta\varphi \right] d\mathbf{r}'. \tag{3.31}$$

Substituting the DCF back into the momentum equation (2.135), we then obtain:

$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{m} k_B T \nabla \log(n) + \frac{1}{m}\mathbf{F}_{\text{ext}} - \frac{q}{m}\nabla\Phi + \frac{k_B T}{m}\nabla C(\mathbf{r}).} \tag{3.32}$$

Our full system of equations can thus be summarized as

$$\frac{\partial n}{\partial t} + \nabla\cdot(n\mathbf{u}) = 0, \tag{3.33}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{m} k_B T \nabla \log(n) + \frac{1}{m}\mathbf{F}_{\text{ext}} - \frac{q}{m}\nabla\Phi + \frac{k_B T}{m}\nabla C(\mathbf{r}), \tag{3.34}$$

$$\frac{\partial T}{\partial t} + \big(T\mathbf{u} - \lambda\nabla T\big)\cdot\nabla\log(n) + 2\nabla T\cdot\mathbf{u} = \nabla\cdot(\lambda\nabla T). \tag{3.35}$$

CHAPTER 4

ANALYSIS OF GOVERNING EQUATIONS

The system of differential equations, while not solvable, can be simplified and investigated analytically. This chapter employs nondimensionalization to reduce the parameter space, and linear stability analysis and dispersion relations to find the approximate linear behavior.

4.1 Nondimensionalization

The main goal of nondimensionalization is to simplify the mathematical equations or expressions describing a problem by removing the dependence on specific units of measurement. This can make the problem easier to analyze, solve, or interpret, and can also provide insights into the behavior of the system under study without the need to specify actual numerical values for the variables and parameters.

Nondimensionalization is a process that involves scaling the variables and parameters of a problem to eliminate or reduce their dimensional units, and express them in dimensionless form. This process is done by rescaling the variables and parameters by suitable reference values, often chosen based on physical or mathematical considerations. Before rescaling, we will absorb the Boltzmann constant into the temperature as $k_BT \rightarrow T$, making it in units of energy, which is common practice. We will find these reference values by scaling

$$\mathbf{r} \rightarrow a_i\mathbf{r}, \qquad t \rightarrow \tau t, \qquad T(\mathbf{r},t) \rightarrow T_0 T(\mathbf{r},t) \tag{4.1}$$

where T_0 is the mean temperature, and τ is some characteristic time scale to be determined. The spatial coordinates have been scaled by the ion sphere radius, which is determined from the mean density of the system

$$n_0 = \left(\frac{3}{4\pi a_i^3}\right) \implies a_i = \left(\frac{3}{4\pi n_0}\right)^{1/3}. \tag{4.2}$$

The ion sphere radius can be interpreted as the radius of the average sphere occupied by an ion. The density, velocity, forces and electrostatic potential fields will then be rescaled as

$$n(\mathbf{r}, t) \to \frac{1}{a_i^3} n(\mathbf{r}, t), \tag{4.3}$$

$$\mathbf{u}(\mathbf{r}, t) \to \frac{a_i}{\tau} \mathbf{u}(\mathbf{r}, t), \tag{4.4}$$

$$\mathbf{F}(\mathbf{r}, t) \to \frac{\Theta}{a_i} \mathbf{F}(\mathbf{r}, t), \tag{4.5}$$

$$\Phi(\mathbf{r}, t) \to \frac{q}{a_i} \Phi(\mathbf{r}, t). \tag{4.6}$$

Upon substituting these scaled variables into the governing equations, we first see that the continuity equation is unchanged

$$\frac{1}{a_i^3 \tau}\left(\frac{\partial}{\partial t} n + \nabla \cdot (n\mathbf{u})\right) = 0 \implies \frac{\partial}{\partial t} n + \nabla \cdot (n\mathbf{u}) = 0. \tag{4.7}$$

Next, the momentum equation can be nondimensionalized as

$$\begin{aligned}
\frac{a_i}{\tau^2}\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) &= -\frac{T_0}{a_i}\frac{1}{m}\left(T\nabla\log(n) - \mathbf{F}_{\text{ext}} + \frac{q^2}{a_i T_0}\nabla\Phi - T\nabla C(\mathbf{r})\right) \\
\implies \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} &= -\frac{T_0 \tau^2}{a_i^2}\frac{1}{m}\left(T\nabla\log(n) - \mathbf{F}_{\text{ext}} + \frac{q^2}{a_i T_0}\nabla\Phi - T\nabla C(\mathbf{r})\right) \\
\implies \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} &= -T\nabla\log(n) + \mathbf{F}_{\text{ext}} - \Gamma\nabla\Phi + T\nabla C(\mathbf{r}),
\end{aligned} \tag{4.8}$$

where we have set the characteristic time scale to be $\tau^2 = \frac{ma_i^2}{T_0}$ and introduced the dimensionless parameter

$$\Gamma \equiv \frac{q^2}{a_i T_0}. \tag{4.9}$$

This parameter is known as the "Coulomb coupling parameter" and is often interpreted as the ratio of the Coulomb energy (q^2/a_i) to the thermal energy (T_0). The correlation contribution $C(\mathbf{r})$ will have the same form, except the inter-particle potential will now be expressed in dimensionless form as

$$\varphi(r) = \frac{\Gamma}{r} e^{-\kappa r}, \quad \kappa \equiv \frac{a_i}{\lambda_s}, \tag{4.10}$$

where λ_s is the screening length. The dimensionless parameter κ determines the strength of electron screening in the system, and in the limit $\kappa \to 0$, the Coulomb potential is recovered.

The temperature evolution equation can be nondimensionalized as

$$\frac{\Theta}{\tau}\left(\frac{\partial T}{\partial t} + \left(T\mathbf{u} - \frac{\tau}{a_i^2}\lambda \nabla T\right) \cdot \nabla \log(n) + 2\nabla T \cdot \mathbf{u}\right) = \frac{\Theta}{a_i^2} \nabla \cdot (\lambda \nabla T) \tag{4.11}$$

$$\implies \frac{\partial T}{\partial t} + \left(T\mathbf{u} - \frac{\tau}{a_i^2}\lambda \nabla T\right) \cdot \nabla \log(n) + 2\nabla T \cdot \mathbf{u} = \frac{\tau}{a_i^2} \nabla \cdot (\lambda \nabla T), \tag{4.12}$$

$$\implies \frac{\partial T}{\partial t} + \left(T\mathbf{u} - \lambda \nabla T\right) \cdot \nabla \log(n) + 2\nabla T \cdot \mathbf{u} = \nabla \cdot (\lambda \nabla T), \tag{4.13}$$

where the thermal conductivity has been scaled by $\lambda \to \frac{a_i^2}{\tau}\lambda$ (note that absorbing the Boltzmann constant into the temperature, λ is now technically a thermal *diffusivity*). This dimensionless thermal diffusivity is sometimes known as a Fourier number (Fo).

Finally, the Poisson Equation can be nondimensionalized as

$$\frac{q}{a_i^3}\left(-\frac{1}{4\pi}\nabla^2\Phi\right) = \frac{1}{a_i^3}\left(q\Delta n\right) \rightarrow -\frac{1}{4\pi}\nabla^2\Phi = \Delta n \tag{4.14}$$

In summary, our final nondimensionalized system is given by

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{u}) = 0, \tag{4.15}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -T\nabla\log(n) + \mathbf{F}_{\text{ext}} - \Gamma\nabla\Phi + T\nabla C(\mathbf{r}), \tag{4.16}$$

$$\frac{\partial T}{\partial t} + \big(T\mathbf{u} - \lambda\nabla T\big)\cdot\nabla\log(n) + 2\nabla T\cdot\mathbf{u} = \nabla\cdot(\lambda\nabla T), \tag{4.17}$$

$$-\frac{1}{4\pi}\nabla^2\Phi = \Delta n, \tag{4.18}$$

where the dynamics are entirely determined by the dimensionless parameters Γ, κ and λ.

4.2 Linearized Equations of Motion

To perform a linear stability analysis, we must first find an equilibrium state. One known equilibrium is

$$n(\mathbf{r},t) = n_0, \qquad \mathbf{u}(\mathbf{r},t) = \mathbf{0}, \qquad T = T_0, \qquad \Phi(\mathbf{r}) = \Phi_0, \tag{4.19}$$

which corresponds to a completely uniform (or quiescent) state. We can perturb about this equilibrium state by adding small perturbations denoted with tildes as

$$n = n_0 + \tilde{n}(\mathbf{r},t), \quad \mathbf{u} = \tilde{\mathbf{u}}(\mathbf{r},t), \quad T = T_0 + \tilde{T}(\mathbf{r},t), \quad \Phi = \Phi_0 + \tilde{\Phi}(\mathbf{r}). \tag{4.20}$$

where $\tilde{n}, \tilde{\mathbf{u}}, \tilde{\Phi}, \tilde{T} \ll 1$. Additionally, in equilibrium, all external forces must be zero, so $\mathbf{F}_{\text{ext}} = \mathbf{0}$. In some systems, the thermal diffusivity will be temperature- and density-dependent, so we perturb about its equilibrium state $\lambda_0 = \lambda(n_0, T_0)$ as well

$$\lambda = \lambda_0 + \tilde{\lambda}, \tag{4.21}$$

where $\tilde{\lambda} \ll 1$. The new perturbed states can be substituted into the evolution equations to get

$$\frac{\partial}{\partial t}(n_0 + \tilde{n}) + \nabla \cdot ((n_0 + \tilde{n})\,\tilde{\mathbf{u}}) = 0 \tag{4.22}$$

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\tilde{\mathbf{u}} \cdot \nabla)\tilde{\mathbf{u}} = \left(T_0 + \tilde{T}\right) \nabla \left(\tilde{C}(\mathbf{r}) - \log(n_0 + \tilde{n})\right) - \Gamma \nabla \left(\Phi_0 + \tilde{\Phi}\right) \tag{4.23}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left(T_0 + \tilde{T}\right) + \left(\left(T_0 + \tilde{T}\right)\tilde{\mathbf{u}} - \left(\lambda_0 + \tilde{\lambda}\right)\nabla\left(T_0 + \tilde{T}\right)\right) \cdot \nabla \log(n_0 + \tilde{n}) \\ + 2\nabla\left(T_0 + \tilde{T}\right) \cdot \tilde{\mathbf{u}} = \nabla \cdot \left(\left(\lambda_0 + \tilde{\lambda}\right)\nabla\left(T_0 + \tilde{T}\right)\right)\end{aligned} \tag{4.24}$$

$$-\frac{1}{4\pi}\nabla^2\left(\Phi_0 + \tilde{\Phi}\right) = \tilde{n}. \tag{4.25}$$

We have also introduced the perturbation associated with the correlational contributions in excess of the mean field

$$\tilde{C}(\mathbf{r}) = \int \tilde{n}(\mathbf{r}')\left[e^{-\left(\beta_0 + \tilde{\beta}\right)\varphi} - 1 + \left(\beta_0 + \tilde{\beta}\right)\varphi\right] d\mathbf{r}', \tag{4.26}$$

where $\beta_0 = 1/T_0$, $\tilde{\beta} = -\beta_0^2\tilde{T}$, and again, $\varphi = \varphi(\mathbf{r} - \mathbf{r}')$. Since the dominant behavior near equilibria is determined by the linear terms, the nonlinear terms can to be disregarded to yield

$$\frac{\partial \tilde{n}}{\partial t} = -n_0 \nabla \cdot \tilde{\mathbf{u}} \tag{4.27}$$

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = -\frac{T_0}{n_0}\nabla\tilde{n} - \Gamma_0 \nabla\tilde{\Phi} + T_0 \nabla\tilde{C} \tag{4.28}$$

$$\frac{\partial \tilde{T}}{\partial t} = \lambda_0 \nabla^2 \tilde{T} \tag{4.29}$$

$$-\frac{1}{4\pi}\nabla^2\tilde{\Phi} = \tilde{n}, \tag{4.30}$$

where the correlation contributions have been linearized to

$$\tilde{C}(\mathbf{r}) = \int \tilde{n}(\mathbf{r}') \left[e^{-\beta_0 \varphi} - 1 + \beta_0 \varphi \right] \, d\mathbf{r}'. \tag{4.31}$$

This linear system describes the approximate behavior near equilibrium up to linear order.

