

DISSERTATION

COMPUTATIONAL MODELING OF LOW-DENSITY ULTRACOLD PLASMAS

Submitted by

Craig Witte

Department of Physics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2017

Doctoral Committee:

Advisor: Jacob L. Roberts

Richard Eykholt

David Krueger

Justin Sambur

Copyright by Craig Witte 2017

All Rights Reserved

## ABSTRACT

### COMPUTATIONAL MODELING OF LOW-DENSITY ULTRACOLD PLASMAS

In this dissertation I describe a number of different computational investigations which I have undertaken during my time at Colorado State University. Perhaps the most significant of my accomplishments was the development of a general molecular dynamic model that simulates a wide variety of physical phenomena in ultracold plasmas (UCPs). This model formed the basis of most of the numerical investigations discussed in this thesis. The model utilized the massively parallel architecture of GPUs to achieve significant computing speed increases (up to 2 orders of magnitude) above traditional single core computing. This increased computing power allowed for each particle in an actual UCP experimental system to be explicitly modeled in simulations.

By using this model, I was able to undertake a number of theoretical investigations into ultracold plasma systems. Chief among these was our labs investigation of electron center-of-mass damping, in which the molecular dynamics model was an essential tool in interpreting the results of the experiment. Originally, it was assumed that this damping would solely be a function of electron-ion collisions. However, the model was able to identify an additional collisionless damping mechanism that was determined to be significant in the first iteration of our experiment. To mitigate this collisionless damping, the model was used to find a new parameter range where this mechanism was negligible. In this new parameter range, the model was an integral part in verifying the achievement of a record low measured UCP electron temperature of  $1.57 \pm 0.28K$  and a record high electron strong coupling parameter,  $\Gamma$ , of  $0.35 \pm 0.08$ . Additionally, the model, along with experimental measurements, was used to verify the breakdown of the standard weak coupling approximation for Coulomb collisions.

The general molecular dynamics model was also used in other contexts. These included the modeling of both the formation process of ultracold plasmas and the thermalization of the electron component of an ultracold plasma. Our modeling of UCP formation is still in

its infancy, and there is still much outstanding work. However, we have already discovered a previously unreported electron heating mechanism that arises from an external electric field being applied during UCP formation. Thermalization modeling showed that the ion density distribution plays a role in the thermalization of electrons in ultracold plasma, a consideration not typically included in plasma modeling. A Gaussian ion density distribution was shown to lead to a slightly faster electron thermalization rate than an equivalent uniform ion density distribution as a result of collisionless effects. Three distinct phases of UCP electron thermalization during formation were identified.

Finally, the dissertation will describe additional computational investigations that preceded the general molecular dynamics model. These include simulations of ultracold plasma ion expansion driven by non-neutrality, as well as an investigation into electron evaporation. To test the effects of non-neutrality on ion expansion, a numerical model was developed that used the King model of the electron to describe the electron distribution for an arbitrary charge imbalance. The model found that increased non-neutrality of the plasma led to the rapid expansion of ions on the plasma exterior, which in turn led to a sharp ion cliff-like spatial structure. Additionally, this rapid expansion led to additional cooling of the electron component of the plasma. The evaporation modeling was used to test the underlying assumptions of previously developed analytical expression for charged particle evaporation. The model used Monte Carlo techniques to simulate the collisions and the evaporation process. The model found that neither of the underlying assumption of the charged particle evaporation expressions held true for typical ultracold plasma parameters and provides a route for computations in spite of the breakdown of these two typical assumptions.

# TABLE OF CONTENTS

Abstract . . . . .	ii
1. Introduction . . . . .	1
1.1 What is a Plasma? . . . . .	2
1.2 The Plasma Parameter Space . . . . .	3
1.2.1 Low Temperature Plasmas . . . . .	5
1.2.2 Astrophysical Plasmas . . . . .	5
1.2.3 Fusion Plasmas . . . . .	6
1.2.4 Warm Dense Matter . . . . .	7
1.2.5 Dusty Plasmas . . . . .	7
1.3 Ultracold Plasmas . . . . .	8
1.3.1 Creation and Lifecycle . . . . .	8
1.3.2 Detection Methods . . . . .	11
1.3.3 Tests of Fundamental Physics . . . . .	14
1.3.4 Strong Coupling Physics . . . . .	16
1.4 Overview of the Thesis . . . . .	19
1.4.1 Chapter 2: Computational Techniques . . . . .	19
1.4.2 Chapter 3: Electron Oscillation Damping . . . . .	20
1.4.3 Chapter 4: Formation Physics . . . . .	20
1.4.4 Chapter 5: Thermalization . . . . .	21
1.4.5 Chapter 6: Evaporation . . . . .	22
1.4.6 Chapter 7: Non-Neutral Expansion . . . . .	22
1.4.7 Chapter 8: Future Work . . . . .	23
1.5 Conclusion . . . . .	24
References . . . . .	25
2. Computational Techniques . . . . .	28
2.1 Theoretical Descriptions of Plasmas . . . . .	28
2.1.1 Single Particle Description . . . . .	29
2.1.2 Kinetic Description . . . . .	30
2.1.3 Two-Fluid Description . . . . .	31
2.1.4 Magneto Hydrodynamics . . . . .	32
2.2 Simulation Types . . . . .	33
2.2.1 Partial Differential Equation Solvers . . . . .	33
2.2.2 Particle in a Cell . . . . .	33
2.2.3 Molecular Simulations . . . . .	34
2.3 Molecular Dynamics . . . . .	35
2.3.1 How Molecular Dynamics Simulations Work . . . . .	36
2.3.2 Integration Techniques . . . . .	37
2.3.3 Energy Conservation and Stability . . . . .	39
2.3.4 Difficulties with Plasma Systems . . . . .	40

2.3.5	Computation Time Reducing Approximations . . . . .	41
2.4	Programing With Graphics Processing Units . . . . .	44
2.4.1	Introdution to GPGPU . . . . .	44
2.4.2	GPU Architecture . . . . .	45
2.4.3	OpenCL Programing Model . . . . .	46
2.4.4	Device-Side Parameterization and Kernel Execution . . . . .	50
2.4.5	PyOpenCL . . . . .	55
2.4.6	General Guidelines for Effective GPU Programing . . . . .	59
2.5	Generalized Molecular Dynamics Model . . . . .	61
2.5.1	Model Overview . . . . .	61
2.5.2	Updater . . . . .	63
2.5.3	Common Forces . . . . .	66
2.6	Conclusion . . . . .	72
References . . . . .		73
3.	Electron Oscillation Damping . . . . .	75
3.1	Coulomb Collisions . . . . .	78
3.1.1	Generalized Solution . . . . .	78
3.1.2	Single Particle Collision Frequency . . . . .	79
3.1.3	Ensemble Collision Frequency . . . . .	83
3.1.4	Strong Coupling Corrections . . . . .	85
3.2	Experimental Overview . . . . .	88
3.2.1	Overview of the Experimental Apparatus . . . . .	88
3.2.2	Experimental Procedure . . . . .	93
3.3	Numerical Modeling . . . . .	96
3.3.1	Overview . . . . .	96
3.3.2	Collision Operator . . . . .	98
3.3.3	Initialization of Model Parameters . . . . .	102
3.3.4	Oscillation Damping . . . . .	106
3.4	Initial Experimental Data Set and Analysis . . . . .	107
3.4.1	Center of Mass De-Phasing . . . . .	108
3.4.2	Magnetic Field Effects . . . . .	115
3.4.3	Verifying the Model . . . . .	117
3.4.4	Finding a Collisional Parameter Range . . . . .	123
3.5	Second Data Set . . . . .	124
3.5.1	Overview of the Measured Results . . . . .	124
3.5.2	Failure of the Collision Operator . . . . .	125
3.5.3	Determining the Electron Temperature . . . . .	127
3.5.4	Testing Collision Assumptions . . . . .	131
3.6	Conclusion . . . . .	134
References . . . . .		136
4.	Modeling Ultracold Plasma Formation . . . . .	138

4.1	Heating Mechanisms . . . . .	140
4.1.1	Continuum Lowering . . . . .	141
4.1.2	Disorder Induced Heating . . . . .	142
4.1.3	Three-Body Recombination . . . . .	143
4.1.4	Cooling Mechanisms . . . . .	146
4.2	Formation Model . . . . .	147
4.3	Results . . . . .	150
4.4	Conclusion . . . . .	152
	References . . . . .	154
5.	Thermalization . . . . .	155
5.1	Motivation . . . . .	156
5.2	Model . . . . .	158
5.2.1	Overview . . . . .	158
5.2.2	Comparing the Distributions . . . . .	160
5.2.3	Quantifying Thermalization . . . . .	162
5.3	Results . . . . .	163
5.3.1	Basic Scaling . . . . .	164
5.3.2	Density Distribution Effects . . . . .	166
5.4	Conclusion . . . . .	175
	References . . . . .	177
6.	Evaporation . . . . .	178
6.1	Evaporation in Ultracold Plasmas . . . . .	179
6.2	Analytical Expressions for Evaporation . . . . .	181
6.2.1	The Atomic Case . . . . .	182
6.2.2	The Charged Particle Case . . . . .	183
6.2.3	Assumptions of Charge Particle Evaporation Expressions . . . . .	185
6.3	Model . . . . .	186
6.3.1	Overview . . . . .	186
6.3.2	Collisions . . . . .	188
6.3.3	Maintaining Equilibrium and the Production Function . . . . .	192
6.4	Results . . . . .	193
6.4.1	Parameter Scaling . . . . .	194
6.4.2	Testing Evaporation Assumptions . . . . .	198
6.5	Conclusion . . . . .	204
	References . . . . .	205
7.	Non-Neutral Expansion . . . . .	207
7.1	Simple Model for Neutral Plasma Expansion . . . . .	208
7.2	Model Components . . . . .	213
7.2.1	Overview of the model . . . . .	213
7.2.2	King Model of the Electron . . . . .	214

7.2.3	Two Point Boundary Value Problems . . . . .	218
7.2.4	Ion Expansion . . . . .	219
7.2.5	Temperature Evolution . . . . .	221
7.3	Model Results . . . . .	222
7.4	Ion Doppler Profiles . . . . .	232
7.5	Conclusion . . . . .	234
References . . . . .		235
8.	Future Work . . . . .	237
8.1	Highly-Magnetized Electron-Ion Collisions . . . . .	238
8.2	Testing Assumptions implied in Plasma Theories . . . . .	239
8.3	Continuum Lowering and Formation Modeling . . . . .	240
References . . . . .		242
A.	Sample Code . . . . .	243

# Chapter 1

## Introduction

Plasmas are charged particle systems whose collective behavior is dominated electric forces present in the system. Plasma systems span many orders of magnitude in both temperature and density leading to many different types of plasma systems, each with their own distinct characteristics, but sharing common aspects associated with the fact that they are plasmas. One particular class of plasma systems is that of ultracold plasmas. Ultracold plasmas (UCPs) are characterized by their low particle densities and very cold particle temperatures. In most circumstances these characteristics are manifested as a cold "ball" of electrons and ions with a spatial extent on the order of a few  $\mu\text{m}$ . UCPs are well controlled and highly tunable systems, which makes them good candidates to probe fundamental plasma physics. Additionally, due to their cold temperatures, UCPs are also excellent systems to test the physics of strongly-coupled plasma systems. This dissertation will cover my research into these systems that I have conducted to over my course of study at Colorado State University.

The goal of this chapter is to provide context as to how ultracold plasma systems fit into the plasma physics ecosystem, as well as to give an overview of what will be discussed in the later chapters contained in this dissertation. To accomplish this task the chapter starts out with a brief discussion of what makes a physical system a plasma. Next, the chapter gives a brief overview of many of the more widely researched plasma systems, and identifies where each of these systems resides in plasma parameter space. Once this context has been provided, the chapter moves into an introduction of ultracold plasmas. This introduction includes a discussion of how ultracold plasmas are formed, their life cycle, general areas of research in ultracold plasma field, and many other topics. Finally, this chapter will provide a preview of the rest of the dissertation by briefly summarizing each of the upcoming chapters.

## 1.1 What is a Plasma?

Plasmas are physical systems composed of a collection of free charged particles. While there are exceptions, most plasmas consist of electrons and one or more types of ions. Plasma systems are typically formed through the ionization of a neutral gas via a variety of different mechanisms (thermal, photo-ionization, field ionization, collisions). In the so-called weak coupling limit, plasmas share some characteristics with neutral gaseous systems (ie compressibility), but the charged nature of plasma particles leads to additional properties as well.

The electro-magnetic forces present in these systems lead to collective particle behavior (screening, plasma oscillations, etc.) unique to plasma systems. Specifically, there are three different fundamental characteristics present in all weakly-coupled plasma systems[1]:

1. The Debye screening length, the characteristic length by which fields are screened in the plasma, is small in comparison to the physical size of the plasma. When true, bulk plasma dynamics dominate the system, and edge effects are minimized.
2. The average inter-particle spacing is small in comparison to the Debye screening length. Without meeting the this condition, Debye screening will not be able to occur.
3. The electron plasma frequency is fast in comparison to the electron-neutral collision frequency. When true, electro-magnetic forces will dominate the plasma dynamics.

These characteristics make plasmas fundamentally different from gases. It is for this reason that plasmas are often referred to as the fourth state of matter.

In the absence of external fields, a plasma can be fully defined by just a small number of fundamental parameters. These are the temperature  $T_s$  and the particle density  $n_s$ , where  $s$  denotes what species of charged particle the quantity is referring to. In the case of a simple two component plasma where  $n_e = n_i$  and  $T_e = T_i$  the entire system can be described by just two quantities. From these two quantities other fundamental plasma parameters can be

derived. One such quantity is the Debye screening length due to the electrons in the plasma,  $\lambda_D$ , which is the characteristic length scale by which the plasma screens out electric fields. This length scale is defined as follows[1]:

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_b T}{n_e q^2}} \quad (1.1)$$

where  $n_e$  is the electron density,  $\epsilon_0$  is the permittivity of free space,  $k_b$  is the Boltzmann constant, and  $q$  is the fundamental charge constant. Another important parameter is the plasma frequency,  $\omega_p$ . This parameter represents the characteristic oscillation frequency of the electron density in a plasma with uniform density.  $\omega_p$  is defined in the following way[1]:

$$\omega_p = \sqrt{\frac{nq^2}{m_e \epsilon_0}} \quad (1.2)$$

where  $m_e$  is the mass of the electron.

## 1.2 The Plasma Parameter Space

Plasma systems span a wide range of physical parameters. Electron temperatures can vary by up to 10 orders of magnitude and charged particle number densities can vary by up to 30 orders of magnitude. This wide range of temperature and density characteristics leads to a number of different types of plasmas, each occupying their own distinct area of parameter space. Many of these plasma types can be seen in Figure 1.1. This section will give a brief overview of some of the different types of plasmas that exist, identify where they are located in parameter space, and discuss some of their unique behavior. This is to help place my work studying ultracold plasmas in the wider context of plasma physics in general.

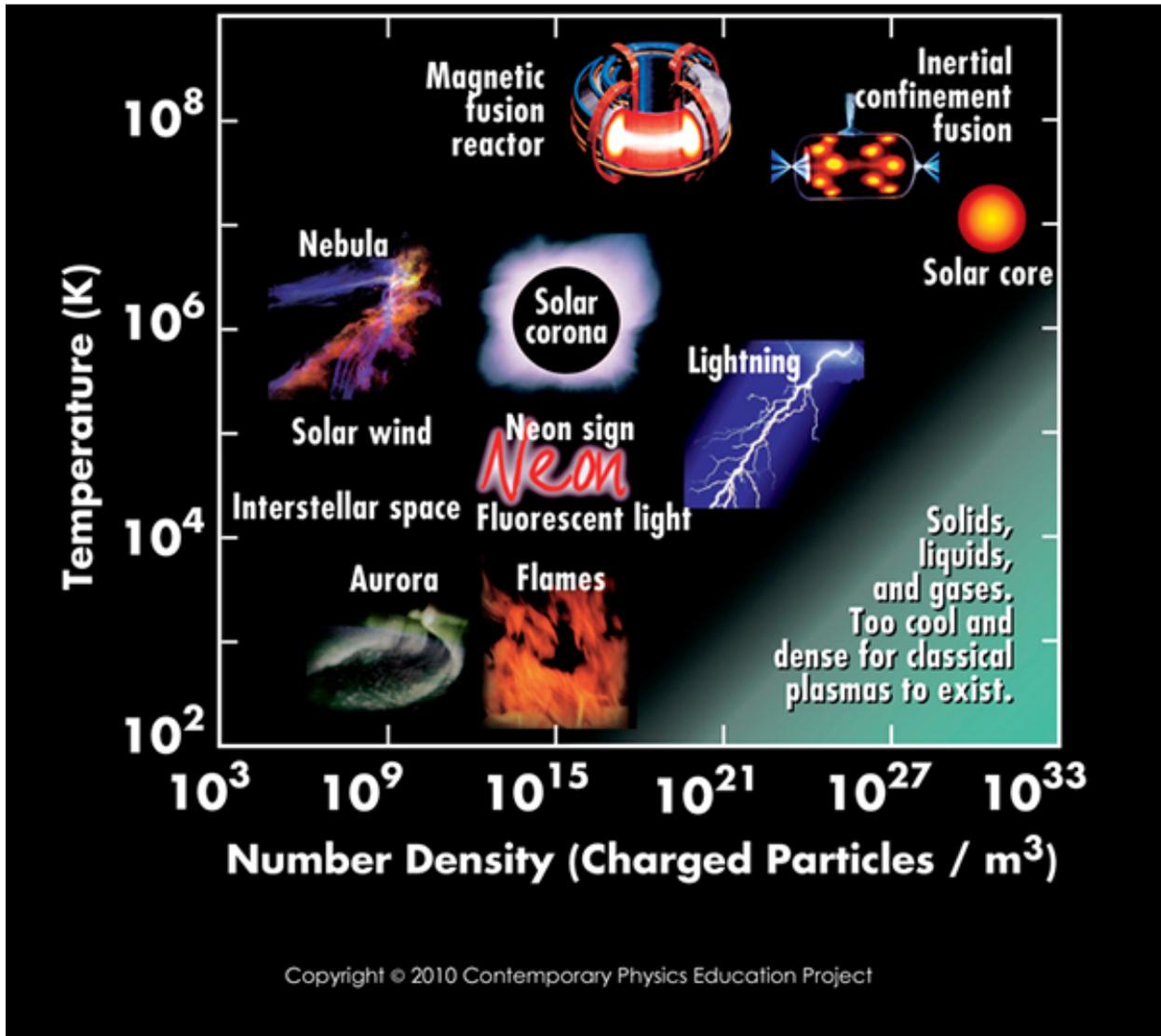


Figure 1.1: The figure shows a vast array of different plasma types and their corresponding temperatures and densities. Due to the large density and temperature range, different types of plasmas can have a wildly different behaviors and physical characteristics.

### 1.2.1 Low Temperature Plasmas

Low temperature plasmas are typically characterized by electron temperatures on the order of single  $eVs$  to tens of  $eVs$ [2, 3]. These systems are often not fully thermalized, allowing for ion temperatures as cold as room temperature. Systems in this parameter range are typified by their applicability in industrial uses. From powering fluorescent lighting to being used in the fabrication of integrated circuits, low temperature plasmas have a wide variety of practical uses. Most plasmas in this regime can be classified as gas discharges. Gas discharges are often used in lighting as well as gain medium for many types of different lasers[4]. Additionally, gas discharges are also frequently used to create an environment where the presence of charges allows for distinct chemical reactions to take place. This process is referred to as plasma chemistry[5], and is used in a variety of applications from etching to water treatment.

### 1.2.2 Astrophysical Plasmas

Plasmas make up over 99% of the visible matter in the universe, so it should come as no surprise that the universe is home to a vast array of different types of plasmas[6]. These include magnetospheres, coronas, accretion disks, interstellar medium, solar cores, the cores of some planets, and many many more. Ranging over many orders of magnitude in both density and temperature, astrophysical plasmas occupy nearly every part of plasma parameter space. Studies of these plasma systems has allowed for the further understanding of many different astrophysical phenomena such as the formation of low density stars, convection in stars, and the spin down of stars. However, there are still a number of active areas of research worth investigating, including magnetorotational instabilities[7] and magnetic reconnection[8] among others.

### 1.2.3 Fusion Plasmas

Perhaps the most widely studied type of plasma, fusion plasmas are characterized by having temperatures and densities that are sufficiently high for fusion reactions to occur[9]. In nature these plasmas occur in the cores of stars where reactions that (most commonly) convert hydrogen ions to helium give off the energy that powers the stars. On earth, similar fusion reactions could lead to a revolution in energy production[10]. If fully realized, fusion reactors would produce little waste and no greenhouse gases, and would have access to near limitless quantities of cheap fuel. Unfortunately, the temperatures necessary for fusion are difficult to confine. In the stars, confinement is achieved by gravitational forces exerted on solar cores by the entire mass of the star. Clearly other confinement methods are needed for fusion to be achieved on earth. Many confinement schemes exist, but the two most prominent are Inertial confinement fusion(ICF)[11] and Magnetic confinement fusion[12].

Magnetic confinement utilizes large magnetic fields to confine fusion plasmas[13]. The most common type of magnetic confinement scheme is the tokamak, a torus with large toroidal magnetic field. Currents, in the toroidal direction, are induced in plasmas inside the tokamak, causing the plasma to heat up to temperatures where fusion is possible. Thus the toroidal magnetic fields are able to confine the plasma current in the toroidal direction.

In the case of ICF fusion, laser pulses are used to rapidly compress a fuel pellet, causing the pellet to reach a high enough density and temperature where fusion reactions can occur. There are two primary ways by which ICF fusion can be initiated. The first, is direct-drive ICF, where the laser beams are focused directly on the fuel pellet, directly causing the pellet implosion. The second method is indirect drive. For this method, the fuel pellet is placed inside a gold cylindrical container, called a hohlraum, which has an opening at each of the end caps. Laser pulses are focused through the hohlraum openings and eventually hit the inside walls of the hohlraum. The laser-hohlraum interaction results in the hohlraum walls emitting x-rays, which strike the fuel pellet causing it to implode and initiating a fusion reaction.

## 1.2.4 Warm Dense Matter

Warm dense matter(WDM) plasmas are characterized by very high particle densities, which are comparable to condensed matter systems, and moderately hot temperatures on the order of 1-100eV[14]. At their high densities, these systems are too cold to be accurately described by standard weakly coupled plasma theory. However, they remain too hot to be described by standard condensed matter physics. This puts WDM squarely in the strongly coupled regime (described in sec 1.3.4), which makes WDM a good candidate to study the fundamental physics associated with strong coupling. WDM systems occur as a byproduct of ICF fusion and are expected to occur naturally in the cores of some large planets, such as Jupiter. In experiments, warm dense matter is typically created by using very high energy laser pulses to uniformly heat pieces of solid state matter.

## 1.2.5 Dusty Plasmas

Dusty plasmas are plasma systems that are dominated by charged "dust" particles[15, 16]. The "dust" particles attract charged particles, resulting in many charges sticking to the surface of the dust particle. Dust particles can range from a nm to a mm in diameter, and can have net charges up to on the order of  $10^3$  fundamental charges. Dusty plasmas exist in many astrophysical contexts such as planetary rings and comet tails. In the laboratory, dusty plasmas can be created by introducing dust particles into a low temperature plasma. Once created, dusty plasma motion is typically slow enough, and the particles are large enough that the system dynamics can be captured via video camera. Since dust particles are highly charged, nearest neighbor potential energy is typically quite large. This typically leads to large strong coupling parameters, making dusty plasmas an interesting system to study strong coupling physics.

## 1.3 Ultracold Plasmas

The work discussed in this thesis is entirely focused Ultracold plasmas (UCPs). These systems lie at the crossroads of atomic physics and plasma physics. Atomic physics techniques are utilized to cool, trap, and ultimately ionize atomic or molecular gases. Once the gas is ionized, a two component plasma (a UCP) is formed. UCP experiments are highly reproducible, typically fit on a single optical table, and operate on easily accessible experimental timescales, making them excellent systems to test fundamental plasma physics. UCPs are characterized by very cold temperatures, with electron temperatures ranging from 1-500K and sub-kelvin ion temperatures, and low particle densities on the order of  $10^6 - 10^{10} \text{ cm}^{-3}$ . The combination of low density and very cold temperature ranges allow for the access to strong coupling regime, making UCPs a viable candidate to study the physics of strong coupling. Because of the focus of this thesis on ultracold plasmas, their properties and behavior will be discussed in more detail in the subsections below.

### 1.3.1 Creation and Lifecycle

Most UCPs are created from ultracold atomic gases[17], but there are other possible sources[18]. Typically gases are cooled and trapped by using Doppler cooling and magneto-optic traps. This process yields atomic temperatures of less than a mK. These low atomic temperatures are essential in reaching the low temperatures that are associated with the ultracold plasma regime. Once the gas is cooled, a laser pulse is used to ionize the gas, creating a UCP. Once created, UCPs have a lifetime of 10s to 100s of  $\mu\text{s}$  depending on the UCP temperature and density. The UCP life cycle contains two distinct phases, the formation phase, and the expansion phase. These two phases will be discussed in further detail in the following paragraphs.

The formation phase of the UCP lifetime starts during ionization process. Typically, the ionization process is a two photon process where the first photon comes from a laser tuned

to the laser cooling transition, and the second photon comes from pulsed dye laser. Dye lasers are tunable lasers with a reasonably narrow linewidth, which allows for the relatively precise tuning of a wide range of electron temperatures. Since UCP ions are many orders of magnitude more massive than the electrons, conservation of momentum and energy dictates that the vast majority of photon energy is transferred to the electron. Thus, the ionizing laser pulse is not a major source of ion heating.

After ionization, there are an equal number of ions and electrons contained in the plasma. This leads to an overall net charge of zero for the whole system. Since the electrons are significantly less massive, and are moving significantly faster than the ions, some of the electrons escape from the system. As electrons escape from the plasma the number of ions begins to exceed the number of electrons, and as a result, a net positive space charge develops. Eventually, this space charge becomes sufficiently large enough to confine the remaining electrons, marking the end of the formation phase. From ionization to electron confinement, the entire formation process usually lasts on the order of a few microseconds. Figure 1.2 illustrates the formation process.

Since the number of ions and electrons are not equal in UCPs, the charge imbalance,  $\delta$  becomes an important parameter in characterizing a UCP, as it can have a significant impact on the time evolution of the plasma (see chapter 7). In our lab this quantity is defined in the following way:

$$\delta = \frac{N_i - N_e}{N_i} \tag{1.3}$$

where  $N_i$  and  $N_e$  are the total number of ions and electrons respectively in the system. Post formation phase  $\delta$  varies substantially with density. High density UCPs lose less than a percent of the total electron number during formation, while lower density UCPs can lose over half of their total electrons number. However, the initial electron escape is not directly a function of density, but rather a function of temperature and the confining potential depth.

The formation phase also yields the thermalization of the UCP electron cloud. Initially after ionization, the electron velocity distribution is very narrow, corresponding to the nar-

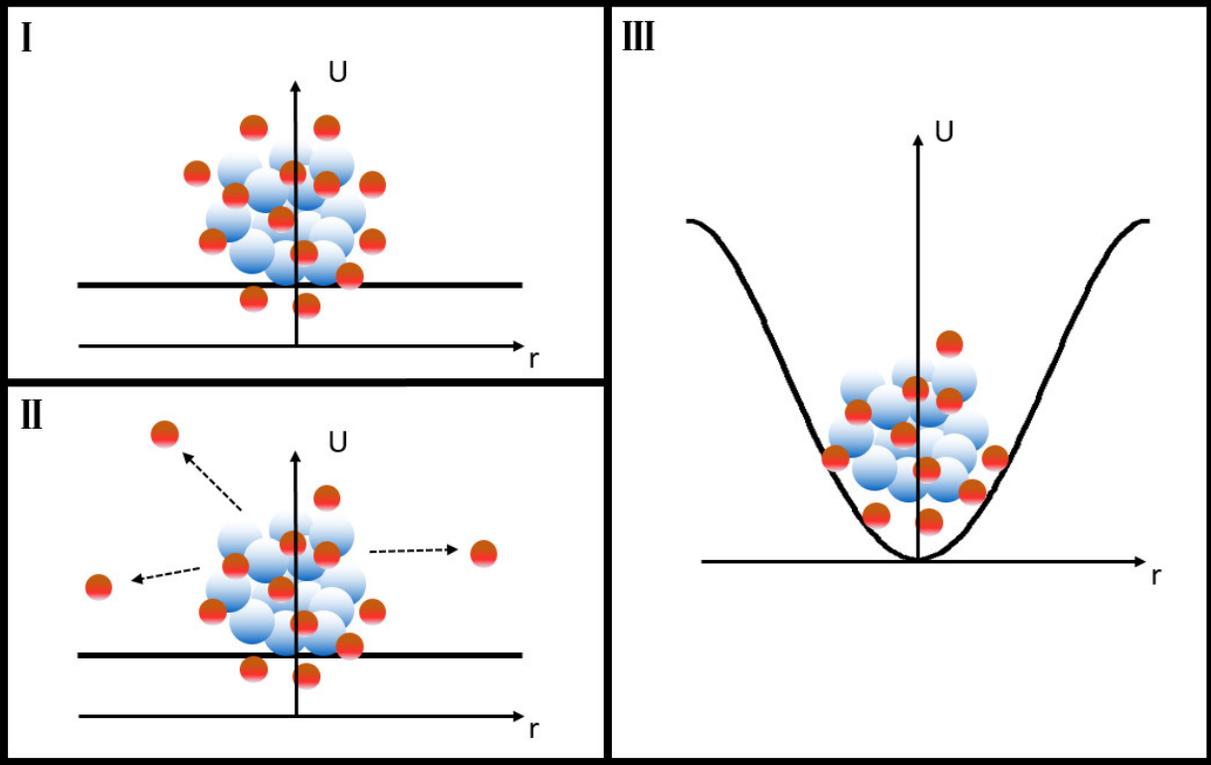


Figure 1.2: Diagram of the ultracold plasma formation process. The red spheres correspond to plasma electrons, and the blue spheres correspond to plasma ions. Initially, since there are an equal number electrons and ions, there is no net confining force holding the plasma particles in place. This is shown in frame I. Since the electrons are much less massive than the ions, a number of electrons initially escape. This process can be seen in frame II. Once these electrons leave, the number of ions in the plasma outnumber the number of remaining electrons, resulting in the development of a net positive space charge. The positive space charge yields a confining potential that traps the remaining electrons. This process can be seen in frame III.

row linewidth of the ionizing laser pulse. However, as time elapses, electron pairs undergo Coulomb collisions, which lead to energy transfer. The net effect of these collisions is to broaden the electron velocity distribution by causing the system to evolve toward thermal equilibrium, i.e. a Maxwellian. By the end of formation, the electron cloud is close to, if not fully in, thermal equilibrium. This fact is often beneficial, since many plasma calculations assume thermal equilibrium.

After the formation phase is over, the UCP will begin to expand. There are a couple of different ways to conceptualize this expansion, but the underlying physics is the same. First, the electron cloud can be thought of as gas exerting pressure on the walls of the confining ions, and causing the ions to expand in the process. Another way to view the situation involves thinking about the electric forces involved. Since a potential confining the electrons exists, that same potential must lead to an anti-confining of the ions. In other words, this potential leads to an outward force acting on the ions that drives ion expansion.

Regardless of the expansion framework used, ion expansion by definition leads to a larger ion spatial extent, which in turn leads to more a shallow confining potential. As the confining potential depth decreases, more electrons are able to escape via the process of evaporation (see chapter 6). Eventually, the ion spatial extent becomes so large that it is unable to confine even a single electron, leading to the end of the plasma life cycle. An illustration of the ion expansion phase can be seen in figure 1.3.

### **1.3.2 Detection Methods**

There are two main techniques that are utilized to detect ultracold plasmas experimentally. These are charged particle detection, and optical absorption imaging. Charged particle detection is perhaps the simpler detection scheme, and is it is also the more versatile method, allowing for the possible detection of all charged particle species in a UCP. Standard implementations of this method do have their limitations though, as charged particles first have to escape the UCPs before they can be detected. There is one notable exception to this,

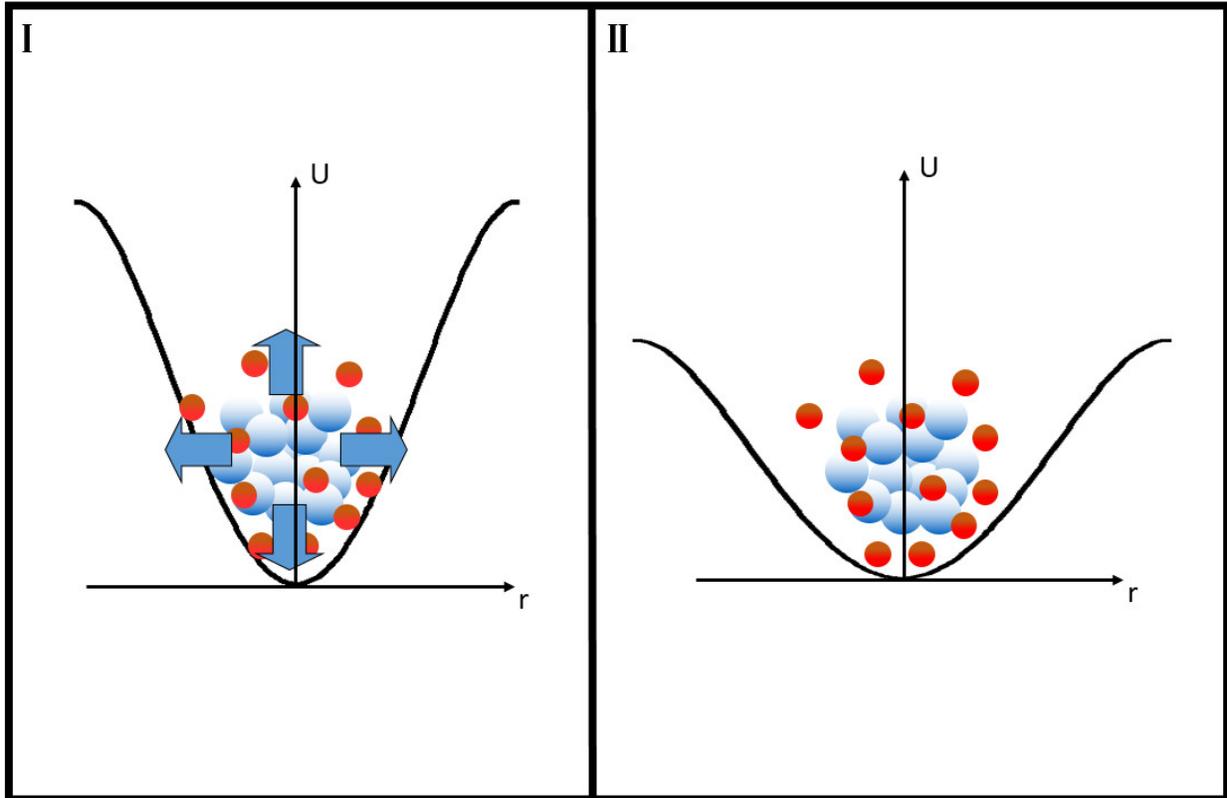


Figure 1.3: Diagram of the expansion phase of the ultracold plasma life cycle. The red spheres correspond to plasma electrons, and the blue spheres correspond to plasma ions. The positive net space charge, originating from the plasma containing more ions than electrons, yields an outward force on the ion component of the plasma. This leads to an expansion of the ion cloud. As the cloud expands, the the confining ion potential becomes more shallow. This allows for additional electrons to escape. Eventually, after the ion cloud expands sufficiently, all of the electrons will be able to escape the plasma. This marks the end of the UCP life cycle.

where charged particle detection is used to spatially resolve a forward propagating ultracold plasma[18]. However, due to the uniqueness of this experiment, the rest of this section will only consider charged particle detection of escaped particles.

In order to effectively use charge particle detection schemes to detect escaping particles, there are a couple of issues that must be accounted for. First, to understand UCP dynamics, a theoretical link needs to be developed between the charge particle escape physics and the underlying UCP dynamics. It is for this reason that charged particle detection schemes typically focus on the electron component of UCP, since electron escape physics can be much more easily linked to the physics underlying UCP evolution. Additionally, in a no external field environment, there is no guarantee that escaped charged particles will come into contact with the detector. For this reason a small bias DC electric field, typically on the order of a few  $V/m$ , is typically applied to the system. While the bias field results in all of the escaped charged particles hitting the detector, it leads to a distortion in shape of the UCP, and also will significantly increase the charge imbalance,  $\delta$ , if large enough. Typically bias field distortions are relatively small, especially in the bulk of the plasma, but they do need to be accounted for. Another challenge is that charged particles don't all follow the same escape path, leading to a variety of different possible flight times. From a detection stand point, this results in an effective temporal averaging of the detection signal, leading to a reduction in resolution. Despite these limitations, charged particle detection methods remain a very valuable UCP diagnostic, and are the only currently demonstrated method for detecting UCP electrons.

Optical absorption imaging allows for the detection of UCP ions with high spatial and temporal resolution[19]. By tuning a laser beam near resonance with an ion transition it is possible to measure the fraction of the laser beam that is absorbed. To do this, the laser beam is imaged onto a ccd camera after passing through the UCP, giving an intensity measurement at each pixel on the camera. This process is then repeated, but in the absence of any UCP. From these two measurements, and optical depth can be calculated. This quantity, defined

as  $OD = \ln(I_0/I)$  where  $I$  and  $I_0$  are the given pixel intensities in the presence or absence of a UCP respectively, yields high resolution information on local ion densities. Furthermore, by varying laser beam detunings, local ion velocity profiles can also be extracted. From these velocity profiles, an ion temperature can then be extracted.

Despite being a powerful technique, optical absorption does have a number of limitations. First, it is only applicable to ions, and thus any experiment looking to study physics associated with the electron component of UCPs will likely need to utilize charged particle detection. Additionally, it is limited to ions with optically accessible transitions. This has typically meant that this technique has been limited to alkaline earth metals such calcium and strontium. These atoms can have a number of decay paths making them more difficult to cool than an atom such as rubidium, and thus add additional experimental complexity.

### 1.3.3 Tests of Fundamental Physics

UCPs offer the ability to test fundamental plasma physics at easily accessible experimental time scales. Such tests can provide further understanding of into UCP systems, as well as grant insight into the physics of other plasma systems. UCPs have been used to study collective mode oscillations. Many experiments have detected electron plasma oscillations[20], and the damping of these oscillations (through both collisional[21] and collisionless[22] mechanisms) has been experimentally measured (see chapter 3). Additionally, more complicated oscillations such as Tonks-Datner resonances[23], ion-acoustic waves[24], and others have been observed in UCP systems.

UCPs also offer the possibility to study fundamental collision processes. One example is three-body recombination, a type of three-body collision, which is a widely studied phenomena in UCPs. Three-body recombination involves a collision between two free electrons and an ion. The collision results in the binding of one of the electrons creating a Rydberg atom. This binding leads to excess energy being transferred to the second electron, causing the heating of electrons in the plasma.

UCPs can also be used to test binary collisions. The rate of electron-electron binary Coulomb collisions in a plasma set the characteristic electron thermalization timescales (see chapter 5). Since thermal equilibrium is an important underlying assumption in many plasma models and experiments, a knowledge of collision rates is necessary for proper experimental design. Additionally, UCPs offer the chance to experimentally test predicted collision rates, as well as predicted strong coupling corrections (see chapter 3). Such tests can offer significant insights into the dynamics of strongly coupled plasma systems. Recent measurements quantifying the rate of ion relaxation in UCP systems demonstrated the influence of ion strong coupling[25]. Since, the ion relaxation rate is driven by ion-ion collisions, these measurements were used to test strong coupling extensions to standard Coulomb collision theory. Other work has measured the damping rate of electron center-of-mass oscillations[21]. This damping was driven by electron-ion collisions, and the results of these measurements could not be explained without the influence of electron strong coupling on the electron-ion rate.[21]. This electron center-of-mass oscillation damping measurement has been a major component of my research and will be discussed extensively in chapter 3. Finally, the small sizes of UCP systems allow for precise experimental control over external electric and magnetic fields, allowing for tests of collisions in magnetized plasmas (see chapter 8).

UCPs can also be used to provide insight into the process of continuum lowering[26]. Continuum lowering arises from a plasma's equilibrium distribution of electrons and ions yielding an electric potential structure that leads to a depression of the ionization potential of neutral atoms. In hotter, denser, and more complex plasmas, continuum lowering leads to an increased rate of collisional ionization. This changes the thermodynamic properties of these plasmas, and can have a significant effect on the transport properties of these plasmas. In ultracold plasmas, continuum lowering can lead to a heating of the electron component of the plasma (see chapter 4). Measuring this heating in UCP systems should, theoretically, provide insight into the underlying mechanisms driving continuum lowering.

Such a measurement could be particularly useful, since there is currently some confusion about the underlying mechanisms driving continuum lowering[27].

UCPs also have similarities to many different physical systems, many of which are substantially more difficult to study experimentally. Perhaps the most striking example of this comes from globular cluster star systems[28, 29]. Globular clusters are densely packed spherical star clusters, which contain thousands, and sometimes millions of stars. At a glance, it is difficult to see what Globular cluster could have in common with UCPs, considering that there is roughly a 60 orders of magnitude mass difference between their constituent parts. However, the shared  $1/r$  potential, gravitational for globular clusters and Coulomb for UCPs, allows for a myriad of similarities. The two systems have similar collisional properties, the  $1/r$  potential leads to binary collisions in both systems being described by the same expressions. Furthermore, binary star formation in globular cluster systems occurs by a process that is quite similar to three-body recombination. Finally, stars in globular clusters are confined by a finite gravitational potential. This allows for high energy stars to escape the cluster in a process analogous to electron evaporation in UCPs. Since globular cluster dynamics play out on an astronomical timescale, and UCPs can be created and studied in a laboratory, experimental UCP systems offer a promising way to learn about globular clusters dynamics.

### 1.3.4 Strong Coupling Physics

Perhaps one of the most useful characteristics of UCP systems, is the ability to probe into the strongly coupled plasma regime. In this regime, nearest neighbor particle correlations, which are not present in traditional weakly couple plasmas, become significant[30, 31]. To denote where in parameter space this regime begins, it is necessary to quantify the degree of strong coupling present in the plasma system. This is done with the strong coupling parameter,  $\Gamma$ , which is defined as:

$$\Gamma = \frac{q^2}{4\pi\epsilon_0 a k_b T} \quad (1.4)$$

where  $a$  is the Wigner-Seitz radius or the average inter-particle spacing, and is defined as  $a = (3/4\pi n)^{1/3}$ . In general, a higher value of  $\Gamma$  signifies a higher degree of strong coupling in a system. Since particle correlations in plasmas are a function of inter-particle Coulomb interactions, it would be expected that the degree of coupling in a plasma systems would increase as inter-particle potential energy becomes a larger share of the total system energy. This is essentially what  $\Gamma$  is quantifying, since  $\Gamma$  is, by definition, the ratio of the average nearest neighbor potential energy,  $q^2/4\pi\epsilon_0 a$ , and average particle kinetic energy,  $k_b T$ .

Most, plasma systems are either sufficiently hot or have sufficiently low densities where values of  $\Gamma \ll 1$ . Such systems exhibit no noticeable effects from inter-particle correlations, and are well described by standard weakly coupled plasma theory. However, as plasma systems become colder and denser, they begin to exhibit strong coupling effects. In certain circumstances this can lead to phase transitions, resulting in typically gaseous plasma exhibiting liquid or even solid like behavior. Typically, the boundary,  $\Gamma = 1$ , is thought to be the line demarcating the transition from the weakly to strongly coupled regime. However, the transition is not quite so abrupt. Certain strong coupling effects (eg. alterations to the coulomb logarithm[33, 34, 35, 36]) can be detected for values of  $\Gamma$  less than 1, while other effects (eg. Wigner crystallization[37]) require values of  $\Gamma \gg 1$ .

UCPs offer the ability to study strongly coupled physics in a unique range of parameter space. As can be seen in Figure 1.2, ultracold plasma systems straddle the boundary of the strongly coupled regime. This allows for UCP experiments to not only probe the strongly coupled regime, but to also investigate the transition region. However, there are limits to the degree of coupling achievable. Heating mechanisms (see Chapter 4), such as continuum lowering[26, 38], disorder induced heating[39], and three-body recombination[40] result in an effective temperature floor for UCP systems. Since UCPs are 2 temperature systems, these temperatures floors are not the same for both the ion and electrons. Typically the

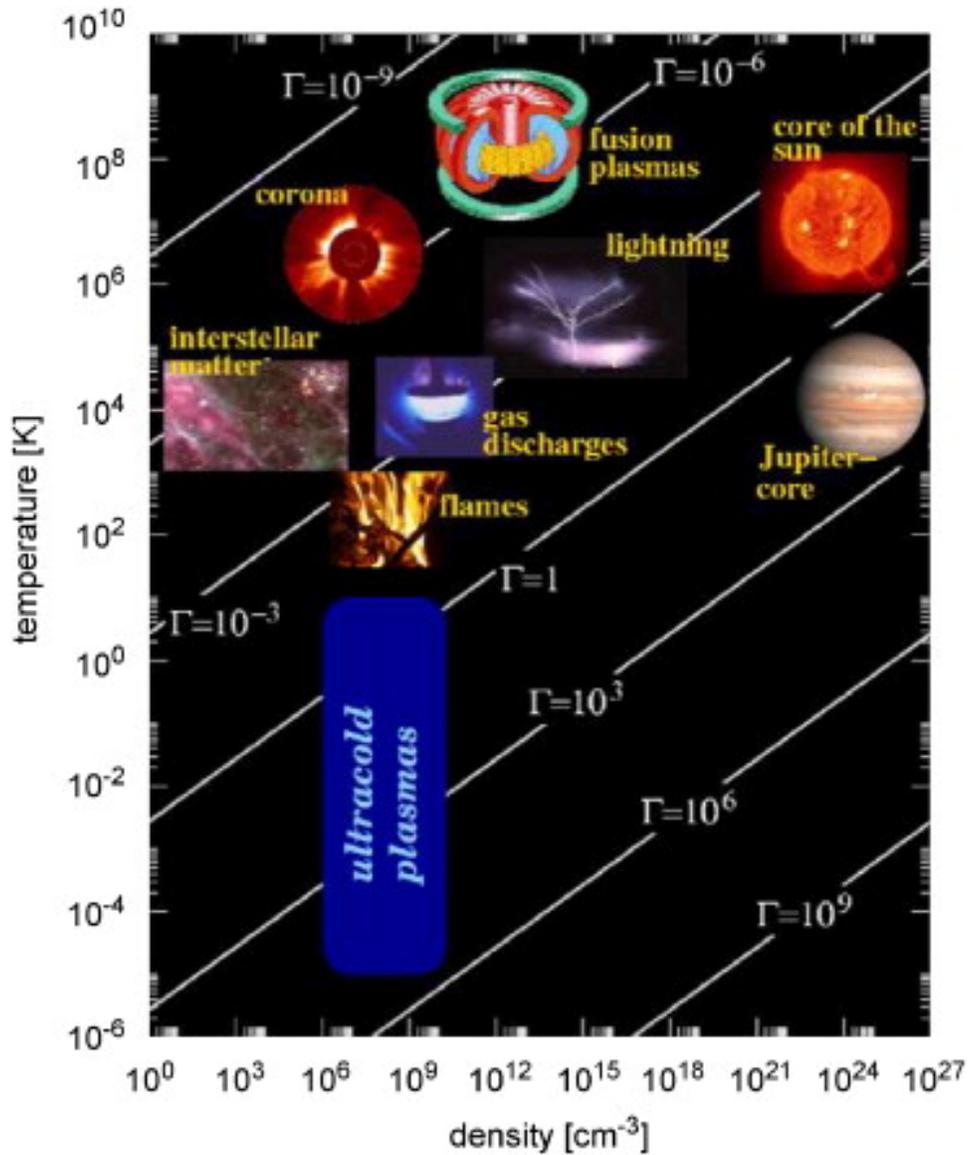


Figure 1.4: The above figure is a modification of figure 1.1 that has been altered to include strongly coupled plasma systems[32]. The diagonal lines denote the effective strong coupling parameter,  $\Gamma$ , which is defined by Eq 1.4. The figure shows that ultracold plasma systems are able to extend well into the strongly coupled regime ( $\Gamma > 1$ ). However, a few caveats need to be made here. First, heating mechanism (discussed in chapter 3) severely limit achievable values of  $\Gamma$ . Second, this figure only corresponds to the ion component of the ultracold plasma. Values of  $\Gamma$  will be even further limited for the electron component of the plasma.

ion component of the UCP is colder than the electron component, and it is not uncommon for UCP systems to exist in which ion coupling,  $\Gamma_i \gg 1$ , and electron coupling,  $\Gamma_e \ll 1$ . Previous theoretical calculations have suggested that electron strong coupling is limited to  $\Gamma_e \leq 0.2$ [41]. However, recent work presented in this thesis (see Chapter 3) has observed  $\Gamma_e \approx 0.35$ .

## 1.4 Overview of the Thesis

The work discussed in this thesis is primarily theoretical in nature. Over my tenure at Colorado State University, I have developed a number of different numerical models that simulate a variety of different physical process that occur in ultracold plasma systems. In this section I will give an overview of the different chapters in this thesis, and discuss the motivations behind each of the chapters that are included.

### 1.4.1 Chapter 2: Computational Techniques

Since the work discussed in this thesis is largely computational, this section is intended to give an overview of computational techniques used both in plasma physics in general, and specifically in my work. First, this chapter will discuss the different ways in which plasmas can be theoretically described. A focus of this discussion will be placed on molecular dynamics simulations, which are the simulation present in the bulk of this work. In this context, typical problems related to plasma simulations will be presented. Next, there will be an overview of the different computational techniques that are used in the numerical evolution of a plasma.