The linearized behavior can be classified to predict the behavior of the solutions in the full nonlinear regime. The density and momentum equations are hyperbolic in nature which means that the solutions to the equations are determined by the propagation of wave-like disturbances through the system. The temperature equation is parabolic which means its solutions are typically diffusive by nature. We shall see this recurrence in expected behavior in the dispersion relations as well as use it to consider various numerical algorithms for simulations.

4.3 Stability Analysis

The linear system can also be analyzed using dispersion relations. Dispersion relations are mathematical relationships that traditionally describe how the frequency and wave vector of a propagating wave are related in continuous media or systems. However, they can be used to describe the growth rate of a given wave vector as well. To begin finding the dispersion relation for the approximate linear system, we let

$$\tilde{n} = \hat{n} e^{\sigma t + i\mathbf{k}\cdot\mathbf{r}}, \quad \tilde{\mathbf{u}} = \hat{\mathbf{u}} e^{\sigma t + i\mathbf{k}\cdot\mathbf{r}}, \quad \tilde{T} = \hat{T} e^{\sigma t + i\mathbf{k}\cdot\mathbf{r}}, \quad \tilde{\Phi} = \hat{\Phi} e^{\sigma t + i\mathbf{k}\cdot\mathbf{r}}. \tag{4.32}$$

Substituting these relations into the linearized system, we see spatial derivatives pull down a factor of $i\mathbf{k}$ while time derivatives pull down a factor of σ as

$$\sigma\hat{n} = -in_0\mathbf{k}\cdot\hat{\mathbf{u}} \tag{4.33}$$

$$\sigma\hat{\mathbf{u}} = -\frac{iT_0}{n_0}\mathbf{k}\hat{n} + iT_0\mathbf{k}\hat{n}\hat{c}^{(2)}(k) = -\left(1 - n_0\hat{c}^{(2)}(k)\right) i\frac{T_0}{n_0}\mathbf{k}\hat{n} \tag{4.34}$$

$$\sigma\hat{T} = -\lambda_0 k^2\hat{T}. \tag{4.35}$$

where $k = |\mathbf{k}|$, $\hat{c}^{(2)}(k)$ is the Fourier transform of the two-body DCF, and the mean field contribution has been reabsorbed back into the two-body DCF. Note that we have made use of the convolution theorem here, which states that the Fourier transform of a convolution between two functions is equivalent to the product of their Fourier transforms. We can immediately see that the temperature equation is decoupled from the remaining system, so $\sigma(k)$ can be solved independently to get

$$\sigma(k) = -\lambda_0 k^2 \tag{4.36}$$

As expected from the linear stability analysis, the temperature is dissipative since $\sigma \leq 0$. The remaining system of algebraic equations can be represented in matrix form as

$$\begin{bmatrix} \sigma & in_0k_x & in_0k_y & in_0k_z \\ \left(1 - n_0\hat{c}^{(2)}\right) i\frac{T_0}{n_0}k_x & \sigma & 0 & 0 \\ \left(1 - n_0\hat{c}^{(2)}\right) i\frac{T_0}{n_0}k_y & 0 & \sigma & 0 \\ \left(1 - n_0\hat{c}^{(2)}\right) i\frac{T_0}{n_0}k_z & 0 & 0 & \sigma \end{bmatrix} \begin{bmatrix} \hat{n} \\ \hat{u}_x \\ \hat{u}_y \\ \hat{u}_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.37}$$

For nontrivial solutions, the determinant of this matrix must be zero, which results in the following equation:

$$\sigma^4 + \sigma^2 \left(1 - n_0 \hat{c}^{(2)}(k)\right) T_0 k^2 = 0. \tag{4.38}$$

Then nonzero solutions will be of the form

$$\sigma(k) = \pm k \sqrt{\left(n_0 \hat{c}^{(2)}(k) - 1\right) T_0}. \tag{4.39}$$

The behavior of this system will be determined by the sign of the radicand, as real values of $\sigma(k)$ correspond to growth/decay, while imaginary values correspond to wave dispersion (hence the namesake of the relation). To gain insight into the sign of $\hat{c}^{(2)}(k)$, we use the Ornstein-Zernike relation [41], which connects the DCF to the radial distribution function $g(r)$ as

$$\hat{h}(k) = \hat{c}^{(2)}(k) + n_0 \hat{h}(k) \hat{c}^{(2)}(k), \tag{4.40}$$

where $h(r) = g(r) - 1$ is the *total* correlation function, and the hats denote Fourier transforms. The radial distribution function determines how many particles are within a distance of r and $r + dr$ away from an average particle relative to the mean density. One can determine $g(r)$ with atomistic simulations or indirectly through scattering experiments using the (static) structure factor through the relationship

$$S(k) = 1 + n_0 \hat{h}(k). \tag{4.41}$$

Combining the Ornstein-Zernike with the structure factor, we see

$$n_0 \hat{c}^{(2)}(k) - 1 = \frac{n_0 \hat{h}(k)}{1 + n_0 \hat{h}(k)} - 1 = -\frac{1}{S(k)}. \tag{4.42}$$

Since the structure is positive by definition $S(k) \geq 0$, the radicand in the dispersion relation will be negative, and thus, $\sigma(k)$ will be purely imaginary. Letting $\sigma(k) = \pm i\omega(k)$, we obtain

$$\omega(k) = k\sqrt{\left(1 - n_0\hat{c}^{(2)}(\mathbf{k})\right) T_0} = \frac{k}{\sqrt{\beta_0 S(k)}}, \tag{4.43}$$

so the evolution of the density and momentum fields is purely dispersive at linear order. This dispersion can be better understood through the phase velocity

$$v_p = \frac{\omega(k)}{k} = \frac{1}{\sqrt{\beta_0 S(k)}}, \tag{4.44}$$

which describes the rate at which a wave with wave number k propagates through the medium. $S(k) \to 1$ for sufficiently large k, so the phase velocity approaches $v_p = \sqrt{T_0}$, which is simply the sound speed of an ideal gas (in dimensionless units).

CHAPTER 5

NUMERICAL IMPLEMENTATION

Given the complexity of the system, we turn to numerical simulation to find a solution, as there is little that can be done with analytical methods when exploring the full solution space. This chapter will discuss the foundational theory for the numerical implementation as well as the issues faced along the way.

We begin by reducing the model complexity. Numerical simulations of fluid equations can require significant computational resources. In light of this high demand, we assume the system is uniform along the y and z directions, which reduces the model to one spatial dimension

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x}(nu) = 0, \tag{5.1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -T\frac{\partial}{\partial x}\left(\log(n)\right) + \Gamma\frac{\partial \Phi}{\partial x} - T\frac{\partial C}{\partial x}, \tag{5.2}$$

$$\frac{\partial T}{\partial t} + \left(Tu - \lambda\frac{\partial T}{\partial x}\right)\frac{\partial}{\partial x}\log(n) + 2\frac{\partial T}{\partial x}u = \lambda\frac{\partial^2 T}{\partial x^2}, \tag{5.3}$$

$$-\frac{1}{4\pi}\frac{\partial^2 \Phi}{\partial x^2} = \Delta n, \tag{5.4}$$

where we have also taken the thermal diffusivity λ to be constant. We note that even if the field variables are in one spatial dimension, the correlation effects must still be aware of all three dimensions at the atomic scale

$$C(x) = \iiint \Delta n(x')\left[e^{-\beta\varphi(r)} - 1 + \beta\varphi(r)\right]\, d\mathbf{r}', \tag{5.5}$$

where $r = |\mathbf{r} - \mathbf{r}'|$. Additionally, we note that the integral notation has been changed to one dimensional integration.

Numerical methods can now be used to convert the equations into a form that a computer can read. There are generally three major types of numerical methods one can implement: finite difference, finite element and finite volume methods. Each type offers a variety of potential benefits. Finite Difference Methods (FDM) discretize the domain into a regular grid of points and approximates the derivatives at each point using difference formulas. The values at the grid points are then updated in time according to the PDE using time-stepping schemes. Finite Element Methods (FEM) discretize the domain into a set of non-overlapping elements and then representing the solution as a basis expansion of piecewise polynomials (sometimes also referred to as "elements"). The method involves minimizing the error in the approximation over the elements and assembling a global system of equations that describes the behavior of the solution across the entire domain. Finite Volume Methods (FVM) integrate the PDE over small control volumes of the discretized domain and then make use of the divergence theorem to convert these volumes into fluxes across surfaces. Because these fluxes are always balanced, the method is conservative, which makes it ideal for preserving conservation laws. For this reason, we will focus our attention on finite volume methods to numerically solve our system.

5.1 Discretization

To define a discretization for the domain $0 \leq x \leq L$ and $0 \leq t \leq T$, we can use a uniform grid with equidistant spacing in both the x and t directions. For the spatial axis, we can divide the interval $[0, L]$ into N equally spaced intervals with a grid spacing of Δx, such that $x_i = i\Delta x$ for $i = 0, 1, 2, \ldots, N$, so there will be a total of $N + 1$ spatial grid points. Similarly for the time axis, we can divide the interval $[0, T]$ into M equally spaced intervals with a time step of Δt, such that $t_j = j\Delta t$ for $j = 0, 1, 2, \ldots, M$, resulting in $M+1$ time steps. With this uniform grid discretization, we can approximate the various velocity moments with discrete functions, where:

$$n(x,t) \approx n_i^j = n(x_i, t_j), \quad u(x,t) \approx u_i^j = u(x_i, t_j), \tag{5.6}$$

$$T(x,t) \approx T_i^j = T(x_i, t_j), \quad \Phi(x) \approx \Phi_i = \Phi(x_i). \tag{5.7}$$

The discrete function n_i^j, u_i^j and T_i^j represent the approximation of the solutions $n(x,t)$, $u(x,t)$ and $T(x,t)$ at the spatial grid point x_i and the time grid point t_j, and Φ_i represents the approximation of $\Phi(x)$ at the spatial grid point x_i. We begin by creating a finite volume method for evaluating the time-dependent solutions and leave the solution of Φ for later (see Modified Thomas Algorithm in Section (5.5)).

For each field at each time step, the quantity in any cell will be updated by the incoming and outgoing fluxes and sources. To preserve conservation, fluxes are computed across cell interfaces and used to update adjacent cells accordingly. Cell interfaces can also be spaced uniformly across the discretized domain. Then for each cell centered at x_i, the left and right interfaces will be $\frac{1}{2}\Delta x$ from x_i located at $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, respectively.

We choose to use a finite volume method (FVM), as it conservatively handles the fluxes and sources within each volume. That is, the FVM implementation would enforce the physical conservation of particle number, momentum, temperature and charge. There are many potential choices for the exact FVM to use, especially with regards to how the fluxes are discretized across the cell interfaces. Besides traditional FVMs that use a Forward Euler scheme for the temporal updating, specialized methods such as Godunov schemes and Roe Solvers have been developed to exploit the hyperbolic nature of these evolution equations by preserving solutions along characteristics.

5.2 Gudonov Schemes

Given the hyperbolic nature of the density and momentum equations, we will choose a numerical scheme that reproduces similar behavior. More specifically, the chosen numerical scheme should propagate information from the fluxes along the characteristics of the solutions. Given the nonlinear nature of the fluxes, we begin by investigating the Riemann Problem.

The Riemann Problem [42] considers a simplified version of a hyperbolic PDE, with initial conditions consisting of two constant states on either side of the interface. These constant states represent the values of the solution variables in the neighboring cells, and the Riemann problem seeks to determine the solution in the region near the interface.

Given two constant initial states, denoted as u_L and u_R, on the left and right sides of the interface, respectively, the Riemann problem seeks to determine the solution

$u(x, t)$ for $x > 0$ and $t > 0$, subject to the initial conditions:

$$u(x,0) = \begin{cases} u_L & \text{if } x < 0, \\ u_R & \text{if } x > 0. \end{cases} \tag{5.8}$$

The initial condition represents a physical situation where the medium undergoes a sudden change, such as a shock wave or a contact discontinuity. A visual representation of the Riemann Problem and its solution is shown in Figure 5.1. The solution for U follows the characteristics found from the fluxes at U_L and U_R. The solution will propogate along these characteristics.

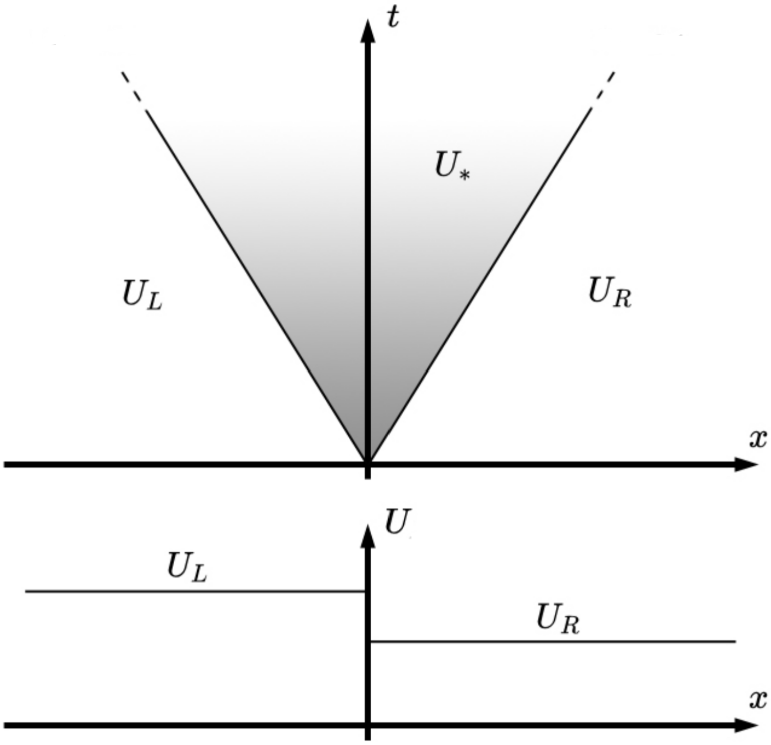


Figure 5.1: U is a piece wise constant function with a jump discontinuity at a cell interface. The evolution of U follows its characteristics as defined by its flux [3].

The Godunov scheme works by solving the Riemann problem at each cell interface to obtain the numerical flux, which is then used to update the solution in each cell at each time step. This method is computationally expensive though, so many approximate Godunov-inspired methods have been developed.

5.3 Roe Solvers

Like Godunov schemes, Roe solvers are based on the solution of Riemann problems at the cell interfaces. However, they use a different approach to computing the numerical fluxes, which allows them to capture shocks and other discontinuities more accurately and efficiently.

The basic idea behind Roe solvers is to linearize the Riemann problem at each cell interface by using a local linearization of the flux function [43]. This leads to a simpler and more efficient algorithm for computing the numerical fluxes, which is given by:

$$f_{i+\frac{1}{2}} = \frac{1}{2}(f(u_i) + f(u_{i+1})) - \frac{1}{2}(A_i + A_{i+1})(u_{i+1} - u_i) \tag{5.9}$$

where $f(u)$ is the flux function, $f_{i+\frac{1}{2}}$ is the flux at the cell interface between cells i and $i+1$, u_i and u_{i+1} are the values of the conserved variables at the cell centers, and A_i and A_{i+1} are the Jacobian matrices of the flux function evaluated at u_i and u_{i+1}, respectively.

The Roe solver has several advantages over Godunov schemes. First, it is computationally more efficient because it avoids the need for solving the Riemann problem exactly at each interface. Second, it is more accurate at capturing shocks and other discontinuities because it uses a linearization of the flux function that takes into account the local flow properties.

One limitation of the Roe solver is that it can suffer from numerical dissipation, which can lead to loss of accuracy in the solution. To address this issue, several variants of the Roe solver have been developed, such as the AUSM (Advection Upstream Splitting Method) [44] and the HLL (Harten-Lax-van Leer) [45] solver. These variants use different techniques to minimize numerical dissipation while maintaining the efficiency and accuracy of the original Roe solver.

5.4 Forward Euler

While we have explored several numerical methods above, a simple Forward Euler scheme may suffice for the present problem. This is because the nonlinearities in the equations do not create enough instability to warrant a more complex scheme. To derive the finite volume method, we first integrate the equations over a particular "volume" (interval) from $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$. For the density evolution equation, this gives us:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial n_i}{\partial t}\, dx + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x} F_{n_i}\, dx = 0 \tag{5.10}$$

where

$$F_{n_i} = n_i u_i \tag{5.11}$$

is the density flux. Evaluating the integral of the flux and interchanging the time derivative on the density, we have

$$\frac{d}{dt}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} n_i\,dx + \left[F^j_{n_{i+\frac{1}{2}}} - F^j_{n_{i-\frac{1}{2}}}\right] = 0 \tag{5.12}$$

We next introduce the average density over the volume given by

$$\overline{n}_i = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} n_i\,dx, \tag{5.13}$$

which will satisfy the ordinary differential equation

$$\frac{d}{dt}\overline{n}_i + \frac{1}{\Delta x}\left[F^j_{n_{i+\frac{1}{2}}} - F^j_{n_{i-\frac{1}{2}}}\right] = 0. \tag{5.14}$$

Up to this point, this relation is still formally exact, and we will now approximate the derivative using a Forward Euler scheme. The solution, n_i^{j+1}, at the next time step can then be found by:

$$\overline{n}_i^{j+1} = \overline{n}_i^{j} - \frac{\Delta t}{\Delta x}\left[F_{n_{i+\frac{1}{2}}} - F_{n_{i-\frac{1}{2}}}\right]. \tag{5.15}$$

To compute the fluxes across a cell interface, we must use numerical approximations as well. If the sign of the flux is known, a simple upwinding scheme can be used in which the flux can be appoximated in the direction the flow is coming from. For example, if the flux is positive (flow to the right), the flux can be interpolated by choosing the left cell as the value, which gives us

$$F_{n_{i+\frac{1}{2}}} \approx F_{n_i}, \qquad F_{n_{i-\frac{1}{2}}} \approx F_{n_{i-1}}. \tag{5.16}$$

Substituting this approximation into the discretization, we get:

$$\overline{n}_i^{j+1} = \overline{n}_i^j - \frac{\Delta t}{\Delta x}\left[F_{n_i} - F_{n_{i-1}}\right]. \tag{5.17}$$

Interestingly, this is equivalent to the Forward Euler-Forward Difference scheme, a finite difference method.

We can repeat this process of integrating over a cell/volume, approximating the time derivatives and the flux, and solving for the solution in the following time step for both the momentum and temperature equations to obtain

$$\begin{aligned}
\overline{u}_i^{j+1} = \overline{u}_i^j - \frac{\Delta t}{\Delta x}\Bigg[&\frac{1}{2}(\overline{u}_i^j)^2 - \frac{1}{2}(\overline{u}_{i-1}^j)^2 - \overline{T}_i^j\left(\log(\overline{n})_i^j - \log(\overline{n})_{i-1}^j\right) \\
&+ \Gamma\left(\Phi_i^j - \Phi_{i-1}^j\right) - \overline{T}_i^j\left(C_i^j - C_{i-1}^j\right)\Bigg], \qquad (5.18)\\
\overline{T}_i^{j+1} = \overline{T}_i^j &+ \frac{\lambda\Delta t}{\Delta x^2}\left[\overline{T}_{i+1}^j - 2\overline{T}_i^j + \overline{T}_{i-1}^j\right] \\
&+ \frac{\Delta t}{\Delta x}\left[\left(\lambda\left(\overline{T}_i^j - \overline{T}_{i-1}^j\right) - \overline{T}_i^j\overline{u}_i^j\right)\left(\log(\overline{n})_i^j - \log(\overline{n})_{i-1}^j\right)\right] \\
&- \frac{\Delta t}{\Delta x}\left[2\overline{u}_i^j\left(\overline{T}_i^j - \overline{T}_{i-1}^j\right)\right]. \qquad (5.19)
\end{aligned}$$

5.5 Modified Thomas Algorithm

The Poisson Equation is solved separately from the other fields since it is an ordinary differential equation independent of time. At time $t = t_j$, we have

$$\overline{\Phi}_{i+1} - 2\overline{\Phi}_i + \overline{\Phi}_{i-1} = -4\pi\left(\overline{n}_i - \overline{n}_0\right) \tag{5.20}$$

where $i = 0, 1, \ldots, N$, and $\overline{\Phi}_{N+1} = \overline{\Phi}_0$ and $\overline{\Phi}_{-1} = \overline{\Phi}_N$ are included to account for periodic boundary conditions. These equations can be combined into a matrix system:

$$\begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 1 & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} \overline{\Phi}_0 \\ \overline{\Phi}_1 \\ \vdots \\ \overline{\Phi}_{N-1} \\ \overline{\Phi}_N \end{bmatrix} = -4\pi \begin{bmatrix} \overline{n}_0 - n_0 \\ \overline{n}_1 - n_0 \\ \vdots \\ \overline{n}_{N-1} - n_0 \\ \overline{n}_N - n_0 \end{bmatrix} \tag{5.21}$$

which can be more concisely written as

$$\mathbf{A}\mathbf{x}_\Phi = -4\pi\mathbf{x}_{\Delta n}, \tag{5.22}$$

where $\mathbf{A}$ is the matrix whose sub- and super-diagonals are 1, the diagonal is -2 and whose $(N, 1)$ and $(1, N)$ entries are 1, $\mathbf{x}_\Phi$ and $\mathbf{x}_{\Delta n}$ are the vectors with components Φ_i and Δn_i, respectively. We then have

$$\mathbf{x}_\Phi = -4\pi\mathbf{A}^{-1}\mathbf{x}_{\Delta n} \tag{5.23}$$

as the solution for $\overline{\Phi}$ at time t_j. However, a matrix inversion is computationally complex. The Modified Thomas Algorithm can be used to exploit the near tridiagonal nature of the matrix to reduce the numerical complexity of the inversion.

First, the matrix $\mathbf{A}$ can be decomposed into a tridiagonal matrix, $\mathbf{B}$, plus the outer product of two vectors, $\mathbf{a}$ and $\mathbf{b}$ such that

$$\mathbf{A} = \mathbf{B} + \mathbf{ab} \tag{5.24}$$

where

$$\mathbf{B} = \begin{bmatrix} -2-1 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & -2-1 \end{bmatrix}, \qquad \mathbf{a} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{5.25}$$

With the constructed tridiagonal matrix $\mathbf{B}$, the Thomas Algorithm can be used to find the solutions of two linear systems,

$$\mathbf{B}\mathbf{x}_1 = \mathbf{x}_{\Delta n}, \qquad \mathbf{B}\mathbf{x}_2 = \mathbf{a}. \tag{5.26}$$

The Thomas Algorithm exploits tridiagonal matrices by performing two sweeps. The first eliminates the subdiagonal while the second sweep eliminates the superdiagonal. We can take the solutions for $\mathbf{x}_1$ and $\mathbf{x}_2$ and reconstruct a solution for $\mathbf{x}_\Phi$. By the Sherman-Morrison formula,

$$\mathbf{x}_\Phi = -4\pi\mathbf{A}^{-1}\mathbf{x}_{\Delta n} \quad = -4\pi\left(\mathbf{B} + \mathbf{ab}\right)^{-1}\mathbf{x}_{\Delta n} = -4\pi\left(\mathbf{x}_1 - \frac{\mathbf{x}_2\mathbf{b}}{1 + \mathbf{x}_2 \cdot \mathbf{b}}\mathbf{x}_1\right) \tag{5.27}$$

The Modified Thomas Algorithm reduces the computational complexity of the matrix inversion from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$.