The chapter will then move into a discussion of the molecular dynamics modeling used in a large portion of this work. Since GPUs were an integral component of this modeling, this chapter will give an overview of gpu architecture and the gpu programming model. Finally, the chapter will discuss the implementation and components of my molecular dynamics model.

### 1.4.2 Chapter 3: Electron Oscillation Damping

Coulomb collisions are fundamental processes that occur in all plasma systems. The rate at which these collisions occur dictates the rate at which a plasma thermalizes. Additionally, many plasma models include collision rate terms that influence the transport properties of a plasma system. Thus knowledge of the collision rate is important to properly understand the behavior of any plasma system. Collision rates are well understood in weakly coupled plasmas, and can be easily calculated. However, this is not the case in strongly coupled plasmas, making further study of the topic necessary.

This chapter will discuss the different electron-ion collision rate measurement experiments that were performed over the last few years in our lab as well as the numerical modeling that was utilized in order to understand our results. Originally, the idea of the experiment was that a plasma oscillation would be induced in the plasma by an external electric field. The rate at which the oscillation damped out was expected to be a function solely of the electron-ion collision rate. However, as will be discussed in this chapter, it was found that collisionless mechanisms could also be a major contributor to oscillation damping. This chapter will discuss where these mechanisms come from, and how they can be mitigated. Finally, the chapter will discuss the experimental conditions utilized to mitigate collisionless effects, and discuss the measured damping rate as well as the modeling that showed that record high electron  $\Gamma_e$  was achieved.

### 1.4.3 Chapter 4: Formation Physics

The physics involved in the formation portion of the UCP lifecycle play a large role in establishing UCP conditions. During this time period, a number of different heating and cooling mechanisms exist, which can have a substantial impact on the electron temperature of the system. As the initial temperatures moves closer to zero, these mechanisms become a more important determinant of the temperatures in the system. This makes these heating mechanisms a fundamental limiter of the degree of strong coupling obtainable in the system.

Understanding and quantifying these mechanisms is necessary for a complete understanding of electron temperatures in UCP systems. In this chapter, I will first outline different heating and cooling mechanisms present in UCP systems, and discuss the underlying theory behind them. Next, I will discuss the molecular dynamic model that I developed in order to study UCP formation physics. Finally, the chapter will give an overview of the model results, as well as quantify the amount of heating expected for different experimental conditions.

One of the main findings of the formation physics model was the role played by external electric fields. In general, the model found that external electric fields are a mechanism for heating in UCP systems. This effect can be quite large, and was shown to be the dominant heating mechanism that exists in the experiment in our lab. This chapter will discuss the origins of this heating, and discuss model results for a variety of different experimental conditions.

#### **1.4.4 Chapter 5: Thermalization**

In chapter 3, the non-collisional mechanism leading to an increased damping of center of mass oscillations is presented. This damping was shown to be the result of the coupling between different oscillatory modes present in the system. When oscillatory modes are coupled, energy is, by definition, being transferred between the coupled modes. In general, energy transfer in a system is a driver of thermalization. Since an understanding of thermalization rates is important in characterizing the behavior of a plasma system, it is important to understand the impact that these collisionless mechanisms have on thermalization.

The work presented in this chapter sets out to determine the impact of which collisionless mechanisms have on the electron thermalization rate. This chapter will discuss the numerical model used to quantify the degree of thermalization caused by collisionless mechanisms, as well as the findings of this model. It was found that these mechanisms resulted in a detectable, but typically small, increase on the electron thermalization rate. Additionally,

the model also identifies three distinct regions of thermalization, which will also be discussed in this chapter.

### **1.4.5 Chapter 6: Evaporation**

Electron evaporation is the process by which highly energetic electrons escape from a UCP. Since charged particle detection is the primary diagnostic used to detect UCP electrons (see section 1.3.2), a thorough understanding of electron evaporation can offer a variety of insights into UCP physics. Evaporation is a well understood phenomenon in ultracold atomic gases, and has been described theoretically by a series of equations. Recently, these equations have been adapted to describe charged particles in anti-hydrogen systems. Naively, one would think that these equations could be used in ultracold plasma systems. However, these expressions have a number of underlying assumptions that are not clearly applicable to UCP systems.

The work described in this chapter tests the underlying assumptions present in the charged particle evaporation expressions. To understand these tests, this chapter will re-derive the evaporation expressions, illustrating where important assumptions are made along the way. Next, I give an overview of the model utilized to test these assumptions. The model found that the evaporation expressions did not accurately predict evaporation in UCP systems, and that neither of the underlying assumptions were valid. The results leading to this conclusion will also be discussed.

### **1.4.6 Chapter 7: Non-Neutral Expansion**

Theoretical treatment of UCP expansion dynamics has been well established for neutral UCPs. However, as the charge imbalance in these systems increases, this treatment begins to fail. Thus, to understand the expansion dynamics of non-neutral UCP systems The model described in this chapter was developed to calculate the expansion of UCP ions in

non-neutral UCPs. This chapter will give an overview of this model, and will discuss the numerical results.

### **1.4.7 Chapter 8: Future Work**

The work discussed in this dissertation has led to many insights into dynamics of ultracold plasma systems. These insights provide a springboard into a number of different potential investigations of physical phenomena. Of these, there are three main future research directions. First, during the work on electron center-of-mass oscillation damping (described in chapter 3), methods for modeling and experimentally measuring the electron center-of-mass oscillation damping were developed. The rate of this damping is expected to decrease in the presence of an strong external magnetic field. By using the previously developed techniques, future research will look to quantify the effect that a strong external magnetic field has on electron center-of-mass oscillation damping. Since this damping is a function of electron-ion collisions, this research will give insights into highly magnetized electron-ion collisions which can have a significant impact on the transport properties of a number of different plasma systems. The second direction of future work will be a theoretical investigation into the underlying assumptions of theoretical predictions of the electron-ion collision rate in strongly coupled plasma systems. The work in chapter 3 showed that these predictions employ assumptions that are inappropriate for our experimental conditions. This direction of future work will test which assumptions are faulty, and quantify how/if they fail. Finally, the last area of future work will be to use the formation model from chapter 4 to quantify the magnitude of the various heating mechanism present in UCP systems. The work will be heavily focused on continuum lowering, and will look to determine whether or not the duration of the ionizing laser pulse has a detectable effect on the magnitude of heating resulting from continuum lowering in UCPs. This chapter will give an overview of these three areas of future work.

## 1.5 Conclusion

Ultracold plasmas occupy a unique range of plasma parameter space, characterized by low particle densities and very low electron and ion temperatures. Ultracold plasmas are good systems to study fundamental plasma physics. Standard plasma phenomena occur on easily accessible timescales, and experimental parameters can be known with fairly high precision. Additionally, their low particle number makes these system particularly amenable to numerical simulations. Ultracold plasmas are also good systems to probe the physics of strongly coupled plasmas. Given their very cold particle temperatures, significant ion strong coupling parameters, as well as moderate electron strong coupling parameters are accessible. Additionally, given that ultracold plasmas straddle the strong/weak coupling boundary, these systems are an excellent candidate to probe the breakdown of the weak coupling approximation in plasma systems.

The work described in this dissertation describes a number of computational investigations into the dynamics of ultracold plasma systems. For most of these investigations the general molecular dynamics model which I developed was employed. This model harnessed the massively parallel nature of modern GPUs to enable the explicit modeling of each individual particle in the ultracold plasma system to be modeled explicitly. By using this model, investigations into the physics of electron center-of-mass oscillations, ultracold plasma formation, and electron thermalization in ultracold plasmas systems were conducted. Additionally, this dissertation contains work that predates the development of the general molecular dynamics model. This includes work developed to describe electron evaporation as well as non-neutral expansion in ultracold plasmas. The work described in thesis has provided a number of different insights into the dynamics of ultracold plasmas as well as other plasma systems.

# References

- [1] P. Bellan, *Fundamentals of Plasma Physics*, Cambridge University Press, 2006.
- [2] R. Hippler, H. Kersten, M. Schmidt, and K. H. Schoenbach, *Low Temperature Plasmas: Fundamentals, Technologies and Techniques*, Wiley, 2nd edition, 2008.
- [3] J. Meichsner, Low temperature plasmas, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 5, Springer, 2005.
- [4] M. Born and T. Markus, Research on modern gas discharge light sources, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 15, Springer, 2005.
- [5] A. Fridman, *Plasma Chemistry*, Cambridge, 2008.
- [6] C. Chiuderi and M. Velli, *Basics of Plasma Astrophysics*, Springer, 2015.
- [7] K. Julien and E. Knobloch, **368**, 1607 (2010).
- [8] M. Yamada, R. Kulsrud, and H. Ji, *Rev. Mod. Phys.* **82**, 603 (2010).
- [9] H. Bosch, Nuclear fusion, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 17, Springer, 2005.
- [10] T. Hamacher, The possible role of nuclear fusion in the 21st century, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 18, Springer, 2005.
- [11] R. Betti and O. Hurricane, *Nature Physics* **12**, 435 (2016).
- [12] J. Ongena, R. Koch, R. Wolf, and H. Zohm, *Nature Physics* **12**, 398 (2016).
- [13] F. Wagner and H. Wobig, Magnetic confinement, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 7, Springer, 2005.
- [14] M. Koenig et al., *Plasma Physics and Controlled Fusion* **47**, B441 (2005).

- [15] A. Melzer, Collective effects in dusty plasmas, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 11, Springer, 2005.
- [16] E. T. Jr, R. L. Merlino, and M. Rosenberg, *Plasma Physics and Controlled Fusion* **54**, 124034 (2012).
- [17] T. C. Killian et al., *Phys. Rev. Lett.* **83**, 4776 (1999).
- [18] J. P. Morrison, C. J. Rennick, J. S. Keller, and E. R. Grant, *Phys. Rev. Lett.* **101**, 205005 (2008).
- [19] T. C. Killian et al., *J. Phys. B: At. Mol. Opt. Phys.* **38**, S351 (2005).
- [20] T. Wilson, W.-T. Chen, and J. Roberts, *Phys. Plasmas*. **20**, 073503 (2013).
- [21] W.-T. Chen, C. Witte, and J. L. Roberts, Submitted to *Phys. Rev. Lett.* (2017).
- [22] W.-T. Chen, C. Witte, and J. L. Roberts, *Physics of Plasmas* **23**, 052101 (2016).
- [23] R. Fletcher, X. Zhang, and S. Rolston, *PHYSICAL REVIEW LETTERS* **96** (2006).
- [24] T. C. Killian, P. McQuillen, T. M. O’Neil, and J. Castro, *PHYSICS OF PLASMAS* **19** (2012).
- [25] G. Bannasch, J. Castro, P. McQuillen, T. Pohl, and T. C. Killian, *Phys. Rev. Lett.* **109**, 185008 (2012).
- [26] J. C. Stewart and K. D. Pyatt, *Astrophys. J.* **144**, 1203 (1966).
- [27] B. Crowley, *High Energy Density Phys.* **13**, 84 (2014).
- [28] I. R. King, *Astron J.* **71** (1966).
- [29] D. Vrinceanu, G. S. Balaraman, and L. A. Collins, *J. Phys. A: Math Theor.* **41**, 425501 (2008).

- [30] S. Ichimaru, *Rev. Mod. Phys.* **54**, 1017 (1982).
- [31] R. Redmer, Strongly coupled plasmas, in *Plasma Physics Confinement, Transport and Collective Effects*, chapter 6, Springer, 2005.
- [32] T. C. Killian, T. Pattard, T. Pohl, and J. M. Rost, *Phys. Reports* **449**, 77 (2007).
- [33] J. Daligault and D. Mozyrsky, *HIGH ENERGY DENSITY PHYSICS* **4**, 58 (2008).
- [34] P. E. Grabowski, M. P. Surh, D. F. Richards, F. R. Graziani, and M. S. Murillo, *Phys. Rev. Lett.* **111**, 215002 (2013).
- [35] S. D. Baalrud, *Phys. Plasmas* **19**, 030701 (2012).
- [36] L. G. Stanton and M. S. Murillo, *Phys. Rev. E* **93**, 043203 (2016).
- [37] H. Thomas et al., *Phys. Rev. Lett.* **73**, 652 (1994).
- [38] G. Ecker and W. Kröll, *Phys. Fluids* **6**, 62 (1963).
- [39] M. Lyon and S. D. Bergeson, *J. Phys. B: At. Mol. Opt. Phys.* **44**, 184014 (2011).
- [40] T. C. Killian et al., *Phys. Rev. Lett.* **86**, 3759 (2001).
- [41] F. Robicheaux and J. D. Hanson, *Phys. Rev. Lett.* **88**, 055002 (2002).

# Chapter 2

## Computational Techniques

Plasmas are multiparticle systems that exhibit collective behavior and long-range interactions. The underlying physics describing these systems is often quite complicated, and in most circumstances, exact analytical solutions describing the dynamics of these systems do not exist. However, in many cases numerical techniques can be utilized to find solutions to new problems, as well as improve the accuracy of existing solutions. Furthermore, with continued and persistent increases in computing power, the number of problems in plasma physics that can be addressed effectively through computation is rapidly increasing. It is for this reason that computational plasma physics is currently a rapidly growing field.

Given that the work presented in this thesis is primarily computational in nature, it is useful to briefly give an overview some of the more relevant techniques that are utilized in computational plasma physics. For this reason, this chapter will summarize some of the different theoretical frameworks utilized to describe plasma systems, and then give an overview of some of the different computational techniques that are used to solve problems in these frameworks. The chapter will also include an overview of some of the most common techniques used to solve many of the problems that are included in this dissertation. This will include a brief discussion of utilizing graphics processing units (GPUs) to solve numerical problems, as well as an outline of the general molecular dynamics model that I developed over the course of my time at Colorado State University.

### 2.1 Theoretical Descriptions of Plasmas

As has been previously discussed, plasmas systems span a wide range of physical parameters. This broad range means that certain theoretical descriptions of plasmas may work well for one specific set of plasma parameters, but not so well for other sets of plasma parameters.

For this reason, there are a number of different plasma descriptions that are commonly used. Theoretical plasma descriptions can be organized on a microscopic to macroscopic continuum. As theoretical treatments move from a microscopic to macroscopic descriptions, the total degrees of freedom in the plasma are reduced at the expense of knowledge about the system. On the microscopic end of the continuum is the single particle description, which explicitly describes the motion each particle in a plasma system. On the macroscopic end of the continuum, magneto hydrodynamics (MHD) describe entire plasma systems as a single fluid.

### 2.1.1 Single Particle Description

As the name implies, the single particle description of a plasma focuses on the dynamics of the individual particles present in plasma systems[1]. For this reason, it is the most fundamental method for modeling the physics involved with plasma systems. In this framework, and in the classical limit, particle motion is governed exclusively by the following equation

$$m \frac{d\vec{r}}{dt} = q(\vec{E}(r) + \vec{v} \times \vec{B}(r)) \quad (2.1)$$

where  $m$  is the particle's mass,  $q$  is the particle's charge,  $\vec{v}$  is the particle's velocity, and  $\vec{E}(r)$  and  $\vec{B}(r)$  are the electric and magnetic field present at the particle's location respectively. The  $\vec{E}(r)$  and  $\vec{B}(r)$  fields are, in general, a combination of externally applied fields and internal fields resulting from charged particles in the plasma system. This framework is typically used to describe single particle behavior (e.g.  $\vec{E} \times \vec{B}$  drift). However, by modeling the individual behavior of every single particle in the system, this framework can be used to describe the collective behavior of an entire plasma system. This is typically done via what is referred to as molecular dynamics simulation, which will be described in detail in section 2.3. Unfortunately, this approach is not viable for most plasma systems, as they contain

too many particles for it to be computationally feasible to simulate them all. Thus other theoretical frameworks are generally necessary.

## 2.1.2 Kinetic Description

The kinetic description of a plasma is next step on the continuum towards a macroscopic description of a plasma. In this framework, phase space distributions of an arbitrarily large number of particles are the basic unit of investigation. In general, the time evolution of a phase space distribution for the  $i$ th plasma component,  $f_i(\vec{r}_i, \vec{v}_i, t)$ , is dictated by the following[2]

$$\frac{\partial f_i}{\partial t} + \vec{v}_i \cdot \frac{\partial f_i}{\partial \vec{r}_i} + \vec{a} \cdot \frac{\partial f_i}{\partial \vec{v}_i} = \left( \frac{\partial f_i}{\partial t} \right)_c \quad (2.2)$$

where  $\left( \frac{\partial f_i}{\partial t} \right)_c$  is the time rate of change of the  $f_i$  due to collisions. Eq. 2.2 is the general formulation of the Boltzmann Transport Equation, and it fully describes the time evolution of an arbitrary distribution function,  $f_i$ , which is experiencing an arbitrary force. In the case of a plasma system, this arbitrary force is described by eq 2.1, and thus by combining eq 2.1 and 2.2 the Boltzmann transport equation for plasma systems can be found:

$$\frac{\delta f_i}{\partial t} + \vec{v} \cdot \frac{\partial f_i}{\partial \vec{r}_i} + \frac{q}{m} (\vec{E} + \vec{v} \times \vec{B}) \cdot \frac{\partial f_i}{\partial \vec{v}_i} = \left( \frac{\partial f_i}{\partial t} \right)_c \quad (2.3)$$

Unfortunately, when  $\left( \frac{\partial f_i}{\partial t} \right)_c$  is specified exactly, eq 2.3 is not solvable. For this reason,  $\left( \frac{\partial f_i}{\partial t} \right)_c$  is typically estimated via some sort of approximation such as BGK[3] or Fokker Planck[4]. In the special case of no collisions, eq 2.3 can be reduced to the Vlasov Equation.

$$\frac{\partial f_i}{\partial t} + \vec{v}_i \cdot \frac{\partial f_i}{\partial \vec{r}_i} + \frac{q_i}{m_i} (\vec{E} + \vec{v}_i \times \vec{B}) \cdot \frac{\partial f_i}{\partial \vec{v}_i} = 0 \quad (2.4)$$

This special case of the Boltzmann Transport Equation, is widely used to describe the dynamics of non-collisional plasma systems.

### 2.1.3 Two-Fluid Description

The Two-Fluid framework allows for plasma systems to be described by two macroscopic fluids[2], with one fluid corresponding to the plasma electrons, and the other system corresponding to the plasma ions. This is accomplished by taking velocity moments of the Boltzmann Transport Equation outlined in eq 2.3. In general these moments are evaluated in the following way

$$\int g(\vec{v}_i) \left( \frac{\partial f_i}{\partial t} + \vec{v}_i \cdot \frac{\partial f_i}{\partial \vec{r}_i} + \frac{q_i}{m_i} (\vec{E} + \vec{v}_i \times \vec{B}) \cdot \frac{\partial f_i}{\partial \vec{v}_i} - \left( \frac{\partial f_i}{\partial t} \right)_c \right) d\vec{v}_i = 0 \quad (2.5)$$

where the function  $g(\vec{v}_i)$  denotes the type of moment that is being evaluated. Evaluating for the zeroth order moment, ( $g(\vec{v}_i)=1$ ), yields the following result

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (n_i \vec{u}_i) = 0 \quad (2.6)$$

where  $n$  is the particle density, and  $\vec{u}$  is the drift velocity. Unfortunately, the drift velocity,  $\vec{u}$  is not known without taking the next order moment.

The result from evaluating the first order moment,  $g(\vec{v}_i) = \vec{v}_i$ , is as follows

$$n_i m_i \frac{d\vec{u}_i}{dt} = n_i q_i (\vec{E} + \vec{u}_i \times \vec{B}) - \nabla P_i - R_i \quad (2.7)$$

where  $R_i$  is the frictional force due to collisions, and  $P_i$  is the pressure. In eq 2.7,  $P_i$  is expressed as a function of the second order moment, and for eq. 2.7 to be fully defined, this moment will need to be fully evaluated. However, the solution to the second order moment will be a function of the 3rd order moment, subsequently the solution to that moment will be a function of the next higher order moment. This process will carry on indefinitely, and thus a method for truncating this cycle must be utilized. This process is called closing the moment hierarchy. Though this process can be achieved via a number of different methods, it typically involves treating the highest order moment, known as the closure moment, as

a constant. An example of this can be seen in the limit of an isothermal plasma, where  $P = n_i k_b T_i$  is a constant, allowing for the hierarchy to be closed with the first order moment.

## 2.1.4 Magneto Hydrodynamics

Magneto Hydrodynamics (MHD) simplifies the Two-Fluid approach by describing the entire plasma as a single fluid instead of two[2]. This requires introduction of the current density

$$\vec{J} = \sum_i n_i q_i \vec{u}_i \quad (2.8)$$

and the center of mass velocity

$$\vec{U} = \frac{1}{\rho} \sum_i n_i m_i \vec{u}_i \quad (2.9)$$

where  $\rho$  is the mass density as described by the following.

$$\rho = \sum_i n_i m_i \quad (2.10)$$

In a method analogous to as to what was seen for the Two-Fluid formulation, velocity moments of eq 2.3 can be found to provide equations describing plasmas in the MHD framework.

The zeroth order moment leads to the mass continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0 \quad (2.11)$$

and the first order moment leads to MHD equation of motion.

$$\rho \left( \frac{\partial \vec{U}}{\partial t} + \vec{U} \cdot \nabla \vec{U} \right) = \vec{J} \times \vec{B} - \nabla P \quad (2.12)$$

## 2.2 Simulation Types

Since plasma systems span a wide range of parameters, and can be described by a number of theoretical frameworks, not all plasma systems can be modeled by a single technique. Certain computational techniques will be more effective at simulating some plasma systems than others. Fortunately there are number of different numerical methods that are available. This section will give an overview of the different types of numerical methods that are used for simulating plasma systems.

### 2.2.1 Partial Differential Equation Solvers

Often, the easiest way to simulate a plasma system is to just integrate the governing equations of that system. For kinetic models, this involves integrating a form of Eq 2.3. For fluid models it involves integrating the moments of a form of Eq. 2.3. Depending on the approximations made, there are a number of different ways that Eq. 2.3 and its resulting moments can be formulated. Since a given differential equation may lend itself more to one type of numerical techniques than another, the best method to solve Eq. 2.3, or any of its derived moments, may vary substantially based on the approximations that are used. Thus, there are numerous different computation techniques that are used to to solve differential equations in computational plasma physics[5, 6, 7]. A full accounting of these methods is well beyond the scope of this dissertation.

### 2.2.2 Particle in a Cell

Particle in a cell (PIC) algorithms are often used to simulate collisionless plasma systems[8]. These simulations chart the evolution of a phase space distribution of a plasma system. To do this, the distribution function is first discretized into series of macro particles, that represent a large number,  $N_0$ , of particles. These particles will have a charge  $q_\sigma N_0$  and mass  $m_\sigma N_0$ , and thus the charge to mass ratio of these macroparticles will be the same as that of their constituent particles.

Once the distribution is discretized and the macroparticles are initialized, the plasma system is allowed to evolve. There are two main components involved in calculating the evolution of the plasma system. First, the distribution of the macroparticles can be used to calculate the electric potential as a function position. This is accomplished by first spatially discretizing the plasma volume into a series of mesh points. Next, an effective charge density, based on macroparticle locations, is assigned to each mesh point. Finally, using a partial differential equation integration scheme, Poisson's equation can be solved, defining the electric potential at all of the mesh points.

Once the electric potential is known at each mesh point, an interpolation scheme can be used to determine the force acting on each of the macroparticles. By integrating these forces, the macroparticles are then allowed to evolve in time, in a manner equivalent to the molecular dynamic simulations (see section 2.3). For stability concerns, these force integrations are limited to small timesteps. For this reason, this process, of solving Poisson's equations and integrating the forces on the macroparticles, needs to be iterated many times.

### **2.2.3 Molecular Simulations**

Molecular simulations are simulations that explicitly model the individual particles of the system[9]. These simulations can be organized into two distinct categories. These are Monte-Carlo models and Molecular Dynamics models. Molecular dynamics simulations calculate the dynamical evolution of physical systems by integrating Newton's 2nd Law for the constituents particles in the system. These types of simulations will be fully discussed in the next section (section 2.3).

Monte Carlo simulations are a class of simulations that use randomly generated data to provide deterministic results[9, 10]. By repeatedly randomly sampling a given probability distribution, Monte-Carlo methods are able to accurately reproduce a distribution which can applied to a number of different numerical problems. Some of the most common applications of the Monte Carlo method are numerical integration, optimization problems, and

inverse problems. Additionally, Monte Carlo techniques can be used in a number of different physics contexts. Some examples of this are simulating radioactive decay, calculating mean macroscopic quantities from a partition function, and simulating random walks in physical systems.

One specific use for Monte Carlo techniques, is to simulate charged particle collisions in plasma systems. Since both the frequency of collisions and the resulting deflection of a collision can be described probabilistically, a Monte Carlo treatment of charged particle collisions can be implemented fairly straightforwardly. Additionally, a Monte Carlo implementation of charged particle collisions is much more computationally efficient than directly integrating Newton's second law, as would be done in a molecular dynamics simulation. Many of the models described in this dissertation incorporate charged particle collisions in this manner. For more information on how this is done see chapter 3 and chapter 5.

## 2.3 Molecular Dynamics

Molecular Dynamics (MD) simulations are simulations in which the dynamic evolution of physical systems are modeled by calculating the evolution of the system's constituent particles[9]. The fact that MD simulations simulate all of system's particles leads to a couple of important consequences. First, this requires a large number of calculations resulting in long computations times. This makes MD simulations untenable for physical systems with a large number of particles. Secondly, since all of the particles in the systems are modeled explicitly, MD simulations would be expected to yield very accurate results. Low-density ultracold plasma systems are characterized by a low number of particles allowing for most MD simulations to be completed in reasonable amount of time. This makes MD a good method for modeling UCPs. It is for this reason that most of the numerical modeling described in this dissertation is molecular dynamics simulations.

### 2.3.1 How Molecular Dynamics Simulations Work

The underlying physics of Molecular Dynamics simulations are not difficult to understand. In the classical limit the time evolution of particles in a system are described by Newton's 2nd law. Formally this is described as

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \vec{F}_i \quad (2.13)$$

where  $m_i$  is the mass of the  $i$ th particle,  $\vec{r}_i$  is the coordinate vector for the  $i$ th particle, and  $\vec{F}_i$  is the net force acting on the  $i$ th particle. While Eq. 2.13 is fairly simple, there are a couple of issues that need to be addressed before it can be fully implemented in a molecular dynamics simulations. First,  $\vec{F}_i$  needs to be calculated explicitly. This involves knowing the functional forms of all of the forces acting upon each particle.

Once  $\vec{F}_i$  is known Eq. 2.13 can be solved. However, in most cases, Eq. 2.13 is rewritten as two first order differential equations

$$\frac{\vec{F}_i}{m_i} = \frac{d\vec{v}_i}{dt} \quad (2.14)$$

$$\vec{v}_i = \frac{d\vec{r}_i}{dt} \quad (2.15)$$

where  $\vec{v}_i$  is the velocity vector of the  $i$ th particle. There are a number of numerical methods that can be used to solve these equations, but they all have a few commonalities. First, all of the quantities are in equations 2.14-2.15 are treated as constants. Since none of these quantities are actually constants, equations 2.14-2.15 are only integrated over time periods short enough where these quantities do not vary significantly. This period of time,  $\Delta t$ , is referred to as a timestep. Typically, MD simulations are interested in physics on timescales  $\gg \Delta t$ , and thus a number of integration steps are needed. By consecutively integrating the equations of motion in this manner, for all of the particles in a physical system, Molecular Dynamics models are able to calculate the dynamic evolution of a physical system.

### 2.3.2 Integration Techniques

There are a number of different integration techniques that can be used in MD simulations[11]. While most of these techniques are fairly simple, a number of thermostat (integrations schemes that force the system to a constant temperature) and barostat integrators (integration schemes that force a constant pressure) do exist. For purposes of this work, the discussion will be limited to simple integration schemes.

Perhaps the most simple integration scheme is Euler's method. This scheme is derived by taking a Taylor expansion of the solutions to Eq. 2.13 around a time  $t_0$ . This yields

$$r_i(t_0 + \Delta t) = r_i(t_0) + v_i(t_0)\Delta t + \frac{F_i\Delta t^2}{2m_i} + \mathcal{O}(\Delta t^3) \quad (2.16)$$

$$v_i(t_0 + \Delta t) = v_i(t_0) + \frac{F_i\Delta t}{m_i} + \mathcal{O}(\Delta t^2) \quad (2.17)$$

where  $r_i(t_0 + \Delta t)$  and  $v_i(t_0 + \Delta t)$  signify the respective position and velocity evolutions that occur in a timestep,  $\Delta t$ . While this method is simple and straight forward to implement, it is rarely used. This is because Euler's method does not properly conserve energy. For this reason, symplectic integration techniques, which by definition conserve energy, are more frequently used.

The most basic symplectic integrator is verlet integration. In addition to properly conserving energy, verlet integration also has the benefit of being time reversible, a property also not present in Euler's method. To determine the velocity and position functions for this scheme, it is necessary to take another Taylor expansion of Eq. 2.13, this as a function of  $-\Delta t$ . This yields the following equation.

$$r_i(t_0 - \Delta t) = r_i(t_0) - v_i(t_0)\Delta t + \frac{F_i\Delta t^2}{2m_i} + \mathcal{O}(\Delta t^3) \quad (2.18)$$

By combining equations 2.16 and 2.18, the functional form of  $r(t_0 + \Delta t)$  can be described as.

$$r_i(t_0 + \Delta t) = 2r_i(t_0) - r_i(t_0 - \Delta t) + \frac{F_i \Delta t^2}{m_i} + \mathcal{O}(\Delta t^3) \quad (2.19)$$

Since Eq. 2.19 does not have a velocity dependence, particle trajectories can be calculated without the knowledge of velocities. However, Eq. 2.19 depends not just on the current particle position, but on the previous position of the particle as well. This is typically inconvenient, and for this reason an alternative construction, the Velocity Verlet algorithm is more typically used. This alternative formulation is as follows.

$$r_i(t_0 + \Delta t) = r(t_0) + v(t_0)\Delta t + \frac{F_i(t_0)\Delta t^2}{2m_i} + \mathcal{O}(\Delta t^3) \quad (2.20)$$

$$v_i(t_0 + \Delta t) = v(t_0) + \frac{(F_i(t_0 + \Delta t) + F_i(t_0))\Delta t}{2m_i} + \mathcal{O}(\Delta t^2) \quad (2.21)$$

where the equations 2.20 and 2.21 are algebraically consistent with Eq. 2.19. By using this formulation, particle trajectories are no longer dependent on their previous positions.

The final integration method that will be discussed is the Leapfrog method. In this method particle positions and velocities are calculated at different discrete moments in time. While, the time spacing,  $\Delta t$ , between two different position calculations is the same as spacing between two different velocity calculations, the velocity calculations are shifted in time away from position calculations by a time of  $\Delta t/2$ . This, in essence, leads to the position calculations "leapfrogging" velocity calculations in time. Formally, this leads to the following functions.

$$r(t_0 + \Delta t) = r(t_0) + v(t_0 + \Delta t/2)\Delta t \quad (2.22)$$

$$v(t_0 + \Delta t/2) = v(t_0 - \Delta t/2) + \frac{F_i(t_0)\Delta t}{m_i} \quad (2.23)$$

Leapfrog integration yields the same results as a the Verlet scheme. It is both symplectic and time-reversible, and requires less computational overhead than the Verlet method. It

is for these reasons, that most of the MD simulations presented in this dissertation use the Leapfrog method of integration.

### 2.3.3 Energy Conservation and Stability

One of the main criteria for determining the stability of a molecular dynamics simulation is quantifying how well it conserves energy. Even with the use of symplectic integrators, accumulated errors will slowly lead to changes in the total energy in molecular dynamics simulations. The magnitude of these errors corresponds directly to the length of a timestep in a simulation, where longer timesteps lead to larger errors. This is a result of integration schemes treating forces as constant over the length of the timestep. The longer the timestep chosen, the worse this constant force approximation becomes and thus the greater the error introduced into the system.

If the errors in a simulation become too large, oscillatory motion can become unstable. While symplectic integration schemes are never able to exactly calculate particle orbits, sufficiently small timesteps will well approximate particle trajectories and frequencies. However, for longer timesteps orbits can become unstable, yielding vast discrepancies between the simulated orbit and the actual orbits.

To avoid simulation instabilities, it is necessary to quantify the size of timesteps at which these instabilities begin to occur. To do this, we can consider a single particle system in a harmonic potential[12]. The force acting on this particle is

$$F = -m\omega^2x \tag{2.24}$$

where  $\omega$  is the frequency of oscillation of the particle. By combining equations 2.22-2.24, and by making a change of variables where  $p = mv$ , the following system of equations can be written.

$$\begin{bmatrix} r(t_0 + \Delta t) \\ p(t_0 + \Delta t) \end{bmatrix} = \begin{bmatrix} 1 - \frac{\omega^2\Delta t^2}{2m} & \frac{\Delta t}{m} \\ \frac{\omega^2\Delta t^3}{4m} - \omega^2\Delta t & 1 - \frac{\omega^2\Delta t^2}{2m} \end{bmatrix} \begin{bmatrix} r(t_0) \\ p(t_0) \end{bmatrix} \tag{2.25}$$

The eigenvalues this system of equations are

$$1 - \frac{\omega^2 \Delta t^2}{2} \pm \frac{\omega \Delta t}{2} \sqrt{\omega^2 \Delta t^2 - 2} \quad (2.26)$$

If  $\omega^2 \Delta t^2 > 2$ , at least one of these eigenvalues will be greater than 0, making the system unstable. If  $\omega^2 \Delta t^2 < 2$ , both of the eigenvalues will be complex and will have a magnitude of 1, making the system stable. For this reason, the stability criterion for a molecular dynamics simulation using the Leapfrog integration scheme is

$$\Delta t < \frac{\sqrt{2}}{\omega} \quad (2.27)$$

where  $\omega$  corresponds to the fastest timescale present in the simulation.

### 2.3.4 Difficulties with Plasma Systems

One of the major difficulties with using the Molecular Dynamics modeling to study plasma systems is the long range nature of the Coulomb force. In Coulomb systems, there exists no natural truncation distance for which particles can be excluded from force calculations. This means that each particle in the system interacts with every other particle in the system. This is especially problematic since as the number of particles in the system,  $N$ , increases, the number of total pairs in the system and thus the number of force calculations increases by a factor of  $N^2$ . This leads to MD simulations of Coulomb systems to quickly become untenable.

The lack of a natural cutoff point in Coulomb systems can be understood by utilizing the following example. Consider a test charge,  $q_T$ , located at the origin surrounded by a charge density,  $\rho(r)$ . The force acting on  $q_T$  is

$$F = \int_V \frac{-q_T \rho(r) \vec{r}}{4\pi \epsilon_0 r^3} d^3r \quad (2.28)$$

where  $\epsilon_0$  is the permittivity of free space. If we assume that  $\rho(r)$  is only nonzero for  $y > 0$  or  $0 < \phi < \pi$ , but otherwise only depends on  $r$ , eq 2.28 yields

$$F = \int_0^\infty \frac{-q_T \rho(r) \hat{y}}{2\epsilon_0} dr \quad (2.29)$$

where the only remaining  $r$  dependence in the system comes from the  $\rho(r)$  term, since the  $r^{-2}$  dependence of the Coulomb force is canceled out by the  $r^2$  dependence of the differential volume element  $d^3r$ . Unless  $\rho(r)$  falls off at a rate faster than  $r^{-1}$ , Eq. 2.29 yields an infinite force. Also, the distance dependence from the origin has disappeared from the integrand. Thus, distant charge density is contributing as much as nearby charge density in determining the total force. These considerations imply that all of the charges in the system make a non-negligible contribution to the force acting on  $q_T$ .

Another common difficulty arising in MD simulations of Coulomb systems is the problem of unlike charges. Most plasmas contain both electrons and ions. Since electrons are attracted to ions, bound states, where an electron orbits an ion, are often created. While this is not a problem for long orbital periods, bound states with short orbital axes are characterized by having very fast orbital frequencies. Bound state orbits orders of magnitude faster than the timescales associated with the simulation's relevant physics can therefore be problematic. If the simulation's timestep size is tuned to match the relevant physics, fast bound state orbits will lead to large conservation of energy errors. If the timesteps are tuned to match the orbits, the simulations will be orders of magnitude slower than what the relevant physics would suggest is necessary.

### 2.3.5 Computation Time Reducing Approximations

Since the  $\mathcal{O}(N^2)$  nature of Coulomb interactions can make Molecular Dynamics simulations untenable for a large number of particles, a number of methods have been developed to increase the speed of computation. One method for accomplishing this is by making force

approximations that yield more favorable number scaling. Another method for accomplishing this is the use of periodic boundary conditions, that allow for a reduction in the total number of particles in the system. This subsection will briefly cover both of these methods.

One of the main classes of force approximation algorithms are tree methods, such as the Barnes-Hut method. These algorithms get their name from the octrees that they employ to structure the particles in a system. Octrees are tree data structures that recursively subdivide a cube into eight evenly spaced cubes. By assigning particles to their proper node in an octree spanning the volume of the system, particles are grouped with other nearby particles. This allows for the force on a given particle from far away particles to be approximated by a multipole expansion. For nearby particles, force calculations are still treated explicitly with the proper Coulomb force. By using this method, computational scaling can be improved to  $\mathcal{O}(N \log(N))$ [13].

The other main method for approximating the Coulomb force comes from the Particle-Mesh method. In a manner similar to PIC codes, the volume of the system is discretized into spatial mesh points. By solving Poisson's equation the electric potential can be solved for at all of the different mesh points, and by taking a derivative the electric field values are found as well. Once the field is known at the mesh points, calculating the forces on the remaining particles becomes an  $\mathcal{O}(N)$  problem. However, this does not include the full computational requirements necessary, since it takes a nontrivial amount of time to solve Poisson's equation.

While these methods can often lead to significant computational speed increases, none of these approximations were used in the simulations discussed in this dissertation. There are a number of reasons for this. First, these algorithms are typically more difficult to program, and sufficient computing power was available to allow for the explicit simulation of every particles in the systems of interest. Second, unlike directly calculating all pair interactions, these algorithms are not naturally parallelizable. This means that considerable effort would need to be expended to adapt these algorithms to accommodate parallel architecture, and

such adaptations would likely be significantly less efficient than their ideal single core scaling would suggest. Finally, while these methods scale more efficiently with increasing particle number, they can require significant computational overhead. Thus for sufficiently low particle numbers these methods can actually be less efficient than straight forwardly calculating all of the Coulomb interactions. This fact, combined with the inefficiencies that would be introduced from parallelization, bring into question how much extra computational efficiency could actually be obtained with these methods. It is for all of these reasons that none of these techniques were utilized in the model discussed later in this chapter.

Periodic boundary conditions are another method used to reduce the computational intensity of plasma simulations[9], by allowing for a large reduction in the number of particles in the system. In a simulation with periodic boundary conditions, the volume of the system is subdivided in to series equally sized cubes. Only the evolution of the particles in a center cube are explicitly calculated, and all of the other cubes are assumed to have the same particle distribution as the center cube. Systems with periodic boundary conditions have two additional considerations not present in MD models without boundary conditions that need to be addressed. The first issue is how to properly treat particles that pass out of the center cube into a neighboring cube. This problem is typically handled by having the particle reappear on the other side of the cube. Formally, this can be described as  $\vec{r}_i \Rightarrow \vec{r}_i - L\hat{n}$ , where  $L$  is the length of the cube, and  $\hat{n}$  is the unit vector normal to the face of the cube that the particle passes through. The second issue is how to properly deal with force contributions from off-center cubes. Formally, the force exerted on the  $i$ th particle in the center cube can be described by

$$\vec{F}_i = \sum_{j \neq i}^N \sum_{k=0}^{N_c} \frac{q_i q_j (\vec{r}_i - \vec{r}_j - \vec{R}_k)}{4\pi\epsilon_0 |\vec{r}_i - \vec{r}_j - \vec{R}_k|^3} \quad (2.30)$$

where  $\vec{r}_j$  is the position vector of the  $j$ th particle in the center cube,  $R_k$  is position vector corresponding to the center of the  $k$ th cube, and  $N_c$  is the total number of cubes in the model.

The biggest limitation of using periodic boundary conditions is that it assumes an isotropic system. Since ultracold plasmas typically have a Gaussian density distribution, it would be inappropriate to use periodic conditions to model such a system. It is for this reason that all of the modeling described in this dissertation uses open boundary conditions.

## 2.4 Programing With Graphics Processing Units

Programing with graphics processing units (GPUs) can offer large speed increases over traditional CPU based programing. This extra computational power does come at the expense of additional complexity and decreased flexibility. This tradeoff is often more than worthwhile, since in some instances GPU based algorithms can be multiple orders of magnitude faster than traditional CPU based single threaded algorithms. GPUs were an integral component in many of the models that I have developed. The speed increases coming from using GPUs allowed for the one-to-one modeling that was a necessary component in finding many of the results that are presented in this dissertation. This section will give an overview of the GPU programming model.

### 2.4.1 Introdtion to GPGPU

The advent of general purpose computing on graphics processing units (GPGPU) is a relatively recent phenomenon[14]. Traditionally, GPUs had been exclusively used for calculating and displaying digital images. While that is still the how most GPUs are used, the scope of problems that are tackled by GPUs is continually expanding. There are two main trends that have led to the expansion of GPGPU. The first of these is the somewhat recent rapid slowdown in speed increases for single CPU. The second factor is the rapid increase in the number of computing cores contained in a single GPU. The combination of these two trends has resulted in modern CPUs not being much faster than CPUs from ten years ago, while Modern GPUs are much faster than their ten year old counterparts. As it stands now, the computing power of a single GPU far exceeds the computing power of a single CPU.

The massively parallel nature of GPU architecture is the reason why GPUs have so much more potential computing power than modern CPUs. A single computing core in a CPU typically has more computing power than a single computing core in a GPU. However, modern GPUs typically have a few thousand computing cores, while a modern CPU rarely has more than 8. This massive difference in the number of computing cores more than makes up for the relative difference in the single core computing power.

While GPUs are more powerful than their CPU counterparts, they are also less flexible. The massively parallel architecture of GPUs requires algorithms to be written in a way that takes of advantage of their parallel structure. Since many computing task do not obviously lend themselves to parallelism, this can be very challenging to accomplish. For this reason, the parallelization of traditionally serial algorithms often results in large increases in algorithm complexity. Even when well thought out, parallelized algorithms rarely result in speed increase comparable to what would naively be expected when just looking at the difference in raw computing power.

There are two main programing languages that are used to program GPUs. These are Cuda and OpenCL. Of the two, Cuda is simpler, and has more supporting documentation and supporting libraries. However, since Cuda was developed by Nvidia, it is only compatible with Nvidia GPUs. Since, the graphics cards in our lab are all made by AMD, I had to use OpenCL for my simulations. The OpenCL programming model will be explored in depth in the upcoming subsections.

## **2.4.2 GPU Architecture**

While there are significant variations in architecture across different GPU models, all GPUs have a number of features in common. The first of these features has to do with memory. All discrete graphics cards have their own RAM, called global memory, and for this reason GPU cores are not able to access system ram directly. This means that if the GPU needs to use information located on system RAM, that information first needs to be

transferred to GPU global memory. There is significant latency involved in this process, so it is best not to do this too often.

GPUs contain a hierarchy of computational elements. While this hierarchy varies substantially with the model, most GPUs have a basic computational unit that contains a few, around 4-5, processing elements as well as some local registers. In Nvidia GPUs these are often called Cuda cores, and in AMD GPUs they are often called stream cores/processors. Clusters of these units are combined with shared memory and sometimes with L1 cache. These clusters are typically called compute units in AMD GPUs, and stream multi-processors in Nvidia GPUs. The shared memory and L1 cache of these clusters allow for the different computing elements to communicate with each other. Compute units in different clusters are unable to do this.

### **2.4.3 OpenCL Programing Model**

The OpenCL computing language was devised to be a single language capable of simultaneously interacting with all different types of devices[17, 18, 14]. Unlike Cuda, OpenCL is not limited to using a subset of GPUs, or even just using GPUs in general. It is capable of utilizing the computing power of any existing processing unit. Furthermore, the OpenCL language has no issues simultaneously deploying the resources of vastly differing processor types. For example, it would be no problem to use an Nvidia GPU in conjunction with both an AMD GPU and an Intel processor to solve a single problem. For this reason, OpenCL offers a great deal of flexibility for solving a variety of problems.

Since OpenCL provides a great deal of flexibility for different hardware configurations, a great degree of abstraction is needed to properly describe the vast hardware possibilities. This abstraction, by necessity, leads to a complicated assortment of different terms and data structures that define the hardware conditions being used by the program. These terms are listed and explained below.

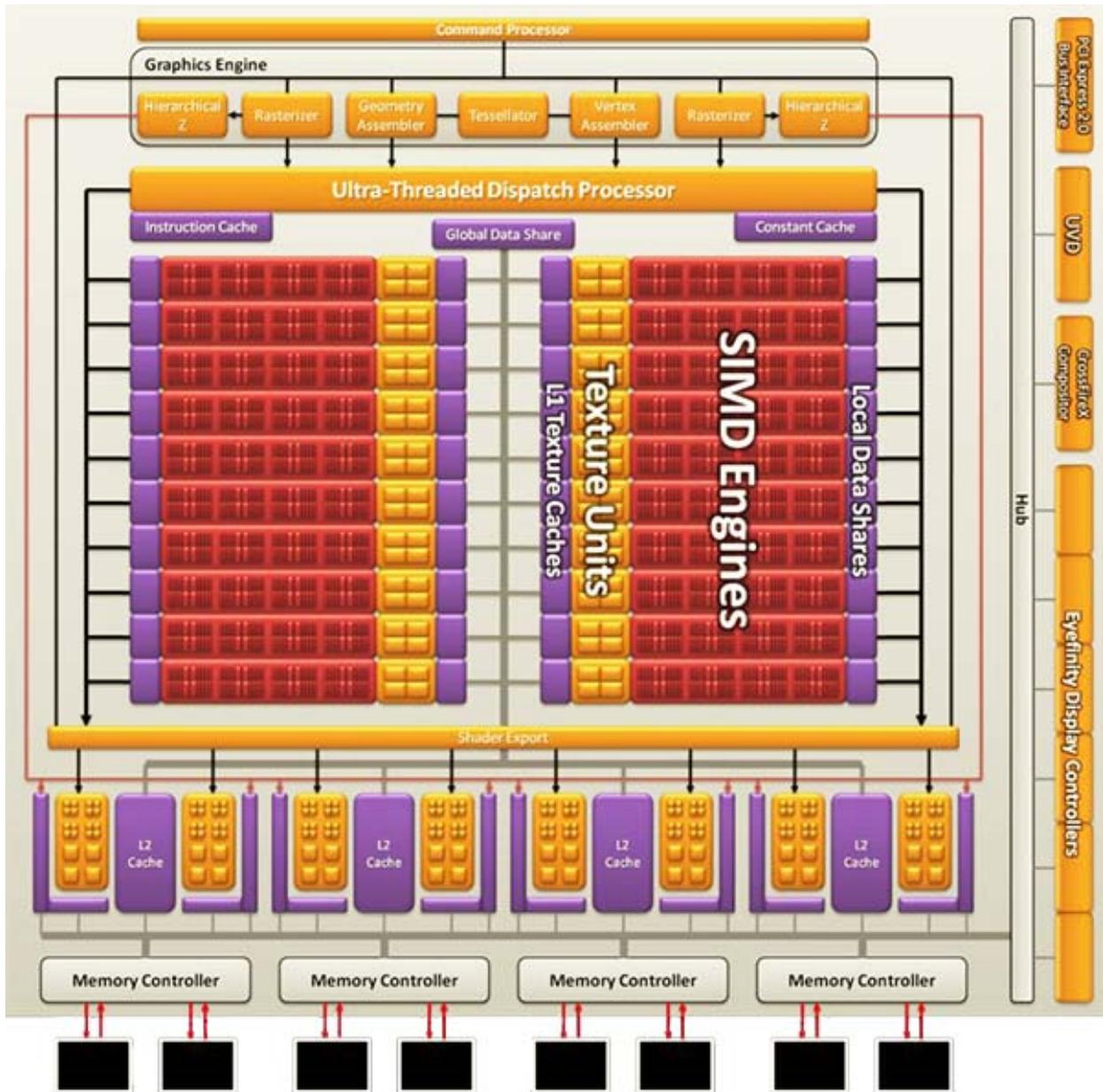


Figure 2.1: Block diagram of a Radeon HD 5870 graphics card[15, 16]. The rectangles in the center of the diagram are SIMD engines which are also known as compute units. As can be seen in the diagram, this particular GPU contains 20 compute units.