5.6 Direct Correlation Function Calculation

Much of the novelty of this project is centered around the effects of the plasma correlations. This section focuses on how the correlation term,

$$\iiint \Delta n(\mathbf{r}')c^{(2)}(\mathbf{r}, \mathbf{r}')\, d\mathbf{r}' \tag{5.28}$$

is numerically integrated. Assuming homogeneity (*i.e.*, translation invariance of the system), the correlation term can be written as a convolution

$$C(\mathbf{r}) = \iiint \Delta n(\mathbf{r}')c^{(2)}(\mathbf{r}-\mathbf{r}')\,d\mathbf{r}' = \iiint \Delta n(\mathbf{r}-\mathbf{r}')c^{(2)}(\mathbf{r}')\,d\mathbf{r}'. \tag{5.29}$$

Given the computational complexity of numerical integration in higher dimensions, we would like to simplify the direct correlation function by reducing the number of integrals.

5.6.1 DCF Calculation By Direct Evaluation

As we are only investigating spatial variations in the x direction, the correlation term can be simplified directly by

$$C(x) = \iiint \Delta n(x-x')c^{(2)}(r')\,d\mathbf{r}', \tag{5.30}$$

where the translational invariance has been used to center the direct correlation function around the $\mathbf{r}'$ coordinate such that $\mathbf{r}-\mathbf{r}' \to \mathbf{r}'$, and the density equation has been reduced in its dimensional dependency. The direct correlation function (DCF) can be simplified using a change of variables to cylindrical coordinates about the x'-axis

$$\begin{aligned} C(x) &= \iiint \Delta n\left(x-x'\right)\left[e^{-\beta(x')\varphi(r')} - 1 + \beta(x')\varphi\left(r'\right)\right]\,d\mathbf{r}' && (5.31)\\ &= \iint \Delta n\left(x-x'\right)w'\left[e^{-\beta(x')\varphi(w')} - 1 + \beta(x')\varphi(w')\right]dw'dx', && (5.32) \end{aligned}$$

where $w' = \sqrt{\rho'^2 + x'^2}$, and the inter-particle potential is given in cylindrical coordinates as

$$\varphi(r) = \frac{\Gamma}{\sqrt{\rho^2+x^2}}e^{-\frac{\kappa}{\sqrt{\rho^2+x^2}}}. \tag{5.33}$$

The integral can be decomposed further, but it is divergent when evaluated over the w' domain *first*. Thus, the correlation term must be discretized in this form and plugged into Equation (5.18). By direct evaluation, the calculation of the correlation term is $\mathcal{O}(N^3)$ at each time step.

5.6.2 DCF Calculation By The Convolution Theorem

In the limit in which β is treated as constant inside the correlation integral, we can use the Convolution Theorem to further reduce computational complexity. The convolution theorem can be employed by

$$\mathcal{F}\{C(x)\} = \mathcal{F}\{\Delta n\,(x)\}\mathcal{F}\{c^{(2)}(r) + \beta\varphi(r)\} \tag{5.34}$$

where $\mathcal{F}$ denotes the Fourier Transform. The use of the Fourier Transform can greatly reduce the numerical complexity of the correlation term by using the Fast Fourier Transform.

The Fourier transform of the density can be simplified by evaluating over the y and z domains by

$$\begin{aligned}
\mathcal{F}\{\Delta n\,(x-x')\} &= \iiint \Delta n\,(x-x')\,e^{-i\mathbf{k}\cdot\mathbf{r}'}\,d\mathbf{r}' \\
&= \int \Delta n\,(x-x')\,e^{-ik_x x'}\,dx' \int e^{-ik_y y'}\,dy' \int e^{-ik_z z'}\,dz' \\
&= (2\pi)^2\delta(k_y)\delta(k_z)\int \Delta n\,(x-x')\,e^{-ik_x x'}\,dx' && (5.35) \\
&= (2\pi)^2\delta(k_y)\delta(k_z)\mathcal{F}_{x'}\{\Delta n\,(x-x')\}, && (5.36)
\end{aligned}$$

where $\mathcal{F}_{x'}$ denotes the one dimensional Fourier Transform, and $\delta(k)$ is a Dirac delta function in Fourier space. The Fourier Transform of the direct correlation function can be simplified by switching to spherical coordinates

$$\begin{aligned}
\mathcal{F}\{c^{(2)}(r')+&\beta\varphi(r')\} \\
&= \iiint \left[c^{(2)}(r') + \beta\varphi(r')\right] e^{-i\mathbf{k}\cdot\mathbf{r}'}\, d\mathbf{r'} && (5.37) \\
&= \int_0^\infty \int_0^\pi \int_0^{2\pi} \left[c^{(2)}(r') + \beta\varphi(r')\right] e^{-ikr'\cos(\phi)} r'^2 \sin(\phi) d\theta' d\phi' dr' && (5.38) \\
&= 2\pi \int_0^\infty \int_0^\pi \left[c^{(2)}(r') + \beta\varphi(r')\right] e^{-ikr'\cos(\phi)} r'^2 \sin(\phi) d\phi' dr' && (5.39) \\
&= \frac{4\pi}{k} \int_0^\infty \left[c^{(2)}(r') + \beta\varphi(r')\right] r' \sin(kr') dr' && (5.40)
\end{aligned}$$

Upon taking the inverse Fourier transform, the delta functions in Equation (5.36) will reduce the magnitude of $\mathbf{k}$ down to $k \to \sqrt{k_x^2 + 0^2 + 0^2} = |k_x|$. We then have a more computationally tractable representation for the correlation integral as

$$C(x) = 8\pi^3 \mathcal{F}^{-1} \left\{ \frac{1}{|k_x|} \mathcal{F}_{x'}\{\Delta n\,(x - x')\} \int_0^\infty \left[c^{(2)}(r') + \beta\varphi(r')\right] r' \sin(|k_x| r') dr' \right\}, \tag{5.41}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform in terms of k_x. The integral over the correlations can be precomputed, which means that only a one-dimensional Fourier transform and its inverse are needed at each time step, making the calculation $\mathcal{O}(N\log(N))$.

CHAPTER 6

RESULTS

This chapter discusses the results of the numerical simulations of the system.

6.1 Simulation Results

The discretized system was simulated in two different modes: with the correlation term and without it. The python code used for the simulation can be found in Appendix A.2. For both simulations, snapshots of the field values were periodically saved to track their behavior over time. To test the response of the system to different initial conditions, various combinations of Gaussian and sinusoidal perturbations about a constant initial value were used for each field. We assume $\lambda = 1$ for all simulations. Additionally, $\Gamma = \kappa = 1$ for this section. The resulting colormaps of each field are shown below in Tables (6.1) and (6.2). It can be seen that the correlations appear to add more structure to the field variables.

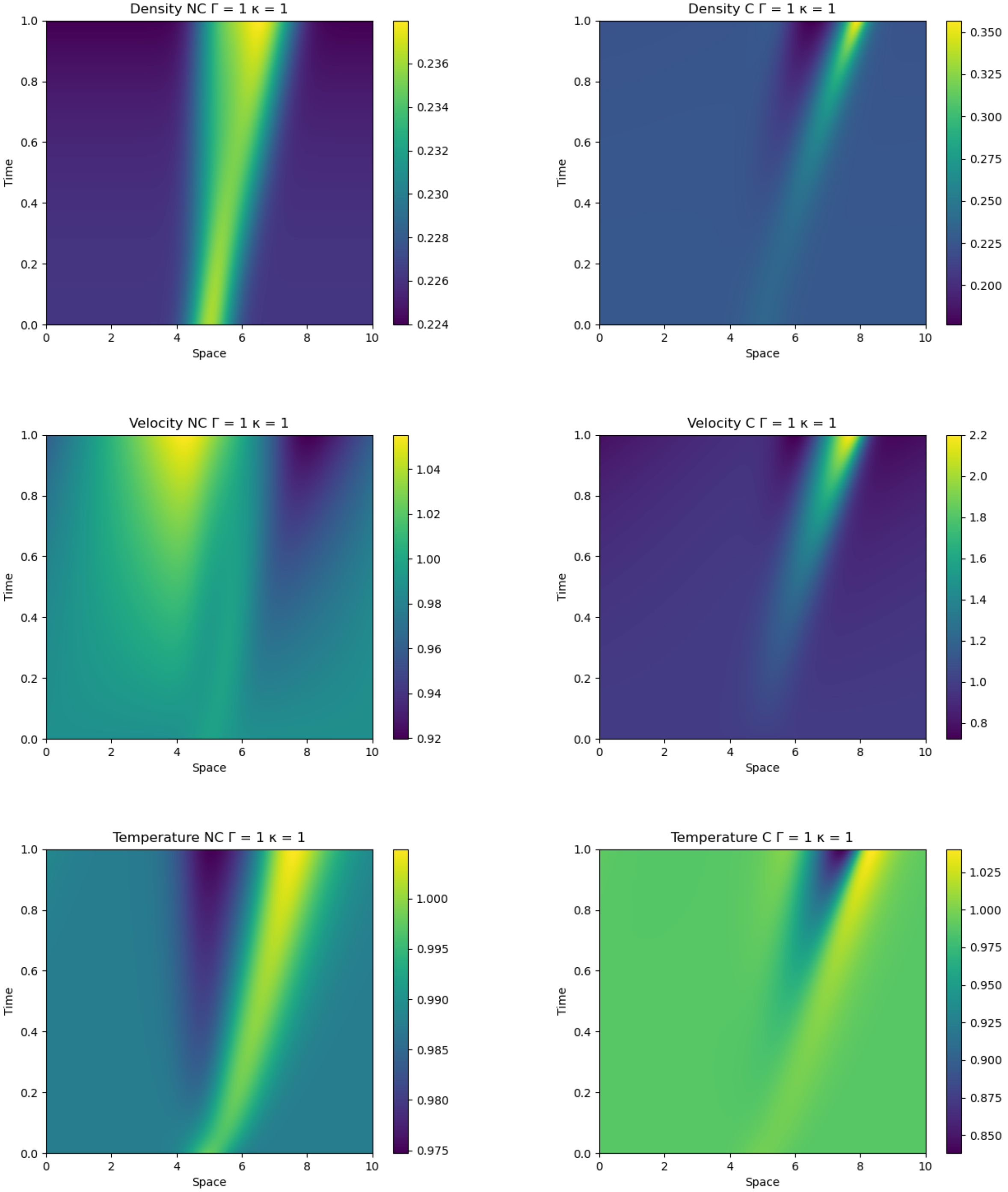


Table 6.1: Colormap of system evolution for $\Gamma = 1$ and $\kappa = 1$ with Gaussian initial conditions. Simulations with *correlations* are denoted with "C", while simulations with *no correlations* are denoted with "NC".