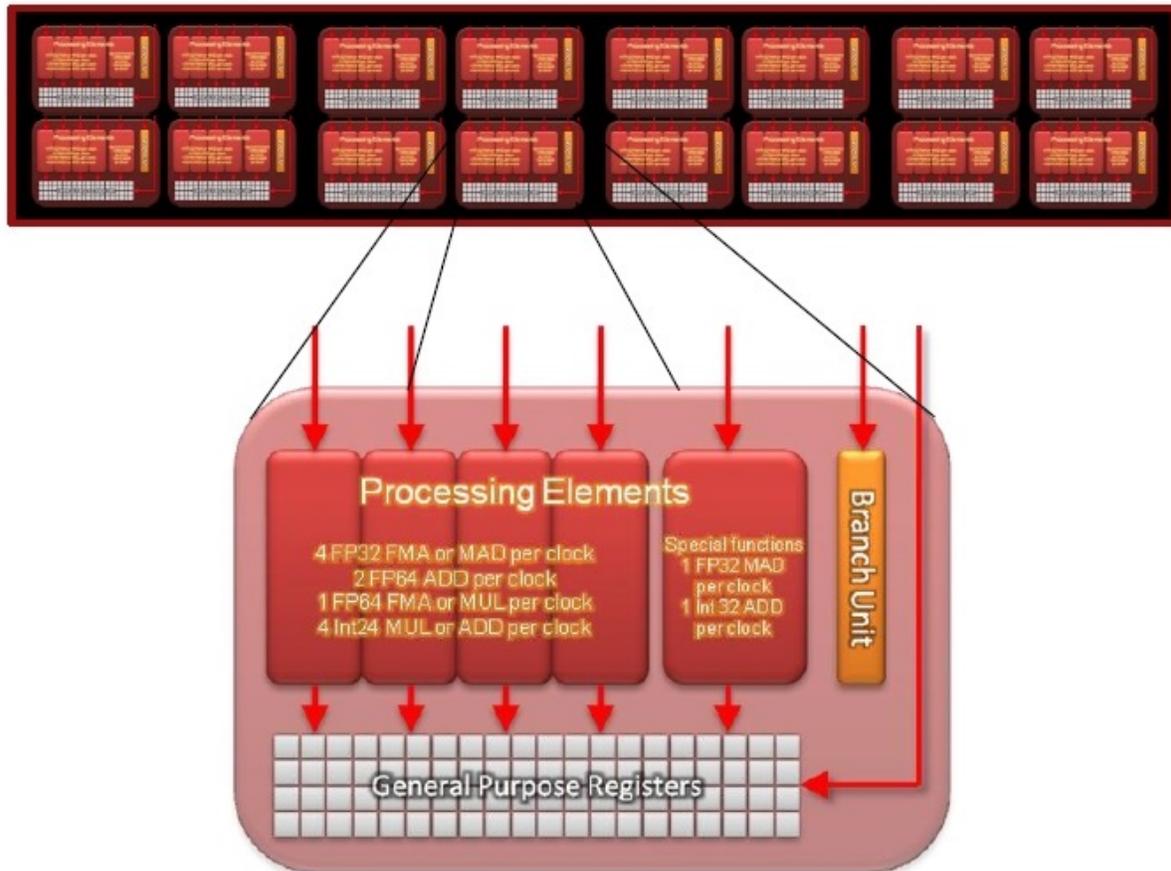


Figure 2.2: Block diagram of one of the SIMD Engine from the HD5870[15, 16], as seen in figure 2.1. The figure shows that a single SIMD Engine, also known as a compute unit, contains 16 stream processors. The figure also shows a zoomed diagram of a single stream processor. It can be seen that each one of these stream processors contains 5 processing elements, and thus an entire compute unit contains 80 processing elements for this particular GPU.

**Host:** The main CPU utilized to organize the flow of an OpenCL program. The host specifies what devices will be in use, and directs their activities.

**Device:** In general, the device is a processing unit of some kind. Typically this is a GPU, but a CPU can also be used. Devices are typically the workhorses of any OpenCL program. At the host's direction, devices will execute kernels and feed the results back to the host. In the OpenCL framework, devices are assumed to be composed of a number of compute units, and each of these compute units are assumed to be composed of a number of processing elements.

**Platform:** Since devices are made by many different companies, they will have different software that governs their behavior. To be able to interface with a device, the proper platform has to be specified. However, before any platform can be used, the necessary software development kit (SDK) needs to be installed.

**Context:** The context data structure accomplishes two main things. First, it establishes a group of devices to work together. Once devices are grouped in a context, OpenCL will treat them as though they are a single pool of computing resources. The second function of the context class is to link devices with their proper platform. Without creating an instance of the context class, devices would not be able to function, since they would not be linked to the appropriate platform. It should be noted that each context can only use a single platform, thus contexts can only contain devices of the same platform type.

**Queue:** The queue is the intermediary by which the host provides tasks to the device. Devices will execute tasks in the order in which they were added to the queue. Each queue is associated with a single context.

Once a context, and its associated queue, have been established the host is able to relay commands to the appropriate devices via the queue. The method by which those commands are made, as well as how they are executed, will be discussed in the next subsection.

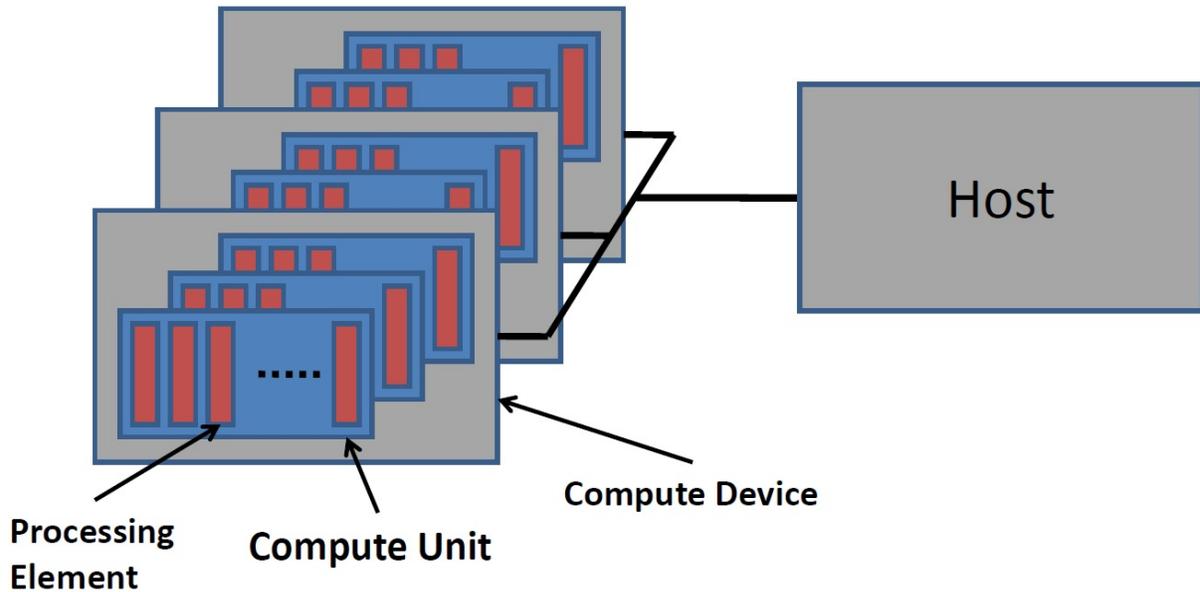


Figure 2.3: A basic diagram of the OpenCL computing model[16]. The host feeds instructions an arbitrary number of devices. In turn the device gives out tasks to a number of compute units which in turn distribute tasks to their component processing elements.

#### 2.4.4 Device-Side Parameterization and Kernel Execution

Kernels are subroutines that are launched by devices during the course of an OpenCL program. Typically the execution of kernels by devices comprises the bulk of the computations in a given OpenCL program, thus making kernels integral to leveraging the power of OpenCL. This subsection will discuss how these kernels are written, and how there are mapped onto devices.

To understand how kernels differ from traditional serial subroutines it is easiest to use an example. Below is a traditional serial implementation of a simple algorithm.

```
void scaler_multiply(float *x, float *y, float k, int N)
{
    int i;
    for(i=0; i<N; i++){
        y[i]=k*x[i];
    }
}
```

This simple C code is used to multiply every element in an array  $x$  of length  $N$  by a constant value  $k$ . The results of this calculation are then stored in the array  $y$ . Since this code

contains a number of calculations that are completely independent from each other, it is easily parallelizable. For this reason, this code can be easily adapted to run as a kernel in the following way.

```
__kernel void scaler_multiply(__global float *x, __global float *y,
float k, int N)
{
    int i=get_global_id(0)+get_global_id(1)*get_global_size(0);
    if(i<=N){
        y[i]=k*x[i];
    }
}
```

There are a number of differences between serial implementation, and the parallel kernel implementation of this code. First is that there is now a `__kernel` at the beginning of the function declaration to signify to the compiler that this is a kernel function. Next, the `*float` data types have been redefined as `__global *float` data types. This means that these pointers are pointing to a location on device global memory, and not to a location on the host's system memory. This is necessary, since the most devices will only have direct access to their own global memory.

Perhaps the biggest change between the two codes, is the absence of any loop in the kernel implementation of `scaler_multiply`. Instead, the kernel will simultaneously execute `N` processing elements. These processing elements are referred to as work items in the OpenCL language. Each of these work items are identified by a single distinct global id. By properly utilizing these ids, each work item can be associated to a single array element.

In OpenCL, work items are represented by a `N` dimensional array, where `N` can be 1-3. The shape of this array can be arbitrarily chosen to be the most convenient for the task at hand. For a given work item, the function `get_global_id(i)`, is used to determine where the work item lies on the `i`th axis of the `N` dimensional work item array. An example of how this function is used can be seen in the `multiply_scaler` kernel, where this function is used to determine the value of the integer, `i`.

```
int i=get_global_id(0)+get_global_id(1)*get_global_size(0);
```

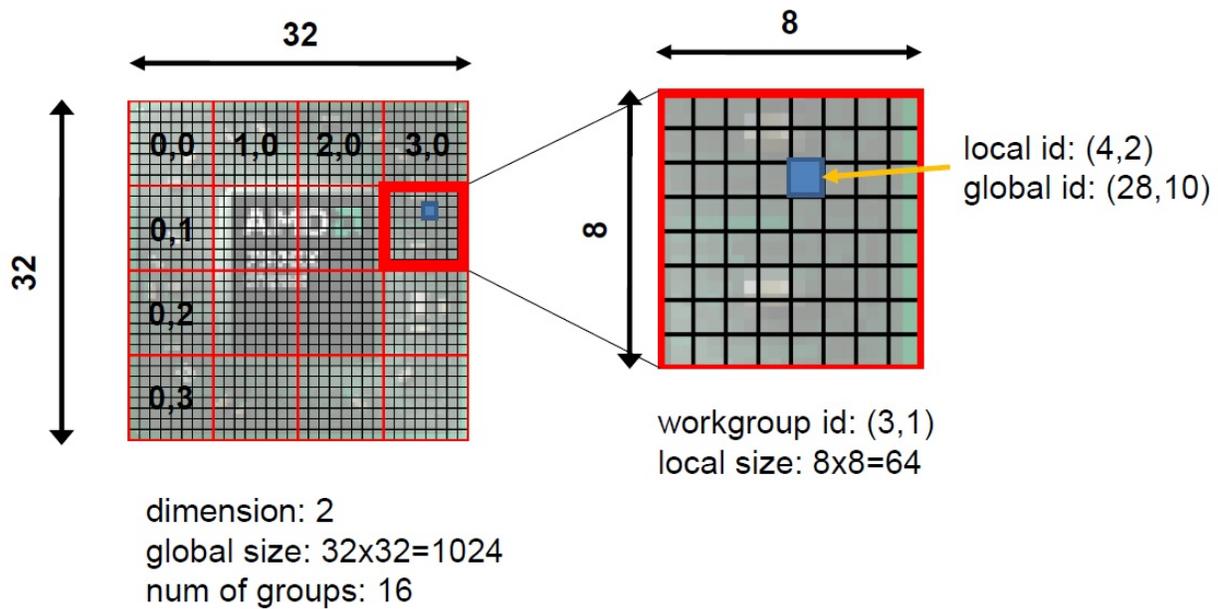


Figure 2.4: Diagram of the launch conditions for a given kernel with 1024 work items and 16 work groups. The launch conditions are configured with work items and work groups oriented in a 2D square array[16]. The small black boxes in the figure represent work items and the larger red boxes represent work groups. In a kernel, workgroups can be identified by their workgroup id, which identifies where they are located on a device. Workitems can be identified either by their global id, which identifies where they are located on a device, or by their local id, which identifies where they are located in a given workgroup. The figure illustrates how these ids are structured for the given launch conditions. It is worth noting that the geometry of this launch condition is arbitrary, and OpenCL allows for 1024 workitems to be launched in any 1-3 dimensional array with 1024 elements.

Here the `get_global_size(0)` function defines length of the 0th axis. If it is assumed that the work items are structured in a 2 dimensional array, it is easy to see that each work item will have a different value of `i`, and that all values of `i` from 0 to `N-1` will be covered. It should be noted that, depending on the geometry of the work items, there may be more than `N` work items that are launched. It is for this reason that the `if` statement in `scaler_multiply` exists, which prevents the kernel from reading from and writing to array elements that don't exist.

Work items in OpenCL are grouped into collections of work items, called work groups. Within a work group work items are able to communicate with each other via shared memory,

whereas work items from different work groups are unable to communicate. Similar to the global work item array, work groups can be composed of an arbitrary geometry of work items that best suits the problem at hand. Additionally, work items in a work group can be identified by their local id, an identification scheme analogous to the global id but limited to a single work group, as well as their global id. For more information on work items and work groups are related see figure 2.4. However, there is one major limitation to work group geometry, and that is that the total number of work items contained in a work group cannot exceed the maximum work group size that is determined by the device. This maximum size is determined by the number of processing elements contained in a single compute unit. If this number were to be exceeded, the the work group would have to span more than one compute unit, and thus would not be able to share shared memory between all of the processing elements in the work group. Since, the concept of work groups is an OpenCL abstraction of generic device hardware, it is possible to chose a work group size that is smaller than the maxium allowed by the device. However, this will typically lead to computational slow downs and is not normally recommended.

OpenCL also assumes an abstract memory hierarchy for generic devices. At the bottom of this hierarchy is the host memory which typically corresponds the host system's RAM. Data from host memory can be directly transfered to a large pool of device memory, known as global memory. Kernels have direct access to global memory, but reading form and writing to global memory is typically computationally costly. For this reason data from global memory can be transfered to smaller and faster memory reservoirs. At the work group level there are reservoirs of memory, called local memory, that are accessible to all the work items in the work group. Additionally, individual work items also have there own small individualized memory source, called private memory. A full diagram of the OpenCL memory model can be seen in figure 2.5.

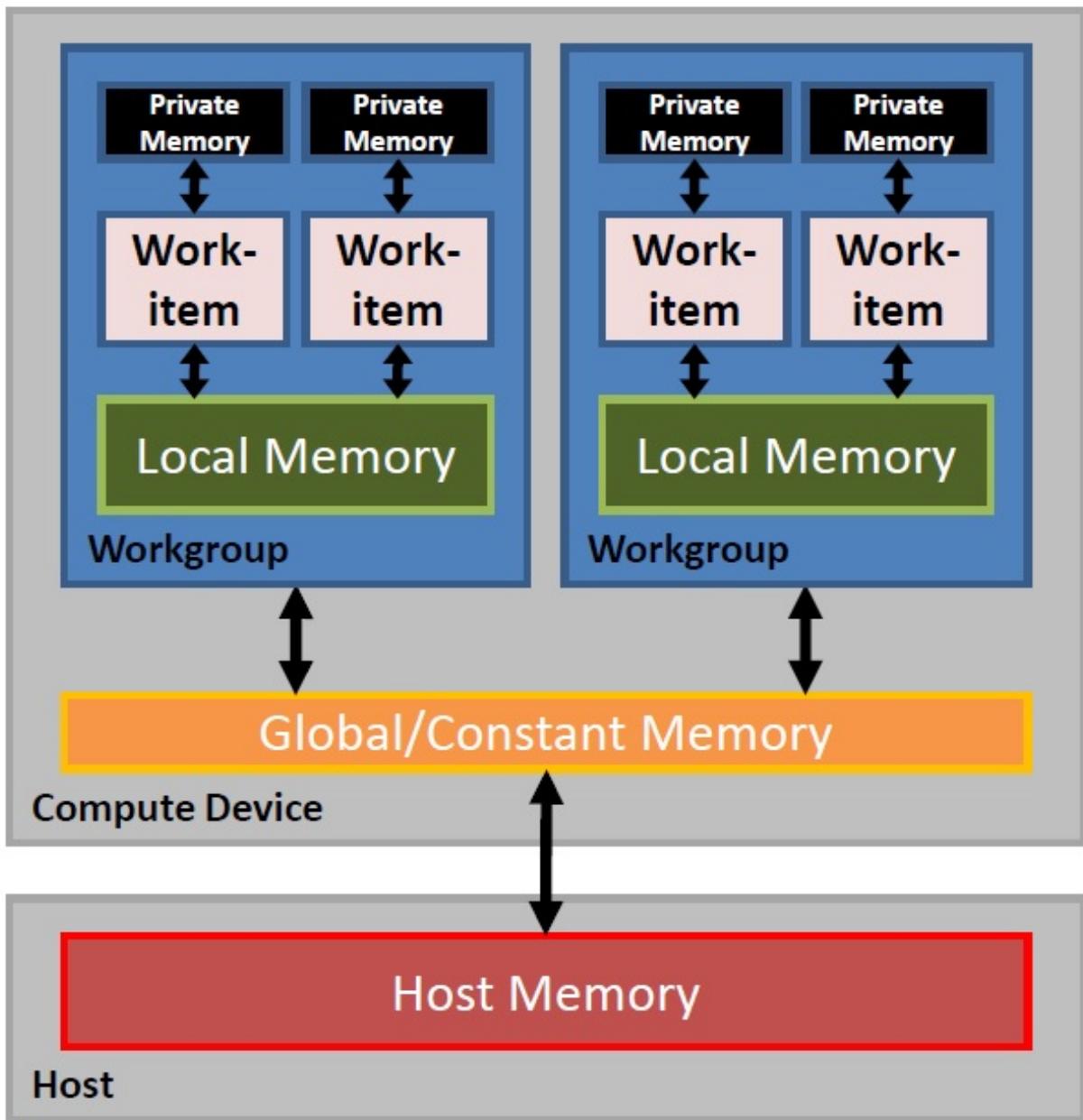


Figure 2.5: Block diagram of the OpenCL memory model[16]. For data to be used by devices it must be first be transferred from host memory to the global memory of a given compute device. OpenCl devices are subdivided into workgroups, and OpenCl designates local memory to each of these workgroups. Workgroups are further subdivided into workitems, each of which is given access to private memory.

## 2.4.5 PyOpenCL

PyOpenCL is a Python wrapper for the OpenCL standard, which allows for the use of OpenCL in a Python environment. Programming in the Python languages offers a number of different advantages. Python is an easy and flexible language that allows for complex programming statements to be made by relatively few lines of code. Furthermore, there are a vast array of easily implementable libraries available for Python, making it relatively easy to tackle new challenges on the Host side of an OpenCL program. The main shortcoming of Python is that it is relatively slow when compared to more traditional languages such as C, C++, and Fortran. However, PyOpenCL only maps the Host side OpenCL functions to the Python framework. For this reason, OpenCL kernels still need to be written in C, meaning that there will not be a drop off in kernel speed using the PyOpenCL framework. Since in an OpenCL program, the vast majority of computational work is offloaded to the devices in the form of Kernels, operating in the Python programming environment should typically have little effect on the speed of an OpenCL program. It is for all of these reasons that all of the OpenCL based numerical models discussed in this dissertation used the PyOpenCL wrapper.

To understand how the PyOpenCL wrapper can be implemented, it is probably easiest to use an example. The following code will be an implementation of the `scaler_multiply` kernel presented in the previous section.

```
#import the necessary libraries
import pyopencl as cl
import pyopencl.array as cl_array
import numpy as np

#establish kernel values in host memory
N=1000000
x=np.ndarray(N, dtype=np.float32)
x.fill(2.)
k=2.
y=np.ndarray(N, dtype=np.float32)

#setup the hardware condtions for OpenCL
ctx = cl.create_some_context()
queue = cl.CommandQueue(ctx)
```

```

#transfer host arrays to the device
xd = cl_array.to_device(queue, x)
yd = cl_array.to_device(queue, y)

#compile the kernel
src="""
__kernel void scaler_multiply(__global float *x,__global float *y,
float k, int N)
{
    int i=get_global_id(0);
    if(i<N){
        y[i]=k*x[i];
    }
}
"""
prog=cl.Program(ctx , src )
prog.build()

#run the program
prog.scaler_multiply(queue ,(N,1) ,(256,1) ,xd.data ,yd.data ,np.float32(k) ,
np.int32(N))

#retrieve the results
cl.enqueue_copy(queue ,y ,yd.data)

```

There is a lot to unpack in this program, so it is best to go step by step. The first portion of the code is used to import the relevant libraries.

```

import pyopencl as cl
import pyopencl.array as cl_array
import numpy as np

```

where `pyopencl` is the main library that handles the Python implementation of OpenCL and `pyopencl.array` is a library that handles the device arrays. This library can be used to directly create arrays on a device, transfer data from a host array to a device array, or to retrieve data from a device array and copy it to an existing Host array. The next portion of the code involves creating required data structures, necessary for `scaler_multiply`, on the host side.

```

N=1000000
x=np.ndarray(N, dtype=np.float32)
x.fill(2.)
k=2.
y=np.ndarray(N, dtype=np.float32)

```

Here N sets the length of the arrays, and k is the scaler factor to be used later in the kernel. x and y are arrays of length N. All of the values of x have been set to 2 while y has been left as an empty buffer. Next, the hardware context and an associated queue are established.

```
ctx = cl.create_some_context()
queue = cl.CommandQueue(ctx)
```

Here, ctx is the context object, and queue is the queue. The function, cl.create\_some\_context, is used to identify the available devices and platforms, and user input is used to select the appropriate values.

Once the appropriate hardware context is established the host arrays can be transferred to device global memory with the following two lines of code.

```
xd = cl_array.to_device(queue, x)
yd = cl_array.to_device(queue, y)
```

Since kernels cannot be run in the Python environment, the kernel has to be compiled separately.

```
src="""
__kernel void scaler_multiply(__global float *x,__global float *y,
float k, int N)
{
    int i=get_global_id(0);
    if(i<N){
        y[i]=k*x[i];
    }
}
"""
prog=cl.Program(ctx,src)
prog.build()
```

The string src contains all of the appropriate text necessary for it to be compiled as a stand-alone program. The instance of the class cl.Program, is used to link to the program in src with the context, ctx, and by calling the function .build() the program in src is compiled.

Once the scaler\_multiply kernel has been compiled, it can be called in the following way.

```
prog.scaler_multiply(queue,(N,1),(256,1),xd.data,
yd.data,np.float32(k),np.int32(N))
```

There are a couple of important things to take away from this line of code. First, the first three arguments in the `scaler_multiply` kernel call are used to specify kernel launch conditions, and are universal to all kernel function calls. The first argument specifies the queue in use, the second and third arguments specifies the size and shape of the work items and work groups respectively. All of the subsequent arguments correspond to the arguments required for the kernel function itself. While python is very flexible with data types, C is much more rigid. For this reason, care needs to be taken to make sure that the appropriate data types are fed to the kernels. This has two main implications for the `scaler_multiply` kernel. First, the kernel requires two array pointers. Pointers are not native to Python, but fortunately PyOpenCL provides the framework by which device pointers can be accessed. For a given device array, `a`, the corresponding pointer for that array is, `a.data`. Thus `xd.data` and `yd.data` are pointers to the device arrays `xd` and `yd` respectively.

The second issue has to do with data type precision. In Python, real numbers and integers default to 64 bit precision, but in C, real number and integer data types defaults to 32 bit precision. For this reason, there needs to be an explicit conversion of 64 bit data types to 32 bit data types. This is accomplished by the `np.float32()` and `np.int32()` functions for real numbers and integers explicitly. After all of the data types were properly converted, and the kernels were launched successfully, the results of the kernel calculations can be seen by retrieving the results array, `yd`. This is done with the last line of code

```
cl.enqueue_copy(queue, y, yd.data)
```

where the `cl.enqueue_copy()` function is used to transfer the device array data back to the corresponding host array.

In short, the process of using PyOpenCL to make numerical calculations can be summarized by the following bullet points.

1. Host side arrays are initialized using numpy.
2. Device contexts and their associated queues are established.

3. Device side arrays are created, and information is copied to these arrays from their corresponding host side arrays.
4. The kernel code is compiled.
5. The kernel is executed with the appropriate launch conditions.
6. The results of the kernel calculation are transferred from the device side arrays back to their corresponding host side arrays.

### 2.4.6 General Guidelines for Effective GPU Programing

GPUs offer the ability to vastly increase the computing power of the computing systems. However, to fully utilize the computational ability of GPUs, there are a number of general guidelines[19, 20, 21]. This subsection will give a brief overview of these guidelines, as well as discuss their relative importance.

For most parallelizable code, the largest speed increases come from simply modifying single threaded code to parallelized GPU code. Without using any optimization techniques, parallelizing single threaded code often leads to order of magnitude increases in computation speed. These increases dwarf the speed increases that can usually be obtained by other optimization techniques, and for this reason developing poorly optimized GPU code is often sufficient for many applications. This is especially true in the context of PyOpenCL. Python has a higher level of abstraction than more traditional languages like C or C++. Additionally, unlike C or C++, Python code is not compiled before execution, requiring the code to be built during runtime. For these reasons, Python is a much slower language than C or C++. However, this is not true of the OpenCL kernels that are executed during a PyopenCL application. These kernels are precompiled just as any piece of C code would be, and thus run just as fast they would in a C environment. This magnifies the difference in computational speed between single threaded host side computations and parallelized device side kernels in PyOpenCL.

While kernel optimization is not always necessary, it can be often be useful. One simple optimization technique is to maximize the number of processing elements that the GPU is using at any given time. By minimizing the amount of time that processing elements lay idle, the GPU is able to work as closely to maximum efficiency as possible. In general, this means two things. First, optimal GPU code should be sufficiently parallelized so that all processing elements are assigned tasks. If some of the processing elements are left idle, the computational capabilities of the GPU are not being fully utilized. For this reason it is often a good idea to think carefully about how a problem can be further subdivided. Second, tasks should be equitably distributed to work items, so that all work items complete their tasks in roughly the same amount of time. If a single work item takes twice as long to complete a task as the other work items in a kernel call, the other work items will remain idle until the final work item has completed its task.

Other optimization techniques have to do how to effectively use memory. In general, GPUs are much better at computing mathematical calculations than they are at reading from and writing to memory. For this reason, a well optimized kernel will maximize the number of arithmetic operations and minimize the number of read and write operations. If multiple read/write calls need to be made from/to the same address in global memory, it is typically a good idea to, at least temporarily, store the data located at that address in local memory. While read and write calls to and from local memory, are still much slower than arithmetic calculations, they are much faster than using global memory. By using local memory effectively, calls to global memory can be minimized leading to faster kernel execution.

Sometimes, calls to global memory are unavoidable. In these cases, there are a couple of techniques that can be used to minimize latency associated with accessing global memory. These include sequential reading from global memory arrays, sizing arrays as to maximize use of faster cache, as well as many others. A thorough discussion of these techniques is

beyond the scope of this dissertation. If the reader is interested, more information can be found in the following references[19, 20, 21].

## 2.5 Generalized Molecular Dynamics Model

One of my main accomplishments during my tenure at Colorado State has been to develop a generalized molecular dynamics model to simulate physical processes in ultracold plasma systems. This model was developed by combining some of the molecular dynamics techniques outlined in section 2.3, with the GPU programming techniques outlined in section 2.4. This section will give an overview of how the model works, and the sorts of problems that it is able to address.

### 2.5.1 Model Overview

My molecular dynamics model provides an excellent method for simulating a variety of the physical processes in ultracold plasma systems. Currently, the model is only capable of utilizing a single GPU at a time. However, this has been more than adequate for most the problems that we have needed to address. With a single GPU the evolution over hundreds of nanoseconds of every particle in the experimental system present in our lab can be calculated in just a few hours. This makes the molecular dynamics model a powerful tool for probing the fundamental physics of UCP systems.

While the structure of the model can change substantially based on the physical processes that the model is trying to simulate, there are a number of similarities that all of the versions of the model have in common. This leads to all of the implementations of the model having the same basic structure. This structure is as follows.

1. The initial parameters of the system are specified. This process involves defining the total number electrons and ions present in the system. Additionally, it is determined whether or not ions in the simulation are treated as discrete particles or as a smooth analytic distribution. Typically, this portion of the simulation will also determine

the temperature and spatial extent of both the ion and electron distributions, but completely specifying these quantities may not be necessary in some instances.

2. Individual particles are assigned positions and velocities. This process typically occurs by randomly generating electrons and/or ions with Gaussian spatial distributions and Maxwellian velocity distributions, but other methods can be used.
3. The forces acting on the particles in the model are determined. All forces present in the model are represented as classes, and all of the force calculations are made on the GPU. During this portion of the simulation, the force classes acting on the particles are initialized, and the instances of those classes are inserted into a force list.
4. A similar procedure is undertaken for particle transformation classes. An example of a transformation class would be the electron-ion collisions class, which models collisions as a rotation of the electron velocity vector (see section 3.2).
5. An instance of the Updater class is initialized. This class is used to sum up the forces acting on each of the particles, and to use that force information to calculate the particles' time evolution.
6. The Updater class' update function is repeatedly called. Each call of this function results in the evolution of the system of particles over a given timestep. This function call will be more fully explained in the next subsection.
7. In between every call of the update function, calculations of the relevant energies in the system are made. This is to make sure that simulation is properly conserving energy.

By following these 7 basic steps, fast and accurate molecular dynamics simulations can be deployed by using relatively few lines of code. Furthermore, this frame work is extremely flexible allowing for wide variety of different physical processes to be simulated. For a generic implementation of this model see appendix A.

## 2.5.2 Updater

The updater class is the workhorse behind the molecular dynamics model. In all model simulations an instance of this class is created and is subsequently used to calculate all of the forces on all of the particles, and to integrate their positions and velocities forward in time. The Updater code can be divided into two parts. The first of these is the constructor function, which is the function that is called when an instance of the class is initialized. A truncated version of this code can be seen below.

```
def __init__(self, ctx, queue, global_size, size=None):
    self.ctx = ctx
    self.queue = queue

    """ Source code for the advance kernel. This kernel is used
    to update a vector (position or velocity) by adding its
    original value to its time derivative times dt """
    src="""
    __kernel void advance(__global float *x, __global float *y,
    __global float *z, __global float *dx, __global float *dy,
    __global float *dz, float dt, int size)
    {
        int i=get_global_id(0);
        i+=get_global_size(0)*get_global_id(1);
        if (i<size)
        {
            x[i]+=dx[i]*dt;
            y[i]+=dy[i]*dt;
            z[i]+=dz[i]*dt;
        }
    }
    """

    self.prog=cl.Program(ctx, src)
    self.prog.build()
    self.global_size=global_size
    self.size=size
    if self.size==None:
        self.size=global_size
    #Allocate global memory for acceleration arrays
    self.axd = cl_array.Array(self.queue, self.global_size, np.float32)
    self.ayd = cl_array.Array(self.queue, self.global_size, np.float32)
    self.azd = cl_array.Array(self.queue, self.global_size, np.float32)
```

Initializing an instance of the Updater class with the above code accomplishes two main things. First, it compiles the advance kernel that will later be used to update both the positions and velocities of particles in the simulation. This kernel updates the components of a generic 3d vector by adding their current value to the value of the appropriate derivative

multiplied by the chosen timestep,  $dt$ . Secondly, this function initializes buffers on global memory. These device arrays will later be used to store acceleration values.

As can be seen in the code, the `__init__` function takes 5 arguments. The first of these is the self argument, which is a required first argument for all class functions in Python. The next two arguments are for an OpenCL context and queue, respectively. Thus, these must be set up before this an instance of this class is initialized. The next two arguments are `global_size` and `size`. The global size argument will determine the how many array elements should be allocated to the array acceleration arrays contained in the Updater class, and the size arguments will determine how many of those elements will be used at a given time. Typically, these two numbers will be the same, but in cases where the total number of particles in the system is changing, they can be different.

The second part of the code contains the update function. This function sums up the forces on every particle, and then integrates their positions and velocities. Additionally, the function also calculates the relevant transforms and applies them to the proper particles. Below is a truncated version of the code behind this function.

```
def update(self, xd, yd, zd, vxd, vyd, vzd, qd, md, forces,
           transforms, t, dt, steps):
    for i in range(steps):
        #Half step position advance
        self.prog.advance(self.queue, (self.size, 1), (256, 1),
                          xd.data, yd.data, zd.data, vxd.data, vyd.data, vzd.data,
                          np.float32(0.5*dt), np.int32(self.size))

        #Half step time advance
        t += 0.5 * dt

        #initialize the acceleration arrays
        self.axd.fill(0.0, self.queue)
        self.ayd.fill(0.0, self.queue)
        self.azd.fill(0.0, self.queue)

        #Cycle through the forces to compute the total acceleration
        #on each of the particles
        for acc in forces:
            acc.computeAcc(xd, yd, zd, vxd, vyd, vzd, qd, md,
                           self.axd, self.ayd, self.azd, t, size=self.size)

        #Compute all of the particle transforms
        for trans in transform:
            trans.computeTrans(xd, yd, zd, vxd, vyd, vzd, qd,
                               md, t, size=self.size)
```

```

#Full step velocity advance
self.prog.advance(self.queue,(self.size,1),(256,1),vxd.data,
vyd.data,vzd.data,self.axd.data,self.ayd.data,self.azd.data,
np.float32(dt),np.int32(self.size))

#Half step position advance
self.prog.advance(self.queue,(self.size,1),(256,1),xd.data,
yd.data,zd.data,vxd.data,vyd.data,vzd.data,np.float32(0.5*dt),
np.int32(self.size))

#Half step time advance
t += 0.5 * dt
return t

```

The update function calculates position and velocity advances by using the advance kernel described earlier. In this version of the code, the update function utilizes the Leapfrog method to evolve the state of the system. This is implemented by first advancing particle positions by half a timestep with the unadvanced velocities. Next the accelerations of the particles are then calculated by summing up all of the applicable forces. Once calculated, these acceleration values are used to advance the velocities a full timestep. Finally, the position vectors are updated the final half timestep by using the updated velocities.

A number of steps need to be taken for the update function to sum the forces on the particles. First, a list of force class objects is passed to the update function as an argument. Each of these force classes has to be structured in a specific way. They all must use kernels to calculate the forces on simulation particles, and the kernel must have been previously compiled before being passed to the update function. They all must have a computeAcc function that accepts device array arguments in the manner illustrated in the above code. Finally, the computeAcc function must use kernels to calculate accelerations and not forces. If these conditions are met the update function is able to iterate through all of the force class objects in the forces list. Once this occurs, the device acceleration arrays will contain the appropriate accelerations values for each particle. An analogous procedure is also followed to perform particle transformations.

### 2.5.3 Common Forces

The molecular dynamics model is able to implement a wide variety of forces. This subsection gives an overview of some the more common forces used in the model.

#### Coulomb Pairwise Interactions

Perhaps the most ubiquitous of all the utilized forces, a variation of the pairwise Coulomb interaction has been utilized in every single simulation using this model. Furthermore, since Coulomb pair interaction calculations scale with  $N^2$ , these calculations are almost exclusively responsible for the total computation time of model simulations. In general, the interaction potential between two charged particles in molecular dynamics is

$$\phi = \frac{q_1 q_2}{4\pi\epsilon_0 r_{eff}} \quad (2.31)$$

where  $r_{eff}$  is the effective distance between the two particles. If  $\phi$  were to be calculated exactly,  $r_{eff}$  would be evaluated as the exact distance between the two particles,  $r$ . However, for sufficiently small values of  $r$ , exact calculations of  $r$  will result in energy conservation issues, as were outlined in section 2.3.4. For this reason, the Coulomb potential is typically altered in such a way where  $r_{eff}$  has a minimum non-zero value,  $\alpha$ , called the softening parameter. The softened Coulomb potential typically expresses the inter-particle separation as either  $r_{eff} = r + \alpha$  or as  $r_{eff} = \sqrt{r^2 + \alpha^2}$ . The later of these formulations diverges less severely from the exact Coulomb potential, and for this reason, this version of the Coulomb potential was what was used in the molecular dynamics simulation outlined in this dissertation. A graph comparing the different variation of the Coulomb potential can be seen in figure 2.6.

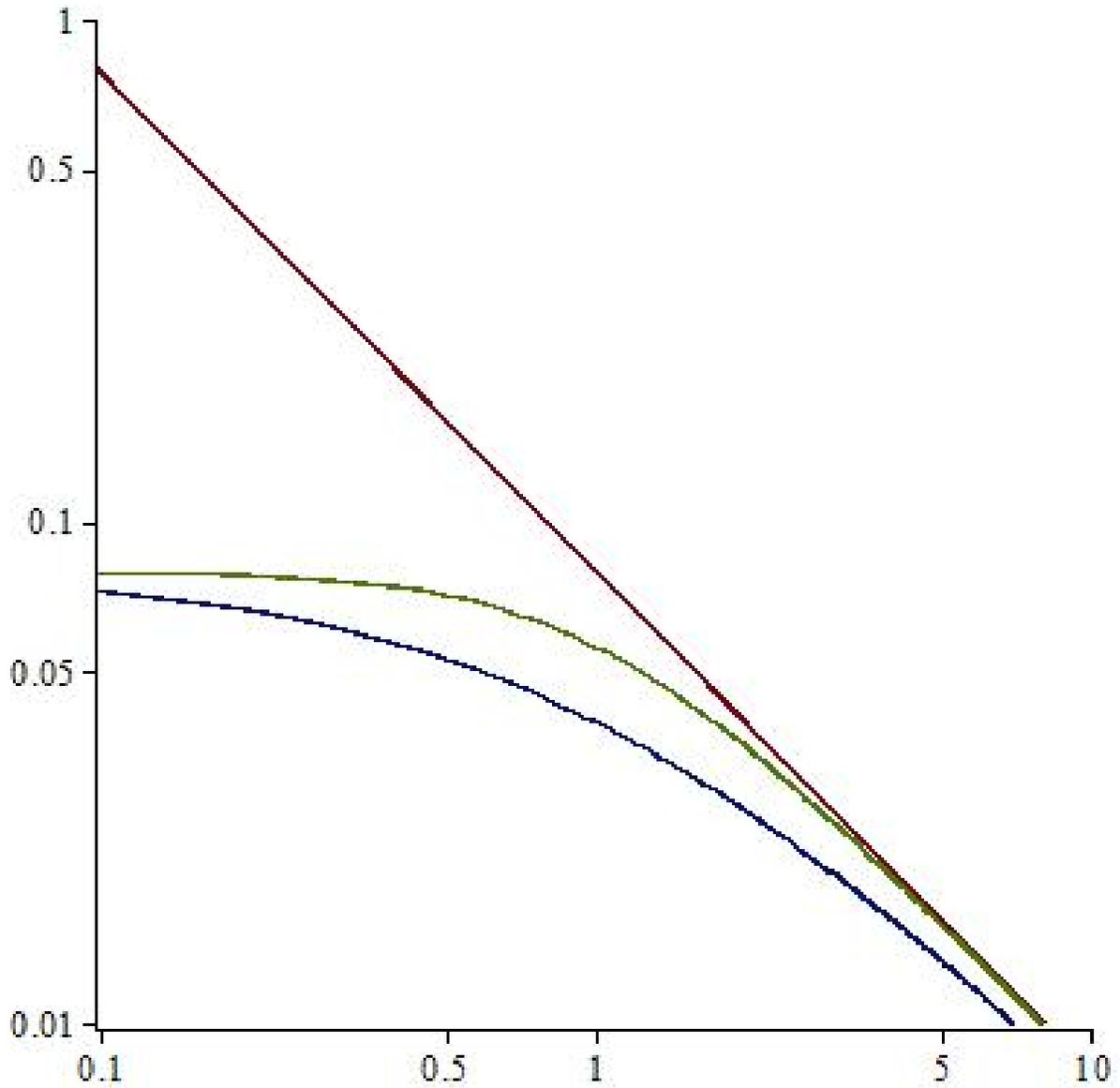


Figure 2.6: Graph of different forms of the Coulomb potential as a function of  $r/\alpha$  in arbitrary units. The red curve represents that the unadjusted Coulomb potential, the blue curve represents a potential where  $U \propto 1/(r + \alpha)$ , and the green curve represents a potential where  $U \propto 1/\sqrt{r^2 + \alpha^2}$ . As can be seen as in the figure, the  $1/\sqrt{r^2 + \alpha^2}$  potential better approximates the unadjusted potential. For this reason, this form of the potential is used in all of the simulations described in this dissertation.

## Ion Forces

In many simulations, it is beneficial to model UCP ions as a smoothly distributed background. There are two main ion distributions that are typically used in the model. These are a uniform distribution and a spherically symmetric Gaussian distribution. From these ion distributions, Gauss' law can be used to generate an analytical force expression as a function of position. The charge density of a uniform density collection of ions is simply  $qn_i$ , where  $n_i$  is the ion density. By using Gauss' law to integrate the charge density, the following force is obtained

$$\vec{E} = \frac{qn_i}{3\epsilon_0} r\hat{r} \quad (2.32)$$

where  $r$  is the distance from the center of what is assumed to be a spherically symmetric ion distribution. For a Gaussian ion distribution, the charge density can be described as the following

$$\rho(r) = \frac{qN_i}{(2\pi\sigma^2)^{\frac{3}{2}}} e^{-\frac{r^2}{2\sigma^2}} \quad (2.33)$$

where  $N_i$  is the total number of ions contained in the distribution, and  $\sigma$  is the characteristic size of a Gaussian distribution. Again, by using Gauss' law the following functional form for the electric field can be calculated.

$$\vec{E} = \left( \operatorname{erf}\left(\frac{r}{\sqrt{2}\sigma}\right) - \sqrt{\frac{2}{\pi}} \frac{r}{\sigma} e^{-\frac{r^2}{2\sigma^2}} \right) \frac{qN_i\hat{r}}{4\pi\epsilon_0 r^2} \quad (2.34)$$

where  $\operatorname{erf}()$  signifies the error function.

## External Electric Fields

Model simulation also would often contain externally applied electric fields. In most cases these were DC electric fields, which are trivial to implement. However, the experimental system in our lab does not have a well known electric field topology, and for this reason it was sometimes necessary to model more complicated field structures. The simplest example

of this would be a linear field of the form

$$\vec{E} = (A_1 + A_2 z)\hat{z} \quad (2.35)$$

where  $A_1$  and  $A_2$  are constants. However, this field violates Laplace's equation,  $\nabla^2\phi = 0$ , and is thus unphysical. To conform with Laplace's equation, linear fields incorporated in the model were given the following functional form.

$$\vec{E} = (A_1 + A_2 z)\hat{z} - A_2 \frac{\rho}{2}\hat{\rho} \quad (2.36)$$

More complicated field structures were used in the model on occasion, but complexity of those fields combined with the difficulty involved making sure that they conformed with Laplace's equation, led to most simulations incorporating either a linear or a DC external field.

## Magnetic Fields

Externally applied uniform magnetic fields were another common force that were frequently included in many different model simulation. Naively, it would be expected that inserting such a force would be rather trivial, and could be accomplished by simply calculating  $\vec{v} \times \vec{B}$ . However, this approach in the absence of very small timesteps would typically lead to the heating of simulation particles. For conditions in which the simulations were typically run, as much as a 10% increase in energy could be observed from  $\vec{v} \times \vec{B}$  calculations over the course of a few hundred nanoseconds.

To understand this formally, consider a particle with charge,  $q$ , and mass,  $m$ , in the presence of an externally applied magnetic field,  $\vec{B} = B\hat{z}$  with no other forces present. If the particle initially starts with a velocity,  $\vec{v} = v_0\hat{x}$ , the velocity of the particle can be described

as

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_0 \begin{bmatrix} \cos(\frac{qB}{m}t) \\ \sin(\frac{qB}{m}t) \end{bmatrix} \quad (2.37)$$

where  $v_x$  and  $v_y$  are the x and y component of the velocity respectively. The forces that are being exerted on this system are as follows.

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = qBv_0 \begin{bmatrix} \sin(\frac{qB}{m}t) \\ -\cos(\frac{qB}{m}t) \end{bmatrix} \quad (2.38)$$

Combining eq 2.37 and eq 2.38 with eq 2.23 yields the Leapfrog velocity integration function for this physical system.

$$\begin{bmatrix} v_x(t + \frac{\Delta t}{2}) \\ v_y(t + \frac{\Delta t}{2}) \end{bmatrix} = v_0 \begin{bmatrix} \cos(\frac{qB}{m}(t - \frac{\Delta t}{2})) + \frac{qB}{m} \sin(\frac{qB}{m}(t - \frac{\Delta t}{2})) \\ \sin(\frac{qB}{m}(t - \frac{\Delta t}{2})) - \frac{qB}{m} \cos(\frac{qB}{m}(t - \frac{\Delta t}{2})) \end{bmatrix} \quad (2.39)$$

In general, the evolution of the kinetic energy in a Leapfrog integration scheme can be written as the following.

$$E\left(t + \frac{\Delta t}{2}\right) = \frac{1}{2}m\left(v(t + \Delta t/2)\right)^2 \quad (2.40)$$

where  $v = \sqrt{v_x^2 + v_y^2}$ . By combining eq 2.39 and eq 2.40, the following expression for kinetic energy is obtained.

$$E\left(t + \frac{\Delta t}{2}\right) = \frac{1}{2}mv_0^2\left(1 + \left(\frac{qB}{m}\right)^2 \Delta_t^2\right) \quad (2.41)$$

The results of eq 2.41 shows that as the system evolves, kinetic energy is added into the system, and thus the system does not conserve energy.

Since explicitly calculating  $\vec{v} \times \vec{B}$  leads to the heating of model particles, another method for implementing the magnetic field needs to be used. Fortunately, a technique exists, called the Boris method, which allows for magnetic forces to be integrated in a energy conserving way[7, 22]. To understand this method consider the Leapfrog velocity integration of a particle

experiencing a generic Lorentz force.

$$v\left(t + \frac{\Delta t}{2}\right) = v\left(t - \frac{\Delta t}{2}\right) + \frac{q}{m} \left( \vec{E} + \frac{v\left(t + \frac{\Delta t}{2}\right) + v\left(t - \frac{\Delta t}{2}\right)}{2} \times \vec{B} \right) \Delta t \quad (2.42)$$

The following change of variables can be made

$$v\left(t - \frac{\Delta t}{2}\right) = v_- - \frac{qE}{m} \frac{\Delta t}{2} \quad (2.43)$$

$$v\left(t + \frac{\Delta t}{2}\right) = v_+ + \frac{qE}{m} \frac{\Delta t}{2} \quad (2.44)$$

This allows eq 2.42 to be rewritten in the following way

$$\frac{\vec{v}_+ - \vec{v}_-}{\Delta t} = \frac{q}{2m} (\vec{v}_+ + \vec{v}_-) \times \vec{B} \quad (2.45)$$

where eq 2.44 is no longer dependent on the electric field. It can be shown that eq 2.44 is purely a rotation, and that this rotation takes the following form

$$\tan\left(\frac{\theta}{2}\right) = \frac{|\vec{v}_+ - \vec{v}_-|}{|\vec{v}_+ + \vec{v}_-|} = \frac{qB\Delta t}{2m} \quad (2.46)$$

where  $\theta$  is the angle between  $\vec{v}_+$  and  $\vec{v}_-$ . For small values of  $\theta$  the value of  $\theta$  can be approximated as  $qB\Delta t/m$ . With this known the Boris method allows for the velocity with the following scheme.

1.  $v(t - \Delta t/2)$  is updated to  $v_-$  via Eq. 2.43.
2.  $v_-$  is rotated by  $qB\Delta t/m$  to obtain  $v_+$ .
3.  $v_+$  is evolved by Eq. 2.44, yielding  $v(t + \Delta t/2)$ .

Since the rotations of a velocity vector by definition preserves the magnitude of the velocity, by taking this approach, the Boris method allows for energy to be conserved. This method can be used in any Leapfrog integration scheme by simply replacing the simpler Eq 2.23.

## 2.6 Conclusion

This chapter introduced the general molecular dynamics model that I developed during my time at Colorado State University, and provided significant context for where this model sits in the landscape of computational plasma techniques. The model uses molecular dynamics techniques to allow for the simulation of many different physical processes in charged particle systems. By using the OpenCL standard, the model is able to harness the massively parallel architecture of modern GPUs. This allows for the model to be able explicitly simulate all of the constituent particles contained in an ultracold plasma in our lab. The model was the backbone for many of the simulations that are described in this dissertation, and has been a vital tool in interpreting a number of different results. While much insight has already been gained from model simulations, we are still just scratching the surface with what the model can help us learn.

# References

- [1] L. Spitzer, *Physics of Fully Ionized Gases*, Dover Publications, Mineola, NY, 2006.
- [2] P. Bellan, *Fundamentals of Plasma Physics*, Cambridge University Press, 2006.
- [3] P. L. Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).
- [4] S. Chandrasekhar, Rev. Mod. Phys. **15**, 1 (1943).
- [5] E. Sonnendrücker, Computational plasma physics, lecture notes, 2015.
- [6] J. J. Y. Hsu, *Visual and Computational Plasma Physics*, World Scientific Publishing, 1st edition, 2015.
- [7] C. Birdsall and A. Langdon, *Plasma Physics via Computer Simulation*, McGraw Hill.
- [8] J. P. Verboncoeur, Plasma Physics and Controlled Fusion **47**, A231 (2005).
- [9] A. Satoh, *Introduction to Practice of Molecular Simulation : Molecular Dynamics, Monte Carlo, Brownian Dynamics, Lattice Boltzmann and Dissipative Particle Dynamics*, Elsevier, 2010.
- [10] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, 3rd edition, 2009.
- [11] U. D. Schiller, An overview of integration schemes for molecular dynamics simulations, lecture notes.
- [12] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC Press, 2011.
- [13] J. Barnes and P. Hut, Nature **324**, 446 (1986).

- [14] A. Munshi, B. R. Gaster, T. G. Mattson, J. Fung, and D. Ginsburg, *OpenCL Programming Guide*, Addison-Wesley, 2010.
- [15] S. Wasson, *AMD's Radeon HD5870 Graphics Processor*, techreport.com, 2009.
- [16] AMD, Introduction to opencl programing, 2010.
- [17] A. Munshi, *The OpenCL Specification*, Khronos OpenCL Working Group, 2011.
- [18] M. Scarpino, *OpenCL in Action: How to Accelerate Graphics and Computation*, Manning Publications Co., 1st edition, 2012.
- [19] G. Ruetsch and M. Fatica, *CUDA Fortran for Scientists and Engineers: Best Practices for Efficient CUDA Fortran Programming*, Morgan Kaufmann, 2014.
- [20] R. Farber, *CUDA Application Design and Development*, Morgan Kaufmann, 2012.
- [21] S. Cook, *CUDA Programming a Developer's Guide to Parallel Computing with GPUs*, Morgan Kaufmann, 2013.
- [22] L. Patacchini and I. Hutchinson, *Journal of Computational Physics* **228**, 2604 (2009).

## Chapter 3

# Electron Oscillation Damping

One of the fundamental properties of plasma systems is the existence of the plasma oscillations. Plasma oscillations are density oscillations of a collection of charged particles, typically electrons, within a plasma system[1]. These oscillations occur when there is a sufficient displacement of charge density. Since plasmas systems have a tendency to remain quasi-neutral, charge density displacements give way to a restoring force that initiates the plasma oscillation. Plasma oscillations have been observed in UCP systems[2], and due to their dependence on particle density, have been used to help characterize UCP systems[2].

In the standard textbook treatment, plasma oscillations are usually considered in the case of uniform density infinite-extent plasmas. In this chapter, however, the focus is on UCPs. In these systems there are several types of electron oscillations that are possible that would involve the restoring forces associated with density displacements. Sufficiently short-wavelength ones would be the most like those in a uniform-density large system. However, the oscillations of interest here involve displacements of the whole electron component. Displacing the electrons as a whole leads to a restoring force that creates an oscillation. This oscillation is evident as an oscillation in the electron center-of-mass. As a result, electron center-of-mass oscillations will be the subject of the rest of this chapter. A diagram of an electron center-of-mass can be seen in figure 3.1.

Without any additional forces, electron center-of-mass oscillations would continue indefinitely. However, in collisionally dominated systems such as UCPs, collisions between charged particles in the system lead to a damping of center of mass oscillations. In two component plasmas like UCPs, there are three distinct types of collisions that can occur, electron-electron, ion-ion, and electron-ion collisions. Neither electron-electron nor ion-ion collisions lead to a damping of electron center-of-mass oscillations. Conservation of momen-

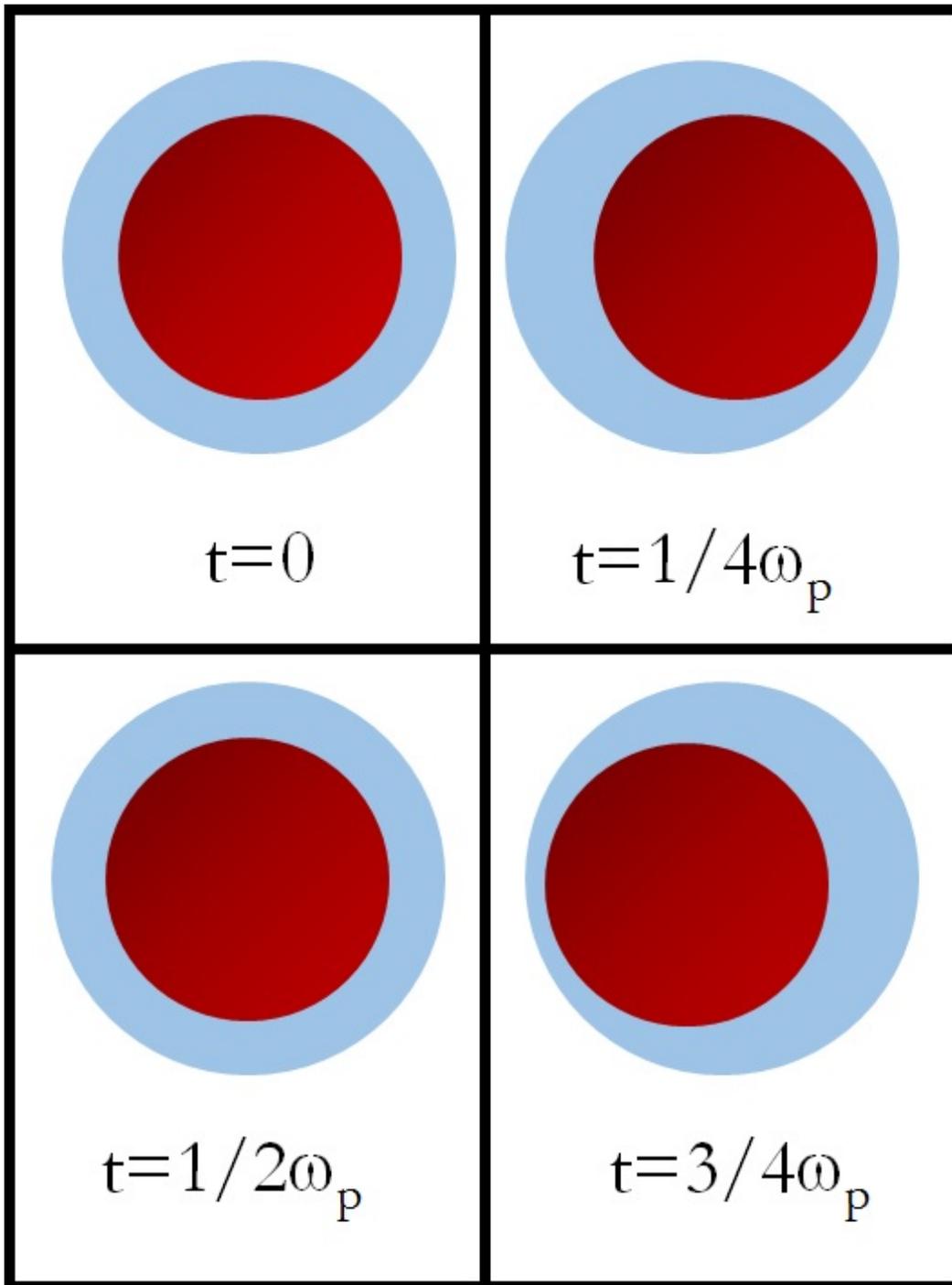


Figure 3.1: Cartoon of electron center-of-mass oscillation in a UCP. The larger blue circle represents UCP ions, and the small ball with the red gradient represents UCP electrons. The figure shows snapshots of the electron cloud at four evenly spaced times during the period of a single plasma oscillation,  $1/\omega_p$ . The figure illustrates that during an electron center-of-mass oscillation the entire electron cloud shifts in position.

tum prevents electron-electron collisions from altering electron center-of-mass motion, and ion-ion collisions operate on much slower timescales and also do not impact the electron component of the plasma. This leaves electron-ion collisions as the only collisional mechanism that leads to the damping of electron center-of-mass oscillations.

If electron-ion collisions were exclusively responsible for electron oscillation damping, experimental UCP systems could be used to study electron-ion collisions through measuring electron center-of-mass oscillation damping. In other words, by measuring the rate by which the electron oscillations damp, it should be possible to extract the electron-ion collision rate. Furthermore, since the electron-ion collision rate is strongly dependent on the electron temperature, such a measurement should be able to be used as a probe of the electron temperature. For low electron temperatures, there are a number of significant heating mechanisms that are difficult to quantify, yielding temperature uncertainties with regard to the final electron temperatures (see chapter 4). For this reason, an independent temperature measurement would be extremely useful in characterizing UCP conditions. Finally, such a measurement should be able to be performed in UCP systems with sufficiently low electron temperatures such that the electrons are strongly coupled. Since electron strong coupling is expected to alter the electron-ion collision rate[3, 4, 5, 6], such a measurement should allow for the detection of electron strong coupling, as well as experimentally quantify its effects on electron-ion collisions.

This chapter will discuss the series of electron oscillation damping measurements that have taken place in our lab. My contributions to these measurements were largely computational, and thus this chapter will be largely focused on the computational side of the measurements that was vital in interpreting the experimental results in a useful way. Modeling in this experiment was necessary to gain a variety of new insights into the relevant underlying physics. Additionally, numerical modeling played a prominent role in choosing experimental conditions, and was essential in interpreting our data. This chapter will discuss all of these applications of modeling, as well as give a high level overview of the model that

was utilized. Additionally, the chapter will also give an introduction to the mathematics behind Coulomb collisions, as well as a brief overview of the experiment.

## 3.1 Coulomb Collisions

Binary Coulomb collisions play an important role in the dynamics of many plasma systems. As charged particles move through plasma systems, they occasionally make close passes by other particles causing a deflection of the particle's trajectory. This section will give an overview of the mathematics involved in modeling these collisions following a standard treatment along the lines of that often found in textbooks on plasma physics.

### 3.1.1 Generalized Solution

To understand Coulomb collisions, consider a system of two charged particles with masses  $m_1$  and  $m_2$  and charges  $q_1$  and  $q_2$ [7]. In the center-of-mass frame, the interaction between the two particles can be described as:

$$\mu\ddot{\vec{r}} = \mu(\ddot{r} - r\dot{\theta}^2)\hat{r} = \frac{q_1q_2\vec{r}}{4\pi\epsilon_0r^3} \quad (3.1)$$

where  $\epsilon_0$  is the permittivity of free space,  $\mu$  is the reduced mass,  $\vec{r}$  is the center of mass vector defined as  $\vec{r}_1 - \vec{r}_2$ , and  $r$  and  $\theta$  are the standard radial coordinates in the center of mass frame. The angular momentum of the system is, by definition,  $\mu r^2\dot{\theta}$ . For a Coulomb collision, the angular momentum can also be described as the constant,  $\mu bv$ , where  $v$  is  $|\dot{r}|$ , and  $b$  is the impact parameter which is explained in figure 3.2. To solve Eq. 3.1 a change of variables is utilized:

$$\frac{d^2u}{d\theta^2} + u = -\frac{q_1q_2}{4\pi\epsilon_0\mu b^2v^2} \quad (3.2)$$

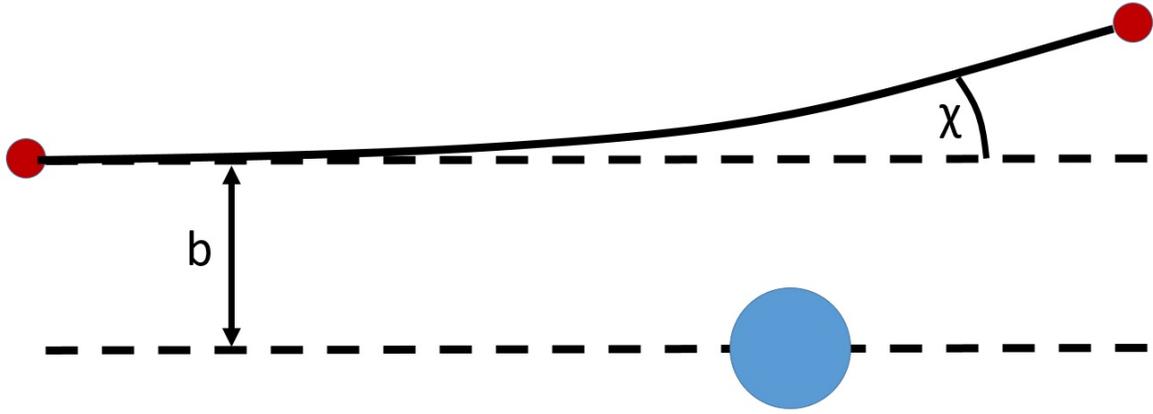


Figure 3.2: Diagram of a binary electron-ion collision. The small red circles correspond to the trajectory of the an electron, and the large blue circle corresponds to the target ion. The impact parameter,  $b$ , quantifies the hypothetical minimum distance between the electron and ion if the electron had not been deflected. For this reason,  $b$  is often referred to as the distance of closest approach. The deflection of the electron is typically quantified by the angle  $\chi$ , defined by the diagram.

where  $u = 1/r$ . The interaction between the two particles can be described entirely by a rotation of the center of mass velocity, and by integrating equation (3.2) the amount of rotation can be determined. This resultant deflection angle,  $\chi$ , is given by:

$$\chi = 2 \tan^{-1} \left( \frac{q_1 q_2}{4\pi\epsilon_0\mu|\vec{v}_1 - \vec{v}_2|^2 b} \right) = 2 \tan^{-1} \left( \frac{b_{90}}{b} \right) \quad (3.3)$$

where  $b_{90}$  is the impact parameter that results in a 90 degree deflection. Equation 3.3 illustrates that Coulomb collision deflection angles can be completely defined by the masses and charges of the colliding particles, the relative velocity between the particles, and the impact parameter,  $b$ .

### 3.1.2 Single Particle Collision Frequency

While Coulomb collisions can result in large deflections, typically, in plasma systems, the large majority of collision events result in very small angle deflections. However, these small angle Coulomb collisions occur on a timescale that is usually much faster than the large

angle collisions. A random walk of small angle collisions can lead to a large net deflection in the momentum of plasma particles. Due to the high probability associated with small angle collisions, it turns out that small angle collisions typically have a larger impact on the momentum deflection of plasma particles than large angle collisions. This is in stark contrast to other systems such as those approximated with hard-sphere collisions where large-angle collisions dominate the net momentum transfer contribution. It will be seen that the influence of small angle collisions in Coulomb systems requires careful treatment that involves consideration of screening and collective effects in a plasma.

In general, Coulomb collisions will cause a random deflection of a particle's momentum such that on average that momentum decreases. The rate at which such collisions cause a particles momentum to damp can be found by properly integrating over all collisions. Explicitly, this can be written as

$$\frac{dp}{dt} = \int \Delta p d\nu \tag{3.4}$$

where  $p$  is the momentum of the projectile,  $d\nu$  is the differential collision rate at which a single particle, known as the projectile, collides with other particles in the distribution, known as target particles, and  $\Delta p$  is the average momentum change resulting from a single collision. For a given projectile,  $d\nu$  is given by

$$d\nu = n\bar{v}_d d\sigma \tag{3.5}$$

where  $n$  is the density of target particles, and  $d\sigma$  is the differential collisional cross-section, which can be found by integrating the differential cross-section,  $d\sigma/db$ .

$$\frac{d\sigma}{db} = 2\pi b \tag{3.6}$$

The quantity,  $\bar{v}_d$ , is the average relative velocity between the projectile and the target particles.

$$\bar{v}_d = \int |\vec{v} - \vec{v}'| f(\vec{v}') d\vec{v}' \quad (3.7)$$

where  $\vec{v}$  is the velocity vector of the projectile, and the  $f(\vec{v}')$  is the velocity distribution of the target particles. In the limit in which the projectile velocity is much greater than the average target particle velocity, Eq. 3.5 can be simplified to

$$d\nu = nd\sigma v \quad (3.8)$$

This high velocity limit is a good approximation for electron-ion collisions in our system since the relative mass differences between the electrons and ions leads to UCP electrons having velocity magnitudes many orders of magnitude greater than UCP ions. However, this high velocity limit is not a great approximation for the electron-electron collisions in UCPs. Since electron-ion collisions, and not electron-electron collisions, are subject of this chapter, this high velocity limit will be used throughout this discussion. To see a derivation of the electron-electron collision rate in the absence of this limit see ref [8].

For a single collision, the total change in momentum can be expressed as:

$$\Delta p = -m_p v (1 - \cos(\chi/2)) \quad (3.9)$$

where  $m_p$  is the mass of the projectile. Since  $\chi$  is a function of  $b$ , Eq. 3.9 can be rewritten as

$$\Delta p = -m_p v \frac{2b_{90}^2}{b^2 + b_{90}^2} \quad (3.10)$$

By combining Eq. 3.6, 3.8, and 3.10, Eq. 3.4 can now be expressed in the following way.

$$\frac{dp}{dt} = -4\pi n m_p v^2 \int b \frac{b_{90}^2}{b^2 + b_{90}^2} db \quad (3.11)$$

Some care of needs to be taken to when determining the limits of integration in Eq. 3.11. In the absence of screening, Coulomb interactions have infinite range, suggesting that the upper limit of integral should be at  $b = \infty$ . However, since the result of the integrand in Eq. 3.12 contains a logarithm, the integral will diverge if an upper cutoff is not established in the limits of integration. Thus an upper cutoff, known as  $b_{max}$ , is typically established. By implementing this cutoff and by assuming  $b_{max} \gg b_{90}$ , evaluating the integral in Eq. 3.11 yields the following equation.

$$\frac{dp}{dt} = -4\pi n\mu v^2 b_{90}^2 \ln\left(\frac{b_{max}}{b_{90}}\right) = -4\pi n\mu v^2 b_{90}^2 \ln(\Lambda) \quad (3.12)$$

where  $\ln(\Lambda)$  is the Coulomb logarithm which is defined as.

$$\ln(\Lambda) = \ln\left(\frac{4\pi\epsilon_0 k_B T_e b_{max}}{q^2}\right) \quad (3.13)$$

This definition is made using approximations assuming a weakly-coupled plasma. Once known, the quantity  $dp/dt$  can be used to define a collision frequency. This collision frequency is as follows.

$$\nu_p \equiv -\frac{1}{p} \frac{dp}{dt} = \frac{4\pi n\mu v b_{90}^2}{m_p} \ln(\Lambda) \quad (3.14)$$

The existence of screening in plasma systems leads to a natural cutoff for the impact parameter. For this reason,  $b_{max}$  is typically defined as being proportional to the debye length,  $\lambda_D$ . Despite  $\lambda_D$  being a natural scaling length, the existence of a sharp cutoff is still unphysical. Unlike screening, which leads to a continuous exponential decrease in the electric potential, the the existence of a sharp upper cutoff leads to a discontinuity. This discontinuity exists since, under these assumptions, an impact parameter of  $b_{max}$  leads to a collision, but an impact parameter of  $b_{max} + db$  does not result in a collision. Despite the unphysical nature of these approximations, previous work investigating the thermalization of electrons and ions has shown that these approximations seem justified given comparisons to molecular dynamics simulations for a  $b_{max} = 0.7\lambda_D$  in the weak coupling limit[3].

### 3.1.3 Ensemble Collision Frequency

Equations 3.1-3.13 cover the collisional behavior of a single particle in a plasma. However, in many circumstances, the collisional behavior of the entire system is also of interest. For instance,  $dp/dt$  of a collection of particles can be found as a volume averaged quantity. In the most general case involving two species of particles,  $dp_1/dt$  can be written as follows

$$\frac{dp_1}{dt} = \int \int -4\pi\mu(\vec{v}_1 - \vec{v}_2)|\vec{v}_1 - \vec{v}_2|b_{90}^2 \ln(\Lambda) f_1(v_1) f_2(v_2) d^3v_1 d^3v_2 \quad (3.15)$$

where  $f_1$  and  $f_2$  are the probability distributions of the first and second species involved in the collision, respectively. By making additional assumptions, Eq. 3.15 can be evaluated using the method found in [7].

The first of these assumptions involves the Coulomb logarithm, which is a function of  $v_1$  and  $v_2$ . However the value of the Coulomb logarithm is typically large in many plasma systems of interest, and due to the functional form of a logarithm it typically remains relatively constant over the range of integration, This allows for the Coulomb logarithm to be treated as a constant, allowing it to be pulled outside of the integral. The second assumption is that both particle species are distributed in a Maxwellian distribution, with a small drift velocity. It is assumed that the drift velocity is much smaller than the thermal velocity of the distribution of the particle species. This allows for an important Taylor expansion to be performed in process of evaluating the integral in Eq. 3.15.

By implementing the previously enumerated assumptions, Eq. 3.15 can be evaluated as

$$\frac{dp_1}{dt} = -\frac{4\pi}{\mu} \frac{2n_1 n_2 v_d}{3\sqrt{2\pi}} \left( \frac{q_1 q_2}{4\pi\epsilon_0} \right)^2 \left( \frac{\mu}{k_b T} \right)^{3/2} \ln(\Lambda) \quad (3.16)$$

where  $v_d$  is the relative drift velocity of the 2 particle distributions. This leads to the following general expression for the volume averaged collision frequency.

$$\bar{\nu}_{12} = -\frac{1}{n_1 m_1 v_d} \frac{dp_1}{dt} = \frac{4\pi}{m_1 \mu} \frac{2n_2}{3\sqrt{2\pi}} \left( \frac{q_1 q_2}{4\pi\epsilon_0} \right)^2 \left( \frac{\mu}{k_b T} \right)^{3/2} \ln(\Lambda) \quad (3.17)$$

In the case of electron-ion collisions, the volume averaged collision frequency is

$$\bar{\nu}_{ei} = \frac{1}{3} \sqrt{\frac{2}{\pi}} \left( \frac{n_i Z q^4}{4\pi\epsilon_0^2 \sqrt{m_e} k_b^{3/2} T^{3/2}} \right) \ln(\Lambda) \quad (3.18)$$

where  $m_e$  is the mass of an electron,  $Z$  is an integer that quantifies the charge of the ion species, and  $n_i$  is the density of ions. In the case of electron-electron collisions, the volume averaged collision frequency is the following.

$$\bar{\nu}_{ee} = \frac{1}{3\sqrt{\pi}} \left( \frac{n_e Z q^4}{4\pi\epsilon_0^2 \sqrt{m_e} k_b^{3/2} T^{3/2}} \right) \ln(\Lambda) \quad (3.19)$$

Equations 3.17-3.19 give a value of  $\bar{\nu}$  that has four main factors. These are a numerical pre-factor, a collection of physical constants, a  $T^{3/2}$  term, and the Coulomb logarithm. The first two components of  $\bar{\nu}$  are constants. These constants can change based on the approximations that are used when deriving a value of  $\bar{\nu}$ . The third part, the  $T^{3/2}$  scaling for  $\bar{\nu}$  is present regardless of what approximations are used. Finally, the fourth part of  $\bar{\nu}$  is the Coulomb logarithm,  $\ln(\Lambda)$ . This term arises from the averaging of all possible collisions. The general formulation of this logarithm remains consistent as long as a plasma system can be considered weakly coupled. However, once a plasma system becomes strongly coupled this general formulation break downs. This breakdown of the Coulomb logarithm will be discussed in the next subsection.

### 3.1.4 Strong Coupling Corrections

The expressions in the previous sections assumed that particle kinetic energy was much larger than nearest neighbor potential energy. However, as the plasma becomes strongly coupled (see chapter 1 for details) this approximation is no longer valid. As degree of strong coupling in the system increases, these expressions start to break down. This breakdown is perhaps easiest seen by rewriting  $\bar{\nu}_{ei}$  in terms of the electron strong coupling parameter, which quantifies the degree of strong coupling in the plasma. This yields the following expression:

$$\bar{\nu}_{ei} = \sqrt{\frac{2}{3\pi}} \omega_p \Gamma^{3/2} \ln(\Lambda) = \sqrt{\frac{2}{3\pi}} \omega_p \Gamma^{3/2} \ln(C\Gamma^{-3/2}) \quad (3.20)$$

where  $\omega_p$  is the plasma frequency,  $C$  is a constant that is determined by the choice of  $b_{max}$ , and  $\Gamma$  is the strong coupling parameter which is defined as  $\Gamma = q^2/4\pi\epsilon_0 ak_b T$ , where  $a$  is the Wigner Seitz radius. The breakdown is most readily seen in the Coulomb logarithm. As the coupling parameter increases,  $\Lambda$  will eventually fall below 1. This leads to the logarithm turning negative, and implying a negative collision rate. This is clearly unphysical, and necessitates an alternative formulation of the Coulomb logarithm for strongly coupled systems.

It is not particularly surprising that the general expression for  $\bar{\nu}_{ei}$  breaks down at significant electron strong coupling. As electron strong coupling in a plasma system increases electron-electron correlations begin to develop. As a result, the development of these correlations alter the screening properties of the plasma. Intuitively, it makes sense that a change in the screening properties of the plasma would alter the collision rate,  $\bar{\nu}_{ei}$ . The calculation of  $\bar{\nu}_{ei}$  involves averaging over all possible collisions by applying a cutoff based on the screening length of the plasma. If the screening properties of the plasma are changed, one would expect that the relevant cutoff would change with it.

Extending the Coulomb collision framework into the strongly coupled regime has received a lot of theoretical interest. A number of these extensions have been developed in previous

work[3, 4, 5, 6]. A common extension is

$$\ln(\bar{\Lambda}) = \ln(1 + \Lambda) \quad (3.21)$$

where  $\ln(\bar{\Lambda})$  is the adjusted Coulomb logarithm. Figure 3.3 illustrates how this extension alters the Coulomb logarithm at high  $\Gamma$ . Experimentally it is quite difficult to resolve the effects that strong coupling has on the Coulomb logarithm, let alone to test whether or these effects are consistent with a certain strong coupling extension (such as the one outlined in Eq. 3.21). This is because, for achievable electron strong coupling parameters in many plasma systems, the expected corrections are typically quite small, and the error of available experimental measurements is often too large to resolve these effects. In fact, in strongly coupled plasma system with negligible interaction with neutrals (such as UCPs) the effects of electron strong coupling on the Coulomb logarithm have yet to be experimentally resolved. One of the main goals of the work outlined in this chapter is to test the validity of the extension shown in Eq. 3.21.

The extension shown in Eq. 3.21 leads to alteration of the effective cutoff,  $b_{max}$ . Since

$$\frac{\bar{b}_{max}}{b_{90}} = 1 + \frac{b_{max}}{b_{90}} = 1 + \frac{0.7\lambda_D}{b_{90}} \quad (3.22)$$

where  $\bar{b}_{max}$  is the altered  $b_{max}$ , and  $b_{max}$  is evaluated as  $0.7\lambda_D$  as per[3], the effective cutoff becomes.

$$\bar{b}_{max} = 0.7\lambda_D + b_{90} \quad (3.23)$$

Equation 3.21 and 3.23 both describe the same physics, but convey it in a different way. By altering the cutoff in this manner, it is possible to simulate collisions in a manner that will effectively recreate the Coulomb logarithm outlined in Eq. 3.21. By utilizing this formulation of  $b_{max}$ , the model described later in this chapter is able to implement strong coupling corrections. This will be described in more detail in section 3.3.2.

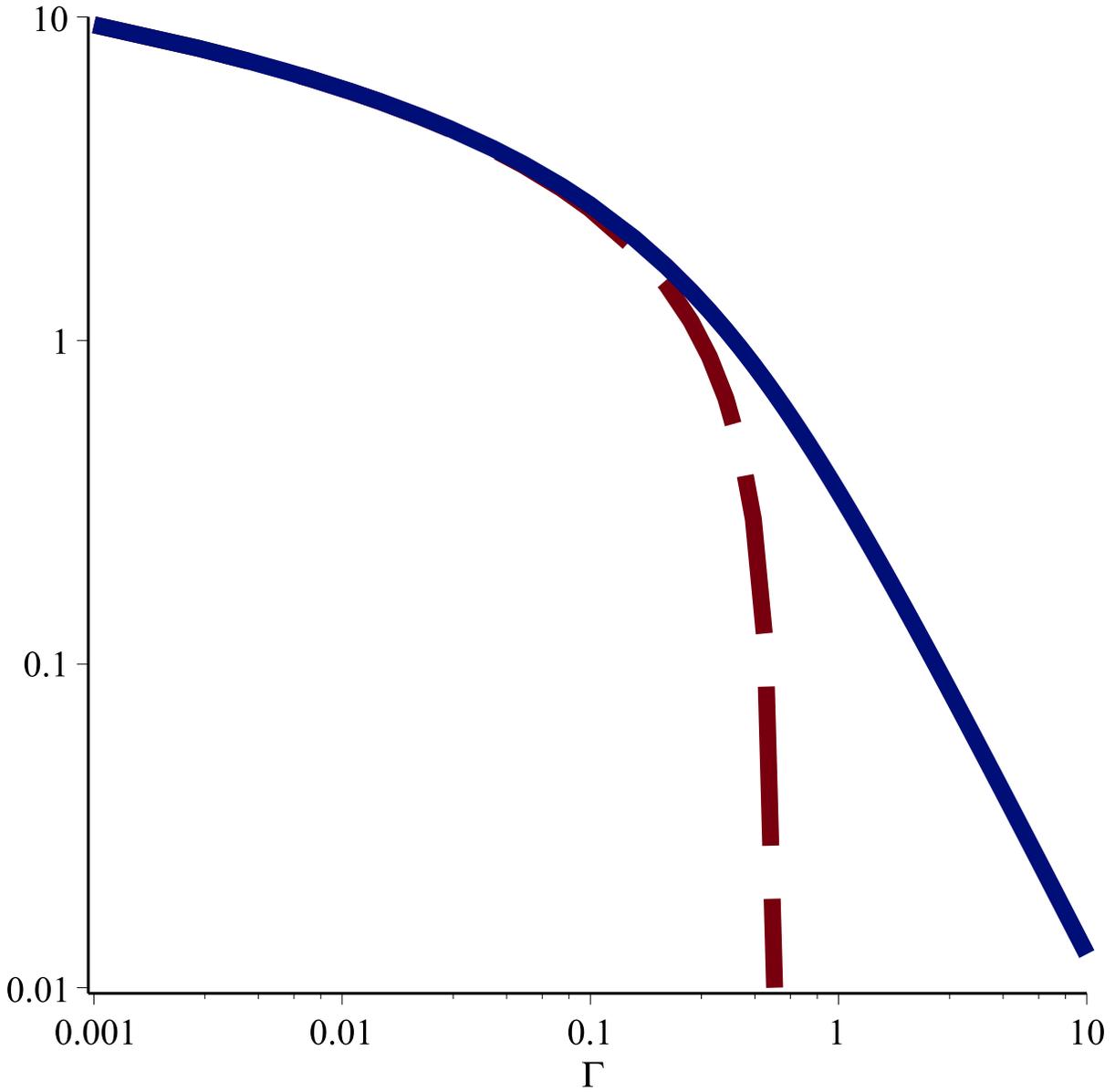


Figure 3.3: Graph of the Coulomb logarithm as a function of  $\Gamma$ . The red dotted line corresponds to the Coulomb logarithm in the weakly coupled limit, and the blue solid line is the Coulomb logarithm with the appropriate strong coupling corrections as seen in Eq. 3.21. The graph illustrates that, in the weakly coupled limit, the two different expressions for the Coulomb logarithm agree. It can also be seen that the two expression begin to diverge around  $\Gamma = 0.1$ , and that the weak coupling approximation has completely failed by the time  $\Gamma = 1$ .

The electron center-of-mass damping rate measurement discussed in the remainder of this chapter provided the opportunity to test the effects of electron strong coupling on the electron-ion collision rate. UCP parameters were cold and dense enough for significant electron strong coupling, and measurement precision was sufficient to resolve the effects of this strong coupling. The next section will give an overview of the experimental method that was used to make these measurements.

## 3.2 Experimental Overview

This section will give an overview of the experimental system located in our lab, as well a summary of the experimental procedure utilized to measure the center-of-mass oscillation damping. Since my contribution to this measurement was mainly computational, this section will be brief. However, if the reader is interested in further details, a more thorough description of the apparatus can be found here[9], and additional discussion on the experimental methodology can be found here[10, 11].

### 3.2.1 Overview of the Experimental Apparatus

Our experimental system contains two distinct sections enclosed in a single vacuum chamber. In the first section of the experiment, atomic physics techniques are utilized to cool and trap  $^{85}\text{Rb}$  atoms in a magneto-optic trap. From there the cooled atoms are transferred to the second section of the experiment via a magnetic trap. Here, the atoms are photoionized, creating a UCP. UCP dynamics can then be studied using charged particle detection (described in 1.3.2). A diagram of the experimental setup can be seen in figure 3.4. Given the focus of this thesis work on computations relevant to ultracold plasmas, more highly detailed descriptions of the experimental apparatus are left to other theses from the research group, in particular those of Truman Wilson[9] and Wei-Ting Chen (in preparation).

The cooling and trapping of atomic Rb is accomplished by utilizing standard techniques for creating a magneto-optic trap (MOT)[12, 13, 9]. Utilizing these techniques, Rb gas is

cooled down from room temperature to around  $100\mu\text{K}$ . This results in a ball of roughly 40 million Rb atoms with a diameter of on the order of a mm.

$^{85}\text{Rb}$  was chosen for this experiment largely due to its properties that make it favorable to cooling and trapping. The  $5S_{1/2}$  to  $5P_{3/2}$  cooling transition requires a laser wavelength of 780nm. This wavelength is easily accessible, and can be generated by inexpensive laser diodes. Furthermore, the  $5P_{3/2}$  state only has two decay paths, requiring only a single repump laser. Thus only two lasers are required for a laser cooling scheme. Also, Rb is magnetically trappable in its ground state, allowing for such a trap to be used in this experiment. Finally, the level structure of atomic Rb allows for a two-step photoionization technique to be straightforwardly applied with reasonable lasers as well.

Once cooled the atomic Rb is loaded into a magnetic trap and transferred to the ionization section of the experiment. The magnetic trap consists of two separated and parallel multiturn coils of wire. These coils are mounted on a automated translation stage, which transfers the trap from the MOT to the ionization region of the experiment. The coils are oriented in an anti-Helmholtz configuration. This configuration yields 0 B field in the center of the trap, and a B field gradient in the radial and axial directions, creating a field minimum with respect to the magnitude of the magnetic field,  $|B|$ , at the center of the trap.

The potential energy of an atom in a magnetic field is:

$$U = g_F m_F \vec{\mu}_B \cdot \vec{B} \tag{3.24}$$

where  $\vec{\mu}_B$  is the Bohr magneton,  $\vec{B}$  is the applied external magnetic field,  $m_f$  is the magnetic quantum number, and  $g_f$  is the g-factor which has a value of  $-1/3$  for the  $F=2$  ground state[9]. While equation (3.24) involves a dot product, for our experimental parameters the orientation of the atoms' spin adiabatically adjusts to changes in orientation of the magnetic field, except for a very small volume at the center of the magnetic trap. Since  $\mu_B$  and  $B$  maintain the same relative alignment, the trapping potential is proportional to  $|B|$ . As can

be seen from Eq. 3.24,  $U$  decreases with decreasing  $\vec{B}$  for negative values of  $m_f$ , and thus the anti-Helmholtz coil configuration leads to the trapping of negative  $m_f$  atoms at the field minimum in the center of the trap. The only purpose of the magnetic trap is to allow the transfer of atoms from the MOT region in the vacuum system to the photoionization region where the UCPs are formed.

Once the atoms are transferred to the ionization region of the vacuum chamber they are ionized by a two photon process. Laser light from the MOT is used to excite the atoms into the  $5P_{3/2}$  state. From there, tunable light with a wavelength near 480nm from a pulsed dye laser is used to ionize the excited atoms. A diagram of this process can be seen in figure 3.5. The dye laser is widely tunable, allowing for a wide range of post-ionization electron energies. Since this post ionization energy is the main factor in determining the post formation electron temperature, it is often convenient to describe electron energies in temperature equivalent units. In these units, up to  $750K \cdot k_b$  of post-ionization energy has been initially imparted to the low-density UCPs in our system and a UCP has still been able to form. Since, for a well thermalized plasma, the average electron energy,  $\langle E \rangle$ , is defined as  $\langle E \rangle = 3/2k_bT_e$ , UCP post formation electron temperatures,  $T_e$ , from roughly 1-500K are obtainable. The lower limit of this range is due to heating mechanisms (see chapter 4) that are independent of ionization energy.

System plasma dynamics are monitored by the use of a charged particle detection scheme. A microchannel plate is located a meter from the UCP, and is used to detect escaping electrons. In order to guide electrons that escape from the UCP to the detector, a series of cylindrical electrodes are inserted into the vacuum chamber. These electrodes provide an electric field configuration that directs escaping electrons towards the microchannel plate. Additionally, these electrodes can be used to alter plasma conditions. Changes in the bias electric field at the location of the UCP can yield changes in the plasma depth. Electric field ramps can lead to changes in the UCP charge imbalance. Resonant RF pulses can be used to induce plasma oscillations. Finally, the experiment also contains a magnetic field coil.

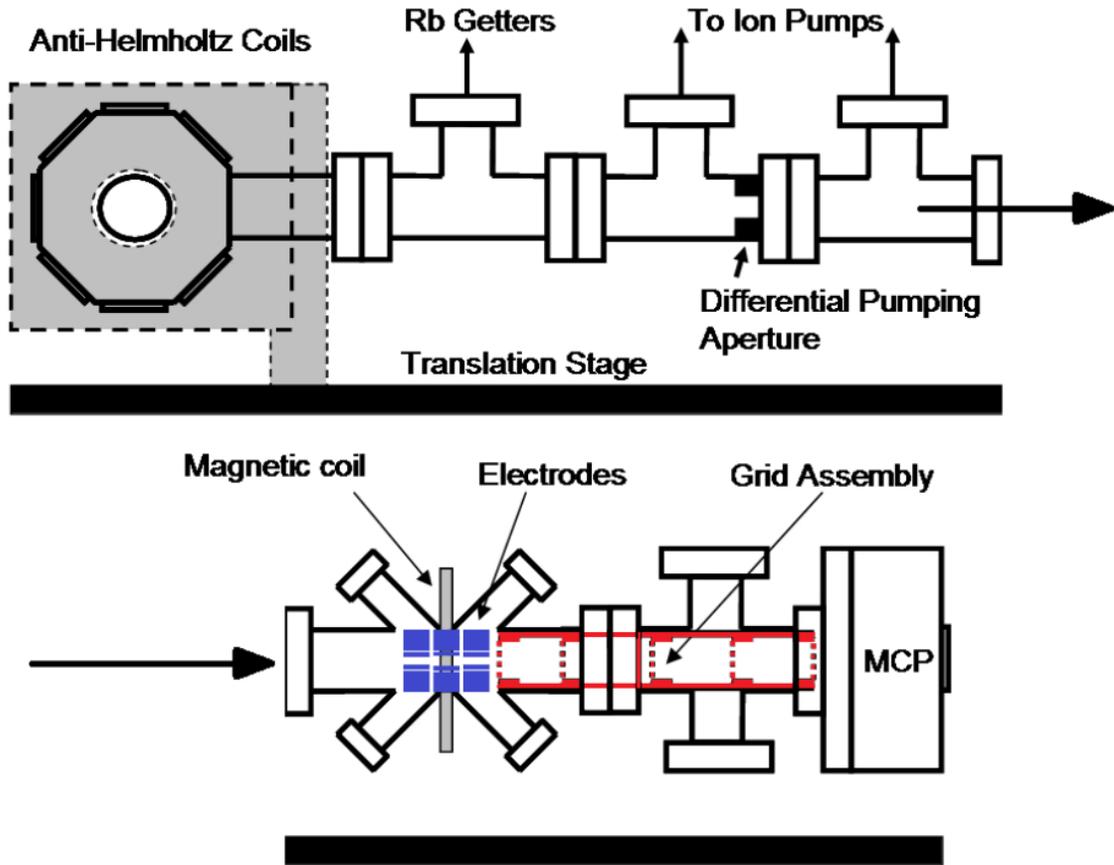


Figure 3.4: A simple diagram of the experimental apparatus in our lab[9]. In the upper left hand corner of the diagram, the at location of the anti-Helmholtz coils in the figure, is where the Magneto-Optic trap is located. At that location atomic Rb is cooled and trapped using now-standard atomic physics techniques. The cooled atoms are then loaded into a magnetic trap created by the Anti-Helmholtz coils. These coils are mounted on a translation stage allowing the atoms to be moved to the second part of the experiment, shown on the bottom of the diagram. Here the atoms are ionized, creating a plasma. The magnetic coil and the electrodes, shown on the lefthand side of the bottom half of the diagram, are used to apply external magnetic and electric fields respectively. The MCP on the bottom lefthand side of the diagram, is a Microchannel plate that is used for charge particle detection, and is the main diagnostic of the experiment.

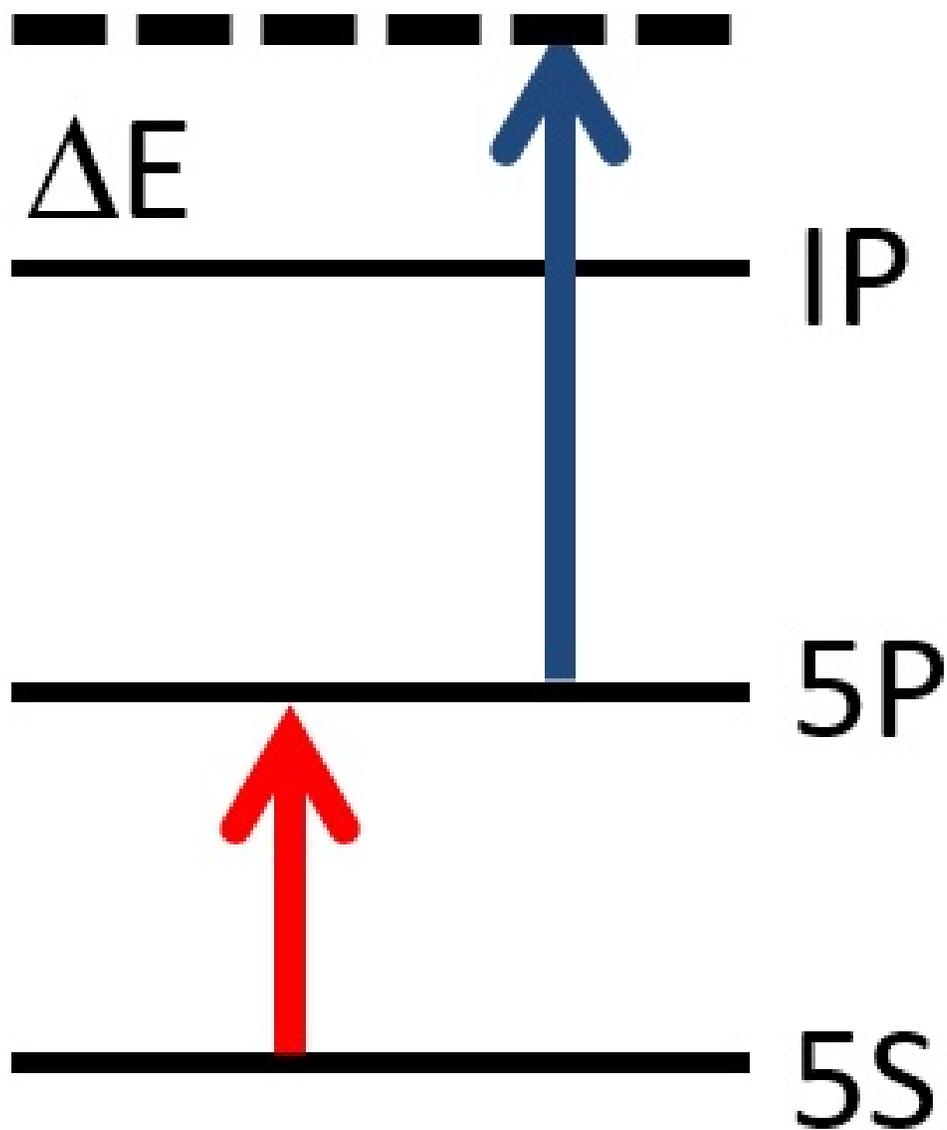


Figure 3.5: Simple diagram of the two-photon ionization process that is used to create a UCP in our lab. The first photon excites the  $^{85}\text{Rb}$  atoms from the 5S ground state to the 5P excited state. The second photon is then able to fully ionize  $^{85}\text{Rb}$  atoms in the 5P excited state. Since the ionizing photons are generated from a dye laser, the frequency of these photons can be altered. By altering the frequency of these photons, the amount of energy imparted to the Rb electrons, in excess of the ionization threshold,  $\Delta E$ , is also altered. This allows for the dye laser to finely tune the initial electron kinetic energy of the UCP.

This coil is used to apply external magnetic fields to the UCP, and can thus be used to test the effects of applied magnetic fields on the plasma.

### 3.2.2 Experimental Procedure

The original goal of the series of experiments described in this chapter was to measure the electron-ion collision rate in the system. To do this, external electric field pulses are used to initiate a collective electron oscillation. The rate at which this oscillation decays can be measured. If it is assumed that the damping of the electron oscillation is caused exclusively by electron-ion collisions, a measurement of the electron oscillation damping rate can be theoretically linked to the electron-ion collision rate. Thus, by measuring the damping rate, a measurement of the electron-ion collision rate can be made.

To experimentally measure the electron oscillation damping rate a two electric field pulse technique was utilized. The reason that a two-pulse technique was used is that the primary diagnostic in the system is the electron escape rate, as measured by the microchannel plate. The first field pulse is used to initiate a plasma oscillation. After a brief delay, a second pulse is then applied to the plasma, resulting in a number of electrons escaping from the plasma. The electron escape is driven by the internal electric fields that are created by the oscillations of the electrons. For electrons in the UCP with just the right positions and velocities, these fields can accelerate them dramatically such that they are driven out of the plasma, creating the signal that is detected. The number of electrons that escape from the plasma scales linearly with the amplitude of the electron center of mass motion. Thus, if the second pulse has a delay time such that it accelerates the electrons in phase with their center-of-mass motion, the oscillation amplitude will be increased by the second pulse and more electrons will escape. Conversely, if the second pulse has a delay time such that the acceleration from the second pulse is out-of-phase with the oscillating electrons, the oscillation amplitude will decrease due to the application of the second pulse and fewer electrons will escape. By

measuring the total number of electrons that escape as a function of delay time between the pulse it is possible to map out the oscillation amplitude as a function of time.

Some work was needed in order to verify the linear scaling of the electron escape signal with the electron oscillation. The evidence of this scaling was observed both in the form of theoretical calculations and experimental data. On the theoretical side, the two pulse experimental procedure was explicitly modeled in both a moment-based model[14], as well as in the hybrid model that will be discussed later in this chapter. In both cases the models generally agreed with the predicted linear scaling. On the experimental side, previous work using RF pulses to induce resonant plasma oscillations[15] can also provide some insight. It was found that the electron escape signal scaled linearly with the amplitude of the RF induced oscillation. While these oscillations are not the same as electron center-of-mass oscillations, it seems likely that the escapes signal from both of these oscillations would scale in the same manner.

For a typical set of conditions, about 20 different delay times were measured. These delays were evenly spaced with a 10ns spacing. This resolution was sufficient to resolve the electron oscillations, which had oscillation periods typically between 50-70ns. To extract a damping rate, the data can be fit to a decaying sinusoid taking the following form

$$Ae^{-kt} \sin(\omega t + B) + C \tag{3.25}$$

where  $A$  is a proportionality constant,  $B$  is a phase factor,  $C$  is an offset,  $\omega$  is the oscillatory frequency, and  $k$  is the decay rate. While all of five of these constants are treated as fit parameters, the damping rate,  $k$ , is the main parameter of interest in terms of extracting the collision rate. The oscillation frequency is useful in determining the UCP density, too. An example of this fit to a sample set of data can be seen in figure 3.6.

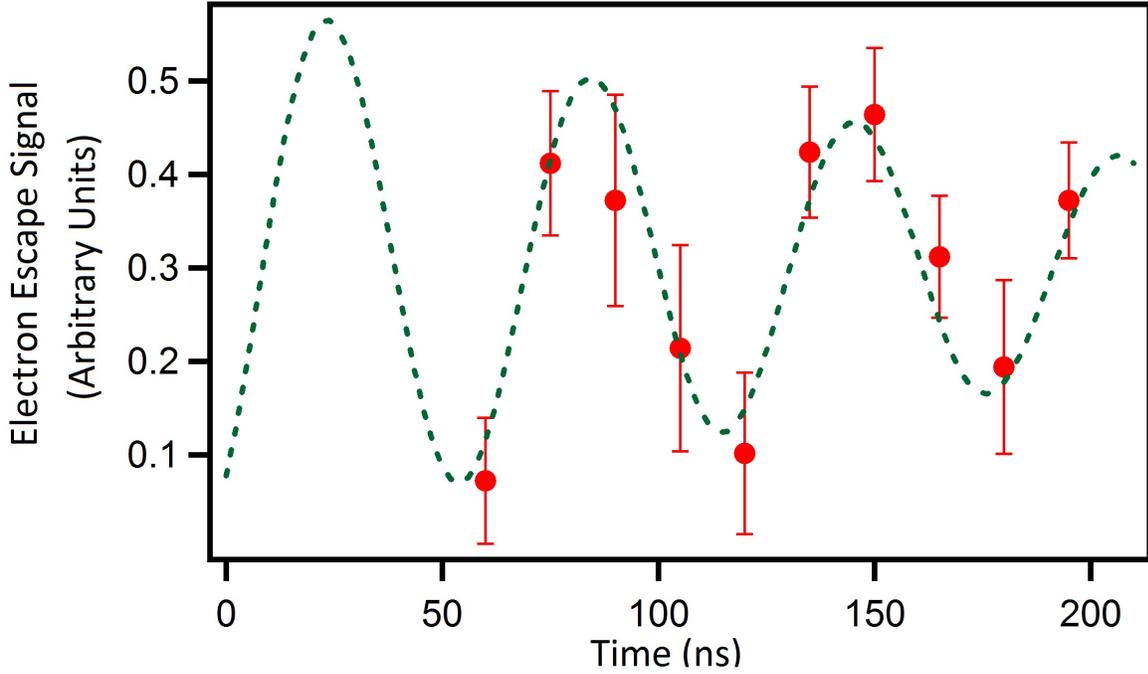


Figure 3.6: An example of an experimental data set. The data points show the electron escape as a function of the time delay between the first and second electric field pulse. The dashed green line is the best fit damped sin wave with fit parameters that are outlined Eq. 3.25. The data points correspond to a UCP with initial electron kinetic energy  $2.26K \cdot k_b$ , and ion and electron numbers of  $5.9 \times 10^4$  and  $2.7 \times 10^4$  respectively.