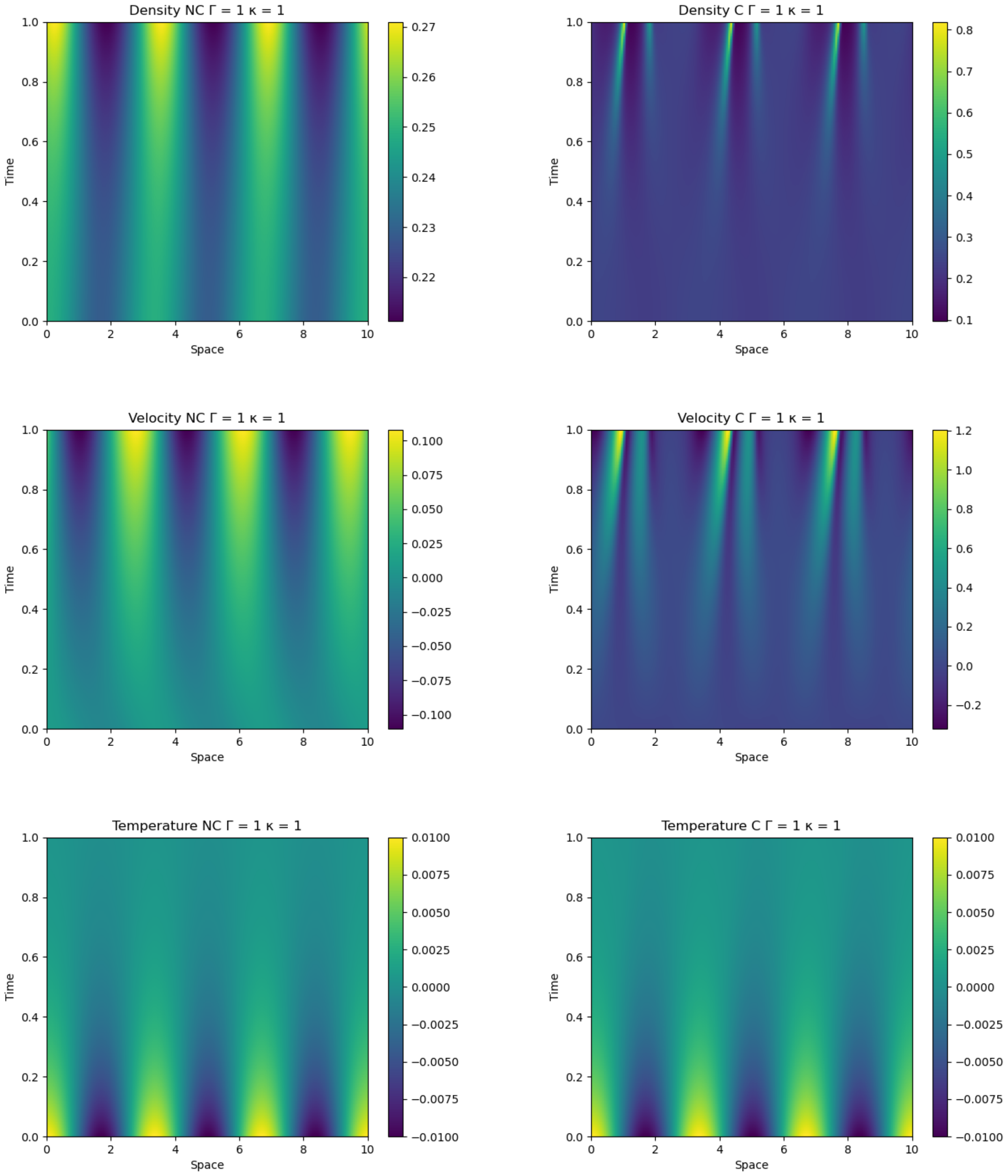


Table 6.2: Colormap of system evolution for $\Gamma = 1$ and $\kappa = 1$ with sinusoidal initial conditions. Simulations with *correlations* are denoted with "C", while simulations with *no correlations* are denoted with "NC".

6.2 Behavior across (Γ, κ) Space

For most plasmas, $\Gamma \ll 1$ and $\kappa \ll 1$. However, strongly-correlated plasmas have higher coupling and larger screening effects, which can affect their behavior. To investigate the effect of these two parameters, we have chosen to sample the parameter space where $\Gamma \in \left\{\frac{1}{10}, 1, 10\right\}$ and $\kappa \in \left\{\frac{1}{2}, 1, 2\right\}$. By examining this sampled parameter space, we can gain insight into how the behavior of the plasma changes as coupling and screening effects increase. The resulting colormap of the density is shown in Figures 6.1 (without correlations) and 6.2 (with correlations). It would appear that either increasing κ or decreasing Γ has the similar effect of increasing the period of oscillations.

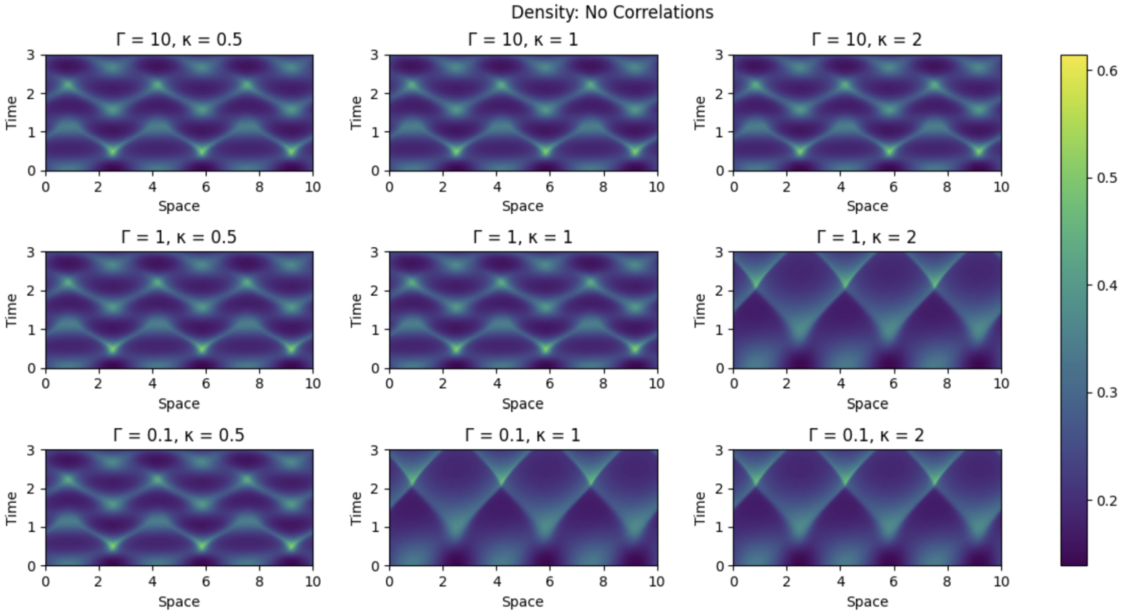


Figure 6.1: Colormaps of density evolution without excess correlations throughout the (Γ, κ) parameter space is shown above. Γ is chosen $\{\frac{1}{10}, 1, 10\}$ with increasing values as bottom to top, and κ is chosen $\{\frac{1}{2}, 1, 2\}$ with increasing values as left to right.

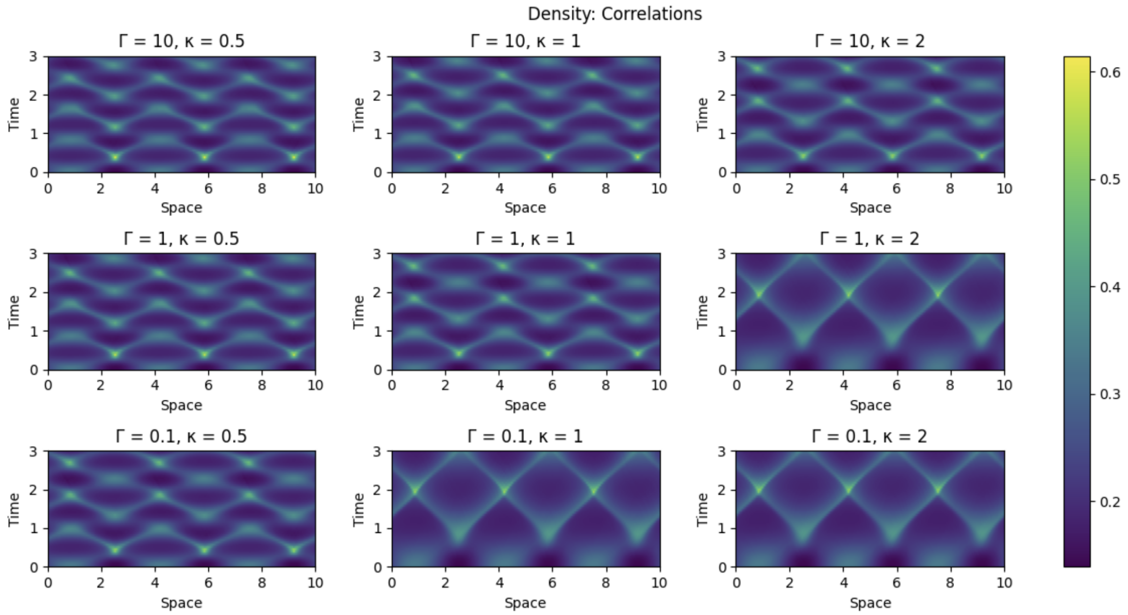
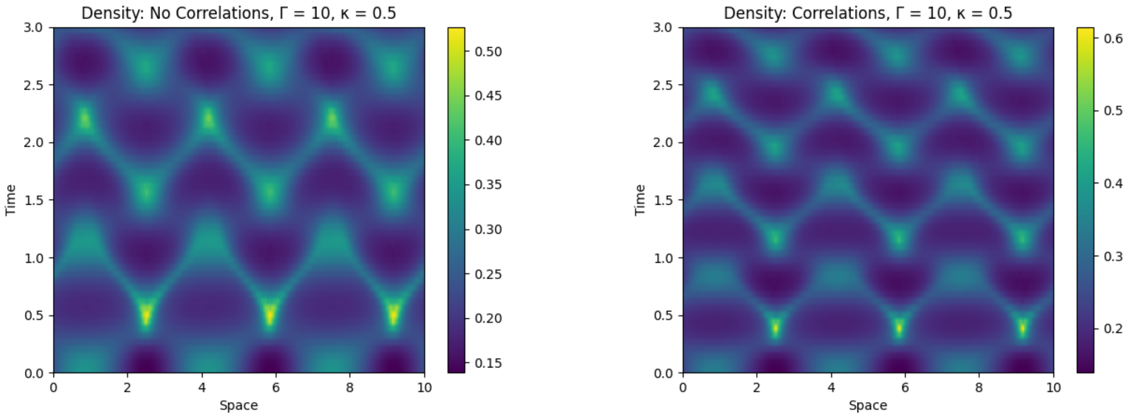


Figure 6.2: Colormaps of density evolution with excess correlations throughout the (Γ, κ) parameter space is shown above. Γ is chosen $\{\frac{1}{10}, 1, 10\}$ with increasing values as bottom to top, and κ is chosen $\{\frac{1}{2}, 1, 2\}$ with increasing values as left to right.

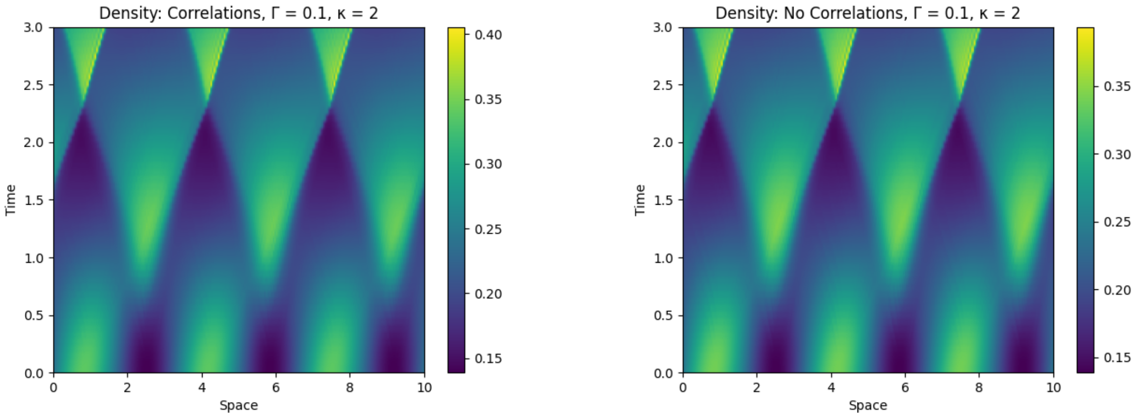
6.3 Effects of the DCF

This section aims to compare two regions of the parameter space, namely the one with large Γ and small κ, and the other with small Γ and large κ, with respect to their plasma behavior. From the simulations, we see that strongly-correlated plasmas deviate the most from non-correlated behavior in the former region, while the latter region exhibits the least deviation. We show the density evolution in these two regimes, with and without correlation effects in Tables 6.3 and 6.4. For each simulation, a sinusoidal initial condition was used. Once again, the addition of the correlations seems to enhance the spatial structure of the standing wave oscillations. Furthermore, the correlations appear to shorten the period of oscillations as well.