## 3.3 Numerical Modeling

In order to extract the electron-ion collision rate as well as the electron temperature from the experiment, it is necessary to theoretically link these quantities to the measured electron oscillation damping rate. To make this link, I utilized numerical modeling to simulate the electron oscillations present in the system. This section will give an overview of this modeling, as well as provide a detailed explanation of the collision operator which is heavily utilized in these calculations.

### 3.3.1 Overview

In general, the numerical models used in this experiment utilized molecular dynamics techniques to simulate the time evolution of a UCP undergoing a plasma oscillation analogous to what occurred in the experiment. In all cases these simulations included explicit calculations of electron-electron and electron-ion interactions. Furthermore, these simulations included particle interactions with external fields when relevant. Most simulations contained an external DC electric field, and many included a constant magnetic field. All of these simulations utilized GPUs, and were structured in the manner outlined in section 2.5.

While some of the simulations utilized a full molecular dynamics treatment for both the electrons and the ions, most of the simulations utilized a hybrid molecular dynamics and Monte Carlo technique. In a full molecular dynamics treatment, both electrons and ions were treated as discrete particles that would evolve in time as a result of the forces calculated between them. By contrast, in my hybrid model, only electrons were treated as discrete particles. Ions are included in terms of their average field effects as a smooth density distribution that is matched to the experimental density distribution. For the purposes of this model, the spatial distribution of the ions does not evolve in time. Model ions are considered to interact with electrons in two ways. First, the collective ion distribution exerts a force on the electrons, as described in section 2.5.3. This force, however, does not account

for the discrete nature of the ions in a UCP, and thus does not capture the physics of electron-ion collisions. Thus, electron-ion collisions need to be inserted into the model in another way. This leads to the second way that model ions interact with model electrons: through a Monte Carlo collision operator. The Monte Carlo collision operator randomly simulates collisions for electrons in the model by rotating the electron velocity vector. This collision operator will be described in greater detail in section 3.3.2.

The hybrid model calculates the time evolution of electrons in the manner described in section 2.5. As with any molecular dynamics simulation, the evolution of the electrons is broken up into many small timesteps over which the force acting on each electron can be considered approximately constant. At each of these timesteps the net force acting on each electron is calculated. By integrating these forces over the appropriate timestep, the positions and velocities of each electron are updated. The method of integration used to update the electrons is known as the leapfrog method, which is discussed in detail in chapter 2. Finally, for each electron the model checks to see if it undergoes an electron-ion collision, and implements a collision if necessary.

The hybrid model offers two main advantages over the full molecular dynamics treatment. The first advantage is that it is substantially more computationally efficient for our range of experimental parameters. In a typical simulation, the total number of discrete particles ( $N \sim 10^5$ ) is sufficiently high so that the Coulomb interaction calculations account for the vast majority of computation time and calculations for the collision operator are trivial. Typical experimental conditions have a charge imbalance of around 50%, and thus removing discrete ions from the model reduces simulation particles by about a factor of 3. Since the Coulomb interaction calculations are an  $\mathcal{O}(N^2)$  calculation, this reduction in the number of particles leads to almost an order of magnitude speed increase by itself. Furthermore, larger timesteps can be used in the simulations that don't involve discrete ions and that results in a faster computation time as well.

The second main advantage of the hybrid model is that it allows for an explicit test of collision physics. In a full molecular dynamics simulation only Coulomb forces are included. Since Coulomb forces can result in many different physical processes (collisions, waves, etc.), it can often be difficult to untangle these different processes in molecular dynamics simulations. Conversely, by explicitly turning off electron-ion collisions in a simulation, the hybrid model can test the precise effect of electron-ion collisions. This property can be used to test the collisionality of different plasma processes. Furthermore, by assuming a weakly coupled scaling for collisions in the hybrid model, results from the model can be compared with low temperature experimental data allowing for the breakdown of the weak coupling approximations to be quantified. Additionally, by altering the collisional scaling, the hybrid model will also allow for the testing of the different strong coupling extensions that were described in section 3.1.4.

### 3.3.2 Collision Operator

The Monte Carlo collision operator is central to the functioning of the hybrid model. The operator is used to determine whether or not an electron undergoes a collision, and to rotate the electron velocity vector in the case that a collision does occur. In general, the operator is implemented in the following way. First, the operator randomly determines whether or not a collision occurs. If a collision does occur, an impact parameter for the collision is randomly generated. From this impact parameter a velocity vector rotation angle can then be calculated. Once this angle is known, the electron velocity vector is then rotated.

To determine whether a collision occurs, the rate at which collisions occur needs to be calculated. By integrating Eq. 3.8, the frequency in which an electron undergoes a collision can be expressed as  $\pi n_i v b_{max}^2$ , where  $n_i$  is the ion density, and  $v$  is the electron velocity. From this expression, the probability that a collision occurs during a time period  $dt$  can be expressed as.

$$P = \pi n_i v b_{max}^2 dt \tag{3.26}$$

A random number is generated for each electron at every timestep and that number is compared to the collision probability in equation (3.26). For such a method to work, timesteps must be small enough so that the probability for a collision to occur is much less than one. This is because for higher collision probabilities, the probability for multiple collisions to occur becomes non-negligible. Since this method can only handle a single collision per time step, time steps associated with higher collision probabilities will systematically undercount the total number of collisions. By choosing a sufficiently small timestep, simulations were designed to make such considerations negligible.

Once a collision is determined to have occurred, an impact parameter,  $b$ , is then calculated. The distribution of all possible impact parameters when normalized can be described by the following.

$$\int_0^{b_{max}} \frac{2bdb}{b_{max}^2} = 1 \quad (3.27)$$

This equation implies the following distribution of the impact parameter

$$\int_0^b b' db' = \frac{b_{max}^2}{2} R \quad (3.28)$$

where  $R$  is a uniformly distributed random number in the range of 0 to 1. Solving this equation for  $b$  leads to the following expression.

$$b = b_{max} \sqrt{R} \quad (3.29)$$

By expressing the impact parameter distribution in this way, an explicit relationship between  $b$  and  $R$  is established. This allows for a random value of  $b$  to be properly calculated with a single random number generation.

Some thought was needed to choose the right value for  $b_{max}$ . In the weak coupling limit the general form of the Coulomb logarithm is.

$$\ln(\Lambda) = \ln\left(\frac{b_{max}}{b_{90}}\right) \quad (3.30)$$

Typically,  $b_{max}$  takes the form of  $C\lambda_D$ , where  $C$  is a constant. However, in most theory papers, such as [3], the Coulomb logarithm takes the following form.

$$\ln\Lambda = \ln\left(C\frac{4\pi\epsilon_0\lambda_D k_b T_e}{q^2}\right) \quad (3.31)$$

Since,  $b_{90}$  is a function of  $v$ , Eq. 3.31 is inconsistent with the standard definition of  $b_{max}$  and Eq. 3.23. This inconsistency comes about from an ad hoc substitution of  $mv^2$  with  $k_b T_e$  in the Coulomb logarithm. The motivation for this substitution comes about from the integral in Eq. 3.15. This substitution allows for the Coulomb logarithm to be pulled outside of the integral, and thus making the math easier.

For many plasma systems this approximation is reasonable, since the Coulomb logarithm remains relatively constant over the full range of system parameters. Typical high-temperature plasmas in astrophysical and magnetic fusion environments have  $\ln(\Lambda) \sim 10$ -20. Under such conditions, a factor of 2 change in  $\Lambda$  changes the Coulomb logarithm by only a few percent. This is not true for UCPs with cold electron temperatures, however, and thus more care needs to be taken. To formally accommodate the usual replacement of  $mv^2$  terms with  $k_B T$  terms that is common in plasma physics,  $b_{max}$  was redefined in the following way.

$$b_{max} = C\lambda\frac{k_b T_e}{mv^2} \quad (3.32)$$

This new formulation allowed for the definition of the Coulomb logarithm to be in agreement with expressions from theory papers and textbook expressions describing electron-ion

collision rates. This in turn allows for the Monte Carlo collision operator to test theory predictions in a consistent manner.

A similar correction can also be made for the strong coupling formulation of  $b_{max}$  outlined in Eq. 3.23 This correction leads to the following expression of  $b_{max}$ .

$$b_{max} = b_{90} + C\lambda \frac{k_b T_e}{mv^2} \quad (3.33)$$

The simulations performed in this chapter generally used this form of  $b_{max}$ . Additionally,  $C$  was chosen to be 0.7, matching the measured value in[3].

Once an impact parameter is randomly generated, the rotation angle,  $\chi$ , can found by evaluating Eq. 3.3. Once  $\chi$  is known the electron velocity vector can be rotated using rotation matrices

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.34)$$

where  $R_z(\theta)$  is a rotation of  $\theta$  around the z-axis, and  $R_y(\theta)$  is an analogous rotation around the y-axis. The rotation of the electron velocity is implemented in the following way

$$\vec{v}' = R_z(\theta)R_y(\phi)R_z(\phi')R_y(\chi) \begin{bmatrix} 0 \\ 0 \\ |\vec{v}_0| \end{bmatrix} \quad (3.35)$$

where  $\vec{v}'$  is the final electron velocity vector,  $\vec{v}_0$  is the initial electron velocity vector,  $\phi'$  is a random rotation orthogonal to the plane of the  $\chi$  rotation, and  $\theta$  and  $\phi$  represent the usual angles in spherical coordinates associated with the initial velocity vector. In the absence of a collision (i.e.  $\chi = 0$ ), the action of Eq. (3.35) is such as to have  $v'$  equal the original velocity

vector in direction and magnitude. The  $\phi'$  rotation is randomly generated from a uniform distribution ranging from 0 to  $2\pi$ .

### 3.3.3 Initialization of Model Parameters

An important component of making the electron oscillation simulations work properly is making sure that the appropriate initial conditions are obtained before the oscillation is induced. Naively, one would think that this could be easily be achieved by simulating the formation of the UCP. However, in our experimental system this takes roughly  $3 \mu s$  to occur, at least to the extent that the system is well-suited for data collection. This would lead to integration times on the order of half a day, drastically increasing simulation time.

Aside from this practical consideration, there are additional challenges. First, the atomic cloud is likely not completely symmetric. This asymmetry can lead to an alteration of the ion potential once the atoms are ionized, which in turn can lead to additional escape paths for electrons, as well as change the degree of disorder induced heating(see chapter 4). However, this is likely a fairly small effect, and almost certainly dwarfed by the second complicating factor: The fact that the external electric field topology is not perfectly known. Since the external field is created by a series of cylindrical electrodes a DC field cannot just be simply created. This is because there are requirements not just to produce the field in the plasma region but also to use a configuration in which the electrons are guided to the detector without striking an electrode. Instead, an approximate DC field is created via superposition of fields from multiple electrodes at the position of interest. Under the conditions at which we have recently collected data, we see evidence that the electric field is consistent and nearly constant over the UCP (and certainly the central region). However, experimental and modeling evidence show that there is a "cliff-like" change to the electric potential outside the plasma region (or very near the edge). The data that lead to this conclusion include electric field calibration measurements combined with measurements of the electron evaporation rate. Without including this "cliff", simulations underestimate the charge imbalance of the

system. It is worth noting that the experimental apparatus has recently been reconfigured with wire grids in a geometry that will drastically reduce this problem.

To compare experiment and model predictions, the electron oscillation model needs to be fed a determination of what the post formation distribution looks like. Experimental diagnostics are able to give most of this information, as electron number, ion number, and the spatial extent of the ion cloud can all be determined experimentally. Furthermore, the frequency of the ionizing dye laser can be used to constrict the range of possible electron temperatures. Finally, by assuming spherical symmetry and a Gaussian ion density distribution, as well as asserting an electron temperature, the initial conditions of the simulation are completely defined. From these initial conditions, modeling can either be carried from time  $t=0$  to the time after plasma formation or initial specifications can be provided at a chosen time (e.g.  $t = 3\mu s$ ) and the simulations carried forward from there. Care was taken to make sure that any measurements that were made were not significantly influenced by our imprecision in our electric field topology determination.

Another important component of properly initializing the UCP is the placement of the discrete electrons. When initializing the electrons, it is important that they are distributed in a way that approximately mirrors their equilibrium distribution. Model electrons are placed into the system by randomly assigning their spatial coordinates via an appropriate underlying distribution. If the model distribution differs substantially from the equilibrium distribution, potential energy in the system will be rapidly converted into kinetic energy, leading to significant heating of the electrons. This necessitates knowledge of an electron distribution that at least approximately matches the correct equilibrium electron distribution.

To insure the proper placement of the electrons, the following approximation is utilized. To be in thermal equilibrium the electron density,  $n_e$  must take the following form.

$$n_e \propto e^{-\frac{q\phi(r)}{k_b T_e}} \quad (3.36)$$

where  $q$  is the fundamental charge constant,  $\phi(r)$  is the electric potential,  $k_b$  is the Boltzmann constant, and  $T_e$  is the electron temperature. The approximation then assumes that the electron distributions is roughly the same shape as the ions' distribution, and is thus a Gaussian

$$n_e \propto e^{-\frac{r^2}{2\sigma^2}} \quad (3.37)$$

where  $\sigma$  is the characteristic Gaussian length. By combining Eq. 3.35 and 3.36,  $\phi$  can be expressed as the following.

$$\phi(r) = \frac{k_b T_e r^2}{2q\sigma} \quad (3.38)$$

By combining Eq. 3.37 with Poisson's equation the total (i.e. net) charge density of the system can be found.

$$\rho(r) = \frac{3k_b T_e \epsilon_0}{q\sigma} \quad (3.39)$$

where  $\rho(r)$  is the total charge density. Since electron temperatures for these simulations are typically near  $T_e = 0$ , a zero temperature approximation can be made. This yields a total charge density of 0 for the UCP, which results in the electron distribution directly mirroring the ion distribution. However, since the total number of electrons is less than the number of ions, a cutoff,  $r_c$ , for the electron spatial distribution must exist. This cutoff is a function of the charge imbalance,  $\delta$ , and can be solved for using the following equation.

$$N_i \int_0^{r_c} n(r) d^3r = (1 - \delta)N_i \quad (3.40)$$

where  $N_i$  is the total number of ions in the UCP,  $n(r)$  is the a normalized Gaussian distribution with a  $\sigma$  that properly corresponds to the ion density, and  $\delta$  is the charge imbalance parameter defined as  $\delta = (N_i - N_e)/N_i$ . Since  $1 - \delta$  is the ratio of the electron number to

the ion number, the integral in Eq. 3.40 corresponds to the total number of electrons in the UCP. Implementing this cutoff yields the final guess electron distribution.

$$n_e(r) = \begin{cases} n(r) & r \leq r_c \\ 0 & r > r_c \end{cases} \quad (3.41)$$

This distribution matches the ion distribution until the cutoff,  $r_c$ , is reached. At this point, the total number of UCP electrons are already contained in the distribution, and thus the electron density falls to 0.

The existence of an external electric field complicates the placement of the electrons in the model. These fields lead to a displacement of the electron component of the UCP. In reality, this displacement results in the shape distortion of the electron cloud. However, since the model only needs an approximately correct distribution, a first order approximation of the electric field induced electron displacement is sufficient. This approximation assumes that there are no shape distortions of the electron cloud. This allows for the electron cloud to be treated as a single object, that is being acted on by two separate forces. These are the force from the electric field, and the force that the ions exert on the displaced electron cloud,  $F_i$ , which can be expressed as a function the electron cloud displacement,  $\Delta z$ . In this approximation, the electron cloud will be in equilibrium when these two forces are equal and opposite, and thus  $\Delta z$  can be solved for by equating the two forces. The E-field force can be simply expressed as  $EN_eq$ , where  $E$  is the external electric field. Evaluating  $F_i(\Delta z)$  is significantly more complicated, as it involves evaluating a six dimensional integral. For this reason,  $\Delta z$  is typically found through an iterative series of guesses and not by explicitly solving for it. Once a reasonable value for  $\Delta z$  has been found, the distribution shown in Eq. 3.41 is shifted by  $\Delta z$  in the  $\hat{z}$  direction, which is the direction of the external electric field.

After the electrons have been randomly placed in the proper distribution, the UCP is allowed to evolve in time. This phase lasts for  $300ns$  for conditions matching those in the experiments, and during this period, the electron cloud evolves from an approximate equi-

librium distribution to the actual equilibrium distribution. Since this process reduces the total potential energy in the UCP, it also leads to the heating of the UCP's electrons. The model typically handles this by applying a one-time energy correction to the electrons. This correction is implemented by altering the total velocity of all of the electrons to obtain the appropriate temperature of the system, as defined by  $1/2m_e v^2 = 3/2k_b T_e$ . However, this method is only effective for small scale corrections. Since the actual electron distribution is a function of the electron temperature, large temperature adjustments can have a detectable effect on the equilibrium electron distribution. For this reason, large corrections often result in substantial post correction heating causing the simulation to miss its temperature target. For full molecular dynamics simulations, the large energy corrections are even more problematic. By significantly reducing the kinetic energy of a particle, free particles can become bound, dramatically altering the system. Thus, care is taken to make sure that the approximate electron distribution is close enough to the equilibrated distribution to avoid these difficulties.

### 3.3.4 Oscillation Damping

After the initialization phase is over, an electron oscillation is induced in the model. This is done by imparting a one time impulse that adds a constant "kick" velocity in the  $\hat{z}$  direction to all of the electrons in the system. For the vast majority of the simulations run, this kick velocity was 6000 m/s to match experimental parameters. Since oscillations in the experimental system were induced by a finite width electric field pulse, it is worth considering if a velocity impulse is really an appropriate way of modeling the oscillations. The main concern is that part of the field pulse will be screened out, and that the velocity kick will not be as significant as what would be expected by a straight forward impulse calculation. Simple scaling arguments suggest that this should not be a problem, as the pulse width of less than 10 ns is roughly an order of magnitude faster than the inverse plasma frequency. However, to be careful, simulations were run where the electric field pulse

was explicitly modeled with its measured time dependence. These simulations reinforced the simple scaling argument, and showed ignorable deviations from equivalent simulations that used a velocity kick.

Unlike the experiment, the model only uses a single kick. In the experiment, the second pulse was used explicitly as a probe of the center of mass velocity in the  $\hat{z}$  direction. Since the model knows the velocity of all of the particles in the simulation, the center of mass velocity can be explicitly calculated without perturbing the system. In principal, a model could be made that more faithfully recreates the experiment, but such a model would suffer from many of the same issues that make modeling the UCP formation so difficult. In particular, the imprecision in the electric field specification would play a role here. Furthermore, for the reasons outlined in section 3.2.2, we expect the two pulse technique to accurately measure the relative center of mass velocity of the system. Thus a more faithful recreation of the experiment is unnecessary.

A damping rate can be extracted from the oscillation in a manner similar to the method outlined in section 3.2.2. By calculating the center of mass  $v_z$  in as a function of time, Eq. 3.25 can be fit to the oscillation. This process will yield a damping rate, as well as an oscillation frequency. It should be pointed out that it is very important that the range of the oscillation that is fit to Eq. 3.25 should directly correspond to the range in which experimental data was taken. The reasons why this is the case will be presented in the next section.

### 3.4 Initial Experimental Data Set and Analysis

The initial set of data that was collected consisted of oscillation data at three different temperatures. The post formation electron temperatures of these sets were estimated to be  $2.87 \pm 0.25\text{K}$ ,  $5.74 \pm 0.32\text{K}$ , and  $14.8 \pm 0.73\text{K}$ [10]. All three of the data sets contained  $2 \times 10^5$  ions with a Gaussian spatial extent of  $\sigma = 720\mu\text{m}$ . At the time in which the oscillation was induced, the experimental system contained roughly  $1.05 \times 10^5$  electrons, which yields a

charge imbalance of  $\delta = 0.475$ . All of the experimental data was collected in the presence of an external DC electric field, as well as a  $9G$  magnetic field. Additional experimental details are presented in Ref. [10] and in Wei-Ting Chen's Ph.D. thesis.

The initial intention in collecting this data was to use the damping rate to measure the electron-ion collision rate. The main assumption involved in doing so is that the damping rate would not be significantly influenced by factors other than the electron-ion collision rate. However, the numerical modeling described in this chapter showed that collisionless mechanisms influenced the damping rate as well. While it was known that collisionless mechanisms had the potential to impact to electron center-of-mass damping rate, initial estimates suggested that these mechanisms would be negligible for our given set of experimental conditions. Unfortunately, the hybrid model, described earlier in this chapter, illustrated that these estimates were inaccurate, and the inaccuracy of these estimates was not discovered until the initial dataset had been collected.

The main collisionless mechanism relevant to the experiment is a process that we call de-phasing. This section will describe how the process of de-phasing results in apparent center-of-mass damping. The experimental and modeling results are still valuable in both elucidating the physics of electron oscillation damping in UCPs as well as indicating the true requirements for parameters necessary to measure the electron-ion collision rate through the oscillation damping rate.

### **3.4.1 Center of Mass De-Phasing**

Collisionless damping resulting from the center of mass de-phasing of the electron cloud arises due to anharmonicities in the ion potential. For a harmonic potential, the oscillation frequency of an electron does not vary as a function of oscillation amplitude. This means that all of the electrons in the distribution oscillate at the same frequency (at least in a non-interacting approximation), and in the case of the experiment all of the electrons oscillate in phase with each other.

In the case of the of an anharmonic potential, this is no longer true. In this potential an electron's oscillation frequency varies as a function of it's oscillation amplitude. If a collective oscillation is induced in the presence of an anharmonic potential, electrons will oscillate at a variety of different frequencies. As the oscillation evolves, the component electrons will begin to move out of phase with each other. While this is not strictly damping, this de-phasing will be generally indistinguishable from damping in our experimental system. Thus, understanding this de-phasing is crucial to properly understanding the collective oscillation damping measurement.

Simulations confirmed the presence of collisionless damping for UCPs with parameters that mirror experimental conditions. This confirmation came in a number of different forms, but perhaps the clearest confirmation came from simulations that did not include the collision operator. These simulations showed a very clear apparent decay in the electron center of mass motion. Since collisions were not present in the simulation, the decay, by definition, had to be the result of collisionless processes. An example of the results of one of these simulations can be seen in figure 3.7.

Additionally, figure 3.7 also showed collapse and revival-type features in the decay of the collective oscillation. This behavior suggests the presence of multiple different oscillatory modes in the system, as well as energy transfer between these differing modes. Naively, the presence of discrete modes would not be expected, as the anharmonicity of the ion potential would be expected to produce a continuous range of frequencies. Additionally, de-phasing from a continuous range of frequencies would not lead to collapse and revival of the collective oscillation but just a collapse. This discrepancy suggests that the ion potential anharmonicity is not the whole story, and that electron-electron interactions play a significant role in collective oscillation dynamics. Rather than a continuous range of frequencies in an anharmonic potential, these interactions lead to motion that is reminiscent of eigenmodes. The fact that there are multiple discrete frequencies does lead to de-phasing, and thus apparent damping, of the center-of-mass motion in the nearly the same way that a continuous

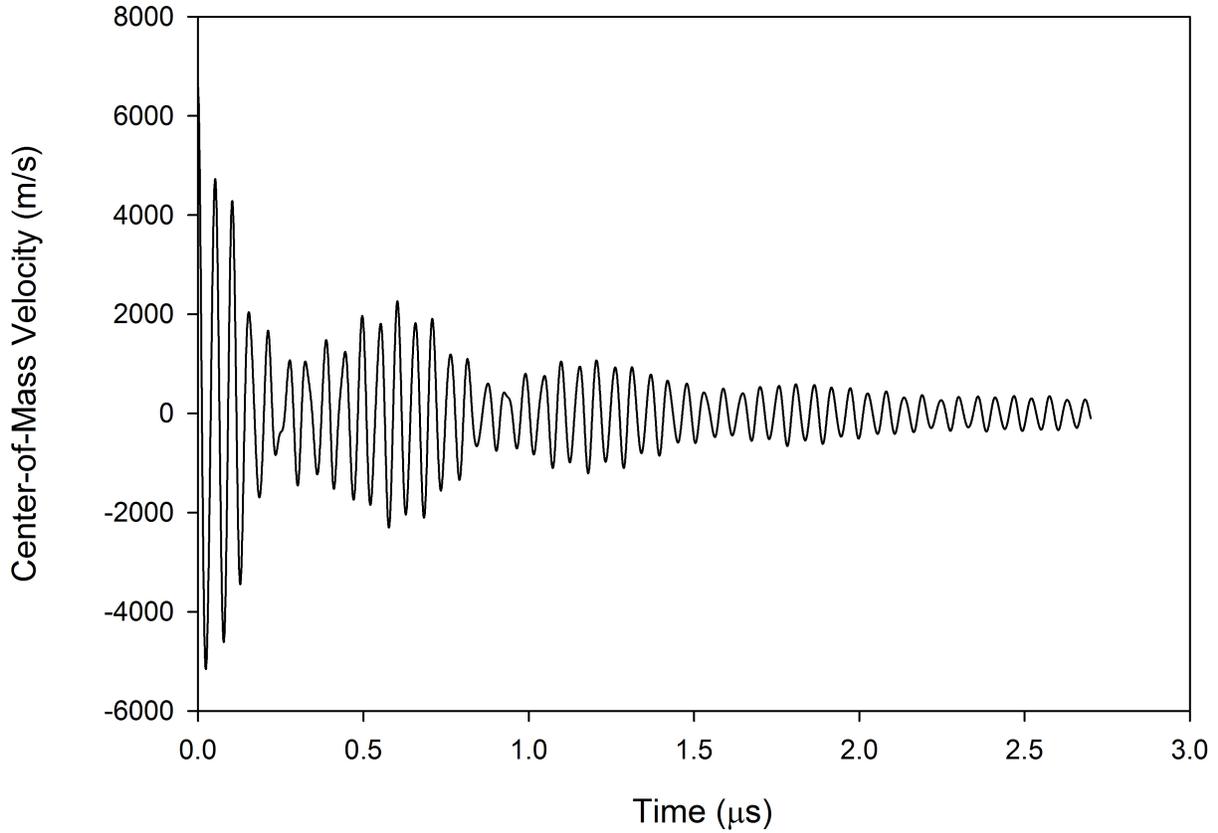


Figure 3.7: UCP electron center-of-mass velocity in the  $\hat{z}$  direction as a function of time, in the absence of electron-ion collisions. Plasma parameters for the simulation mirrored experimental conditions for  $T_e = 3K$  with  $\delta = 0.45$ . The ion numbers was 200,000, and the applied DC electric field was 7V/m. The figure illustrates the partial collapse and revival of the electron center-of-mass oscillation, and suggests coupling between different oscillatory modes.

range of frequencies in an anharmonic potential does. In practice, it is likely that both effects are in play and the observed de-phasing is a complicated combination of the two.

The presence of collisionless damping was also observed by changing simulation conditions. By altering certain parameters, not expected to impact the collision rate, changes were observed in overall damping rate of the oscillation. Three main variables were shown to alter the degree of collisionless damping. The first two, the charge imbalance,  $\delta$ , and the external electric field, will be discussed in this subsection. The third factor, the magnetic field, will be discussed in the next subsection.

The effective potential of the ion charge distribution can be described by

$$\phi_i(r) = \frac{qN_i \operatorname{erf}\left(\frac{\sqrt{2}}{2}r\right)}{2\epsilon_0 r} \quad (3.42)$$

where  $\phi_i$  is the electric potential resulting from the ion cloud. Around the origin,  $\phi_i$  roughly approximates a harmonic potential. However, as  $r$  moves away from the origin,  $\phi_i$  begins to diverge from a harmonic potential. An example of this divergence can be seen in figure 3.8. Section 3.3.3 illustrated that as the charge imbalance,  $\delta$ , in the UCP decreases, the spatial extent of the electron equilibrium distribution increases. Since the anharmonicity of  $\phi_i$  varies as a function of  $r$ , the charge imbalance of a UCP contributes to degree of collisionless damping in the system. This suggests that as the charge imbalance of a UCP decreases, the degree of collisionless damping increases. This conclusion was confirmed by simulations calculating the collective oscillation damping rate as the function charge imbalance. The results of these simulation can be seen in figure 3.9.

This same basic physics is what leads to electric field induced collisionless damping. As was explained in section 3.3.3, an external electric field leads to a displacement of the electron cloud. This displacement leads the electron component to be pushed more into an anharmonic region of the ion potential, which in turn leads to more collisionless damping. Additionally, since the magnitude of the displacement increases with the strength of the elec-

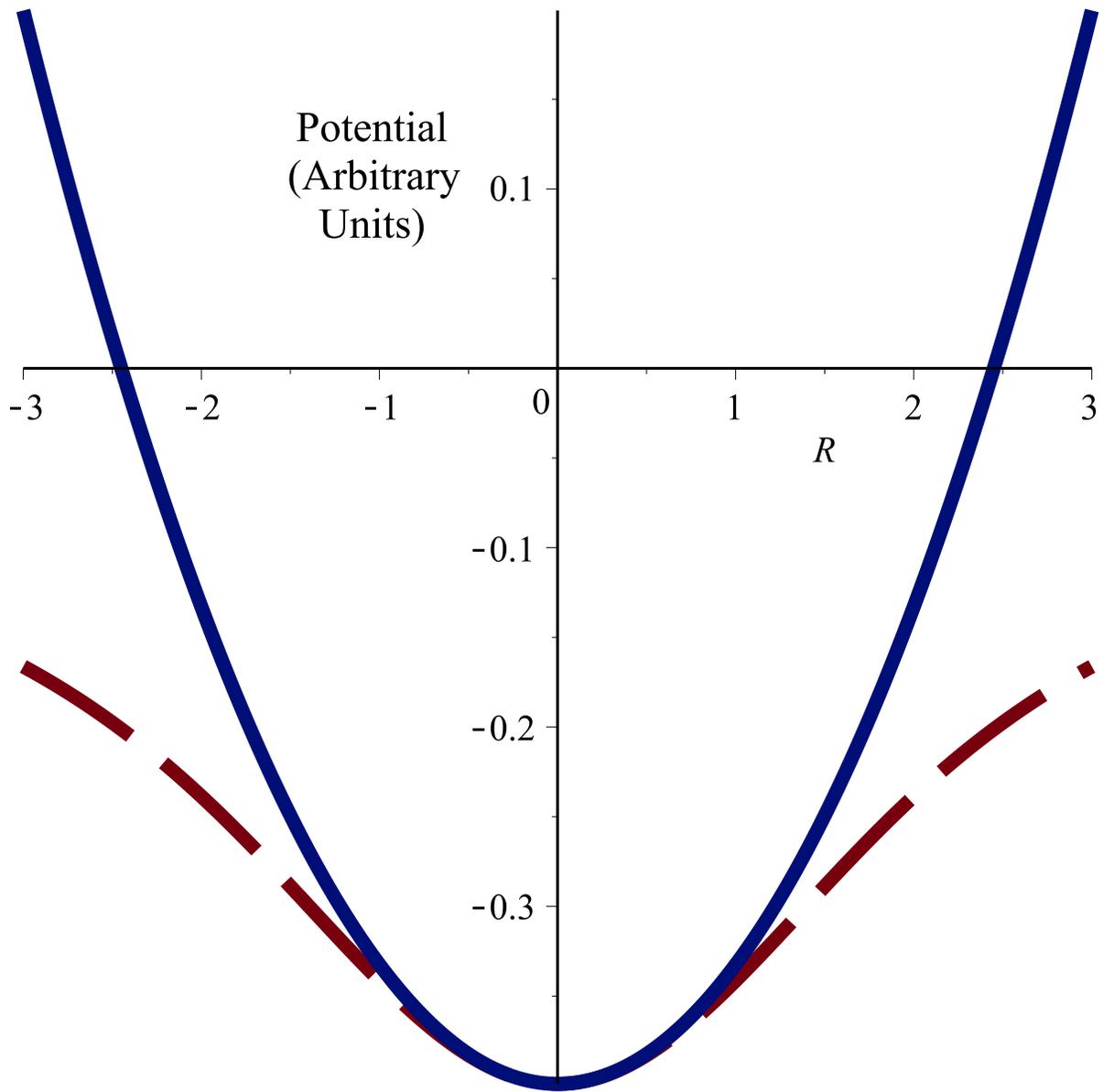


Figure 3.8: Graph of the electric potential due to the ions in the UCP as function of position. The red curve represents the electric potential due to a Gaussian distribution of ions. The blue curve is a simple harmonic potential, derived from a Taylor expansion of the Gaussian ion potential.  $R$  is the normalized radial coordinate defined as  $r/\sigma$  where  $\sigma$  is the standard Gaussian scale size. As can be seen from the figure, the Gaussian potential remains harmonic near the origin, but begins to deviate as it moves away from the origin.

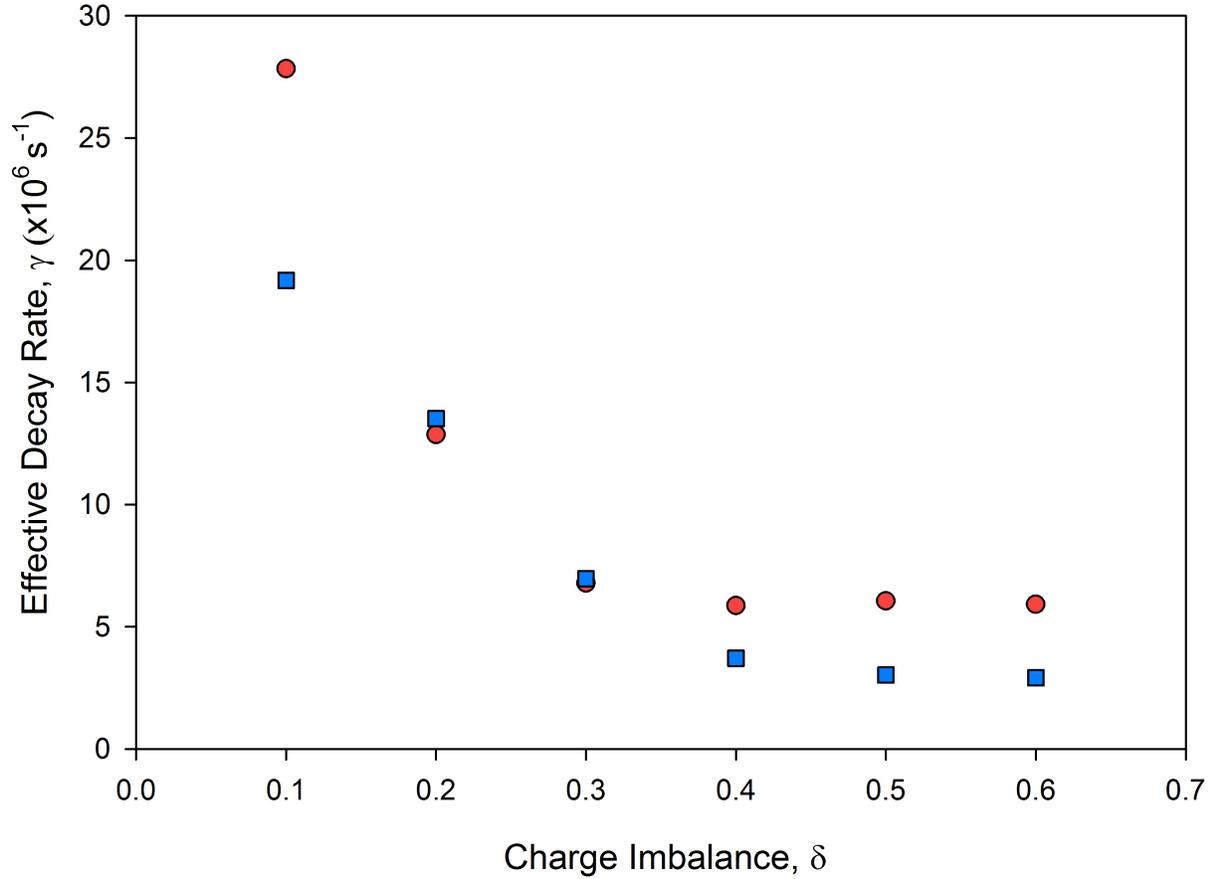


Figure 3.9: The impact of charge imbalance  $\delta$  on the effective damping rate,  $\gamma$ . Simulations were run with 20000 ions, an appropriate number of electrons as defined by  $\delta$ . No DC electric field was applied, and electron-ion collisions were included in the calculation. The electron temperature was 3K for red circles and 15K for blue squares. To extract  $\gamma$ , a damped sinusoid was fit to the first five oscillations of the center-of-mass motion. The figure shows the sharp increase in  $\gamma$  as the UCP becomes more neutral.

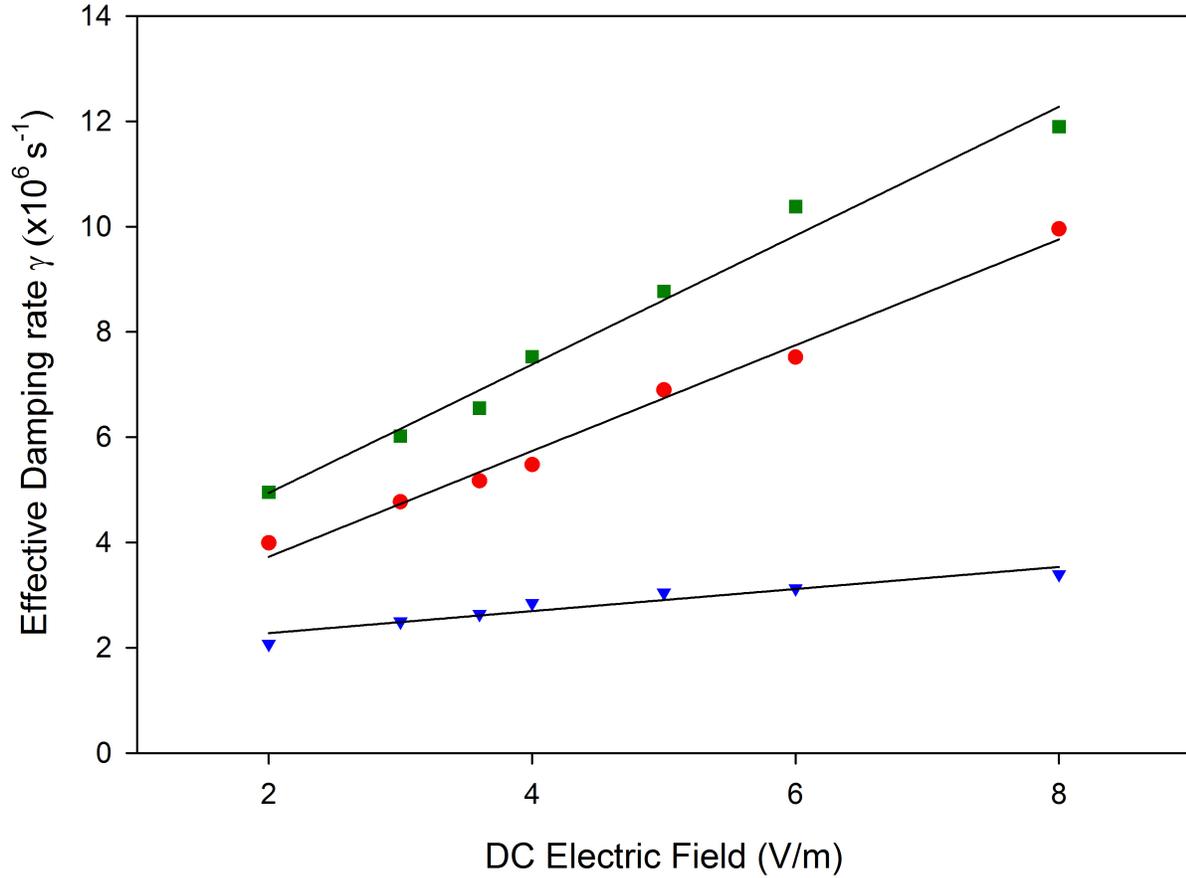


Figure 3.10: Damping rate vs. applied DC electric field for three different electron temperatures (green square, 3K, red circle, 6K, and blue triangle, 15K). In this calculation, electron-ion collisions were included. The charge imbalance  $\delta$  was 0.45, and the ion number was 200000. The damping rate increases with an increase of applied DC electric field, and decreases with increasing electron temperature.

tric field, the collisionless effects present in the UCP increase with increasing field strength. This behavior is confirmed by simulations run at a variety of different DC field strengths, and at three different electron temperatures. The results of these simulations can be seen in figure 3.10.

Clearly, collisionless damping is a significant contributor in this range of plasma parameters, as altering the plasma towards a more neutral charge imbalance should not increase the electron-ion collision rate substantially. The electric fields applied are on the order of typical

average internal electric fields in the plasma, but they too had a large effect on the apparent damping rate. Thus, measuring the electron-ion collision component of the damping rate seems daunting where these collisionless mechanisms are present. Care must be taken to avoid parameters where the collisionless damping rate is significant.

### 3.4.2 Magnetic Field Effects

While the experimental data sets were collected in the presence of a magnetic field, initial modeling did not include an external magnetic field. It was thought that since electron collective oscillations are induced in the direction of the magnetic field, the field would not have significant impact on this motion. Moreover, the degree of magnetization with regard to its impact on electron-ion collisions can be characterized by the ratio of a typical Larmor radius of an electron in the magnetic field to the  $b_{90}$  deflection parameter. For the coldest conditions,  $b_{90}$  was about a factor of six or so less than the Larmor radius of electrons with the most-probable velocity. For hotter temperatures, the ratio was even larger. Thus, substantial contributions were simply not expected, especially for the hotter electrons. However, initial modeling with magnetic fields showed the presence of additional collective modes, as can be seen in figure 3.7. Collisionless damping would not be expected to be characterized by the  $b_{90}$  parameter, but likely the much larger Debye screening length that is comparable to the Larmor radius for our conditions. Plus, these additional modes likely involved motion not exclusively limited to the  $\hat{z}$  direction, it seemed likely that an external magnetic field could impact the motion of these oscillations, as well as either impede or enhance energy transfer between the modes present in the system.

For this reason, a magnetic field was added to the model. When included, the magnetic field could sometimes lead to significant changes in the oscillation damping. Most common among these changes was that the magnetic field led to a preservation of late  $\hat{z}$  oscillations. An example of this can be seen in figure 3.11. The severity of the B-field induced collisionless effects strongly correlated with the anharmonicity of the ion potential experienced by the

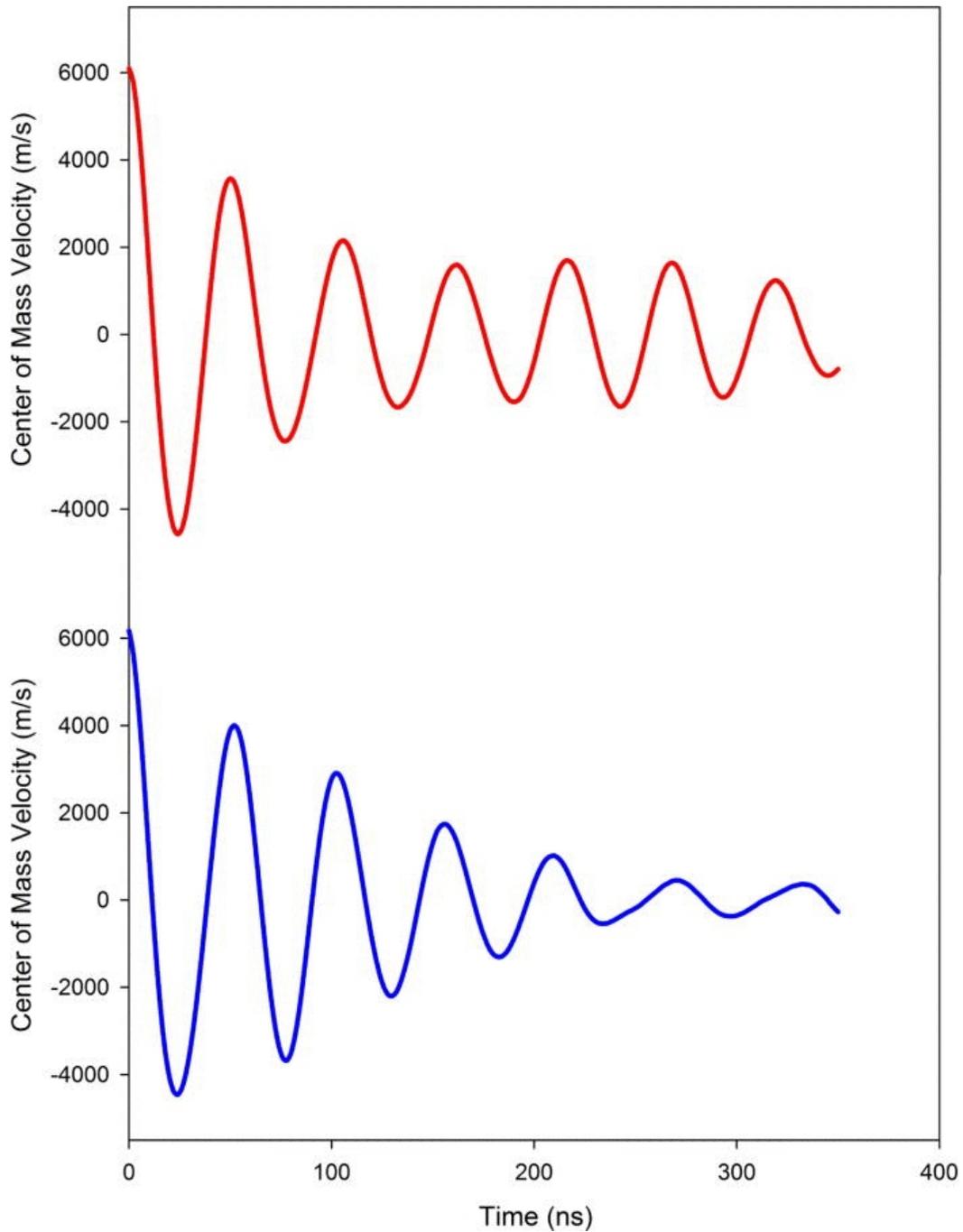


Figure 3.11: Graph showing the effect of an external magnetic field on the center-of-mass damping of the electron cloud. The top curve shows the calculated center-of-mass oscillation of the electron cloud for a simulation that included a 9G magnetic field. The bottom curve shows the calculated center-of-mass oscillation of the electron cloud in the absence of an external magnetic field. With the exception of the magnetic field, both simulations were run with identical parameters. The figure illustrates that the inclusion of an external magnetic field can have a significant effect on the damping of the electron collective motion.

UCP electrons. For plasma parameter ranges dominated by the collisionless effects caused by ion potential anharmonicity, the inclusion of a B-field significantly altered the resultant oscillation. Conversely, for collisionally dominated plasma parameters, the inclusion of a magnetic field had little to no impact on the result of the simulation.

A possible explanation for the influence of the magnetic field is as follows. As the electron cloud oscillates, the cloud is displaced from the center of the ion distribution. Due to this displacement, parts of the electron distribution are surrounded by a lower ion density than what they experience in equilibrium. This change in the surrounding charge density results in a net force that pushes the electrons apart. A substantial portion of this force is in the  $\hat{\rho}$  direction, and in the absence of a magnetic field, likely feeds other oscillatory modes like the ones present in figure 3.7. However, the presence of a  $\hat{z}$  magnetic field will impede the motion of electrons in the  $\hat{\rho}$  direction. This impediment will have the result of reducing the coupling between the different oscillatory modes. Since other modes will not be able to siphon energy away from the main  $\hat{z}$  oscillation, this oscillation will damp more slowly than it would typically in the absence of a magnetic field. In any case, the comparison of the typical Larmor radius to the Debye screening length suggests that the magnetic field very likely should have an impact on the collisionless damping rate, and simulations demonstrated that it did.

### 3.4.3 Verifying the Model

The previous subsections have illustrated the ways in which collisionless effective damping can arise in collective oscillations of UCP electrons. However, up until this point, the evidence for these collisionless effects has only been theoretical in nature. If the collisionless effects illustrated in the model accurately describe what is occurring in the experiment, the experimental data should agree with the model results. However, for a number of reasons, the two results are not easily comparable. This subsection will discuss the different steps

necessary in order to make a proper comparison between the model and experiment, as well as discuss the results of this comparison.

Before any discussion of how the model was compared to the experimental data, it is necessary to give an overview of how the measured data were structured. The data map the oscillations of the electron cloud at three different temperatures. At each temperature 4-5 different oscillation curves were collected. Each oscillation curve consists of roughly 20 different data points, which occur at evenly spaced points in time. Finally, each one of these data points is the average of 16 different experimental measurements.

One challenge to comparing the model to the experiment is that in the experiment the spatial size of the UCP varies from shot to shot. This size variations leads to density variation which in turn, as can be seen in the expression for the plasma frequency, leads to a variation in the collective oscillation frequency. This results in the distortion of oscillation damping in the following way. Each data point in an experimentally collected oscillation curve is an average of many measurements. Since each measurement is taken from an oscillation with a slightly different frequency, the measurements, when averaged, will not be added together constructively. This will lead to a systematic underestimation of the amplitude of the oscillation peaks. Furthermore, since phase differences resulting from frequency variation increase as a function of time, this effect will be larger for late time data points than for early time data points. The net result is that the experimental data will overestimate the amount of damping that exists in the system.

This effect can be expressed formally in the following way. In general, the functional form of an arbitrary oscillation can be represented as  $F(\omega, t)$ , and the curve,  $F'(t)$  is defined as  $1/N \sum F_i(\omega_i, t)$ . If all of the  $\omega_i$ s equal the center frequency  $\omega_0$ ,  $F(\omega_0, t) = F'(t)$ . However, if  $1/N \sum \omega_i = \omega_0$  but not all of the  $\omega_i$ s are the same,  $F(\omega_0, t)$  will not equal  $F'(t)t$ . This is because the different oscillatory frequencies will lead deconstructive interference. In general, the phase difference between two oscillations,  $F(\omega_1, t)$  and  $F(\omega_2, t)$ , that are in phase with each other at time  $t = 0$  is defined as  $(\omega_2 - \omega_1)t$ . A similar result can be derived if the values

of the  $\omega_i$ s are normally distributed around a central frequency,  $\omega_0$ , with a Gaussian  $\sigma$  of  $\sigma_\omega$ , the resulting phase variation,  $\sigma_\phi(t)$  of the  $F_i$  curves can be defined as

$$\sigma_\phi(t) = \sigma_\omega t \quad (3.43)$$

where the phase variation is a function of time. In the limit of an infinite number of  $F_i$ s,  $F'(t)$  can be written in terms of a convolution integral.

$$F'(t) = \int_{-\infty}^{\infty} \sqrt{\frac{1}{2\pi\sigma_\phi^2(t)}} F(\omega_0, \tau) \exp\left(-\frac{(t-\tau)^2}{2\sigma_\phi^2(t)}\right) d\tau \quad (3.44)$$

To properly compare the model to the experimental results, it is necessary to compensate for the effects caused by frequency variation. To do this, it is necessary to know the impact that frequency variation would have on model results. This was accomplished by utilizing Eq. 3.44 where  $F(\omega_0, \tau)$  is the resultant model curve for a specific electron temperature. To simplify the effects of frequency variation it is possible to parameterize these effects with a single exponential decay constant. For a given model result,  $F$ , and a given frequency variation,  $\sigma_\omega$ ,  $F'$  can be well approximated as

$$F'(t) = F(\omega_0, t) e^{-\Delta(\sigma_\omega)t} \quad (3.45)$$

where  $\Delta(\sigma_\omega)$  is the decay constant used to approximate the effects of frequency variation.

To quantify the effects of frequency variation,  $\Delta$  was explicitly calculated as a function  $\sigma_\omega$ . This was done in the following manner. For a given model result,  $F$ ,  $F'$  is calculated exactly for a given  $\sigma_\omega$  using the relationship in Eq. 3.43. Once this was done, a fitting routine was used in conjunction with Eq. 3.44 to extract a best fit  $\Delta$ . This process was repeated for a number of discrete values of  $\sigma_\omega$ , and a 2nd order interpolating polynomial was used to fully approximate  $\Delta(\sigma_\omega)$ . This procedure was performed for model results representing all

three temperature conditions, which would later be used in fitting the experimental data to the model.

The complexity of the magnetic field simulations introduced another complication to a direct comparison between the model and the experimental results. These simulations could no longer be well approximated by decaying sinusoids, preventing a direct comparison of model and experimental damping rates. Thus another method of comparison was needed. In theory, the experimental data could be fit directly to the model result. However, since the experimental parameters were not known exactly, such a comparison would not be expected to yield an agreement. For any comparison between the model and the experimental data to yield agreement, some variation in the model oscillation needs to be allowed. Incorporating model variation into any fitting routine would be completely untenable without first parameterizing the model oscillations with a simple analytical function.

Fortunately, magnetic field oscillations could be well approximated by a sinusoid with 2 distinct decay constants. This led to the model oscillation being parameterized by the following functional form

$$(Ae^{-k_1(\omega)t} + Be^{-k_2(\omega)t})\cos(\omega t + \phi) \quad (3.46)$$

where  $A$ ,  $B$ ,  $k_1$ ,  $k_2$ ,  $\omega$  and  $\phi$  were treated as fit parameters. This functional form allows for quick initial decay resulting from the first time constant, and a slower decay rate at late times resulting from the second decay constant. Model results for all three temperature conditions are fit to Eq. 3.46. The best fit values of  $A$  and  $B$  for a given temperature condition did not vary as a function of the UCP density. However, the best fit values of for  $k_1$  and  $k_2$  did. To handle this, model simulations were run over a range of densities corresponding to the density uncertainty present in the experiment. Linear fits were then conducted for the values of the decay constant as a function of frequency. This allowed for  $k_1(\omega)$  and  $k_2(\omega)$  to be fully defined as a function of  $\omega$  for all three temperature conditions.

Once the values of  $A$ ,  $B$ ,  $k_1(\omega)$ ,  $k_2(\omega)$  had been determined. The experimental data curves were fit to the following functional form

$$D(Ae^{-k_1(\omega)t} + Be^{-k_2(\omega)t})e^{-k_{aux}t}\cos(\omega t + \phi) \quad (3.47)$$

Where  $D$ ,  $\phi$ ,  $\omega$ , and  $k_{aux}$  are treated as fit parameters, and all of the other parameters were determined from the model by the process described in the preceding paragraph. The main difference between Eq. 3.46 and Eq. 3.47 is the inclusion of an additional decay constant,  $k_{aux}$ . This additional decay constant is included to account for any discrepancies between the model and the data. If  $k_{aux}$  is positive, the data damp faster than the model. If  $k_{aux}$  is negative, the data damp more slowly than the model.

To compare the model to the experiment, the concept of an effective decay rate,  $k_{eff}$ , was introduced. This allowed both model and experimental oscillations to be characterized by a single decay rate as was done for the non-magnetic field simulations. For the experimental curves,  $k_{eff}$  was defined by the following

$$(A + B)e^{-k_{eff}t_0} = (Ae^{-k_1(\omega)t_0} + Be^{-k_2(\omega)t_0})e^{-k_{aux}t_0} \quad (3.48)$$

where  $t_0$  is set to a convenient but arbitrary time of  $150ns$ . For each data curve, the best fit value of  $k_{eff}$  was found. The  $k_{eff}$ s were then averaged for each temperature condition. The appropriate error bars were then determined by adding the statistical errors and the systematic errors in quadrature.

A similar procedure was used to determine a  $k_{eff}$  for the model points. Since model curves and the experimental data are related by Eq. 3.45,  $k_{aux}$  should be in statistical agreement with  $\Delta(\sigma_\omega)$  if the model and the experiment agree. For this reason, model values of  $k_{eff}$  were calculated using the following functional form

$$(A + B)e^{-k_{eff}t_0} = (Ae^{-k_1(\omega)t_0} + Be^{-k_2(\omega)t_0})e^{-\Delta_i(\sigma_\omega)t_0} \quad (3.49)$$

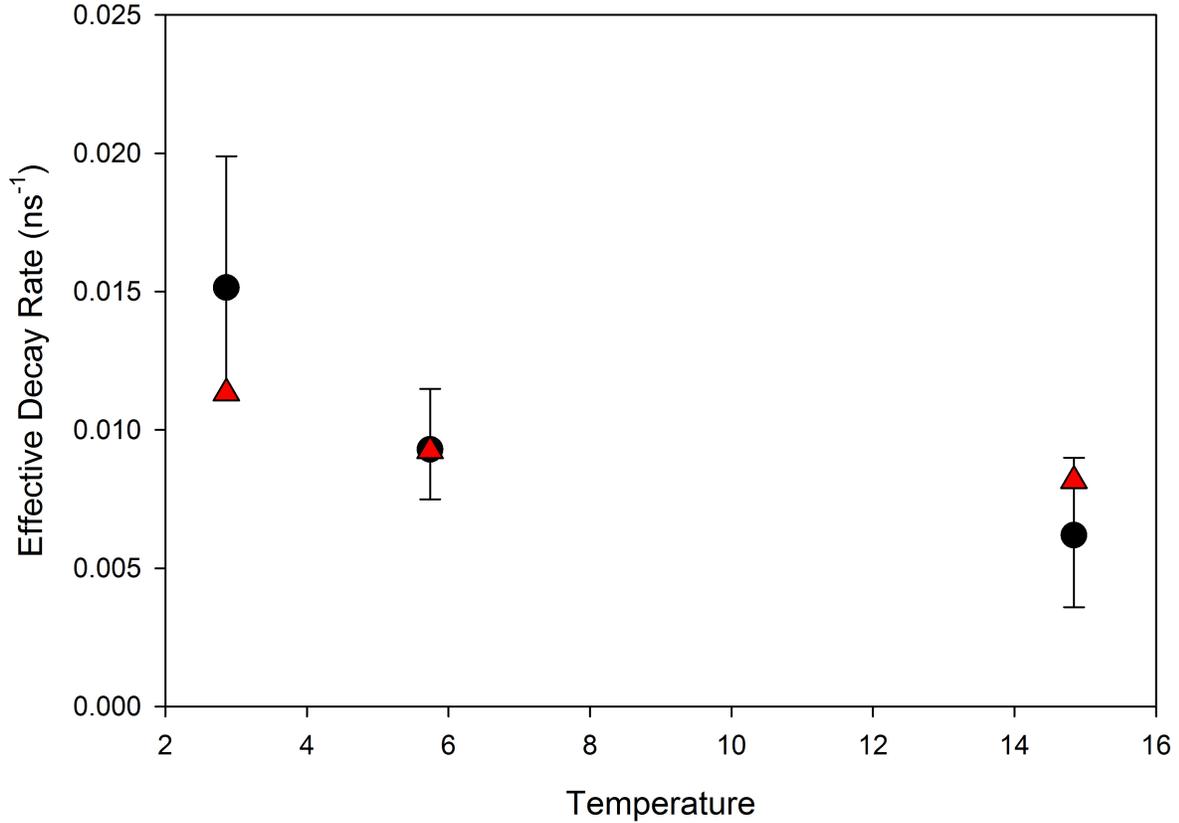


Figure 3.12: Effective decay constant,  $k_{eff}$ , as a function of temperature, as defined in Eq. 3.48 and Eq. 3.49. The black dots, and the accompanying errors bars, represent the experimentally measured values of  $k_{eff}$ . The red triangles represent the values of  $k_{eff}$  as determined by the model. As can be seen in the figure, the model and experimental values of  $k_{eff}$  are in statistical agreement.

where  $\Delta_i(\sigma_\omega)$  is the polynomial describing the effective damping caused by frequency variation for the  $i$ th temperature condition. Model values for  $k_{eff}$  were then fit to the previously determined experimental values for  $k_{eff}$ , by using  $\sigma_\omega$  is the only fitting parameter. The results of this fit can be seen in figure 3.12 As can be seen in the figure, the model predictions for  $k_{eff}$  agree with the experimental results. Furthermore, the best fit value for  $\sigma_\omega$  was found to be 7.3%, which was in strong agreement with the variation that was observed in the experiment. The combination of these two results strongly suggests that collisionless

damping mechanisms play a prominent role in our experimental system, and that the model that I have developed effectively captures that role.

### 3.4.4 Finding a Collisional Parameter Range

The initial set of experimental data provided a variety of different insights into how collisionless mechanism can impact collective electron oscillations in an UCP. However, the data was not able to answer any of the questions that we had initially set out to answer. These questions included, what is the electron-ion collision rate? how cold are the electrons? and what are the effects of strong-coupling in the system coupling present in the system? The answers to all of these questions were obscured by the presence of collisionless mechanisms. Thus, in order to answer these questions, the experiment has to be conducted in a parameter range where collisionless mechanisms are negligible.

It is not necessarily obvious where this parameter range is located, and thus a search needed to be conducted. Due to the relative speed of simulations as compared to the rate that experimental data could be collected and analyzed, this search was conducted numerically. By running simulations with electron-ion collisions on and again with electron-ion collisions off, the relative collisionality of a given set of parameters could be determined.

In general, there were two main things that increased the collisionality of a set of parameters. These were a reduction in the external electric field, and an increase in the charge imbalance,  $\delta$ . Unfortunately, these two levers are typically at odds with each other, since an increase in the charge imbalance is typically brought about experimentally by an increase in the external electric field. However, charge imbalance can also be increased by reducing the total number of ions in the UCP. Since a reduction in ion number leads to a small confining force, a smaller external electric field can be utilized to obtain the same charge imbalance. In addition, it is possible to change the electric field in time. The UCP can be formed at a higher electric field, and then that electric field can be decreased before the electron oscillation measurements are conducted. These were the methods that were ultimately employed

to find a collisionally dominated UCP parameter range. The precise details of this parameter range are outlined in the next section.

## 3.5 Second Data Set

In order to mitigate the collisionless effects present in the first data set, the second data set was moved to a different parameter range. This range was characterized by a lower ion number, smaller spatial size, and a smaller external electric field at the time the electron oscillation damping was measured. This section will discuss this new parameter range, the numerical modeling used to analyze a second data set taken under new conditions, and the results of this analysis.

### 3.5.1 Overview of the Measured Results

The second data set consisted of two distinct temperature conditions. The first condition had an initial electron kinetic energy of  $\Delta E = 2.26K \cdot k_b$ . Data points for this temperature condition corresponded to a UCP with 62,000 ions, a charge imbalance of  $\delta = 0.434$ , and a Gaussian size of  $\sigma = 625\mu m$ . The second condition had an initial electron kinetic energy of  $\Delta E = 0.1K \cdot k_b$ . Data points for this temperature condition corresponded to a UCP with 72,000 ions, a charge imbalance of  $\delta = 0.462$ , and a Gaussian size of  $\sigma = 630\mu m$ . For the remainder of this chapter, the  $\Delta E = 2.26K \cdot k_b$  data point will be referred to as the hot point, and the  $\Delta E = 0.1K \cdot k_b$  data point will be referred to as the cold point. All data points were collected in the presence of an external DC electric field, as well as 9G magnetic field.

By switching to this new parameter range, the damping of the electron center of mass motion can now be accurately described by a single damping rate as described by the Eq 3.24. The damping rate for the hot point was measured to be  $3.72 \pm 0.79\mu s^{-1}$ , and the damping rate for the cold point was measured to be  $8.53 \pm 1.54\mu s^{-1}$ . These rates correspond to specific electron temperatures. Since post formation heating should be substantial for these

temperature conditions, these rates will correspond to temperatures that are significantly higher than what would be expected from what their respective initial ionization energy would suggest. The method utilized for determining the proper corresponding temperatures will be the subject of the rest of this section.

### 3.5.2 Failure of the Collision Operator

The original method to determine the appropriate electron temperatures from the measured damping rates was to use model simulations that were analogous to those utilized for the first data set. However, the hybrid model, using a collision operator which assumed the weak coupling limit, was unable to reproduce a damping rate that was, at any temperature condition, large enough to match the cold experimental results. A plot of the cold data's measured damping rate compared to model calculated damping rates, in the weakly coupled limit, as a function of temperature can be seen in figure 3.13. As can be seen in figure 3.13, the model predicts a electron oscillation damping rate that peaks around  $T_e = 2K$  at a value that less than half of the measured value of the cold point.

This result was not particularly surprising, since, at low electron temperatures, strong coupling effects are expected to increase the electron-ion collision rate. The result suggested that the electron temperature at the cold point was indeed cold enough that strong coupling is present in the experimental system, and that strong coupling needed to be included in the collision operator in the model. However, incorporating strong coupling into collision operator via Eq. 3.33, only resulted in a small increase in the damping, around 30%, which was not a large enough effect to bring the model into agreement with the measured result. Dynamic screening[16] was another potential effect that could lead to an increase in the damping rate, as the presence of an electron oscillation is predicted to alter the value of  $b_{max}$ . However, this too was a small effect,  $< 10\%$ , which was not nearly large enough to bring the model into agreement with the results.

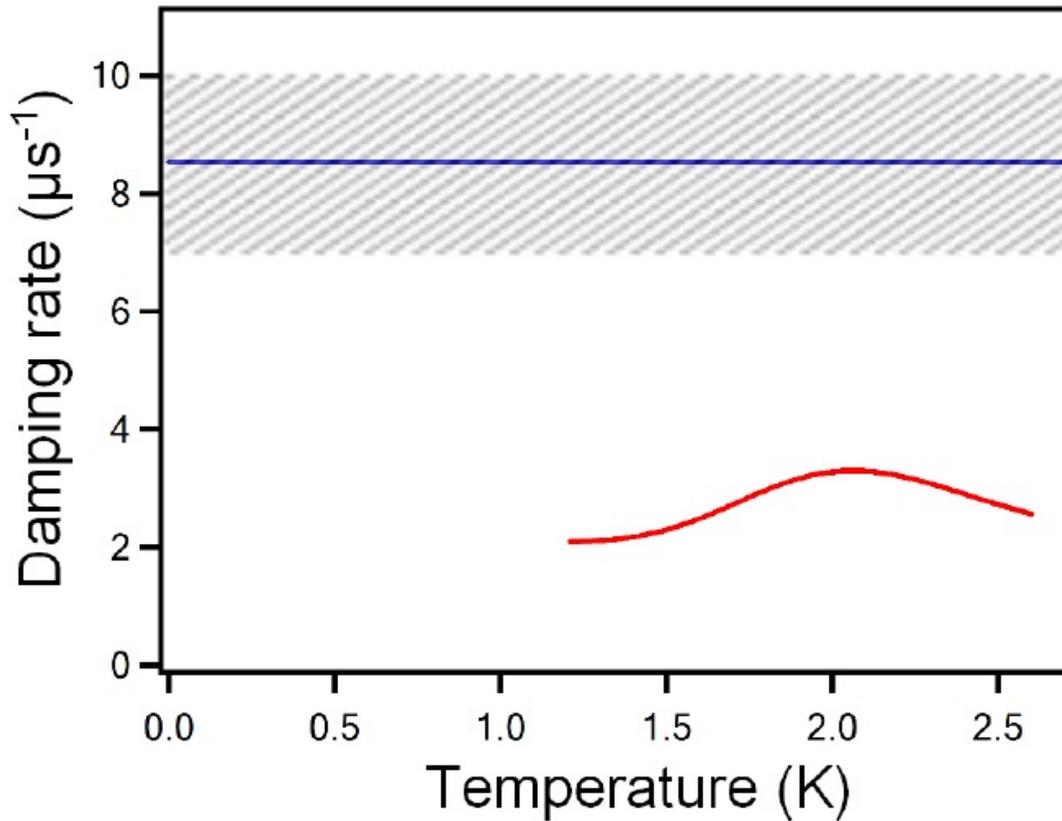


Figure 3.13: Graph comparing the damping rate of the electron cloud calculated by the hybrid model, to the experimentally measured damping rate of the cold point. The red curve represents the hybrid model calculations of the damping rate, using a weak coupling collision operator, as a function of temperature. The blue line represents the experimentally measured damping rate of the cold point, and the gray shaded region represents the  $1 \sigma$  confidence interval of this measurement. The figure shows that the hybrid model is not in agreement with the experimentally measured damping rate for any temperature condition.

Thus, agreement between the hybrid model and experimental results was unable to be achieved. This suggested that one of two possible things must be occurring. Either there was a systematic error in the experiment, or the collision operator did not properly predict collisions in the strongly coupled regime. To test the second possibility, it was decided that a full molecular dynamics simulation, one in which the both the electrons and ions were modeled as discrete particles, needed to be run. By using this method, a collision rate that was in agreement with the experimental results was able to be obtained. Thus it was possible to use full MD modeling to determine the electron temperatures a both the hot and cold data point. The method by which this was accomplished will be described in the next subsection.

### 3.5.3 Determining the Electron Temperature

By utilizing full molecular dynamics modeling, insights into the experimental electron temperature can be gained. Simulations were used to calculate the electron oscillation decay rate as a function of electron temperature,  $\kappa_i(T)$ , where  $i$  denotes the  $i$ th set of experimental conditions. cursory glances at the functional forms of  $\kappa_i(T)$ , yield a number of key insights about the experimentally determined decay rates. First, with the full MD simulation it was clear that a model electron temperature exists that will yield a decay rate that matches the experimentally determined decay rate. Second, the experimentally determined decay rates correspond to electron temperatures that are higher than what is implied by initial ionization energies. This means, if the model is correct, that there were significant post ionization electron heating mechanisms. Finally, the amount of heating implied by the experimental results was in agreement with what would be expected. For these reasons, initial analysis suggested that the full molecular dynamics model seemed to properly describe the experimental system. Since this was the case, the functional forms,  $\kappa_i(T)$ , can be used to determine the post formation heating, as well as the final electron temperatures of the experimental system.

The biggest challenge in determining the post formation heating of the electrons is that the amount of electron heating should vary as a function of the initial ionization energy, where lower ionization energies are expected to lead to more post formation heating. For this reason, the cold point was expected to undergo more post formation heating than the hot point. In theory, the amount of heating present for both of these data points could be estimated by summing up different estimates for the various heating cooling mechanisms present in the system. Since this would involve combining several estimates, the precision of the resulting determination would suffer. Thus, such a method seemed unlikely to produce an estimate of the heating that would lie within the experimentally measured precision.

For this reason, we opted to use the formation model that I had recently developed to quantify the amount of heating that existed in the experiment. The details and the preliminary results of this model are complicated enough that they deserve their own chapter. These will be discussed in the next chapter. However, the model can be summarized as follows. The model uses full molecular dynamics techniques to simulate both the ions and electrons in the plasma. Particles in the model are initialized where electrons are bound to a corresponding ion to classically simulate a Rb atom. The model simulates the photo-ionization process by systematically applying a velocity impulse to each of the electrons in the direction away from the corresponding ion. The magnitude of this velocity impulse is set to correspond to the energy of the ionizing laser pulse. After all of the electrons were ionized, the plasma is allowed to evolve for 3  $\mu s$  to correspond to the typical length of UCP formation. By calculating the average electron kinetic energy as a function of time, the model is able to calculate the amount of heating present in the present.

Preliminary results from the formation model suggested that there was a discrepancy between the full MD model and the experimental results, as the formation model was unable to produce sufficiently high amount of heating for the experimental results come into agreement with the calculated values for  $\kappa_i(T)$ . However, these preliminary formation results did not include an external electric field, which was present in the experimental measurements.

When an external electric was incorporated into the formation model it was found that the presence of an external electric field was a significant heating mechanism. In fact, the formation model calculated that the external electric field was largest contributor to electron heating for the conditions present in our experimental system by a large margin. A more thorough discussion of the electric field heating and the calculations used to quantify it can be seen in chapter 4.

With the inclusion of an external electric field, the formation model predicted sufficiently large amounts of heating to bring the experimental results into agreement with the full MD calculations. Just like how the full MD simulations were used to create an analytical functional form for the calculated damping rate,  $\kappa_i(T)$ , formation simulations could be used to calculate an analytical functional form for the final electron temperatures as a function of the external electric field. By running formation simulations for both the hot and cold conditions with the calibrated external electric fields, the final electron temperatures for both conditions,  $T_i(E)$ , could be evaluated. This allowed for the calculated damping rate from the full MD simulations to be expressed as,  $\kappa_i(T_i(E))$ . From this formulation, the statistical agreement between the model and the experiment could be determined from the following  $\chi^2$  expression

$$\chi^2 = \sum_i \left( \frac{\kappa_i(T_i(E)) - \kappa_{0i}}{\sigma_i} \right)^2 \quad (3.50)$$

where  $\kappa_{0i}$  and  $\sigma_i$  are the measured decay rate and uncertainty for the  $i$ th data point respectively. Evaluating Eq. 3.50 yielded a  $\chi^2$  value of 1.18, which strongly suggested that these points were in statistical agreement with the experimentally measured values. A graph comparing the best fit model values to the experimentally measured values can be seen in figure 3.14.

Once the full molecular dynamics model had been verified, it could then be used to determine the temperature of the electrons in the experiment. This was accomplished by simply determining the temperature that corresponded to the measured rate via the previ-

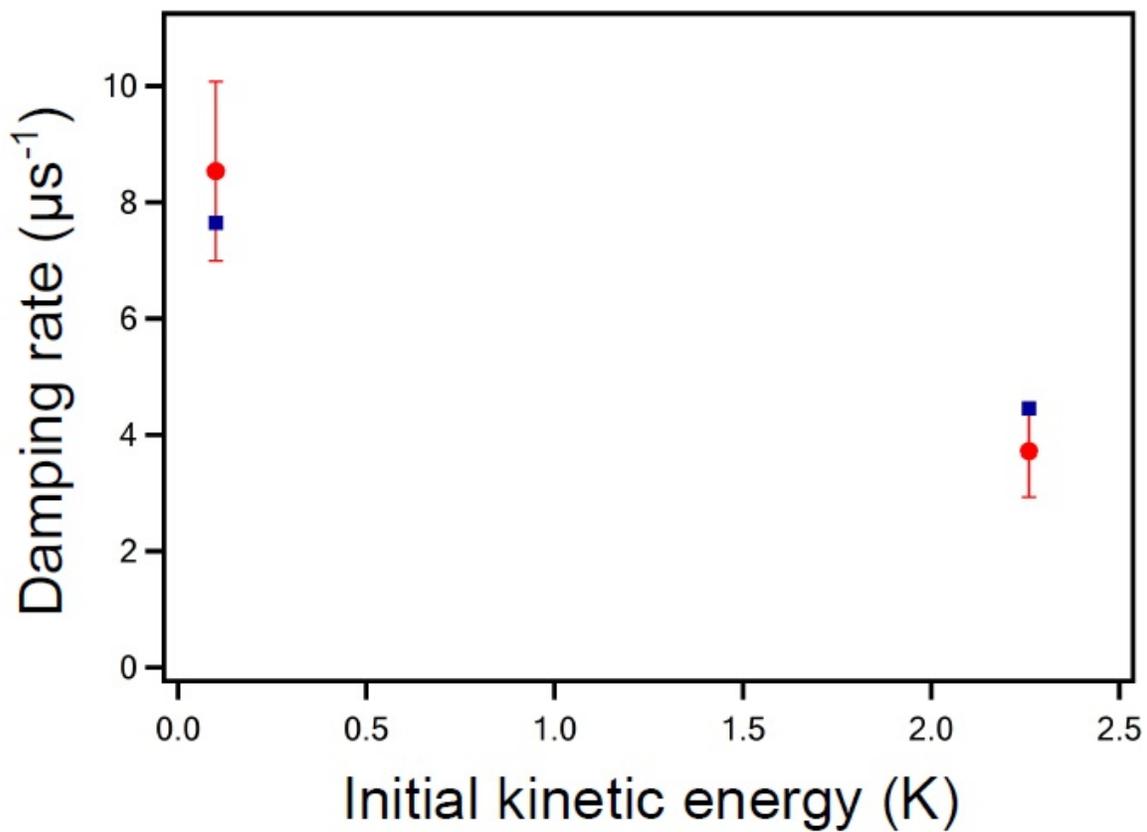


Figure 3.14: Decay rate of collective electron oscillations as a function of initial kinetic energy. The red dots represent the experimentally determined decay rates. The blue squares represent the calculated decay rates from full molecular dynamics simulations. The graph shows that the model calculations agree with the experimentally measured decay rates.

ously calculated analytic expression for  $\kappa_i(T)$ . This resulted in a cold point temperature of  $1.57 \pm 0.28\text{K}$ , which in turn corresponded to a average density  $\Gamma$  of  $0.35 \pm 0.8$ . This result corresponded to a record low electron temperature as well as a record high value of the electron  $\Gamma$  in UCP systems. Furthermore, the measured value of the electron  $\Gamma$  exceeded a predicted steady-state theoretical limit of  $\Gamma < 0.2$ [17]. This doesn't indicate that Ref. [17] is incorrect, but rather that we can make measurements early enough before  $\Gamma$  becomes limited to less than 0.2 due to three-body recombination.

### 3.5.4 Testing Collision Assumptions

While the experimental data agreed with the molecular dynamics simulations, there is still disagreement between the data and the hybrid model simulations. Since the data are in agreement with a the fundamental model, it seems highly probable that the disagreement is the fault of the hybrid model and not of the experiment. Also, by its nature the full molecular dynamics simulation is more complete than the hybrid calculation. In order to determine where the hybrid model falters, it is important to investigate the validity of the underlying assumptions of the model. Because there is disagreement between the model and the data, it seems reasonable that there must be at least one assumption that is not justified. Determining which assumption(s) are invalid will help improve the model in the future as well as give insight to the underlying physics of collisions in strongly coupled plasmas.

There are four main standard assumptions for Coulomb collisions that are applied in the hybrid model and in similar theories throughout plasma physics. These are:

1. Collisions can be modeled via a Rutherford scattering process with a cutoff parameter  $b_{max}$  based on  $\lambda_D$ . The details of this approximation are given in section 3.1 and section 3.3.2.
2. All collisions in the system can be model by successive binary collisions. This is known as the binary collision approximation.

3. It is reasonable to make a substitution of a thermal velocity in the Coulomb logarithm for an individual electron velocity. This approximation allows the Coulomb logarithm to be treated as a constant, and lets it be moved outside the integral in Eq. 3.15.
4. It is assumed that ions are spatially uncorrelated.

Of these four assumptions, assumption 3 seems to be the most suspect for our experimental conditions.

The justification for assumption 3 is that the Coulomb logarithm for most plasma systems is fairly large. This means that variation within the argument of the Coulomb logarithm leads to little variation in the overall value of the Coulomb logarithm, making the logarithm roughly constant over variations in electron velocity. Such an assumption seems likely to be a poor assumption for our experimental conditions. This proved to be the case as relaxing this assumption in the model, as well as assuming that a cutoff that incorporates strong coupling (Eq. 3.23), led to closing the gap between the hybrid model and experimental data by over 50%. The results of this change can be seen in figure 3.15. For further details on how the relaxation of assumption 3 was implemented see Wei-Ting Chen's thesis.

While relaxing assumption 3 led to a significant closing of the gap between the data and the hybrid model, there is still a gap of about 25% between the two. This means that at least one of the other assumptions is still problematic. Of these assumptions, assumption 4 is by far the easiest to test. By increasing the ion masses in the simulation by an order of magnitude the ions preserve their initial random positions for a longer time. If ion-ion correlations were important, preserving initial randomness should produce an observable change in the damping time. However, simulations with differing ion masses showed results which were indistinguishable. This suggests that assumption 4 is indeed valid for our experimental conditions.

This leaves assumptions one and two as the remaining possibilities. However, since these assumptions are significantly more challenging to verify, they remain outside of the scope of

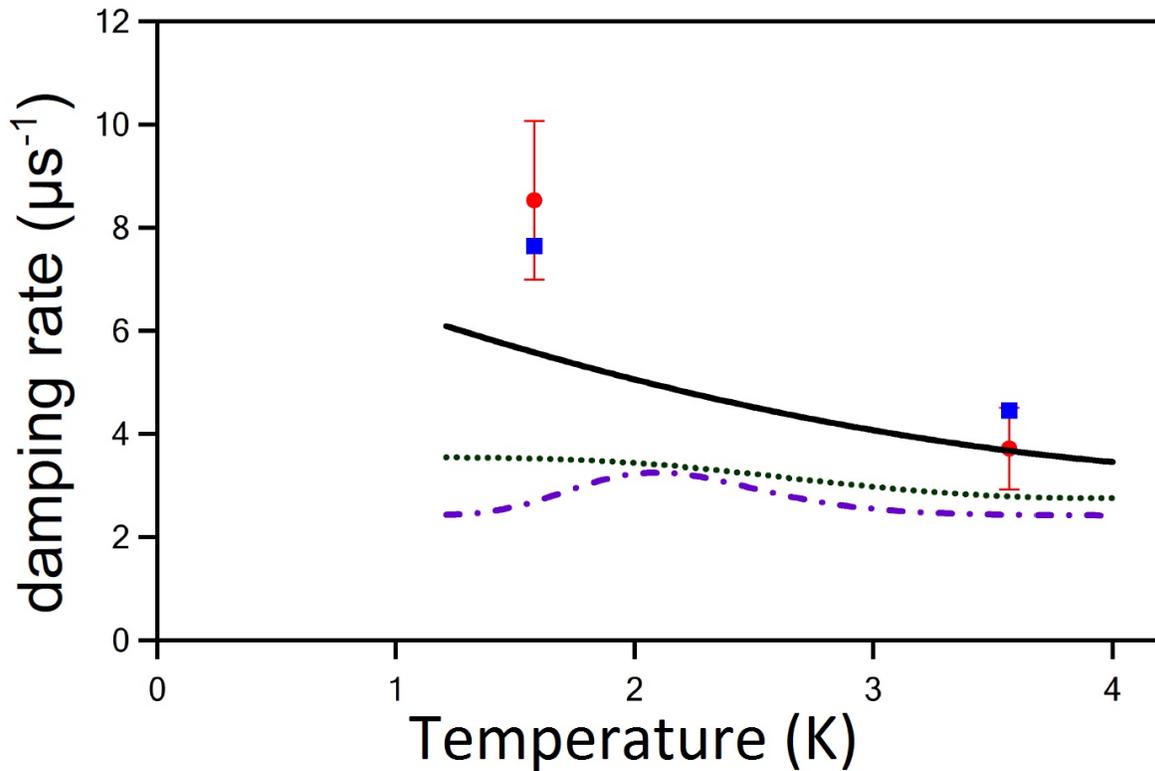


Figure 3.15: Comparisons between measured damping rates and theoretical calculations. The red circles are the measured damping rate. The blue squares are molecular dynamics results from section 3.5.3. All of the lines represent hybrid model calculation for different formulation so Coulomb logarithm. The dash-dotted purple line is weak coupling result. The green dotted line is uses strong coupling extensions outlined in Eq 3.23. The solid black line implements the strong coupling extension from Eq. 3.23 as well as relaxes assumption 3 from section 3.5.4. As can be seen from the graph, relaxing assumption 3 significantly closes the gap between the hybrid model and the experimental results.

the work described in this chapter. For more information on how these assumptions will be tested in the future, see chapter 8.

## 3.6 Conclusion

In this chapter I gave an overview of how the numerical modeling was used to simulate electron center-of-mass damping, and how it was utilized to interpret data from two different experimental datasets. The main numerical model that I developed, the hybrid model, was able to model each electron present in the electron center-of-mass oscillation process over timescales on the order of an hour. Additionally, by explicitly modeling the electron-ion collisions in the system with a monte carlo collision operator, the model could also be used to test the explicit form of the electron ion collision rate,  $\bar{\nu}_{ei}$ .

The model proved to be an invaluable tool for interpreting the data from the electron center-of-mass oscillation experiment. Using numerical modeling, the presence of a collisionless mechanism in the first dataset, which we call de-phasing, was established. De-phasing arises from the anharmonicity in the confining ion potential, and led to collisionless electron center-of-mass damping. Since the presence of de-phasing in the first dataset made it difficult to extract a measurement of the electron-ion collision rate, modeling was again used to find a new parameter range where the effects of de-phasing would be mitigated. The second dataset was taken in this regime.

For the second dataset, numerical modeling was integral in a number of different findings. First, modeling established that the colder data condition was characterized by an electron temperature of  $1.57 \pm 0.28K$  and electron  $\Gamma$  of  $0.35 \pm 0.08$ . These conditions correspond to a record low electron temperature and record high electron  $\Gamma$  in UCP systems. Furthermore, the measured electron  $\Gamma$  exceeds a previously predicted limit of  $\Gamma = 0.2$ . Second, modeling was used to confirm the existence of a previously unreported heating mechanism: external electric field heating. Without including this heating mechanism in the model, the model would not have agreed with experimental measurements. This heating mechanism will be

discussed further in the next chapter. Finally, modeling was used to confirm the breakdown of the weak coupling approximation for electron-ion collisions. Furthermore, modeling also showed that strong coupling extensions to the weak coupling collisional theory, when applied using the typical assumptions, failed to predict the correct damping rate as well. This suggested that one of the typical assumptions must not apply to our experimental conditions.

The chapter reports some preliminary work on testing these assumptions. One of these assumptions, a substitution of a thermal velocity in the Coulomb logarithm for an individual electron velocity, proved to be highly problematic. By relaxing this assumption the gap between the predicted damping rates was closed significantly. However, a gap still remains, and either the assumption of Rutherford scattering or the binary collision approximation are the culprit. Testing these two assumptions is the subject of future work.

# References

- [1] P. Bellan, *Fundamentals of Plasma Physics*, Cambridge University Press, 2006.
- [2] T. Wilson, W.-T. Chen, and J. L. Roberts, Phys. Rev A **87**, 013410 (2013).
- [3] J. Daligault and D. Mozyrsky, HIGH ENERGY DENSITY PHYSICS **4**, 58 (2008).
- [4] P. E. Grabowski, M. P. Surh, D. F. Richards, F. R. Graziani, and M. S. Murillo, Phys. Rev. Lett. **111**, 215002 (2013).
- [5] S. D. Baalrud, Phys. Plasmas **19**, 030701 (2012).
- [6] L. G. Stanton and M. S. Murillo, Phys. Rev. E **93**, 043203 (2016).
- [7] I. Hutchinson, Collisions in plasmas, lecture notes.
- [8] P. Mulser, F. Cornolti, E. Besuelle, and R. Schneider, Phys. Rev. E **63**, 016406 (2000).
- [9] T. Wilson, *Dynamics of Low-Density Ultracold Plasmas in Externally Applied Electric and Magnetic Fields*, PhD thesis, Colorado State University, 2013.
- [10] W.-T. Chen, C. Witte, and J. L. Roberts, Physics of Plasmas **23**, 052101 (2016).
- [11] W.-T. Chen, C. Witte, and J. L. Roberts, Submitted to Phys. Rev. Lett. (2017).
- [12] E. L. Raab, M. Prentiss, A. Cable, S. Chu, and D. E. Pritchard, Phys. Rev. Lett. **59**, 2631 (1987).
- [13] W. Demtröder, *Laser Spectroscopy: Basic Concepts and Instrumentation*, Springer, 2nd edition, 1996.
- [14] J. Guthrie and J. Roberts, Journal of Physics B: Atomic, Molecular and Optical Physics **49**, 045701 (2016).
- [15] T. Wilson, W.-T. Chen, and J. Roberts, Phys. Plasmas. **20**, 073503 (2013).

[16] P. Mulser, F. Cornolti, E. Besuelle, and R. Schneider, Phys Rev E **63**, 016406 (2000).

[17] F. Robicheaux and J. D. Hanson, Phys. Rev. Lett. **88**, 055002 (2002).

## Chapter 4

# Modeling Ultracold Plasma Formation

The formation phase is the first portion of the lifecycle of ultracold plasma systems. During this phase, electrons in the plasma escape, change velocities through collisions, and/or adjust their spatial distributions until a state of quasi-equilibrium is reached. Experiments that take place in our lab (such as the electron center-of-mass damping measurements detailed in the last chapter) most often collect data directly after this formation phase. This is because at that time the state of the plasma is best known, since the plasma is a state of equilibrium and very little ion expansion has occurred.

One post-formation parameter of particular interest is the UCP electron temperature. The main determinant of the electron temperature is the kinetic energy imparted to the electrons via the photo-ionization laser. This is especially true for plasmas with high initial electron temperatures, since the electron temperature can be well approximated solely from the wavelength of the ionizing laser pulse. However, as initial electron temperatures begin to approach zero, a number of heating mechanisms begin to become significant that can substantially alter the initial electron temperatures. These mechanisms impart more uncertainty into the initial electron temperature as the ionizing laser energy falls.

In our lab we are interested in probing the electron strongly-coupled plasma regime, which corresponds to low initial electron temperatures where uncertainty caused by heating mechanisms is high. Since there are currently no known simple techniques for experimentally determining the electron temperature of a UCP system, it is important to be able to properly quantify these different mechanisms. While there are a number of theoretical predictions for each of these mechanisms, many of these theories have not been experimentally verified. Most of these theories are derived in a manner intended to be sufficiently general to be able to span many orders of magnitude in both electron temperature and density. As such,

very few theories include the external electric and magnetic fields that are included in the experiments in our lab. Additionally, many of these theories include approximations that are not applicable to our experimental system. Finally, in most cases, these heating mechanisms are treated individually, and thus little insight into how these different heating mechanisms add together can be gained from these works. It is for all of these reasons that even with all of the available theoretical work, that the electron temperatures of low temperature UCPs are challenging to determine.

To address this uncertainty, I have developed a UCP formation model that is specifically tailored to the experimental conditions present in our lab and can be used to determine the post-formation conditions that describe UCPs in our experimental system. This model utilizes the molecular dynamics techniques described in chapter 2 and chapter 3 to simulate the formation phase of UCPs. Through these simulations the model is able to predict how the different heating mechanism are compounded together, but without additional work is unable to distinguish between the individual contributions of the different mechanisms. This model is a critical resource for determining post-formation conditions for experiments in our lab, but further work is needed to usefully extend this model's predictive power to other plasma systems.

In this chapter I will first discuss the different heating (and cooling) mechanisms that are present in ultracold plasma systems, as well as the current research that surrounds each of these mechanisms. Next I will describe the basics behind the formation model that I have developed, and discuss how this model differs from the modeling described in chapter 3. After that, I will discuss the results of that I have obtained from this model. These results include the discovery of a previously undiscovered electron heating mechanism in UCPs caused by an external electric field, as well as a discussion of the results that were used to generate a temperature for the electron oscillation damping measurement described in chapter 3. Finally, the chapter will end with concluding thoughts and a discussion of the many areas of future investigation that can be pursued with this model.

## 4.1 Heating Mechanisms

In ultracold plasma systems, a number of different physical mechanisms exist that lead to the heating of UCP electrons. These mechanisms include continuum lowering[1], disorder induced heating[2], and three-body recombination[3]. Electron heating mechanisms result in two main consequences for ultracold plasma systems. First, heating mechanisms, by definition lead to higher electron temperatures, which in turn leads to lower strong coupling parameters. Additionally, since many heating mechanisms are more severe at lower electron temperatures, these mechanisms naturally lead to a maximum coupling parameter. To determine what this maximum coupling parameter is for a given set of parameters, it is necessary to be able to accurately estimate the magnitude of the different heating mechanisms. Additionally, a thorough understanding of the origins of each of these mechanism may give insight into different techniques that can be used to minimize or outright prevent additional heating.

Second, heating mechanisms impart additional uncertainty into the electron temperature. In the absence of heating mechanisms, the electron temperature is solely determined by the wavelength of the ionizing laser pulse. In this scenario, the uncertainty in the laser wavelength is the sole determinant of the uncertainty in the electron temperature. However, with the presence of heating mechanisms, the magnitude of these effects needs to be estimated. Since each estimate has an associated uncertainty, the inclusion of heating effects leads to added uncertainty in the electron temperature. Additionally, it is not always obvious exactly how these different heating mechanism combine, adding yet another layer of uncertainty. Overall, electron temperature uncertainty due heating mechanisms can, for certain parameter ranges, be the primary driver of electron temperature uncertainty in ultracold plasma systems.

For these reasons electron heating mechanisms in ultracold plasmas have been a widely studied topic. This section will give an overview of the underlying physics behind these mechanisms, as well as discuss some research around each of these mechanisms.

### 4.1.1 Continuum Lowering

Continuum lowering is a physical process that can play an important role in many plasma systems. Continuum lowering arises from a plasma's equilibrium distribution of electrons and ions yielding an electric potential structure that leads to a depression of the ionization potential of neutral atoms. In ultracold plasma systems, this leads to an energy difference between a photon in resonance with an atom's vacuum ionization potential and the actual depressed ionization potential. Thus, when the atom is ionized, this energy difference is imparted onto the ionized electron in the form of kinetic energy. This leads to a heating of the UCP's electron component during the UCP formation process. In other hotter, denser, and more complex plasmas, continuum lowering leads to an increased rate of collisional ionization. This changes the thermodynamic properties of these plasmas, altering their dynamical behavior.

There are two main theoretical predictions quantifying the magnitude of the continuum lowering. These are the Stewart and Pyatt prediction (SP)[1] and the Ecker-Kröll prediction (EK)[4]. Historically, Stewart and Pyatt has been the preferred model as its derivation has been considered to be more fundamentally sound[5]. However, recent experiments using free electron lasers to ionize solid-state aluminum[6, 7], have observed continuum lowering results that were more consistent with Ecker-Kröll than Stewart and Pyatt. The results for these experiments were a bit troubling for the field, since Stewart and Pyatt had been the preferred model, and much work had been accomplished relying on results from this model[5]. To further confuse matters, another recent experiment studying shocked aluminum[8], measured results that were consistent with the prediction of Stewart and Pyatt and inconsistent with the prediction of Ecker-Kröll.

The contradictory results of recent experiments have introduced a large degree of uncertainty into the study of continuum lowering. This uncertainty, coupled with the importance of continuum lowering in defining the thermodynamic properties of many plasma systems, has made continuum lowering a lively area of investigation in recent years. Given the well controlled nature of ultracold plasmas, these systems could provide a useful platform to study continuum lowering. Such a line of investigation is likely an area of future research in our lab.

### 4.1.2 Disorder Induced Heating

Disorder induced heating is another heating mechanism that occurs during the formation process of an ultracold plasma. When the atoms are ionized during the UCP creation process, the resulting charged particles become subjected to electric forces. The location of these particles is essentially random, and thus not at an electric potential energy minimum. As such, the plasma particles' potential energy is rapidly converted into kinetic energy during the formation of a UCP, especially for closely-spaced pairs. This leads to a heating of both the electron and ion component of the UCP. This heating occurs in UCPs on timescales on the order of the plasma period,  $\tau_p$ , for both the electron and ion component of the plasma.

Since disorder-induced heating arises from excess potential energy, the magnitude of this heating scales can be naively expected to scale with the nearest neighbor potential energy,  $U_N$ . formally this energy can be written as

$$U_N = \frac{q_1 q_2}{4\pi\epsilon_0 a} \quad (4.1)$$

where  $q_1$  and  $q_2$  are the charges of the two neighboring particles,  $\epsilon_0$  is the permittivity of free space, and  $a$  is the Wigner-Seitz radius. Since the electron strong coupling parameter  $\Gamma_e$  also scales with  $U_N$ , the heating resulting from disorder induced heating would be expected to scale with  $\Gamma_e$  as well. This suggests that disorder induced heating results in an upper

limit on the electron strong coupling constant,  $\Gamma_e$ . Computer simulations have shown that this limit is approximately  $\Gamma_e \approx 1$ [9].

### 4.1.3 Three-Body Recombination

Another electron heating mechanism is the process of three body recombination. Three body recombination involves a collision between two free electrons and an a single ion. During the collision, one of the electrons becomes bound to the ion creating a high energy Rydberg atom, and causing the electron to lose energy in the process. The excess energy is thus transfered to the other electron in the form of additional kinetic energy. The three body recombination process has two main immediate consequences. First, the recombined electron and ion pair is now a neutral atom. Thus, there is a loss of a plasma electron and ion. Secondly, the additional kinetic energy imparted to the second electron increases the total energy of the electron component of the plasma. Through thermalizing electron-electron collisions, this increase in kinetic energy will ultimately yield an increase in the electron temperature. Thus three-body-recombination can be a significant heating mechanism for UCP electrons.

Three-body-recombination is expected to scale very strongly with the electron temperature. In the weakly-coupled limit the three-body-recombination rate is proportional to  $T_e^{-9/2}$ [3]. This effective temperature scaling can be illustrated by the following scaling argument. The rate at which an electron collides for with an ion is  $nv\sigma$ , where  $n$  is the density of ions,  $v$  is the velocity of of the electron, and  $\sigma$  is the collisional cross-section for a electron-ion collision. In the standard theory for Coulomb collisions, the electron-ion collisional cross-section for substantial collisions is defined  $\sigma = \pi b_{90}^2$ , where  $b_{90}$  is the impact parameter associated with 90 degree deflection, (discussed in more detail in chapter 3) and defined as  $b_{90} = q^2/(4\pi\epsilon_0 k_B T_e)$ . Since  $T_e$  is defined by the expression,  $3/2k_B T_e = 1/2mv^2$ ,  $v$  is proportional to  $\sqrt{T_e}$ , making the collision rate proportional to  $nT_e^{-3/2}$ .

For a three body recombination event to occur a second electron needs to be nearby the colliding electron and ion pair. In the standard treatment of Coulomb collisions, collisions occur when a pair of particles pass within a distance less than  $b_{90}$ . The probability that a second electron is within a distance  $b_{90}$  of the colliding electron-ion pair is  $b_{90}^3/a^3$  which is proportional to  $nT_e^{-3}$ . By multiplying this probability with the original electron-ion collision rate, the effective three-body recombination rate of  $n^2T_e^{-9/2}$  is obtained. This simple scaling argument has been validated by a more rigorous theoretical treatment which yields the three-body recombination rate of[10]

$$K_{TBR} = 3.8 \times 10^{-9} \cdot n_e^2 T_e^{-9/2} \quad (4.2)$$

where  $n_e$  is the electron density in units of  $cm^{-3}$ , and  $T_e$  is the electron temperature in units of  $K$ .

The three-body-recombination rate can be rewritten in terms of the electron strong coupling constant,  $\Gamma_e$ , and the plasma period, in the following way

$$K_{TBR} \propto \frac{\Gamma_e^{9/2}}{\tau_p} \quad (4.3)$$

where  $\tau_p$  is the plasma period.  $\tau_p$  is defined as

$$\tau_p = \sqrt{\frac{m_e \epsilon_0}{n_e q^2}} \quad (4.4)$$

where  $q$  is the fundamental charge constant,  $m_e$  is the mass of an electron, and  $\epsilon_0$  is the permittivity of free space, the electron strong coupling constant is defined as

$$\Gamma_e = \frac{q^2}{4\pi\epsilon_0 a k_b T_e} \quad (4.5)$$

where  $a$  is the Wigner-Seitz radius. Since most plasma processes occur on timescales proportional to  $\tau_p$ , Eq. 4.3 shows that the three-body-recombination rate is determined by the electron strong coupling parameter,  $\Gamma_e$ . This suggests that three-body-recombination ultimately limits the degree of strong coupling in UCPs. This limit has been confirmed theoretically, and has been shown to be  $\Gamma_e = 0.2$ [11].