Density: No Correlations, Γ = 10, κ = 0.5

Density: Correlations, Γ = 10, κ = 0.5

Table 6.3: The colormaps of the density for $\Gamma = 10$ and $\kappa = \frac{1}{2}$ without (left) and with (right) correlation effects are compared above.


Density: Correlations, Γ = 0.1, κ = 2

Density: No Correlations, Γ = 0.1, κ = 2

Table 6.4: The colormaps of the density for $\Gamma = \frac{1}{10}$ and $\kappa = 2$ without (left) and with (right) correlation effects are compared above.

We compared these two regions of the parameter space by computing the relative density, which is defined as the difference between the density with and without correlations divided by the mean density. The colormap of the relative density (as seen in Table 6.5) reveal that the largest deviations occur in the region of large Γ and small κ, while the smallest deviations occur in the region of small Γ and large κ.

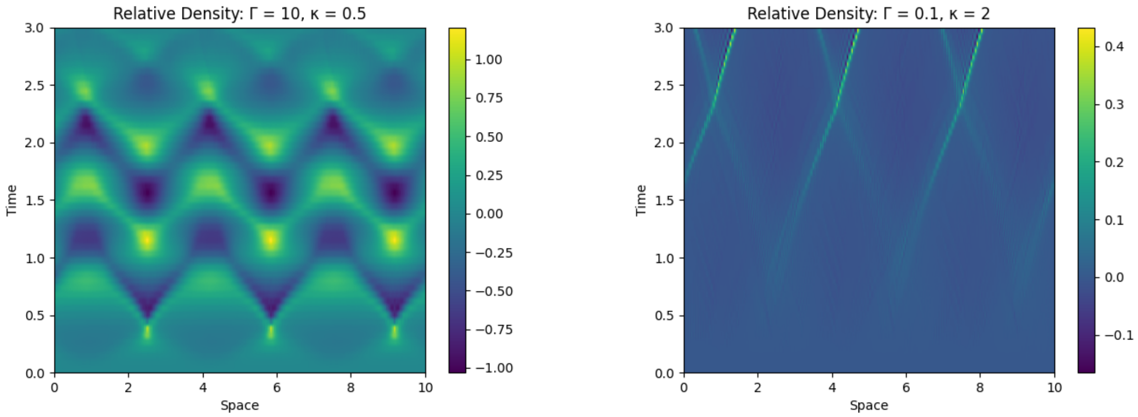


Table 6.5: The colormaps of the relative density for $\Gamma = 10$ and $\kappa = \frac{1}{2}$ (left) and $\Gamma = .1$ and $\kappa = 2$ (right) are compared above. There is a larger deviation between the systems with and without excess correlations for strongly-coupled plasmas (left) than wealky-coupled plasmas (right).

Based on the relative density colormaps, we observe that the deviation between the behavior of strongly correlated plasmas throughout the (Γ, κ) parameter space is consistent with the expected theory. Furthermore, the figures show that the relative density indicate a significant deviation in the strongly correlated regime. Specifically, we observe a large deviation in behavior in the strongly correlated regime and little deviation in the weakly correlated regime, as expected.

CHAPTER 7

CONCLUSION & OUTLOOK

This chapter summarizes the results of the thesis as well as where more research can be done.

7.1 Summary of Results

In this thesis, we have expanded the use of Hydrodynamic Density Functional Theory for modeling strongly-correlated plasmas. By first considering a system of particles interacting through classical Newtonian mechanics, we derived the Liouville Equation, which describes the time evolution of the N-body probability distribution function. We then successively integrated out degrees of freedom, which created the BBGKY hierarchy, until we were left with the evolution of the one-body distrubtion function (BY-1). Centralized statistical velocity moments of this equation were taken to derive the evolution of the density, momentum and stress of the one-body distribution function.

Taking moments of BY-1 generated its own hierarchy of equations, and we derived closures for these equations by comparing them to the equilibrium behavior of the fluids (through Maxwellian distributions). The stress tensor evolution equation was replaced by a temperature equation, which was closed through the implementation of Fourier's Law. Additionally, we used classical Density Functional Theory (DFT) to quantify plasma correlations that were not included in the original evolution equations. In particular, the free energy (which describes the available work in a closed, isothermal system) was fully described with the help of a correlation expansion. Various models were explored to describe the direct correlation function for modeling the correlation physics in the governing equations. A cluster expansion was chosen to approximate the direct correlation function.

Once the model was closed, the dynamics were determined by three parameters $(\Gamma, \kappa, \lambda)$ through nondimensionalization. We then performed a linear stability analysis, which generated an approximate linear model. We used dispersion relations to find determine the qualitative nature of the solutions. The temperature evolution was found to be decoupled from the density and momentum dynamics at linear order.

We then implemented a numerical solution of the model using a finite volume scheme to discretize the governing equations. We simulated the system with and without the correlation contributions for a variety of initial conditions, including Gaussian, sinusoidal, and constant functions. We sampled the parameter space, where $\Gamma \in \left\{\frac{1}{10}, 1, 10\right\}$ and $\kappa \in \left\{\frac{1}{2}, 1, 2\right\}$, to study the plasma behavior as coupling and screening effects increase. We found that strongly-correlated plasmas had the largest deviation from non-correlated behavior for large Γ and small κ and the smallest deviation for small Γ and large κ. We used the relative density to quantify the deviation between plasmas with and without correlations and found that the deviation was significant in the strongly-correlated regime, as expected. Overall, our results were consistent with the expected theory, and we gained insight into the behavior of strongly-correlated plasmas through our modeling and simulations.

7.2 What's Next?

There are several areas where more research can be done, including implementing different correlation models, analyzing the mode-coupling physics of the nonlinearities and extending the theory to mixtures. Many choices were made with how to model the correlations, and other decisions can be explored in future research. We may consider expanding the model to account for more than two-body correlation effects. Additionally, other correlation models (besides cluster expansions and the Mean Field approximation) can be compared and contrasted with each other. On a more fundamental level, we may consider non-homogeneous direct correlation functions. The simulation plots alluded to possible mode-coupling of the nonlinear waves. This coupling can be analyzed through perturbation theory in future research. Lastly, we assumed a single species fluid throughout this thesis. However, many areas of interest in fluid mechanics deal with mixtures. This research can be extended to consider how a mixture of fluids interact.

BIBLIOGRAPHY

[1] A. J. Archer, “Dynamical density functional theory for molecular and colloidal fluids: A microscopic approach to fluid mechanics,” *The Journal of chemical physics*, vol. 130, no. 1, p. 014509, 2009.

[2] A. Diaw and M. S. Murillo, “Generalized hydrodynamics model for strongly coupled plasmas,” *Physical Review E*, vol. 92, no. 1, p. 013107, 2015.

[3] W. Commons, “File:riemann problem.jpg — wikimedia commons, the free media repository,” 2020. [Online; accessed 21-May-2023].

[4] W. Kohn and L. J. Sham, “Self-consistent equations including exchange and correlation effects,” *Physical review*, vol. 140, no. 4A, p. A1133, 1965.

[5] E. Fermi, “Un metodo statistico per la determinazione di alcune priorieta dellatome,” *Rend. Accad. Naz. Lincei*, vol. 6, no. 602-607, p. 32, 1927.

[6] D. R. Hartree, “The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, pp. 89–110, Cambridge university press, 1928.

[7] P. Hohenberg and W. Kohn, “Inhomogeneous electron gas,” *Physical review*, vol. 136, no. 3B, p. B864, 1964.

[8] C. Ebner, W. Saam, and D. Stroud, “Density-functional theory of simple classical fluids. i. surfaces,” *Physical Review A*, vol. 14, no. 6, p. 2264, 1976.

[9] R. Evans, “The nature of the liquid-vapour interface and other topics in the statistical mechanics of non-uniform, classical fluids,” *Advances in physics*, vol. 28, no. 2, pp. 143–200, 1979.

[10] H. v. Helmholtz, “On the thermodynamics of chemical processes,” *Physical Memoirs*, vol. 1, pp. 121–157, 1882.

[11] J.-P. Hansen and I. R. McDonald, *Theory of simple liquids: with applications to soft matter.* Academic press, 2013.

[12] J. E. Mayer and E. Montroll, “Molecular distribution,” *The Journal of Chemical Physics*, vol. 9, no. 1, pp. 2–16, 1941.

[13] J. P. Perdew, R. G. Parr, M. Levy, and J. L. Balduz Jr, “Density-functional theory for fractional particle number: Derivative discontinuities of the energy,” *Physical Review Letters*, vol. 49, no. 23, p. 1691, 1982.

[14] U. M. B. Marconi and P. Tarazona, “Dynamic density functional theory of fluids,” *The Journal of chemical physics*, vol. 110, no. 16, pp. 8032–8044, 1999.

[15] U. M. B. Marconi and S. Melchionna, “Dynamic density functional theory versus kinetic theory of simple fluids,” *Journal of Physics: Condensed Matter*, vol. 22, no. 36, p. 364110, 2010.

[16] H. Löwen, “Density functional theory: from statics to dynamics,” *Journal of Physics: Condensed Matter*, vol. 15, pp. V1–V3, 2003.

[17] A. J. Archer and R. Evans, “Dynamical density functional theory and its application to spinodal decomposition,” *The Journal of chemical physics*, vol. 121, no. 9, pp. 4246–4254, 2004.

[18] A. J. Archer, R. Evans, and M. Schmidt, “Dynamical density functional theory,” *Journal of Physics: Condensed Matter*, vol. 16, no. 45, pp. S5291–S5316, 2004.

[19] B. Clapeyron, “Mémoire sur la théorie de la chaleur,” *Journal de l’École Polytechnique*, vol. 14, pp. 116–131, 1831.

[20] L. Euler, “Principes généraux du mouvement des fluides,” *Mémoires de l’académie des sciences de Berlin*, vol. 11, pp. 274–315, 1757.

[21] C. L. Navier, “Mémoire sur les lois du mouvement des fluides,” *Ann. Chim. Phys.*, vol. 19, p. 244, 1821.

[22] G. G. Stokes, “On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids,” in *Mathematical and Physical Papers*, pp. 75–129, Cambridge University Press, 1880.

[23] R. Evans, “Density functionals in the theory of non-uniform fluids,” in *Fundamentals of Inhomogeneous Fluids*, pp. 85–175, Marcel Dekker, 1992.

[24] R. L. Liboff, *Kinetic theory: classical, quantum, and relativistic descriptions.* Springer Science & Business Media, 2003.

[25] N. N. Bogoliubov, “Kinetic equations,” *Journal of Physics USSR*, vol. 10, no. 3, pp. 265–274, 1946.

[26] J. Yvon, *La théorie statistique des fluides et l’équation d’état*, vol. 203. Hermann & cie, 1935.

[27] J. G. Kirkwood, "The statistical mechanical theory of transport processes i. general theory," *The Journal of Chemical Physics*, vol. 14, no. 3, pp. 180–201, 1946.