There are a number of possible methods that could be potentially used to circumvent the  $\Gamma_e = 0.2$  limit. First, a large external magnetic field could be applied to the UCP. The Coulomb collision rate is predicted to fall significantly in the presence of a magnetic field[12]. Since three-body recombination is a function of these collisions, a reduction in the collision rate should subsequently lead to a reduction in the three-body recombination rate. This should in turn reduce the electron heating rate caused by three-body recombination leading to a lower steady state electron temperature as well as a higher  $\Gamma_e$ . As of this dissertation, this line of research has yet to be experimentally investigated by the ultracold plasma community. However, applying a strong magnetic field to the ultracold plasma experiment in our lab is one of the many planned directions for future work. Such a field would likely be initially used to measure electron-ion collisions in a magnetized environment, but could likely be used to measure the effects of large magnetic fields on Rydberg formation as well.

Another possibility for circumventing the  $\Gamma_e = 0.2$  limit is to probe the UCP early in its lifetime. The three-body recombination limit of  $\Gamma_e = 0.2$  is a steady state limit, and thus the limit does not apply to ultracold plasma systems that have yet to reach a steady state  $\Gamma_e$ . Since it can take on the order of  $100\tau_p$  to reach a  $\Gamma_e$  steady state, and since UCP formation can often occur on faster timescales, it should be possible to probe a well thermalized UCP with  $\Gamma_e$  greater than 0.2. Recent work, conducted by myself as well as others in my lab, have confirmed that this limit can indeed be broken, as we were able to measure a UCP with  $\Gamma_e = 0.35 \pm 0.08$ . This measurement is discussed in further detail in chapter 3.

One final consideration for three-body recombination, is the inevitable breakdown of the three-body recombination rate at low temperatures. According to Eq. 4.2, the three-body

recombination rate begins to approach infinity as the electron temperature approaches 0. An infinite collision rate is of course unphysical, and thus the formulation of the three-body recombination rate expressed in Eq. 4.2 must eventually break down. As of this dissertation, there have yet to be any experimental confirmations of the breakdown of the three-body recombination rate in UCP system. However, one theory posited a three-body recombination rate that was proportional to  $\Gamma_e^{9/2} e^{-\lambda\Gamma_e}$  [13], where  $\lambda$  is a constant for  $Z = 1$  ions that is determined by molecular dynamics simulations.

#### 4.1.4 Cooling Mechanisms

In addition to the previously listed heating mechanisms, a number of electron cooling mechanisms exist. The two most prominent of these mechanisms are cooling from adiabatic expansion and cooling resulting from electron evaporation. For certain plasma parameters, these cooling mechanisms can have a substantial impact on the electron temperature. For this reason, a thorough understanding of these cooling mechanisms is often necessary to accurately predict the electron temperatures of UCP systems.

Electron cooling from adiabatic expansion occurs during the expansion phase of the plasma. As the plasma evolves, electrons exert pressure on the confining UCP ions, accelerating the ions and leading to the expansion of the ion cloud. As the volume of the ion cloud expands, the electron volume increases, and the pressure on the ions decreases. This leads to the cooling of the electron cloud. Furthermore, since the rate of ion expansion increases as time elapses, the effects of adiabatic expansion on the electron temperature become more pronounced later in the plasma lifetime. For this reason, the effects of adiabatic cooling on the electron temperature are negligible directly after the formation of the plasma. This allows for measurements taken directly after UCP formation to ignore the contributions from adiabatic cooling. The mechanisms describing adiabatic cooling will be more thoroughly discussed in chapter 7.

Electron evaporation is the process by which electrons escape from the confining ion potential. Since electrons require a sufficiently high amount of kinetic energy to escape from the plasma, this process leads to the high-energy electrons selectively leaving the plasma. This leads to a decrease in the average electron kinetic energy, and thus a decrease in the electron temperature. Unlike adiabatic cooling, the rate of evaporation is greatest during the formation phase of the UCP. For this reason evaporation is likely an important consideration when determining the electron temperature of a UCP at any point during its lifetime. This is especially true for low density plasmas like the ones created in our laboratory. Since electron evaporation is a function of the potential depth, electron evaporation plays a more significant role in the dynamics of low density plasmas, where large charge imbalances are needed to create sufficiently deep ion confining potentials. The dynamics of evaporation will be more thoroughly discussed in chapter 6.

## 4.2 Formation Model

I have developed a numerical model that can be used to model the formation phase of ultracold plasma systems. This model uses the same GPU accelerated full molecular dynamics treatment that has been described in the previous two chapters. Where this model differs is in how the model is initialized. While simulations discussed in chapter three started after the formation phase, where the electron cloud was already in thermal equilibrium, charged particles in this model are initialized as ions with a corresponding bound electron (essentially as an atom), and the ionization process is explicitly modeled. Ionization is modeled classically by sequentially imparting kinetic energy to a given electron in an amount that is determined by the chosen ionization photon energy. Once an electron has been ionized, that electron, as well as its corresponding ion, are subjected to the standard Coulomb forces and are allowed to evolve in time in manner consistent with previously discussed full MD simulations. This section will discuss the details of this model.

Since this model uses a full molecular dynamics treatment, both electrons and ions are modeled explicitly in the model. Ions are initially imparted with zero kinetic energy, and are randomly placed corresponding to a Gaussian distribution. For each ion, a corresponding electron is placed at a fixed distance  $\alpha$  away from the ion, and is randomly assigned a polar and azimuthal angle. For simulations described in this section, the orbiting radius  $\alpha$  was set to be equal to the  $\alpha$  from the softened inter-particle Coulomb potential  $U = q_1q_2/4\pi\epsilon_0\sqrt{r^2 + \alpha^2}$  that is used in this simulation as well as all of the other GPU-accelerated models discussed in this dissertation. The results shown in the next section use an  $\alpha$  that has been set equal to 500nm. As a check, simulations were also run with an  $\alpha$  of 100nm, and these results proved to be indistinguishable from the  $\alpha=500$ nm results. This check strongly suggests that heating results from this model are likely independent from the choice of  $\alpha$ , or at least in the case where  $\alpha < 500$ nm.

Once the initial placement of all the electrons and ions pairs is determined, the ionization process begins. Ionization is simulated dynamically in the model, with a finite time width laser pulse. This leads to the sequential ionization of electrons in the model, with the duration between electron ionizations being evenly spaced in time. This time interval is determined by the length of the laser pulse divided by the number of electron and ion pairs in the UCP. For the simulations that will be discussed in the results section the duration of the laser pulse was set to 10ns to correspond to experimental conditions.

When an electron is ionized, the process is modeled by instantaneously imparting kinetic energy to the electron, where the resultant velocity vector is pointed directly away from the electron's corresponding ion. The amount of kinetic energy imparted is determined by the ionizing laser pulse and by conservation of energy. The explicit amount of kinetic energy is determined by the following equation:

$$\Delta KE_i + U_B + \sum_j^{i-2} U_{(i-1)j} + \sum_j^{i-2} U_{ij} = \Delta E_L \quad (4.6)$$

where  $\Delta KE_i$  is the kinetic energy imparted to the  $i$ th charged particle.  $U_B$  is the binding energy of the electron-ion pair, and, in general,  $U_{ij}$  is the Coulomb inter-particle potential energy between the  $i$ th and  $j$ th charged particle. Charged particles in the model are stored in an array in an AB pattern where ions correspond to A and electrons correspond to B, and each sequential AB corresponds to a given electron-ion pair. As such, the  $U_{(i-1)j}$  sum corresponds to all of the ion potential energy pairs, the  $U_{ij}$  sum corresponds to all the electron potential energy pairs, and the  $i - 2$  upper limit in both sums means that only previously ionized electron-ion pairs are considered in this energy calculation. Finally, the  $\Delta E_L$  term is the energy of a single ionizing photon minus the ionizing threshold for a given atom. Overall, this equation should lead to an appropriate ionization potential depression, and in the absence of other charged particles should lead to post-ionization electron energies of  $\Delta E_L$ .

Electron-ion pairs in the model are only subjected to Coulomb forces once they have undergone the ionization process. Care needed to be taken to preserve energy conservation for electrons immediately after ionization. Initial electron orbits were fixed at 500nm, and initial ion positions were typically on the order of  $10^{-4}$  to  $10^{-3}m$  from the origin. This results in there being at least a three orders of magnitude difference between the coordinate position of electrons and ions, and the difference between their positions, which in turn results in there being a six orders of magnitude between the square of this difference and the square of their coordinates. This leads to significant roundoff error when using floating point precision to explicitly calculate Coulomb forces, and thus double precision is necessary.

Additionally, our experiment contains a DC external electric field, which should be included in the model. For this reason, a DC electric field force is also included in the model. The inclusion of an external electric field also alters the conservation of energy equation in the following way

$$\Delta KE_i + U_B + \sum_j^{i-2} U_{(i-1)j} + \sum_j^{i-2} U_{ij} + q\vec{E} \cdot (\vec{r}_{i-1} - \vec{r}_i) = \Delta E_L \quad (4.7)$$

By modifying Eq. 4.6 in this manner, an external electric field can be explicitly implemented into the code. Since the electron-ion initial separation is relatively small when ionized, this electric field contribution is not large.

### 4.3 Results

As of this dissertation, the surface has only been scratched in terms of the possible uses for this model. Currently, the model has only been used to predict the electron temperature for the experimental conditions from the second dataset of electron center-of-mass damping measurement discussed in chapter 3. From this analysis, we were able to find agreement between numerical modeling and the data collected in the second dataset. Additionally, the model allowed the observation of a previously undiscovered heating mechanism resulting from an applied external electric field. Without the inclusion of this mechanism, the simulations did not predict a sufficient amount of heating for agreement with the data.

To be able to interpret results from the model, a method needs to be devised as to how to determine the electron temperature from the results. The method that was chosen was to use the average electron kinetic energy to determine the electron temperature. Formally, this can be expressed as

$$1/2m_e \langle v_e^2 \rangle = 3/2k_bT_e \quad (4.8)$$

where  $m_e$  is the electron mass,  $T_e$  is the electron temperature, and  $\langle v_e^2 \rangle$  is the average squared electron velocity in the distribution. An example of these calculations can be seen in figure 4.1.

Some care needs to be taken to make sure that electron evaporation is properly accounted for in the simulation. When an electron escapes from the confining ion potential, it will be consistently accelerated by the external electric field. If these electrons are included in the electron temperature calculation, the electron temperature will be overestimated. It is for

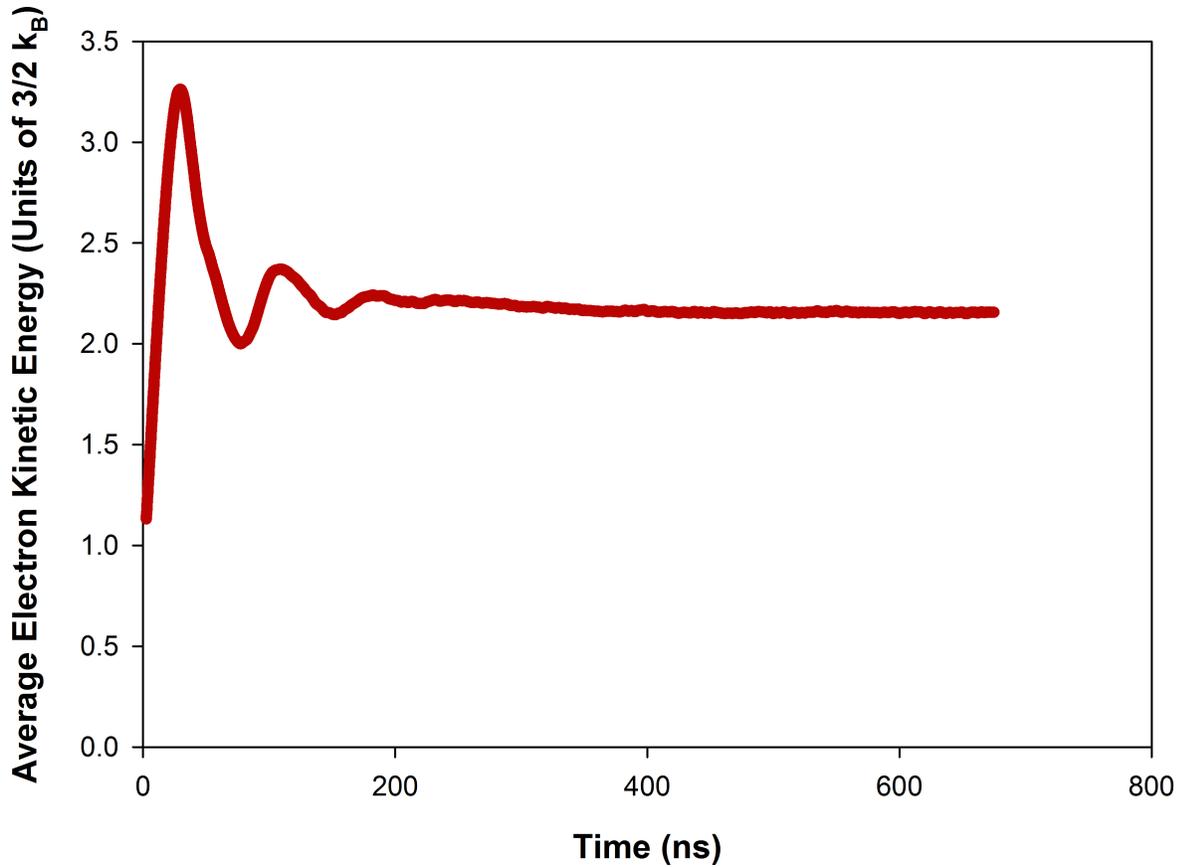


Figure 4.1: Plot of the average electron kinetic energy as a function of time. The plot represents a simulation of a UCP corresponding to the cold data point (72000 ions,  $\Delta E_L = 0.1k_b$ ,  $\sigma = 630\mu m$ ) in the presence of a  $3 V/m$  external electric field. The graphs shows that after some initial electron kinetic energy oscillations, the final electron temperature stabilizes at roughly 2.15K.

these reason that electrons that are more than  $4 \sigma$  away from the origin in position are excluded from temperature calculations.

One of the major findings from this model was the discovery of the external electric field induced electron heating mechanism. When an ultracold plasma is formed, the external electric field induces an electric dipole in the plasma, which is brought about from a shift in the electron cloud center-of-mass. For this shift to occur, electric potential energy due to the interaction between the electric field and the electron cloud is transferred to electron kinetic energy. This leads to a heating of the UCP electron component in a manner analogous to disorder induced heating. The magnitude of the shift in the electron cloud center-of-mass is roughly proportional to the strength of the applied field. As such, the amount of electric field heating increases with the strength of the applied electric field. Predictions of the total electron heating as a function of the applied electric field can be seen for UCPs parameters mirroring the cold and hot point from the second data set of the electron center-of-mass damping in figure 4.2.

## 4.4 Conclusion

There are many heating mechanisms that impact UCP electron temperatures that occur during the formation of the plasma. After giving an overview of these mechanisms, this chapter gave an overview of the model that I developed to quantitatively predict the magnitude of the heating of these mechanisms. Results from this model were used to help characterize the electron center of mass damping measurements discussed in the last chapter. Additionally, this model was used to find a previously undiscovered heating mechanism, electron heating due to an external electric field, without which we would have been unable to explain the second dataset from chapter 3. Finally, while this model has already proved integral to understanding data from the electron center-of-mass damping measurements, there are still many more uses for this model. The Future work chapter has some examples of these potential uses.

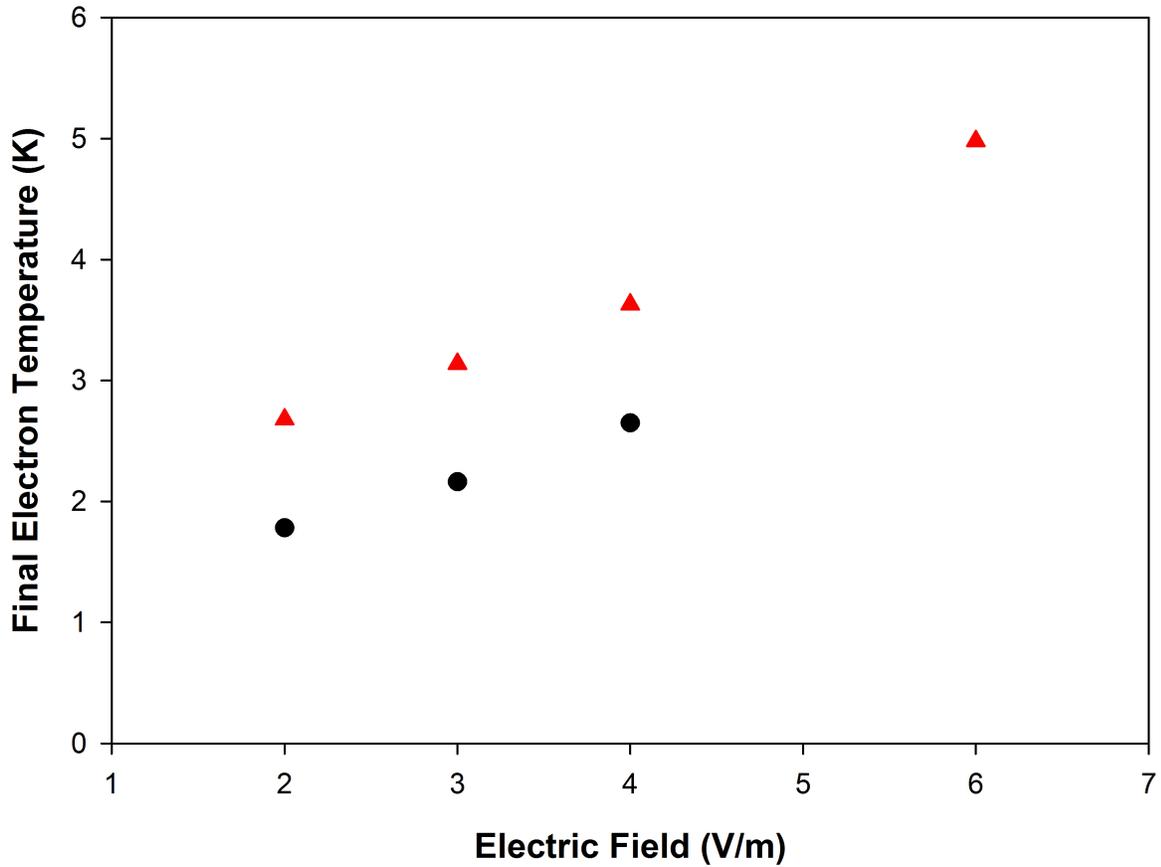


Figure 4.2: Plot of the final electron temperatures as a function applied external electric field for two different sets of conditions. The black circles correspond to the cold data point (72000 ions,  $\Delta E_L = 0.1k_b$ ,  $\sigma = 630\mu m$ ) and the red triangles corresponds to the hot data point (62000 ions,  $\Delta E_L = 2.25k_b$ ,  $\sigma = 625\mu m$ ). The graphs shows that as the applied external electric field is increased, the amount of electron heating increases.

# References

- [1] J. C. Stewart and K. D. Pyatt, *Astrophys. J.* **144**, 1203 (1966).
- [2] M. Lyon and S. D. Bergeson, *J. Phys. B: At. Mol. Opt. Phys.* **44**, 184014 (2011).
- [3] T. C. Killian et al., *Phys. Rev. Lett.* **86**, 3759 (2001).
- [4] G. Ecker and W. Kröll, *Phys. Fluids* **6**, 62 (1963).
- [5] B. Crowley, *High Energy Density Phys.* **13**, 84 (2014).
- [6] O. Ciricosta et al., *Phys. Rev. Lett.* **109**, 065002 (2012).
- [7] T. Preston et al., *High Energy Dens. Phys.* **9**, 258 (2013).
- [8] D. J. Hoarty et al., *Phys. Rev. Lett.* **110**, 265003 (2013).
- [9] S. G. Kuzmin and T. M. O'Neil, *Phys. Plasmas* **9**, 3743 (2002).
- [10] T. C. Killian, T. Pattard, T. Pohl, and J. M. Rost, *Phys. Reports* **449**, 77 (2007).
- [11] F. Robicheaux and J. D. Hanson, *Phys. Rev. Lett.* **88**, 055002 (2002).
- [12] M. E. Glinsky, T. M. O'Neil, M. N. Rosenbluth, K. Tsuruta, and S. Ichimaru, *Phys. Fluids B.* **4**, 1156 (1992).
- [13] A. V. Lankin and G. E. Norman, *J. Phys. A: Math. Theor.* **42**, 214042 (2009).

# Chapter 5

## Thermalization

Thermalization of the electron component in ultracold plasmas is an interesting area of study. Electron thermal equilibrium is an important underlying assumption in many UCP and other plasma systems. For instance, it is an important assumption in characterizing electron evaporation in a UCP, and more generally it underlies the determination of whether a plasma is in the hydrodynamic or kinetic regime[1]. Electron thermal equilibrium is typically hard to verify experimentally (in UCP systems or in plasma systems in general), and thus the existence of thermal equilibrium is often determined theoretically.

For the purposes of this chapter, electron thermalization is more specifically defined as the evolution of the electron velocity distribution towards a Maxwellian. Typically, this process is assumed to be driven by electron-electron collisions. As electron-electron collisions occur, energy is exchanged between colliding electron pairs. Over time this energy exchange leads to the electron velocity distribution evolving to a Maxwell-Boltzmann distribution. Collisions are not the only thermalization mechanism, however. For example, wave propagation in plasmas can also lead to significant energy transfer, and thus contribute to thermalization[2].

In addition, other thermalization mechanisms are possible. In the case of UCPs, which contain both ions and electrons, it is possible that electron-ion interactions may contribute to the thermalization process. The work detailed in chapter 3 has shown that the shape of the ion spatial distribution can have a substantial impact on electron oscillation damping in UCPs. As ion spatial distributions deviate from a uniform distribution, the ion potential becomes more anharmonic. This anharmonicity leads to additional collisionless effective "damping." This collisionless energy transfer would be naively expected to increase the thermalization rate of the electron component of the UCP since it represents another way besides electron-electron collisions for energy to be transferred. Since this collisionless

damping would be substantial under the formation conditions for most UCPs, it is reasonable to assume that it would have a role in the initial thermalization of the electrons. How significant an effect this is, however, is hard to estimate.

In this chapter I will discuss the numerical model that I developed to simulate the electron thermalization process both in the presence of a uniform ion distribution and a Gaussian ion distribution. The purpose of this model was to theoretically investigate the electron thermalization process in UCPs for typical experimental parameters shortly after the UCP formation. Furthermore, this work was used to determine whether the shape of the electron and ion spatial distribution has any significant impact on the thermalization process due to the collisionless mechanisms described above.

It was found that the ion density distribution had a small but detectable impact on the overall electron thermalization rate, with electrons in the presence of a Gaussian distribution thermalizing faster than in the uniform case. More generally, the results from the model were used to identify three distinct phases of the electron thermalization process. Finally, it was found that regardless of the ion density distribution, the thermalization process scaled with the standard electron-electron collision timescale, as would be naively expected for the case in which thermalization is entirely driven by electron-electron timescale. Overall, the results of the work discussed in this chapter suggest that for most applications, the ion distribution will not be an important consideration for determining if a plasma is thermalized. However, for some applications (such as evaporation) that depend on a well thermalized electron kinetic energy tail, ion density distribution effects may be important.

## 5.1 Motivation

During the investigation into the electron center-of-mass damping (discussed in Chapter 3), it was found that electron center-of-mass damping could arise from collisionless mechanisms in the UCP. These mechanisms were, in general, associated with ion density non-uniformity, and confining potential anharmonicity. Perhaps the most striking example of

these collisionless mechanisms at work can be seen in figure 3.7 (in chapter 3) which shows a damping of the electron center-of-mass from a simulation that explicitly excluded electron-electron collisions (the mechanism previously assumed to be the sole driver of center-of-mass damping). Additionally, figure 3.7 shows the collapse and revival of the this center-of-mass oscillation. This suggests some form of mode-coupling or at least quasi-mode-coupling is present in the system.

One hypothesis for this mode coupling is as follows. As the electron center-of-mass oscillates, it moves from areas of relatively high ion density to areas of relatively low ion density. When in the low ion density areas, the density of the electron cloud exceeds that of the ions. This in turn leads to a net force on the electrons which is, at least partially, pointed outward in the radial direction. Through this basic phenomena, the electron center-of-mass oscillation is able to couple to different radial modes. Simulations of this damping in the presence of an external magnetic field lend credence to this hypothesis. The presence of an external magnetic field along the axial direction would serve to impede electron motion in the radial direction, and would thus be expected to reduce the hypothesized mode coupling. This in turn would suggest that the presence of an axial magnetic field would lead to the preservation of electron center-of-mass oscillation. The fact that this is exactly what simulations show is significant evidence that this sort of mode coupling is occurring in the system.

If this sort of mode coupling is occurring, it is by necessity resulting in the transfer of energy between these modes. While electron-electron collisions are the typically considered the driver of electron thermalization in UCPs, in general a mechanism that leads to energy transfer would be expected to drive a system toward thermal equilibrium. This begs the question, does the mode coupling that appears to be arising from ion density non-uniformity lead to an appreciably faster thermalization of the electron cloud in UCPs? This question will be addressed in the upcoming sections.

## 5.2 Model

I have developed a numerical model to simulate the electron thermalization process in UCPs. UCP electrons are initialized to all have the same fixed velocity magnitude which corresponds to a given electron initial kinetic energy. The electron cloud is allowed to evolve and is tracked using molecular dynamics techniques, and the degree of thermalization can be computed at any timestep from the electron velocity distribution. By allowing electrons to evolve both in the presence of uniform and non-uniform ion distributions, the model is used to determine how the ion distribution impacts electron thermalization.

### 5.2.1 Overview

To test the effect of collisionless energy transfer mechanisms on the thermalization process, it is useful to look at two different ion spatial distributions. The first is a Gaussian distribution, which was chosen because it mirrors typical experimental conditions. This density distribution is defined as

$$n = n_0 e^{-r^2/2\sigma^2} \quad (5.1)$$

where  $n_0$  corresponds to the peak density distribution,  $r$  is the standard radial coordinate, and  $\sigma$  is the standard Gaussian spatial size. The second distribution is a uniform distribution, which was chosen because the collisionless energy transfer seen in the Gaussian case is absent for a uniform distribution and thus a uniform distribution provides a baseline for comparison. In the absence of collisionless energy transfer mechanisms, we would expect that if the average density in both types of spatial distributions is the same, the thermalization rate would be the same if it were only due to electron collisions alone. For computational efficiency, these ion distributions are modeled as a smooth background.

Unlike the ions, UCP electrons are modeled individually and are evolved in time in via molecular dynamics calculations. Treating the electrons in this manner allows for us to monitor the evolution of the electron velocity distribution, giving insight into the thermalization

process. The model includes electron-ion collisions as well, though. Since ions in the system are modeled as a smooth background, electron-ion collisions are implemented via Monte Carlo collision operator[3]. This general model structure is nearly identical to the hybrid molecular dynamics monte carlo model discussed in chapter 3. The calculations described in this section were confined to regions of weak plasma coupling.

The model also utilizes an adjusted electron-electron interaction potential:

$$U = \frac{q^2}{4\pi\epsilon_0\sqrt{r^2 + \alpha^2}} \quad (5.2)$$

Where  $q$  is the fundamental electron charge  $\epsilon_0$  is the permittivity of free space, and  $r$  is the distance between the two electrons. The softening parameter,  $\alpha$ , is added to the Coulomb potential to help preserve energy conservation. While repulsive interactions do not strictly require an adjusted Coulomb potential, the inclusion of this adjusted potential allowed for the use of larger timesteps without causing energy conservation problems from close collisions. The value of  $\alpha$  was chosen, however, so that its presence does not significantly affect the quantities of interest.

Again, ions in the model are treated as a smooth ion background that is fixed in time. This implementation allows for the average electron-ion interaction to be calculated by just a single analytical force interaction per electron per timestep. This increases computational efficiency by reducing the electron-ion interaction from an  $O(N^2)$  to an  $O(N)$  problem. Additionally, this implementation avoids the complication of bound states. Deeply bound electrons can have very fast orbits, which require very small timesteps to conserve energy, and by avoiding these states larger timesteps can be utilized resulting in shorter run times.

Electron-ion collisions are modeled via random binary Coulomb collisions. These collisions have been incorporated into the model via a Monte Carlo collision operator which assumes stationary and infinitely massive ions. The range of these binary collisions are truncated via a cut-off parameter that is based on the Debye length. This cutoff parameter is

calculated using the electron temperature and the local ion density, and is implemented, following typical practices, in order to avoid divergences when averaging over all Coulomb collisions. At each time step, the probability of each electron undergoing an electron-ion collision is calculated. This probability is given as

$$P = n\sigma v dt \tag{5.3}$$

where  $n$  is the density of ions,  $v$  is the electron velocity, and  $dt$  is the length of the timestep. If a collision occurs, an impact parameter is randomly chosen and the electron velocity is deflected by amount determined by Rutherford scattering.

As mentioned above the model utilizes two different ion spatial distributions, a Gaussian and a uniform distribution. Since UCP ions are many orders of magnitude more massive than electrons, it reasonable to assume little motion from the ions over the relevant electron thermalization timescales. This allows for the ion distributions to be modeled as fixed quantities over the duration of the model. For consistency, initial electron spatial distributions mirror the distributions of the ions, and initial electron positions are generated randomly with an appropriate distribution. All electrons are initially given the same velocity magnitude corresponding to a chosen thermal energy,  $3/2k_bT$ , but their velocity direction is randomized.

### 5.2.2 Comparing the Distributions

Some thought was needed in choosing the exact analytical form for the two ion distributions. This was straightforward in the Gaussian case. Since the ion charge density falls off sufficiently fast, the Gaussian ion distribution could be described simply by an untruncated Gaussian distribution as seen in Eq. (5.1). The best method for describing a uniform ion distribution was a little less obvious. One possibility would be to assume a uniform density ball of ions, where the total number of ion could be determined by the chosen density and the

radius of the ball. While such a description may initially appear reasonable, it is ultimately inappropriate for this work. By characterizing the ions in this manner, a density, as well as a density gradient, discontinuity is introduced at the edge of the ion ball. Since the purpose of this work is to quantify effects from ion density non-uniformity, and the uniform ion density is meant to be used as a baseline, such a discontinuity is unacceptable.

For this reason, it was instead decided to analytically describe the uniform ion distribution as a uniform charge density with an infinite spatial extent. At first, such a description may seem inappropriate, since it essentially assumes an infinite number of ions in the system, and creates a mismatch in ion number between the two ion distributions. However, since this ion distribution is spherically symmetric, Gauss' Law states that electrons within a certain radius from the origin,  $R$ , will only experience a net force from ions that are also contained within  $R$ . Thus, by initializing the electrons' spatial coordinates such that they are forced to be contained within  $R$ , the electrons are effectively subject to the previously discussed ball distribution, without being subjected to the density discontinuity.

The differences between these distributions lead to a couple subtleties, and some care needed to be taken to make sure that a proper comparison between simulations of the two distributions were made. First, there is the question of what density should be used. Since the uniform distribution consists of a single density, and the Gaussian distribution consists of a distribution of densities, the answer to this question is not immediately obvious. One possibility is to calculate the average density of of the Gaussian distribution as defined by

$$\langle n \rangle = \frac{1}{N} \int_0^\infty n^2(r) d^3r \quad (5.4)$$

where  $\langle n \rangle$  is the average density,  $n(r)$  is the particle density, and  $N$  is the total number of particles in the corresponding density distribution,  $n(r)$ . Once  $\langle n \rangle$  is calculated for the Gaussian case, the density of the uniform distribution to be equal to that value. This sets the average densities of both distributions equal. This method is ultimately what is used in

this model, but properly calculating the Gaussian average density is more challenging than what one would initially expect due to the presence of electron evaporation in the Gaussian case.

Since, in the Gaussian case, the confining ion potential is finite, some electrons in the UCP are able to escape from the plasma via evaporation. This complicates a proper comparison between the Gaussian and the Uniform distribution in two main ways. First, evaporation leads to the development of a charge imbalance in the system. In turn, this charge imbalance leads to a deviation from an ideal Gaussian electron distribution, that must be accounted for when calculating the average electron density. Second, since high-energy electrons are more likely to escape via electron evaporation, the electron evaporation process leads to an energy loss, which results in a cooling of the electron cloud.

In order to account for the effects of evaporation, simulations first needed to be run in order to quantify the magnitude of these effects. First, a search spanning a small range of temperature space needed to be undertaken. The purpose of this search was to determine what initial electron energy led to desired electron temperature after the initial cooling caused by evaporation had been accounted for. This meant that the initial energy given to electrons in the Gaussian case was actually higher (typically by 10-20%) than in the uniform case. After this temperature is determined, the number of electrons that initially leave the plasma is determined. From this a charge imbalance is calculated and an effective electron distribution is approximated (using the  $T=0$  approximation discussed in chapter 3) allowing for the an average density to be calculated.

### 5.2.3 Quantifying Thermalization

In order to monitor the thermalization process, it is necessary is to be able to quantify how thermalized the electron cloud is. There are a number of different ways in which this can be done. One possibility is to fit the electron velocity distribution to an appropriate Maxwellian. By doing this, the resultant  $\chi^2$  value can be used as the metric by which the

degree of thermalization is measured. Another method would be to describe the velocity distribution by its moments, and then compare these moments to what would be expected for a Maxwellian. In general,  $i$ th order moment of a distribution,  $f(x)$ , can be written as

$$\mu_i = \int_{-\infty}^{\infty} (x - \langle x \rangle)^i f(x) dx \quad (5.5)$$

where  $\mu_i$  is the  $i$ th order moment, and  $\langle x \rangle$  is the mean value of the distribution,  $f(x)$ .

For this work, it was decided to utilize a metric based on the second order moment of the electron velocity distribution, also known as the variance. This metric is defined as follows

$$\nu = A\sqrt{\sigma^2}/\bar{v} \quad (5.6)$$

where  $\bar{v}$  is the average electron velocity and  $A$  is a normalization constant. The variance of electron velocity distribution,  $\sigma^2$ , is defined as

$$\sigma^2 = \int_0^{\infty} (v - \bar{v})^2 f(v, t) dv \quad (5.7)$$

where  $f(v, t)$  is the electron velocity distribution as a function of time. Here,  $\nu$  is essentially a normalized square root of the variance of the electron velocity distribution. The inclusion of  $\bar{v}$  in  $\nu$  allows for a direct comparison of  $\nu$ s at different  $T$ s.  $A$  is chosen so that a  $\nu$  of 1 corresponds to a perfect Maxwellian.

## 5.3 Results

Model simulations with 60,000 electrons were run across a range of electron temperatures (5K-50K). To corresponds with a typical set of low-density UCP conditions[4, 5, 3, 6], average ion and electron densities were set at  $2.5 \times 10^{12} m^{-3}$ . This corresponds to a Gaussian size,  $\sigma$  of  $800\mu m$ , and a radius,  $R$ , of a uniform distribution of 1.248mm.

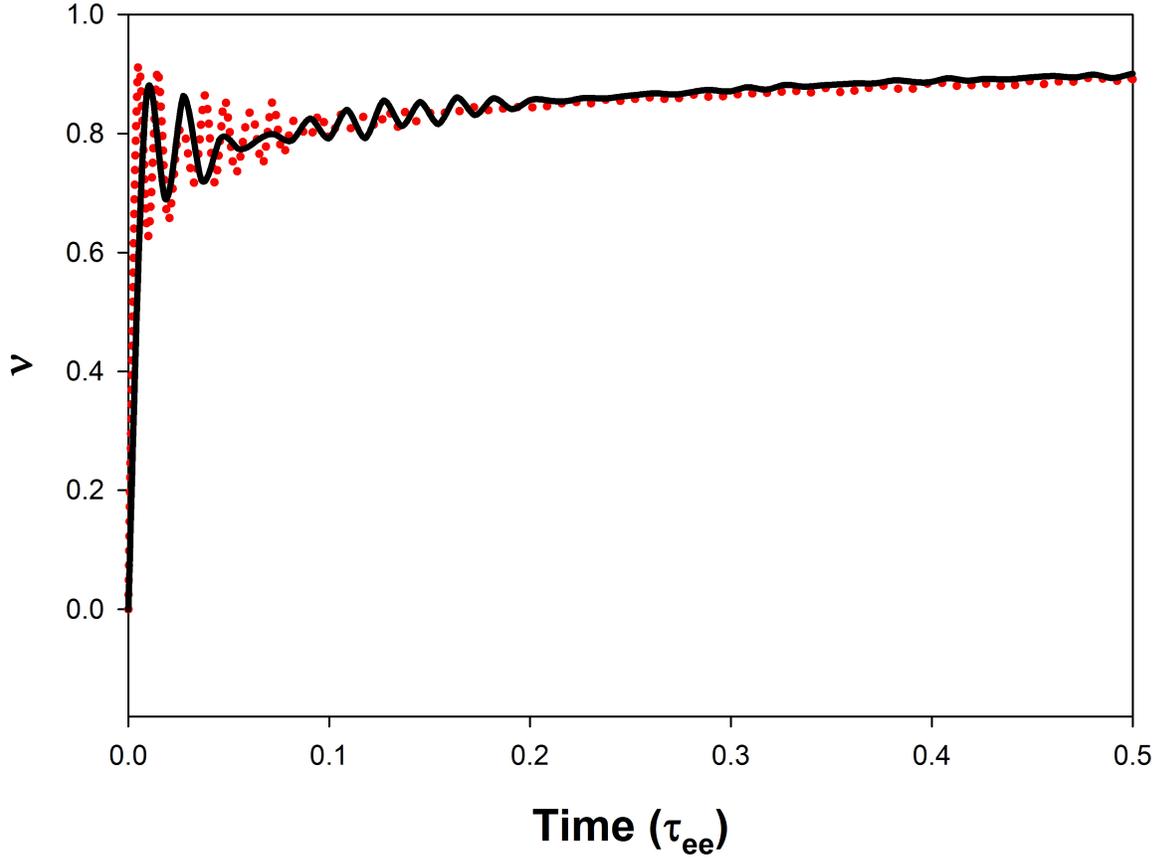


Figure 5.1: (Placeholder Graph). Plot of  $\nu$  as function  $t/\tau_{ee}$  for two different temperature conditions. The orange curve corresponds to a  $T = 30K$  electron cloud, and the gray curve corresponds to a  $T = 50K$  electron cloud. Since the two curves overlap, the simulations results confirm that thermalization do scale with the characteristic electron-electron collision time,  $\tau_{ee}$ .

### 5.3.1 Basic Scaling

Since electron thermalization is typically thought to be driven via electron-electron collisions, UCP thermalization is assumed to scale with the electron-electron collision rate, and approximated to occur within the characteristic collision timescale. This characteristic collisional timescale,  $\tau_{ee}$ , is given by the following expression[7]

$$\tau_{ee} = \frac{1}{\nu_{ee}} = 3\sqrt{\pi} \left( \frac{4\pi\epsilon_0^2 \sqrt{m_e} k_b^{3/2} T^{3/2}}{n_e q^4 \ln(\Lambda)} \right) \quad (5.8)$$

where  $\nu_{ee}$  is the characteristic electron-electron collision frequency,  $\epsilon_0$  is the permittivity of free space,  $n_e$  is the electron density,  $q$  is the fundamental charge constant,  $T$  is the electron temperature,  $k_b$  is the Boltzmann constant. Eq. 5.8 also contains the Coulomb logarithm term,  $ln(\Lambda)$ . This term arises from averaging over all possible collisions, and is defined as

$$ln(\Lambda) = ln\left(\frac{C\lambda_d}{b_{90}}\right) \quad (5.9)$$

where  $\lambda_d$  is the Debye screening length,  $b_{90}$  is the characteristic impact parameter defined as  $b_{90} = q^2/4\pi\epsilon_0k_bT$ , and  $C$  is a constant which sets the cutoff. In this work  $C$  is set to 0.765 to remain consistent with other theoretical work[8, 9, 10]. For the UCP parameters simulated in this work  $\tau_{ee}$  scales as

$$\tau_{ee} \sim \frac{1.555 \times 10^{-7}T^{3/2}}{ln(2T^{3/2})} \quad (5.10)$$

Our model simulations confirmed this expected scaling outlines. Simulations were run over range of different electron temperatures, and  $\nu$  was calculated as a function of time. These simulations showed that  $\nu(t/\tau_{ee}(T))$  remained largely invariant with respect to  $T$ . An example of this can be seen in figure 5.1. This scaling remained consistent for both the uniform ion density case as well as the Gaussian ion density case.

The fact that electron thermalization scales with the electron collision time in the presence of an anharmonic Gaussian ion potential seems to suggest that the ion density does not play a significant role in the electron thermalization process. However, it is still possible that the ion density non-uniformity still leads to collisionless thermalization that also happens to scale with the typical collision timescales. The results of the next subsection indicate that this is indeed the case, where it can be seen that the distribution of ion density has an effect on the electron thermalization rate.

### 5.3.2 Density Distribution Effects

Simulation results were used to compare the two different density distributions to determine if density non-uniformity played a significant role in the thermalization process. By calculating the value of  $\nu$  as a function of time for simulations representing both of these distributions, such a comparison can be made. The results of this comparison were in line with expectations outlined in the introduction of this chapter. Electrons in the presence of a Gaussian ion distribution thermalized faster than an equivalent electron cloud in the presence of a uniform ion distribution for all of the cases simulated. However, while detectable, the effect was fairly small in all cases. An example of such a comparison can be seen in figure 5.2.

Figure 5.2 shows pronounced oscillations of  $\nu$ . These oscillations arise from collective breathing modes (oscillations of the rms electron spatial coordinate). At early times, these oscillations are more pronounced, but as time elapses these oscillations damp. As the electron cloud oscillates the density of the cloud changes. In theory the electron thermalization rate should change with these density changes. However the time averaged density remains constant over the thermalization process. Since the expected thermalization rate scales linearly with density, the oscillation of the electron cloud was not expected to significantly impact the electron thermalization rate.

To gather more insight into the electron thermalization process, it is useful to look at instantaneous thermalization rates. The most straightforward way for this would be to take the time derivative of  $\nu$ . However, such a calculation would be dominated by the rapid fluctuations in  $\nu$  due to the presence of breathing modes in the system and other noise. This necessitates another approach, and it was chosen that the thermalization process would be parameterized as an exponential-type relaxation toward thermal equilibrium. Thus, thermalization is described as

$$\nu = 1 - e^{-\kappa(t)} \quad (5.11)$$

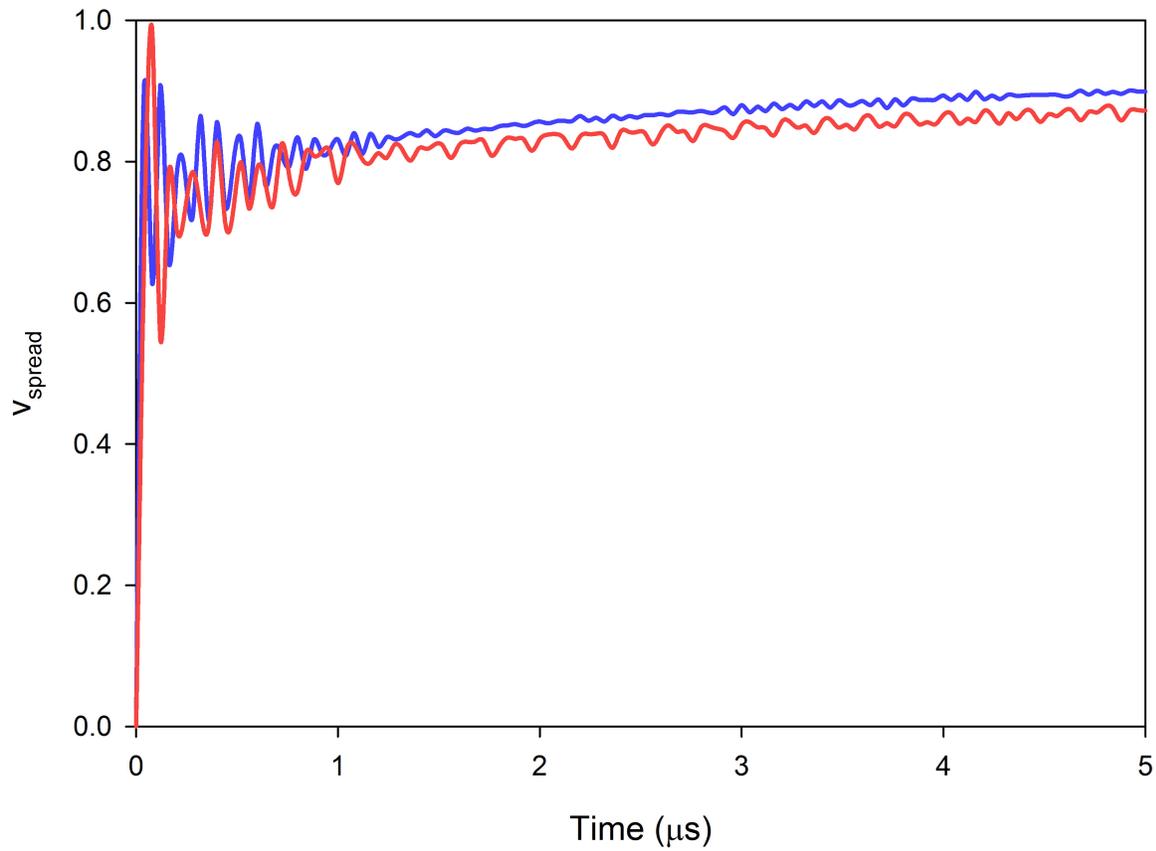


Figure 5.2: The time evolution of the electron velocity spread parameter,  $\nu$ . The blue curve corresponds to electrons evolving in the presence of the Gaussian ion distribution, and the red curve corresponds to electrons to evolving in the presence of a uniform ion distribution. Electrons in both curves have a temperature of 30K. As can be seen in the graph, electrons in the presence of a Gaussian ion distribution thermalize detectably faster than electrons in the presence of a uniform ion distribution.

where  $\kappa(t)$  can be used to quantify the degree of thermalization in the system, and  $\dot{\kappa}(t)$  can be thought of as the instantaneous thermalization rate. By assuming this functional form of  $\nu$ , a qualitative comparison can be made between different thermalization curves. By taking natural logarithm of Eq. (5.11) the following expression is obtained:

$$\kappa(t) = -\ln(1 - \nu) \quad (5.12)$$

By taking the time derivative of Eq. 5.12, an expression for  $\dot{\kappa}(t)$  can be derived. This value is as follows

$$\dot{\kappa}(t) = \frac{\dot{\nu}(t)}{1 - \nu} \quad (5.13)$$

The value of  $\dot{\kappa}(t)$  is thus an instantaneous thermalization rate, and can be calculated from Eq. 5.13.

Equations (5.12) and (5.13) give two possible methods by which the  $\dot{\kappa}(t)$  can be found. Each of these approaches has its own advantages. Eq. (5.13) is perhaps the most straightforward method, as it is direct calculation of  $\dot{\kappa}(t)$ . By explicitly plotting  $\dot{\kappa}(t)$  as a function of time, it is immediately obvious that the electron thermalization rate has very large peak at the beginning of the simulation, and that after this peak the thermalization rate rapidly falls off. An example of this can be seen in figure 5.3.

While Eq (5.13) provides the most straight forward way to plot  $\dot{\kappa}(t)$ , it can be be problematic. Since  $\dot{\kappa}(t)$  is a function of  $\nu$  and  $\dot{\nu}$ , fast oscillations in  $\nu$  will lead to large oscillations in  $\dot{\nu}$ , which in turn lead large oscillations in  $\dot{\kappa}(t)$ . As can be seen in figure 5.2 for all simulations  $\nu$  undergoes a number of these fast oscillations. Thus, when Eq. (5.13) is explicitly used to calculate  $\dot{\kappa}(t)$ , the result yields very noisy values of  $\dot{\kappa}(t)$ . In principle, these oscillations could be smoothed out by taking local averages. However, given the large ratio between the oscillation amplitudes and the average local  $\dot{\kappa}(t)$  value, as well as the finite resolution of the simulation, such a method can lead to large errors. Such errors were particularly pronounced

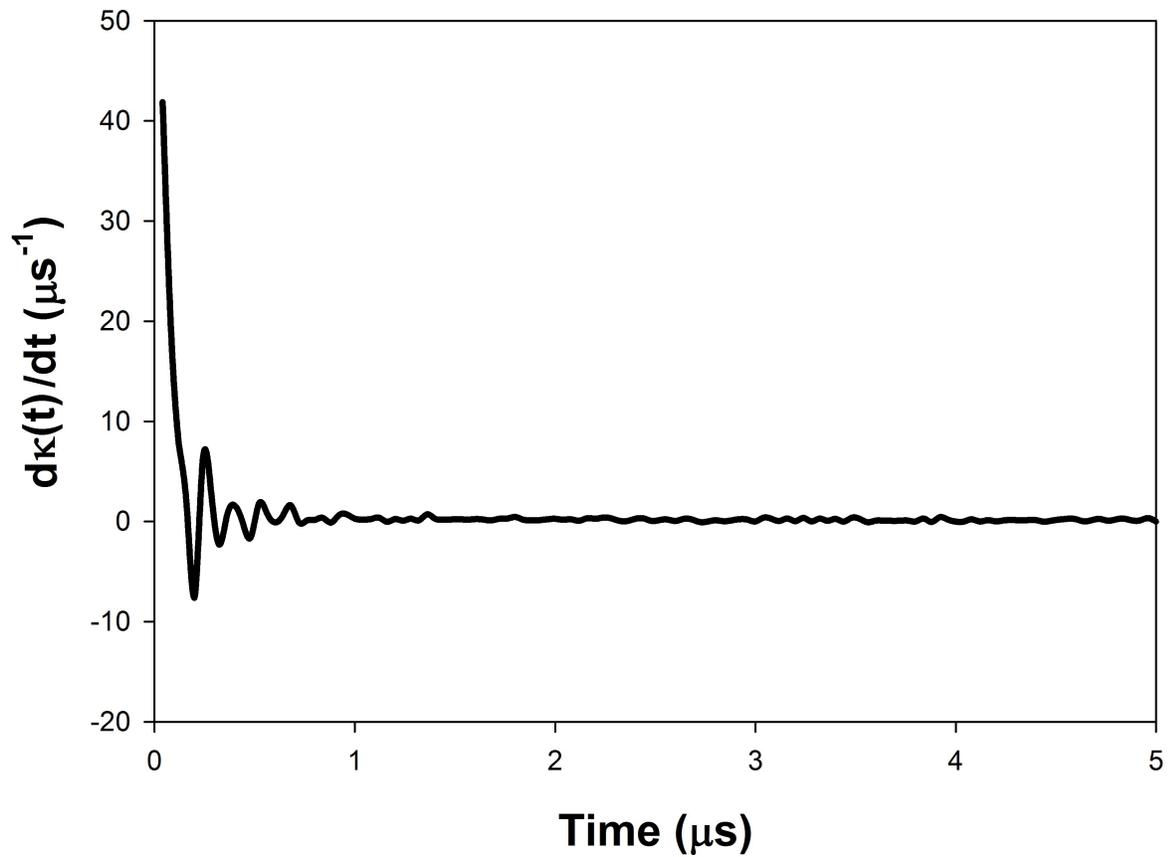


Figure 5.3: (Placeholder Graph) Plot of the effective thermalization rate,  $\dot{\kappa}(t)$ , as a function of time for a 30K plasma in the presence of a uniform density distribution. The curve is calculated first by smoothing  $\nu$  and then using Eq. (5.13). The graph shows that  $\dot{\kappa}(t)$  has a large peak at the beginning of the simulation, and that the thermalization rate falls off rapidly after that peak.

during the later times in the simulation (After the thermalization rate peak illustrated in figure 5.3).