[28] J. G. Kirkwood, "The statistical mechanical theory of transport processes ii. transport in gases," *The Journal of Chemical Physics*, vol. 15, no. 1, pp. 72–76, 1947.

[29] M. Born and H. S. Green, "A general kinetic theory of liquids i. the molecular distribution functions," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 188, no. 1012, pp. 10–18, 1946.

[30] A. Vlasov, "Theory of vibrational properties of electron gas and its applications," *Uch. Rec. MSU*, 1945.

[31] A. A. Vlasov, "The vibrational properties of an electron gas," *Soviet Physics Uspekhi*, vol. 10, p. 721, jun 1968.

[32] A. D. Fokker, "Die mittlere energie rotierender elektrischer dipole im strahlungsfeld," *Annalen der Physik*, vol. 43, no. 14, p. 912, 1914.

[33] M. Planck, "Die naturwissenschaften und die erkenntnistheorie," *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, p. 324, 1917.

[34] A. Lenard, "A plasma theory of terrestrial radiation belts," *Annalen der Physik*, vol. 10, no. 5, p. 390, 1960.

[35] R. Balescu, "Equilibrium and nonequilibrium statistical mechanics," *Physics of Fluids*, vol. 3, no. 1, p. 52, 1960.

[36] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems," *Physical review*, vol. 94, no. 3, p. 511, 1954.

[37] K. S. Singwi, M. P. Tosi, R. H. Land, and A. Sjölander, "Electron correlations at metallic densities," *Phys. Rev.*, vol. 176, pp. 589–599, Dec 1968.

[38] T. Ramakrishnan and M. Yussouff, "First-principles order-parameter theory of freezing," *Physical Review B*, vol. 19, no. 5, p. 2775, 1979.

[39] E. E. Salpeter, "On mayer's theory of cluster expansions," *Annals of Physics*, vol. 5, no. 3, pp. 183–223, 1958.

[40] P. Debye, "The theory of electrolytes. i. lowering of freezing point and related phenomena," *Physikalische Zeitschrift*, vol. 24, pp. 185–206, 1923.

[41] L. S. Ornstein, “Accidental deviations of density and opalescence at the critical point of a single substance,” *Proc. Akad. Sci.*, vol. 17, p. 793, 1914.

[42] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, vol. 31. Cambridge university press, 2002.

[43] P. L. Roe, “Approximate riemann solvers, parameter vectors, and difference schemes,” *Journal of computational physics*, vol. 43, no. 2, pp. 357–372, 1981.

[44] M.-S. Liou and C. J. Steffen Jr, “A new flux splitting scheme,” *Journal of Computational physics*, vol. 107, no. 1, pp. 23–39, 1993.

[45] A. Harten, P. D. Lax, and B. v. Leer, “On upstream differencing and godunov-type schemes for hyperbolic conservation laws,” *SIAM review*, vol. 25, no. 1, pp. 35–61, 1983.

[46] L. Boltzmann, “Studien über das gleichgewicht der lebendigen kraft zwischen bewegten materiellen punkten,” *Wiener Berichte*, vol. 58, pp. 517–560, 1868.

APPENDIX A

APPENDIX

A.1 Comparing HDFT to Known Results

Hydrodynamic density functional theory can recover several known results from simpler models.

We may consider a model that ignores all correlations and external forces. Then the free energy can be written as

$$\mathscr{F} \approx \mathscr{F}_{\mathrm{id}}. \tag{A.1}$$

Then Equation (2.63) simplifies to

$$\frac{1}{n}\nabla p = k_B T \nabla \log(n) \tag{A.2}$$

Solving for p in the limit where temperature is constant yields

$$p = n k_B T \tag{A.3}$$

which is the Ideal Gas Law [19]. The Ideal Gas Law approximates the behavior of a gas as a system of spherical particles that collide perfectly elastically. This behavior is to be expected in the case where no correlations or external forces are expected.

As mentioned above, the mean field approach attempts to capture the average behavior of all the particle interactions. We may also consider a model that includes the mean field approximation but no external forces. Then the free energy can be written as

$$\mathscr{F} \approx \mathscr{F}_{\text{id}} + \mathscr{F}_{\text{ex}}, \tag{A.4}$$

and from Equation (2.101), we get

$$\mu = \; k_B T \log(\Lambda^3 n) + \varphi(\mathbf{r} - \mathbf{r}'). \tag{A.5}$$

Substituting in the electric field found from Equation (3.11), we get

$$\mu = \; k_B T \log(\Lambda^3 n) + q\Phi. \tag{A.6}$$

We can now solve for n:

$$n = n_0 e^{-\beta q \Phi}. \tag{A.7}$$

where

$$n_0 = \frac{e^{\beta\mu}}{\Lambda^3} \tag{A.8}$$

is the fugacity. The Boltzmann Distribution [46] is recovered in Equation (A.7). The Boltzmann Distribution is a probability distribution function that describes the probability of a system being in a particular state. It states that the probability of a system being in a certain state is proportional to the the inverted exponential of the energy of the state times the temperature.

A.2 Simulation Code

This section details the python script that was used to simulate the evolution equations.

```
import numpy as np

import matplotlib.pyplot as plt

import time


# Space Domain

Xpts = int(2e2)

Xlngth = 10

dx = Xlngth / Xpts

X = np.linspace(0, Xlngth - dx, Xpts)


# Time Domain

Tpts = int(1.2e3) + 1

Tlngth = 1

T = np.linspace(0, Tlngth, Tpts)[:-1]

dt = Tlngth / (Tpts - 1)


lmbd = dt / dx


# Snapshots

snaps = 100 + 1

cursnap = 0

Y = np.linspace(0, Tlngth, snaps)
```

```
xx, yy = np.meshgrid(X, Y, sparse=False, indexing='xy')

# Spatial Frequency Domain
k_fft_norm = 2 * np.pi / (Xpts * dx)
k = k_fft_norm * np.linspace(-Xpts / 2, Xpts / 2 - 1, Xpts)

# Parameters
mean_n = 3 / (4 * np.pi)
Gamma_0 = 1
kappa_0 = 1
therm_cond = 1
q = 1

# Integration Domain/Extensions
num_dom_ext = 5
Xppts = int(num_dom_ext * Xpts)
xp_min = -int(num_dom_ext/2)
xp_max = int(num_dom_ext/2 + 1)
Xpts_ext = int(num_dom_ext * Xpts)
Xp_ar = np.linspace(xp_min, xp_max, Xpts_ext)
rhop_min = 0
```

```
rhop_max = int(np.sqrt(2) * int(num_dom_ext/2 + 1))  # TODO:
    Check max(sqrt(y^2+z^2)) = sqrt(2)*max(y')?
Rhopts_ext = int(num_dom_ext * Xpts)
Rhop_ar = np.linspace(xp_min, xp_max, Xpts_ext)
drhop = xp_max/(Xpts_ext-1)
Xp, Rhop = np.meshgrid(Xp_ar, Rhop_ar)
Rp = np.sqrt(Xp**2 + Rhop**2)  # Define |r'| = sqrt(rho'^2 + x
    '^2)

u = q**2 / Rp * np.exp(-kappa_0*Rp) # Define Potential u

# Memory Allocation
n, ntot, nint, nflux, nfluxtot = np.zeros((2, Xpts)), np.zeros((
    snaps, 2, Xpts)), np.zeros((snaps, 2)), \
                                 np.zeros((2, Xpts)), np.zeros((
                                     snaps, 2, Xpts))
v, vtot, vint, vflux, vfluxtot = np.zeros((2, Xpts)), np.zeros((
    snaps, 2, Xpts)), np.zeros((snaps, 2)), \
                                 np.zeros((2, Xpts)), np.zeros((
                                     snaps, 2, Xpts))
e, etot, eint, eflux, efluxtot = np.zeros((2, Xpts)), np.zeros((
    snaps, 2, Xpts)), np.zeros((snaps, 2)), \
                              np.zeros((2, Xpts)), np.zeros((
                                snaps, 2, Xpts))
phi, phimtx, phitot = np.zeros((2, Xpts)), np.zeros((2, Xpts)),
   np.zeros((snaps, 2, Xpts))


# Memory Allocation - Correlations
# TODO: Dynamic to Static Memory


# Initial Condition
ICchoice = np.array([1, 1, 1])  # Gaussian = 0, Wave = 1, Random
   = 2, Constant = 3, Zero = 4
perturb_ampltd = [.01, .01, .01]
gaus_lngth = .05 * Xlngth
IC_freq = 2 * np.pi / Xlngth
ICdict = np.array([
    np.array([
        mean_n * np.ones(Xpts) + perturb_ampltd[0] * np.exp(-((X
           - Xlngth/2) ** 2) / (2 * gaus_lngth ** 2)) -
           perturb_ampltd[0] * np.sqrt(2 * np.pi) * gaus_lngth,
        mean_n * np.ones(Xpts) + perturb_ampltd[0] * np.cos(3 *
           IC_freq * X),
        mean_n * np.ones(Xpts) + perturb_ampltd[0] * np.random.
            random(Xpts),
        mean_n * np.ones(Xpts),
        np.zeros(Xpts)
    ]),
    np.array([
        np.ones(Xpts) + perturb_ampltd[1] * np.exp(-((X - Xlngth
            /2) ** 2) / (2 * gaus_lngth ** 2)) - perturb_ampltd[1]
             * np.sqrt(2 * np.pi) * gaus_lngth,
        np.zeros(Xpts) + perturb_ampltd[1] * np.cos(3 * IC_freq *
             X),
        np.zeros(Xpts) + perturb_ampltd[1] * np.random.random(
            Xpts),
        np.ones(Xpts),
        np.zeros(Xpts)
    ]),
    np.array([
        np.ones(Xpts) + perturb_ampltd[2] * np.exp(-((X - Xlngth
            /2) ** 2) / (2 * gaus_lngth ** 2)) - perturb_ampltd[2]
             * np.sqrt(2 * np.pi) * gaus_lngth,
        np.zeros(Xpts) + perturb_ampltd[2] * np.cos(3 * IC_freq *
             X),
        np.zeros(Xpts) + perturb_ampltd[2] * np.random.random(
            Xpts),
        np.ones(Xpts),
        np.zeros(Xpts)
    ])
])

n[:] = ICdict[0, ICchoice[0]]
v[:] = ICdict[1, ICchoice[1]]
e[:] = ICdict[2, ICchoice[2]]



# Shift to Left
def l(array): # Has Been Checked
    return np.roll(array, 1, axis=-1)



# Shift to Right
def r(array): # Has Been Checked
    return np.roll(array, -1, axis=-1)
```

```
# Derivative
def ddx(array):  # Has Been Checked
    return (array - l(array)) / dx



# Second Derivative
def d2dx2(array):  # Has Been Checked
    return (r(array) - 2 * array + l(array)) / (dx * dx)



# Check Nans
def checknan(array, name, time):
    if np.isnan(array.any()):
        print(”Nan␣value␣at␣” + name + ”␣at␣tt␣=␣” + str(time))
        return True
    return False



# Take Snapshot
def savesnap(cursnap):
    ntot[cursnap] = n
    vtot[cursnap] = v
    etot[cursnap] = e
    phitot[cursnap] = phi
    nfluxtot[cursnap] = nflux
    vfluxtot[cursnap] = vflux
    efluxtot[cursnap] = eflux
    nint[cursnap] = np.trapz(n, X)
    vint[cursnap] = np.trapz(v, X)
    eint[cursnap] = np.trapz(e, X)


# Solve Phi
def solvephi(den):
    phi = np.zeros((2, Xpts))
    # phimtx * phi = b
    # Define b
    b = 3 - 4 * np.pi * dx * dx * den
    b = b - np.mean(b)
    # First sweep
    phimtx[:, 0] = -0.5
    b[:, 0] = -0.5 * b[:, 0]
    for ii in range(1, Xpts - 1):
        phimtx[:, ii] = -1 / (2 + phimtx[:, ii - 1])
```

```
            b[:, ii] = (b[:, ii - 1] - b[:, ii]) / (2 + phimtx[:, ii - 1])
        # Second sweep
        phi[:, 0] = b[:, Xpts - 1] - b[:, Xpts - 2]
        for ii in range(1, Xpts - 1):  # TODO: CHECK INDICES
            phi[:, ii] = (b[:, ii - 1] - phi[:, ii - 1]) / phimtx[:, ii - 1]

        return phi


def meyer_correlations(den, temp):
    # Define c(|r'|)
    e_ext = np.tile(temp, num_dom_ext)
    c = np.exp(-u / e_ext[1]) - 1 + u / e_ext[1]
    c_fft = np.fft.fftn(c)

    # Integrate rho'*F(c(r')) drho'
    rho_integrand = Rhop * c_fft
    c_int = np.trapz(rho_integrand, axis=1, dx=drhop)

    # n(x-x') - mean_n
    delta_n_xp = np.tile(den, num_dom_ext) - mean_n
    delta_n_fft = np.fft.fft(delta_n_xp)

    corr_fft = c_int * delta_n_fft
    corr = np.fft.ifftn(corr_fft)
    corr = corr.real[int(num_dom_ext/2)*Xpts:Xppts-int(
        num_dom_ext/2)*Xpts]
    return corr



start = time.time()
for tt in range(Tpts):

    phi = solvephi(n)

    # Flux
    nflux = dt * ddx(n * v)
    vflux = dt * (ddx(.5*v*v) + e*ddx(n)/n - Gamma_0*ddx(phi))
    eflux = dt * (- therm_cond*d2dx2(e) - therm_cond*ddx(n)/n*ddx
        (e) + v*e*ddx(n)/n + 2*v*ddx(e))

    # Correlations
```

```
vcorr = np.zeros((2, Xpts))
vcorr[1] = dt*e[1]*ddx(meyer_correlations(n[1], e[1]))


# Store Values
if (tt % int(Tpts / (snaps - 1))) == 0:
    savesnap(cursnap)
    cursnap += 1


# Solve
n = n - nflux
v = v - vflux - vcorr
e = e - eflux


# Check Nans
ncheck, vcheck, echeck = checknan(n, ”n”, tt), checknan(v, ”n
   ”, tt), checknan(e, ”n”, tt)
nfluxcheck, vfluxcheck, efluxcheck = checknan(nflux, ”n”, tt)
   , checknan(vflux, ”n”, tt), checknan(eflux, ”n”, tt)
if ncheck or vcheck or echeck or nfluxcheck or vfluxcheck or
   efluxcheck:
    exit()
    # Track Progress
    if tt % int(Tpts / 10) == 0:
        print(str(round(100 * tt / Tpts)) + "%␣Done")
end = time.time()



def data_visualize(*args):  # Has Been Checked
    sys_data_to_visualize = np.array([np.swapaxes(array, 0, 1)
       for array in args])
    return sys_data_to_visualize



def data_names(var_name, *args):
    cornames = ["NC", "C"]
    if var_name.strip():
        var_name = "␣" + var_name
    names = np.array([np.array([name + var_name + "␣" + corname
       for corname in cornames]) for name in args])
    return names



# Reformat data for plotting
```

```
syssnap = data_visualize(ntot, vtot, etot, phitot) # Shape: (4,
    2, Tpts, Xpts)
syssnapflux = data_visualize(nfluxtot, vfluxtot, efluxtot) #
    Shape: (3, 2, Tpts, Xpts)
syssnapint = data_visualize(nint, vint, eint) # Shape: (3, 2,
    Tpts)

syssnapname = data_names("", "Density", "Velocity", "Temperature"
    , "Electrostatic␣Potential")
syssnapfluxname = data_names("Flux", "Density", "Velocity", "
    Temperature")
syssnapintname = data_names("Integral", "Density", "Velocity", "
    Temperature")



# Plotting
def plot_snaps(data, name, xaxis, yaxis):
    fig = plt.figure()
    ax = fig.add_subplot(1, 1, 1)
    for kk in range(len(data)):
        ax.plot(X, data[kk], label="T␣=␣" + str(round(kk * dt *
            Tlngth / snaps, 2)))
    ax.set_title(name + "␣  0 ␣=␣" + str(Gamma_0) + "␣  _0 ␣=␣" +
        str(kappa_0))
    ax.set_xlabel(xaxis)
    ax.set_ylabel(yaxis)
    ax.legend()

    return ax



syssnap_eq, syssnap_cor, syssnap_time, syssnap_lngth = syssnap.
    shape
syssnapflux_eq, syssnapflux_cor, syssnapflux_time,
    syssnapflux_lngth = syssnapflux.shape
syssnapint_eq, syssnapint_cor, syssnapint_time = syssnapint.shape

# Plot Fields
# syssnapplots = [[0 for ii in range(syssnap_cor)] for jj in
    range(syssnap_eq)]
# for ii in range(syssnap_eq):
#     for jj in range(syssnap_cor):
#         syssnapplots[ii][jj] = plot_snaps(syssnap[ii, jj],
    syssnapname[ii, jj], "Space", syssnapname[ii, jj])
```

```
# Plot Fluxes
# syssnapfluxplots = []
# for ii in range(syssnapflux_eq):
#     for jj in range(syssnapflux_cor):
#         plot_snaps(syssnapflux[ii, jj], syssnapfluxname[ii, jj
  ], "Space", syssnapfluxname[ii, jj])



# Subplot: Individual Fields
def subplot(data, name, xaxis, title):
    row, col, plot, _ = data.shape
    fig, axs = plt.subplots(row, col)
    for ii in range(row):
        for jj in range(col):
            for kk in range(plot):
                axs[ii, jj].plot(X, data[ii, jj, kk], label="T␣=␣
                    " + str(round(kk * dt * Tlngth / snaps, 2)))
            axs[ii, jj].set_title(name[ii, jj] + "␣ 0 ␣=␣" + str(
              Gamma_0) + "␣ _0 ␣=␣" + str(kappa_0))
            axs[ii, jj].set_xlabel(xaxis[ii][jj])
            axs[ii, jj].set_ylabel(name[ii][jj])
    plt.subplots_adjust(hspace=0.5)
    fig.suptitle(title)


# subplot(syssnap, syssnapname, [["Space" for jj in range(
    syssnap_cor)] for ii in range(syssnap_eq)], "Fields")

# subplot(syssnapflux, syssnapfluxname, [["Space" for jj in range
    (syssnapflux_cor)] for ii in range(syssnapflux_eq)], "Flux
    Fields")


# Subplots: Compile Plots Using Axes
def subplot_plots(plots, title):
    row = len(plots)
    col = max(len(plots[ii]) for ii in range(row))
    fig, axs = plt.subplots(row, col)
    for ii in range(row):
        for jj in range(col):
            axs[ii, jj] = plots[ii][jj]
    plt.subplots_adjust(hspace=0.5)
    fig.suptitle(title)
```

```
# subplot_plots(syssnapplots, ”Fields”)

# subplot_plots(syssnapfluxplots, ”Flux Fields”)


# Subplot: Color Plots
def subplot_imshow(data, name, xaxis, title):
    row, col, plot, _ = data.shape
    fig, axs = plt.subplots(row, col)
    for ii in range(row):
        for jj in range(col):
            for kk in range(plot):
                axs[ii, jj].plot(X, data[ii, jj, kk], label=”T␣=␣
                    ” + str(round(kk * dt * Tlngth / snaps, 2)))
            axs[ii, jj].set_title(name[ii, jj] + ”␣  0 ␣=␣” + str(
                Gamma_0) + ”␣  _0 ␣=␣” + str(kappa_0))
            axs[ii, jj].set_xlabel(xaxis[ii][jj])
            axs[ii, jj].set_ylabel(name[ii][jj])
    plt.subplots_adjust(hspace=0.5)
    fig.suptitle(title)
```

```
# Plot Integrals, Ensure Conservation
def plot_conserved(data, name):
    plt.figure()
    plt.plot(np.arange(snaps), data)
    plt.title(name + "␣ ␣=␣" + str(Gamma_0) + "␣ ␣=␣" + str(
        kappa_0))
    plt.xlabel("Time")
    plt.ylabel(name)



# Plots: Conservation of Fields
# for ii in range(len(syssnapint)):
#     for jj in range(len(syssnapint[ii])):
#         plot_conserved(syssnapint[ii, jj], syssnapintname[ii,
    jj])

# Plots: Integral Difference of Fields Over Time
# for ii in range(len(syssnapint)):
#     plot_conserved(syssnapint[ii, 0] - syssnapint[ii, 1], "
    Integral Difference")
```

```
# subplot(syssnapint, syssnapintname, [["Time" for jj in range(
    syssnapint_cor)] for ii in range(syssnapint_eq)])



# ImShow
def imshow(data, name, xaxis, yaxis):
    plt.figure()
    clr = plt.imshow(data, aspect='auto', origin='lower', extent
        =(0, Xlngth, 0, Tlngth))
    plt.colorbar()
    plt.title(name + "␣ ␣=␣" + str(Gamma_0) + "␣ ␣=␣" + str(
        kappa_0))
    plt.xlabel(xaxis)
    plt.ylabel(yaxis)



# Color Plots: Fields
for ii in range(len(syssnap)):
    for jj in range(len(syssnap[ii])):
        imshow(syssnap[ii, jj], syssnapname[ii, jj], "Space", "
            Time")
```

```
# Color Plots: Fields
for ii in range(len(syssnapflux)):
    for jj in range(len(syssnapflux[ii])):
        imshow(syssnapflux[ii, jj], syssnapfluxname[ii, jj], "
            Space", "Time")

# Color Plots: Differences
for ii in range(len(syssnap)):
    imshow(syssnap[ii, 0] - syssnap[ii, 1], "Difference", "Space"
        , "Time")

# FFT
nsnap = syssnap[0]
for ii in range(len(nsnap)):
    n_fft = nsnap[ii] - np.mean(nsnap[ii])
    n_fft = np.swapaxes(n_fft, 0, 1)

    n_fft_flip = np.flip(n_fft, axis=1)
    n_fft = np.hstack((n_fft, n_fft_flip))

    fft = np.fft.fft2(n_fft, axes=(0, 1))
    fft = np.fft.fftshift(fft, axes=(0, 1))
    fft = np.transpose(fft)

    # Reflect  # TODO: If fft axis i length odd/even -> Add 1 in
        index -> reflect1[:, :int(fftlength1 / 2)+1]
    fftlength0, fftlength1 = fft.shape
    fft1_1st_half, fft1_2nd_half = fft[:int(fftlength0 / 2)], fft
        [int(fftlength0 / 2):]
    reflect1 = (np.flip(fft1_1st_half, axis=0) + fft1_2nd_half) /
        2  # Reflect Bottom -> Up
    reflect1_1st_half, reflect1_2nd_half = reflect1[:, :int(
        fftlength1 / 2)], reflect1[:, int(fftlength1 / 2):]
    fft_avg = (np.flip(reflect1_1st_half, axis=1) +
        reflect1_2nd_half) / 2  # Reflect Left -> Right

    # Plot FFT
    plt.figure()
    plt.imshow(np.abs(fft_avg), aspect='auto', origin='lower',
        extent=(-0.5, snaps - 0.5, -0.5, Xpts - 0.5))
    plt.colorbar()
    plt.xlabel("Spatial␣Frequency␣(k)")
    plt.ylabel("Dispersion␣(   )")
```

```
    plt.title("FFT:    0 = " + str(Gamma_0) + "    0 = " + str(
        kappa_0))

end = time.time()
print("Total Time: ", end - start, " seconds")
# print("Off By:", np.round(100*(end - start - (elapsed_time * T
    / 100))/(end - start), 2), " %")

plt.show()

if __name__ == "__main__":
    print()
```