For this reason, the method for determining  $\dot{\kappa}(t)$  given by Eq. (5.12) is a better choice for determining  $\dot{\kappa}(t)$  after the thermalization rate peak. Figure 5.4 uses this method to plot of  $\dot{\kappa}(t)$  for a Gaussian and a uniform density UCP with 30K electrons as a function of time. The slope of these curves corresponds to the instantaneous thermalization rate,  $\dot{\kappa}(t)$ , to a good approximation. Figure 5.5 shows a curve analogous to figure 5.4 but with 50K electrons. The main difference between two plots is that the 50K case thermalizes slower, as was shown in figure 5.1.

Between figure 5.3 and 5.4 there it can be seen that there exists three distinct regions of thermalization. The first phase occurs during the large  $\dot{\kappa}(t)$  peak illustrated in figure 5.3. This phase of electron thermalization is very short lived, and typically occurs over timescales that are less than a few plasma periods. However, most of the electron thermalization occurs during this phase of the thermalization process. Figure 5.6 shows an example of the electron velocity distribution at the end of this phase.

To understand why this occurs, it is useful to consider two velocity distribution limits. In the case of a delta function distribution, all collisions will lead to an increase in the variance of the distribution and thus an increase in  $\nu$ . Alternatively, in the case of a Maxwellian distribution, the principle of detailed balance prevents collisions from imparting any additional variance into the system. In general, this implies that single collisions impart more variance into low variance distributions than they impart into high variance distributions, and this explains the rapid thermalization that occurs during the initial phase.

The second and third phases are more visible in figure 5.4 than in 5.3. In figure 5.4, the start of the second phase of electron thermalization begins directly after the initial thermalization rate peak, and lasts roughly 10-20 plasma periods. The hallmark of this phase is that the ion distribution non-uniformity plays a significant role electron thermalization process, where the ion non-uniformity accelerates the thermalization of the electron cloud. This

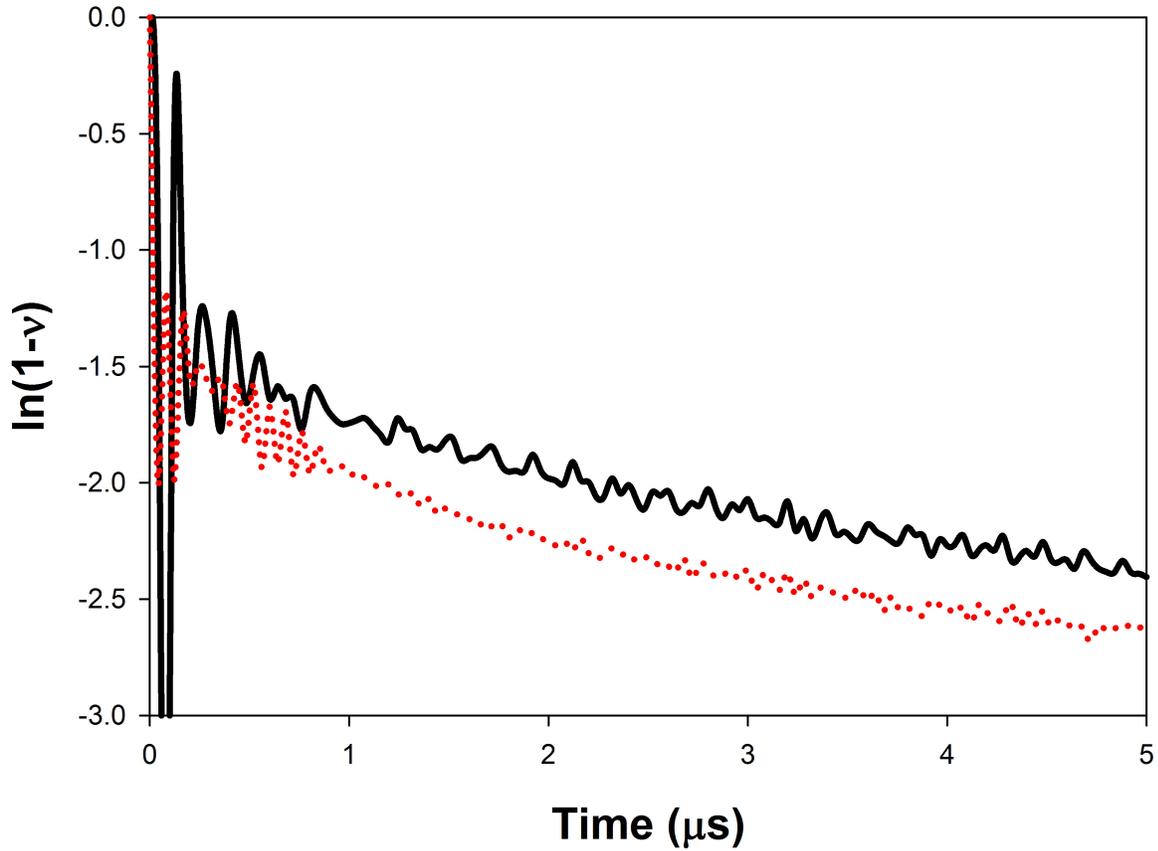


Figure 5.4: Plots illustrating the time evolution of  $\ln(1-\nu)$ . The red dotted curve to electrons evolving in the presence of the Gaussian ion distribution, and the black curve corresponds to electrons to evolving in the presence of a uniform ion distribution. Electrons in both curves have a temperature of 30K, and density of  $2.5 \times 10^{12} m^{-3}$  which corresponds to a plasma period of roughly 70ns. The instantaneous slopes,  $\dot{\kappa}(t)$ , of these curves represent an effective thermalization rate. Furthermore, the plots show evidence of the second and third phase of electron thermalization. The characteristics of these regions are described in the main body of the chapter.

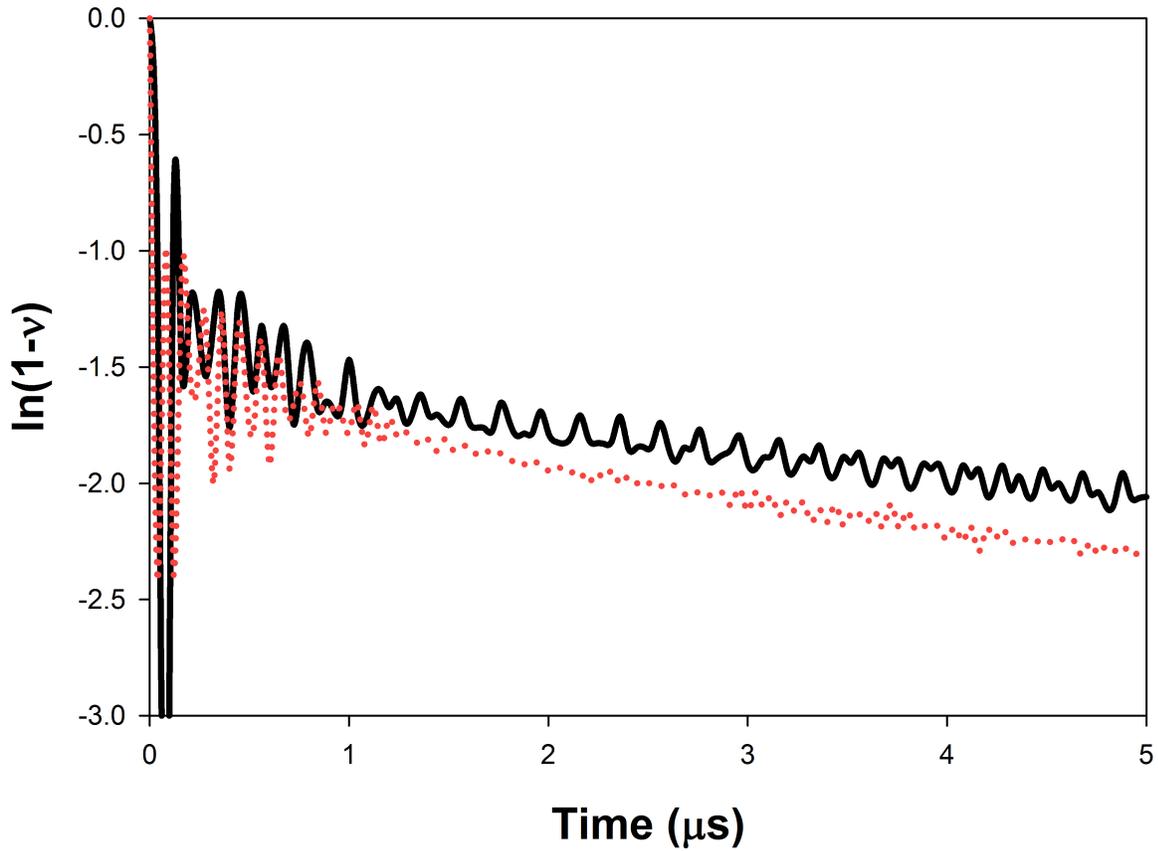


Figure 5.5: Plots illustrating the time evolution of  $\ln(1 - \nu)$ . The red dotted curve corresponds to electrons evolving in the presence of the Gaussian ion distribution, and the black curve corresponds to electrons to evolving in the presence of a uniform ion distribution. Electrons in both curves have a temperature of 50K, and density of  $2.5 \times 10^{12} m^{-3}$  which corresponds to a plasma period of roughly 70ns. The instantaneous slopes,  $\dot{\kappa}(t)$ , of these curves represent an effective thermalization rate. Unlike the previous figure, this plot only shows evidence of the second thermalizaion phase. This is do the  $T^{3/2}$  scaling leading to a slower thermalization process for the hotter 50K case shown in this plot.

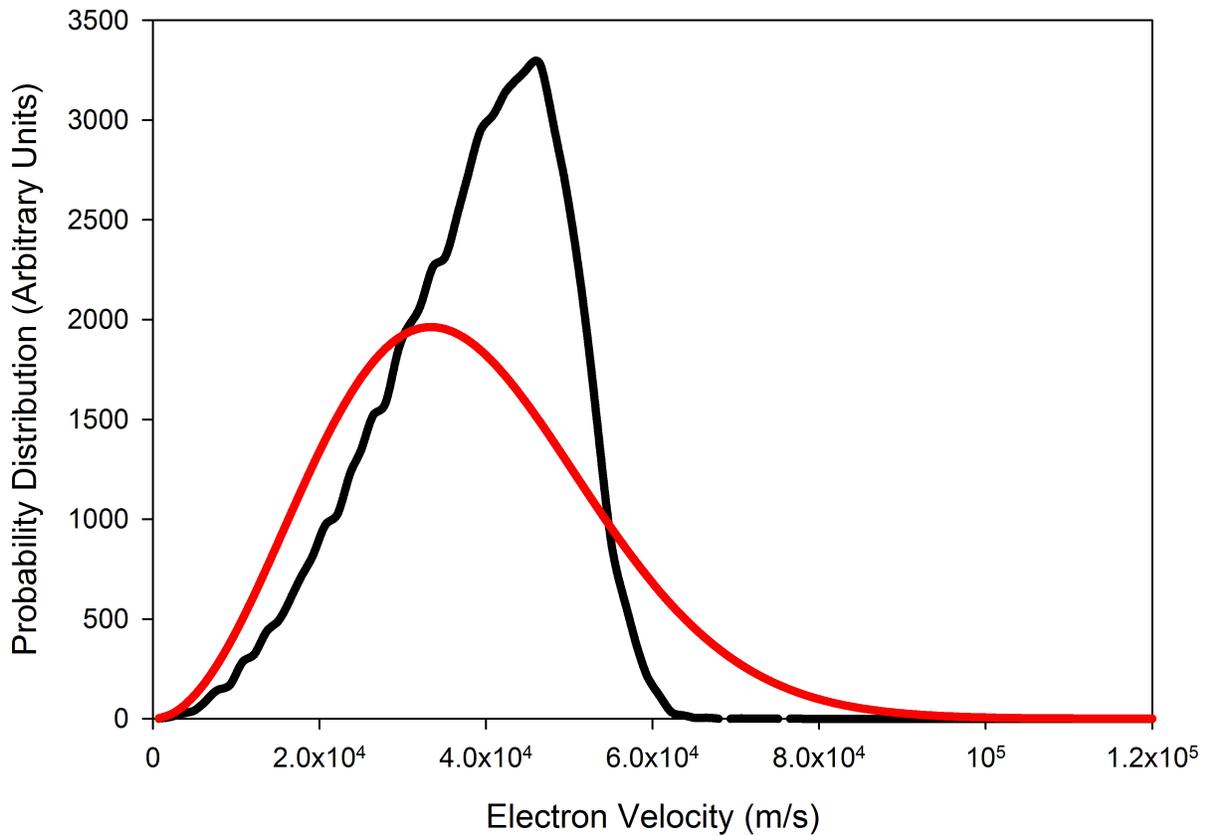


Figure 5.6: Plot of the electron velocity distribution after the electron cloud has been allowed to evolve for 40ns. The black curve is the actual electron velocity distribution, and the red curve is the best fit Maxwellian distribution. While the electron velocity distribution is still far from being fully Maxwellian, the majority of velocity spread has already been imparted into the electrons cloud after this short period of time.

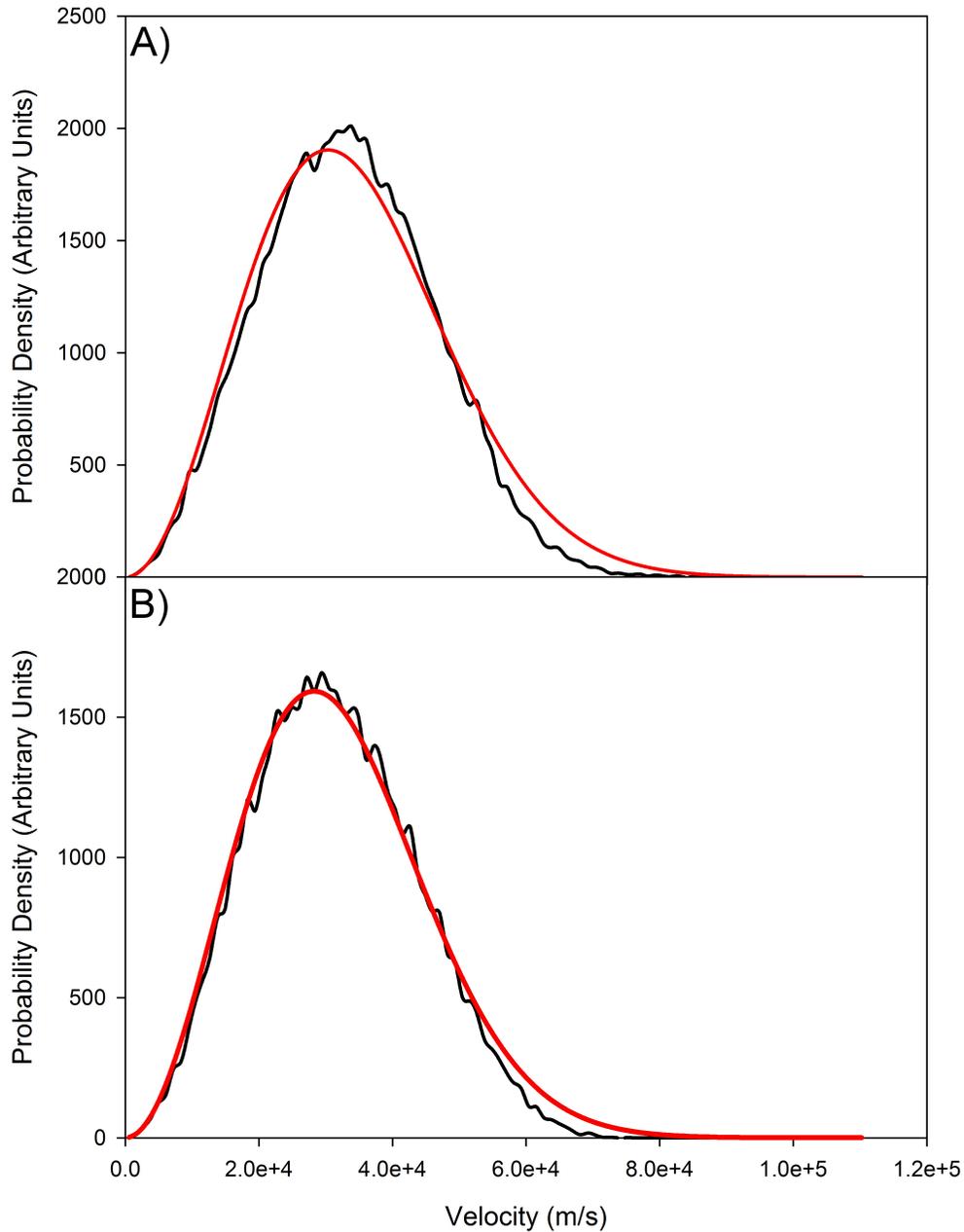


Figure 5.7: Plot of the electron velocity distribution after the electron cloud has been allowed to evolve for  $5\mu s$ . The black curves correspond to the actual velocity distribution, and the red curves correspond to the best fit Maxwellian distribution. The curves in A correspond to an electron cloud that has evolved in the presence of a uniform ion distribution, and the curves in B correspond to an electron cloud which has evolved in the presence of a Gaussian ion distribution. The figure shows that electrons thermalize slightly faster in the presence of a Gaussian ion distribution than in the presence of a uniform distribution. This effect is particularly noticeable in the high-energy tail of the distribution, which is noticeably more filled out in the case of the Gaussian ion distribution.

increased thermalization rate is likely due to energy transfer caused by the mode coupling discussed in section 5.1. This effect is illustrated in figure 5.4, where it can be seen that the slope for the Gaussian curve is noticeably steeper for the Uniform curve.

Interestingly, this difference in thermalization rates does not persist indefinitely. Rather, there is a point when the thermalization rate between the two cases again becomes similar. In the third and final phase of the thermalization, the ion distribution no longer seems to be important. This can be seen in figure 5.4, where the slopes of the two curves are approximately equal, suggesting that the different ion distributions are no longer playing a role in the thermalization process, and that, at this point, thermalization is entirely driven by electron-electron collisions.

Overall, the impact of the ion distribution is noticeable but not particularly dramatic. An example of the differences between the two cases at late times can be seen in figure 5.7. The fact that there is a density distribution dependence does indicate that collisionless mechanisms play a role in energy transfer and thus thermalization. This is interesting from a fundamental point of view. However, from a more practical point of view, the size of the contributions of the collisionless factors is small compared to the overall scale of the thermalization of the electrons. Thus, it is unlikely to be a critical consideration in UCP experiments or interpretation of those experiments for our UCP parameters. There is, however, value in understanding that the timescale for the upper tail of a Maxwellian distribution is determined by the collision rate, as this can be a critical consideration in calculating electron evaporation rates (see chapter 6).

## 5.4 Conclusion

A numerical model was developed to theoretically investigate the thermalization of the electrons in ultracold plasmas. We identified three main regions of thermalization. Additionally, it was found that non-uniformity of the ion spatial distribution led to a perceptibly faster electron thermalization. This demonstrates that collisionless mechanisms indeed to

play a role in ultracold plasma thermalization. This faster thermalization will likely have little impact on physics where only average electron energy is important. However, it will likely be an important consideration for physics where the electron energy distribution, and specifically the high energy Maxwellian tail, plays an important role. To be honest, we anticipated a larger relative role of the collisionless effects given their dominance in determining the center-of-mass damping rate. In phase two of the thermalization, there really is a strong difference between the Gaussian and Uniform cases in terms of the  $\dot{\kappa}(t)$  parameter. What was not originally anticipated is that the bulk of the thermalization would occur very rapidly, diminishing the role of collisionless mechanisms on the absolute amount of thermalization. Despite this unanticipated result, we gained valuable information about assumptions about the high-energy velocity tail at early times in the UCPs in our experiments and do see that collisionless mechanisms are not completely ignorable in the overall thermalization physics.

# References

- [1] P. Bellan, *Fundamentals of Plasma Physics*, Cambridge University Press, 2006.
- [2] J. D. Jackson, *Classical Electrodynamics*, Wiley, 2nd edition, 1975.
- [3] W.-T. Chen, C. Witte, and J. L. Roberts, *Physics of Plasmas* **23**, 052101 (2016).
- [4] T. Wilson, W.-T. Chen, and J. Roberts, *Phys. Plasmas*. **20**, 073503 (2013).
- [5] T. Wilson, W.-T. Chen, and J. L. Roberts, *Phys. Rev A* **87**, 013410 (2013).
- [6] W.-T. Chen, C. Witte, and J. L. Roberts, Submitted to *Phys. Rev. Lett.* (2017).
- [7] P. Mulser, F. Cornolti, E. Besuelle, and R. Schneider, *Phys Rev E* **63**, 016406 (2000).
- [8] J. Daligault and D. Mozyrsky, *HIGH ENERGY DENSITY PHYSICS* **4**, 58 (2008).
- [9] L. S. Brown, D. L. Preston, and R. S. Jr., *Phys. Rep.* **410**, 237 (2005).
- [10] P. E. Grabowski, M. P. Surh, D. F. Richards, F. R. Graziani, and M. S. Murillo, *Phys. Rev. Lett.* **111**, 215002 (2013).

# Chapter 6

## Evaporation

In ultracold plasmas, evaporation is the process by which electrons escape the confining ion potential and leave the plasma. Evaporation plays an important role in the evolution of ultracold plasma systems, and the rate at which it occurs ultimately determines the lifetime of the plasma. Additionally, the electron evaporation rate is a directly measurable quantity, and thus can offer other many insights into the dynamics of UCPs.

In order maximize the information obtained from evaporation measurements, a thorough understanding of the mechanics of evaporation is needed. Ideally, there would be a set of analytical expressions describing the functional form of the electron evaporation rate as a function of other UCP parameters. In the context of ultracold atomic gases, such expressions have long been established[1]. More recently, the ALPHA collaboration has adapted these expressions to be applicable to the antiproton plasmas present in their experimental system[2]. Naively, it would be expected that such expressions would also be applicable to UCP systems. However, these expressions make two assumptions that are not clearly applicable to UCPs.

First, it is implicitly assumed that the average electron mean free path is much greater than the spatial size of the UCP. In other words, once an electron gains enough energy to escape the plasma, it will do so. This greatly simplifies the calculation of the evaporation rate because there is no need to consider the actual spatial path of an escaping electron. Second, this treatment simplifies the consideration of Coulomb collisions by assuming that only small-angle collisions contribute significantly to the evaporation rate. This assumption is rooted in the fact that in many plasmas small-angle collisions do indeed dominate collisional processes. By considering only such small-angle collisions, again the calculation of the evaporation rate

is simplified through having to consider collisions for electrons with one particular velocity rather than integrating over multiple velocities.

In order to test the validity of these assumptions, I developed a numerical model to calculate the rate of evaporation in ultracold plasma systems. By comparing the model results with the evaporation expressions it was found that neither assumption is valid under a typical set of UCP conditions. Additionally, I was able to characterize the degree to which these assumptions are broken. These results indicate that a simple analytic description of the UCP electron evaporation rate would be highly challenging to formulate, and that a numerical model, such as the one presented in this chapter, would be needed for accurate predictions.

Section 6.1 gives an overview of evaporation in the context of ultracold plasma systems. Section 6.2 discusses the theoretical treatment of evaporation in ultracold atomic systems, how this treatment has previously been adapted to plasma systems, and potential issues that could arise from applying this adapted treatment to ultracold plasmas. Section 6.3 gives an overview of how the evaporation model functions. In Section 6.4, I report the numerical results of the model, as well as use these results to test whether previous plasma evaporation treatment can be applied to ultracold plasma systems. Finally, section 6.5 presents my conclusions and the possibilities for future work.

## 6.1 Evaporation in Ultracold Plasmas

Evaporation is an important mechanism in ultracold plasma systems, and plays a significant role in their evolution. When electrons escape via evaporation, they carry away energy from the system. Since an evaporating electron requires sufficient kinetic energy to escape the confining ion potential, only higher energy electrons are able to escape via evaporation. Thus, evaporation leads to a cooling of the plasma's electron component[3, 4]. Furthermore, since the rate of UCP ion expansion is a function of the electron temperature, this cooling results in a change in the dynamical evolution of the ultracold plasma

system[3, 4, 5] as is discussed in the chapter 7. Additionally, electron evaporation leads to higher levels of charge imbalance, resulting in additional Coulomb forces that distort the plasma expansion[4].

One interesting application of evaporation is the possibility of forcibly inducing evaporation with the purpose of tuning UCP parameters to a desired value. By applying an external electric field, the potential confining the electrons is lowered and the rate of evaporation increases. By altering the strength of the electric field, the rate of evaporation can thus be tuned. This method can be used to experimentally set the charge imbalance of the plasma system, allowing for the exploration of parameter ranges spanning the collisional and collisionless regimes (see chapter 3). Additionally, since evaporation leads to a cooling of the electron component of the UCP, forced evaporation offers possible access to previously inaccessible temperatures. Since strong coupling becomes more significant at cold temperatures, this technique also offers the possibility of studying UCP systems with a greater degree of electron strong coupling.

Electron evaporation can also be used as a tool to probe other properties of UCP systems. By using charged particle detection techniques to detect the escaping electrons, the electron evaporation rate can be directly measured experimentally. In fact, charged particle detection is the main method by which the electron component of the UCP is probed experimentally (see chapter 1 for further details). Since evaporation is a directly measurable property, a thorough understanding of the underlying mechanisms of evaporation should allow for significant insight into the properties of experimental UCP systems.

One possibility is to use electron evaporation to probe the temperature of the electron cloud. Techniques for measuring UCP ion temperatures are well established[6], but such techniques are not applicable to UCP electrons, and typically electron temperatures are estimated based on theoretical interpretations of ion expansion[7] or derived from other plasma properties[8, 9]. In the absence of other effects, electron temperatures would be solely determined by the photon energy of the ionizing laser pulse. However, such a simple estimation

ignores various heating mechanisms such as continuum lowering, disorder induced heating, and three-body recombination[10, 11, 12]. Theoretical predictions quantifying these heating mechanisms do exist, and can be incorporated into temperature estimates (see chapter 4), but tests of the predicted heating rates are not available across all UCP parameter ranges. Furthermore, it is not clear how well these heating predictions extend into the strongly coupled regime.

The development of an independent electron temperature measurement would offer the ability to test the net temperature change due to the previously listed effects. UCP electron evaporation is a good candidate for such measurement over a wide range of UCP conditions, since there is a strong electron temperature dependence expected in the electron evaporation rate, making the measurement of the rate potentially sensitive to the electron temperature. However, before evaporation can be used to measure the electron temperature, a robust theoretical link between the electron evaporation rate and the temperature of the electron cloud needs to be developed.

## 6.2 Analytical Expressions for Evaporation

As the previous section discusses, a robust theoretical description of evaporation in ultracold plasma systems would provide a wide variety experimental insights. Ideally this would come in the form of analytical expressions that provide a direct relationship between the evaporation rate and other UCP parameters. Fortunately, such expressions exist in the context of ultracold atomic systems and have been adapted to correspond to charged particle systems. However, the charged particle adaptation contains two assumptions which are not clearly valid for ultracold plasmas. This section will discuss the atomic evaporation theory and how it can be modified to fit charged particle systems. Additionally, this section will outline the assumptions in the charged particle treatment of evaporation, and discuss reasons why they may be problematic.

### 6.2.1 The Atomic Case

Evaporation is a commonly used experimental technique in ultracold atomic physics, where collections of atoms are routinely trapped inside potential wells produced by external optical or magnetic fields[1, 13, 14, 15]. In these systems, evaporation occurs when an atom gains sufficient energy to be able to escape from the trapping potential. The threshold energy necessary for escape is equivalent to the overall depth of the trapping potential, and will be henceforth referred to as the potential barrier,  $U$ .

While confined, atoms in the cloud collide with each other. These collisions lead to energy being transferred between the atoms and can lead to an atom acquiring energy greater than the barrier. If an atom is assumed to escape once its energy exceeds the barrier,  $U$ , the evaporation rate would in that case be the rate at which collisions excite atoms above the barrier.

Integrating over all possible elastic collisions in a thermal gas is difficult, making the determination of the evaporation rate a non-trivial endeavor. Traditionally, in neutral atoms, this difficulty has been mitigated by utilizing the principle of detailed balance. Since collisions with neutral atoms are predominantly large angle collisions, it is reasonable to assume that the vast majority of collisions involving an atom with energy exceeding  $U$  will cause that atom to lose energy and fall back below the barrier. Detailed balance indicates that the rate at which particles fall below barrier is equivalent to the rate at which atoms are excited above the barrier in thermal equilibrium. This leads to the following approximate general expression for the rate of change in particle number from evaporation[1]:

$$\dot{N} = -n\sigma N_{he}v_U = -N_{he}\nu_{col} \quad (6.1)$$

where  $n$  is the particle density,  $\sigma$  is the collisional cross-section,  $N_{he}$  is the number of high energy atoms with energy exceeding the barrier,  $v_U$  is the atom velocity that corresponds to the barrier energy, and  $\nu_{col}$  is the average collision frequency for high energy atoms in the

distribution. If the atoms are assumed to be distributed in a Maxwellian with three degrees of freedom, and the barrier energy to average thermal energy ratio is sufficiently high, the fraction of atoms above the barrier is approximately  $2\sqrt{W/\pi}e^{-W}$ . In this limit Eq. 6.1 takes the following form[1]:

$$\dot{N} = -n\sigma W e^{-W} \bar{v} N \quad (6.2)$$

where  $\bar{v}$  is the average velocity in the particle distribution and  $W$  is the scaled barrier height defined as  $U/k_bT$ .

### 6.2.2 The Charged Particle Case

In plasmas, the presence of charged particles substantially alters the dynamics of evaporation. In general, Coulomb collisions have a much larger interaction range than the hard sphere collisions involved in collisions of neutrals. This leads to a much larger collisional cross-section, which in turn results in a substantially higher frequency of collisions. Furthermore, Coulomb collision cross-sections are velocity dependent, necessitating additional care when calculating  $\dot{N}$ . Finally, the Coulomb interaction leads predominantly to small angle deflections from collisions[16]. This last point has a profound impact with respect to the previous detailed balance assumptions, since it is no longer reasonable to assume that a collision will knock a high energy particle below the barrier. In a recent adaptation of atomic evaporation theory to charged particles, an alternative assumption is made. It was assumed that only a small amount of energy is transferred between colliding particles. In the context of electron evaporation, this allows for the evaporation rate to be expressed in the following way[17]:

$$\dot{N} = \left( \frac{dN}{dt} \right) \Big|_{v=v_U} = \left( \frac{dN}{dv} \frac{dv}{dt} \right) \Big|_{v=v_U} \quad (6.3)$$

where  $dN/dv$  is simply the electrons' velocity distribution, and  $dv/dt$  is a velocity damping rate. In this treatment, the  $dN/dt$  term in Eq. 6.3 is assumed to be the rate at which electrons fall below the barrier. From detailed balance considerations analogous to the atomic case, this rate has to be equal to the rate by which electron move above the barrier, and thus the evaporation rate in general. The small energy transfer assumptions enters Eq. 6.3 by evaluating  $dN/dt$  explicitly at the barrier. By not including additional velocities, this formulation precludes the possibility of significant energy transfer collisions playing an important role in evaporation in charged particle systems. In addition, the assumption that the mean-free path is long compared to the system size is implicit in Eq. (3) as well in that no reference is made to the spatial location of electrons in computing the evaporation rate.

For a Maxwell Boltzmann distribution with three degrees of freedom,  $\frac{dN}{dv}$  takes the following form[17]:

$$\left(\frac{dN}{dv}\right) = 4\pi N v^2 \left(\frac{m}{2\pi k_b T}\right)^{3/2} e^{-\frac{mv^2}{2k_b T}} \quad (6.4)$$

In the limit of  $v$  being much larger than the average thermal velocity,  $dv/dt$  can be calculated via conservation of momentum. The result of this calculation is[17]:

$$\frac{dv}{dt} = -\frac{2e^4 n \ln(\Lambda)}{4\pi\epsilon_0^2 \mu^2 v^2} \quad (6.5)$$

where  $e$  is the fundamental electron charge,  $\mu$  is the reduced mass of the two colliding particles, and  $\ln(\Lambda)$  is the Coulomb logarithm that results from averaging over all possible Coulomb collision angles in the typical treatment of Coulomb collisions in a plasma. Combining equations 6.3-6.5 yields the following expression for evaporation[17]:

$$\dot{N} = -N e^{-w} \frac{\sqrt{2} e^4 n \ln(\Lambda)}{\pi^{3/2} \epsilon_0^2 \sqrt{\mu} (k_b T)^{3/2}} \quad (6.6)$$

### 6.2.3 Assumptions of Charge Particle Evaporation Expressions

As mentioned above, Eq. 6.6 has two underlying assumptions which are not obviously applicable to UCPs. First, it is assumed that once an electron's energy exceeds  $U$ , it immediately escapes from the plasma. However, it is easy to imagine that once a electron is excited above the barrier subsequent collisions could knock that electron back below the barrier before it is able to escape. Implicitly, immediate escape assumes that the average mean free path for an electron is much larger than the spatial size of the UCP. A naive estimate of the mean free path,  $c$ , would simply be  $v^2/\frac{dv}{dt}$ , the velocity times the effective velocity damping time constant. Such an estimate would yield a mean free path for typical experimental conditions roughly an order of magnitude larger than typical UCP sizes, apparently satisfying the prior assumption.

However, such an estimate implies an average electron slowing of  $1/e$  is the relevant amount of velocity decrease for determining an effective mean free path for electrons with kinetic energy just greater than  $U$ . Considering that the overwhelming number of electrons will be substantially closer to the barrier than a factor of  $e$ , it seems likely that the average electron will have to travel a significantly shorter distance than the estimated mean free path before it falls below the barrier due to collisions. Thus, this naive estimate is likely not relevant for evaporation considerations. Without a more sophisticated calculation, it is not intermediately obvious what the relevant mean free path is and thus it is unclear whether this underlying assumption is indeed met. We provide such a calculation below.

Secondly, it is not clear whether the assumed functional form in Eq. 6.3 accounts for all collisions appropriately. The function only includes evaporation contributions from electrons with kinetic energy right at the barrier, and ignores the contributions from large angle collisions for higher-energy electrons not right at the barrier. It is unclear whether the contributions from these electrons are negligible for a typical set of experimental conditions. Furthermore, the assumed evaporation function assumes an average velocity slowing. Con-

sidering that the evaporation rate is the sum of discrete electron escapes, it is unclear whether utilizing an average slowing rate is appropriate.

## 6.3 Model

I have developed a model to calculate the rate of electron evaporation in ultracold plasma systems. By assuming thermal equilibrium, the model calculates the dynamical evolution of high-energy electrons. As time elapses, these electrons undergo Coulomb collisions which are simulated by Monte Carlo techniques. By averaging the rate by which these electrons escape from the plasma, the model is able to calculate the instantaneous rate of electron evaporation as a function of potential depth and electron temperature. This section describes the workings of this model in detail.

### 6.3.1 Overview

In principle, a full molecular dynamics model of UCP electrons could be constructed to determine the electron evaporation rate in UCPs, but the necessary  $O(N^2)$  force calculations naively required make such a model computationally expensive. However, by assuming that the UCP electrons are in thermal equilibrium, approximations can be made that lead to drastically faster computational run times. Electrons in thermal equilibrium are, by definition, distributed in a Maxwell Boltzmann distribution. In the limit of  $U \gg k_b T$  only electrons in the upper tail of the Maxwellian distribution are able to escape from the confining potential. To take advantage of this fact, our simulation only tracks position and velocity vectors for electrons with a total velocity magnitude above a certain tracking threshold. Threshold energies were chosen to be low enough to not interfere with evaporation mechanics, but to be high enough to minimize computation time. A diagram illustrating the relationship between the tracking threshold and the overall electron distribution is given in Figure 6.1.

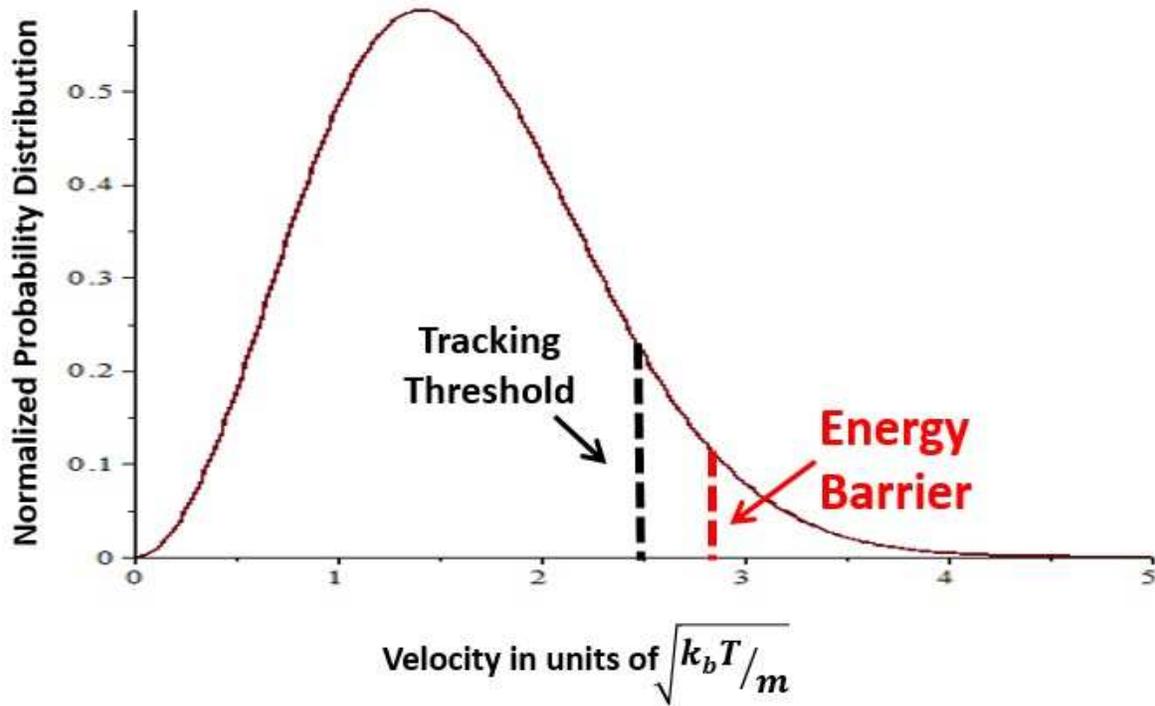


Figure 6.1: A diagram illustrating the different velocity thresholds present in our model. The black dashed line corresponds to the tracking threshold. Electrons to the right of this line are tracked in the model, and electrons to the left are not tracked in the model. The red dashed line represents the energy barrier. Electrons to the right of this line are able to escape from the plasma and electrons to the left are not.

The simulation consists of a series of small timesteps. In each timestep, tracked electrons are moved in space in a direction and at a speed determined by their velocity. After this movement, collision probabilities for collisions with both electrons and ions are calculated. Random numbers are generated to see if a collision occurs. If it does, the electron velocity is changed by the collision. The paragraphs below provide the details of these collision calculations.

The tracked electrons in the model are uniformly distributed across the volume of a sphere, with the surface of the sphere acting as a “hard wall” potential barrier. When an electron comes in contact with the barrier, it will either escape if it has sufficient energy, or be reflected back toward the center if it does not. The rate at which electrons escape from the system, the evaporation rate, can thus be calculated. It should be noted that by treating the barrier as an infinitely thin shell, the assumption of plasma neutrality is effectively built into the model. In an actual UCP system the potential well arises from a layer of unpaired ions that line the exterior of the plasma. The thickness of this layer varies as a function of the ratio of UCP ions to electrons. As this ratio grows larger the ion layer becomes thicker, and in the limit of perfect neutrality this layer becomes infinitely thin. While a finite layer might modify the evaporation rate somewhat, for this article’s purpose of testing the two main assumptions of Eq. 6.6 it represents an unnecessary complication and so an infinitely thin layer is assumed.

### **6.3.2 Collisions**

The thermal equilibrium assumption also allows for direct force calculations to be approximated by a series of random Coulomb collisions. Tracked electrons are assumed to be in contact with a reservoir of Maxwellian electrons, representing the complete electron distribution in the UCP. Tracked electrons collide with the reservoir electrons via a Monte Carlo collision operator, leading to changes in the momentum and energy of the tracked electron. In a similar fashion, tracked electrons also collide with a reservoir of infinitely massive

UCP ions, where treating the ions as having infinite mass is a reasonable approximation in UCPs. By utilizing such a method, an  $O(N^2)$  process is reduced to an  $O(N)$  process, greatly reducing computation time.

Occasionally, collisions cause tracked electrons to lose sufficient energy so as to fall below the tacking threshold. When this occurs, these electrons are discarded from the simulation. To maintain detailed balance, a certain number of additional highly energetic electrons are generated at chosen time intervals. The process by which this was implemented will be discussed later in this section.

Tracked electrons move in a constant trajectory until they undergo a collision. Whether or not a collision occurs is determined randomly. In binary collisional theory, the probability of a collision occurring in a time period,  $dt$ , is  $n\sigma \langle v_d \rangle dt$ , where  $n$  is the particle density,  $\sigma$  is the collisional cross section, and  $\langle v_d \rangle$  is the average difference velocity between a tracked electron and particles in either the electron or ion distribution. For electron-electron collisions,  $\langle v_d \rangle$  can be expressed as a function of tracked electron velocity,  $v_e$ :

$$\langle v_d \rangle = \frac{k_b T \operatorname{erf}\left(v_e \sqrt{\frac{m}{2k_b T}}\right)}{mv_e} + \operatorname{erf}\left(v_e \sqrt{\frac{m}{2k_b T}}\right)v_e + \sqrt{\frac{2}{\pi}} e^{-\frac{mv_e^2}{2k_b T}} \quad (6.7)$$

For electron-ion collisions,  $\langle v_d \rangle$  is set simply equal to  $v_e$ , since UCP electron velocities are typically many orders of magnitude larger than ion velocities.

A simple method for implementing random collisions would be to repeatedly randomly determine whether or not a collision occurs in consecutive short time intervals. However, this method is computationally inefficient, requiring many random number calls per collision. Instead, the model determines the time until the next collision from the following probability distribution:

$$P(t) = n\sigma \langle v_d \rangle e^{-n\sigma \langle v_d \rangle t} dt \quad (6.8)$$

where  $P(t)$  is the probability that the next collision will occur between time  $t$  and  $t + dt$ . To generate random collision times, the above equation can be rewritten as a function of a randomly generated number:

$$t = -\frac{\ln(R)}{n\sigma \langle v_d \rangle} \quad (6.9)$$

where  $R$  is a random number between 0 and 1. For each collision, an electron-ion and electron-electron collision time are randomly generated, and the shorter time is used.

All collisions in the model are assumed to be elastic. When an elastic collision occurs, the center of mass velocity of the two colliding particles is rotated by the angle  $\chi$ , changing the momentum of each particle[16]. The angle  $\chi$  is defined by the following relationship:

$$\chi = 2\tan^{-1}\left(\frac{q_1q_2}{4\pi\epsilon_0\mu|\vec{v}_1 - \vec{v}_2|^2b}\right) \quad (6.10)$$

where  $\mu$  is the reduced mass,  $b$  is the impact parameter,  $q_1$  and  $q_2$  are the charges of the two particles, and  $\vec{v}_1$  and  $\vec{v}_2$  are the 2 particle velocities. Following the standard treatment for Coulomb collisions in a plasma, impact parameters are assumed to not exceed a maximum cutoff. For the purposes of this work, the standard  $\lambda_D$  cutoff is assumed, where  $\lambda_D$  is the Debye length.

Once a collision has been determined to have occurred the model needs to determine an appropriate deflection angle,  $\chi$ . This requires knowing the value of the impact parameter,  $b$ , which the the model randomly generates. The probability that for a given collision the impact parameter is between  $b$  and  $b + db$  is given by the following expression:

$$P(b) = \begin{cases} \frac{2b}{\lambda_D^2} & 0 < b \leq \lambda_D \\ 0 & b > \lambda_D \end{cases} \quad (6.11)$$

By integrating this probability distribution and inverting the result,  $b$  can be written as function of a random number:

$$b = \lambda_D \sqrt{R} \quad (6.12)$$

where  $R$  is again a random number between 0 and 1.

Since ions in the model are assumed to be stationary, the deflection angle,  $\chi$ , for electron-ion collisions is solely a function of this generated impact parameter. However, for electron-electron collisions  $\chi$  is also a function of the relative velocity of the two colliding electrons,  $|\vec{v}_1 - \vec{v}_2|$ . The relative velocity,  $|\vec{v}_1 - \vec{v}_2|$ , probability distribution as a function of  $\vec{v}_1, \vec{v}_2$  is as follows:

$$P(\vec{v}_1, \vec{v}_2) \propto |\vec{v}_2 - \vec{v}_1| e^{-\frac{m\vec{v}_1 \cdot \vec{v}_1}{2kT}} e^{-\frac{m\vec{v}_2 \cdot \vec{v}_2}{2kT}} d^3 v_1 d^3 v_2 \quad (6.13)$$

Since, in the context of the model,  $\vec{v}_1 = \vec{v}_e$ , Eq. 6.13 can be reduced to exclusively a  $\vec{v}_2$  probability distribution:

$$P(\vec{v}_2) \propto |\vec{v}_2 - \vec{v}_e| e^{-\frac{m\vec{v}_2 \cdot \vec{v}_2}{2kT}} d^3 v_2 \quad (6.14)$$

In principle,  $\vec{v}_2$  could be written as a function of a random number by integrating and inverting Eq. 6.14 in a manner analogous to Eq 6.8 and Eq. 6.11. However, this approach yields an unwieldy and computationally inefficient expression for  $\vec{v}_2$ , and thus an alternative procedure for generating  $\vec{v}_2$  is used.

The first step of this procedure is randomly generating the vector components of  $\vec{v}_2$  by assuming  $\vec{v}_2$  corresponds to a Maxwellian. In practice this means randomly generating values of  $v_{2x}$ ,  $v_{2y}$ , and  $v_{2z}$  by drawing from a normal distribution with a Gaussian  $\sigma$  of  $\sqrt{k_b T/m}$ . Next, these values can be inserted into Eq. 6.14 in the following manner:

$$P(\vec{v}_2) = A \sqrt{((v_{2x} - \vec{v}_e \cdot \hat{x})^2 + (v_{2y} - \vec{v}_e \cdot \hat{y})^2 + (v_{2z} - \vec{v}_e \cdot \hat{z})^2)} e^{-\frac{v_{2x}^2 + v_{2y}^2 + v_{2z}^2}{2kT}} \quad (6.15)$$

where  $A$  is a proportionality constant which is set to be sufficiently low such that  $P(\vec{v}_2)$  is always less than 1. Evaluating Eq. 6.15 calculates the relative probability that the randomly

generated vector components of  $\vec{v}_2$  would occur in a collision with an electron with velocity,  $\vec{v}_e$ . Next, another random number,  $R$ , ranging from 0 to 1, is generated, and is compared to the relative probability calculated by Eq. 6.15. If  $R$  is greater than determined value of  $P(\vec{v}_2)$ , new vector components for  $\vec{v}_2$  are generated and the process is restarted. Otherwise, if  $R$  is less than  $P(\vec{v}_2)$ , the value of  $\vec{v}_2$  is kept. This procedure was extensively tested, and was shown to accurately reproduce the appropriate  $\vec{v}_2$  probability distributions over a wide range of tracking velocities. Once a value for  $\vec{v}_2$  and the impact parameter,  $b$ , are known, the deflection angle,  $\chi$ , and thus the collision is fully determined.

### 6.3.3 Maintaining Equilibrium and the Production Function

As mentioned previously, modeling these collisions will result in electrons' energies falling below the tracking energy threshold. Relatively quickly, this would mean that no electrons above the tracking energy would remain. In steady-state, the fraction of tracked electrons should be approximately constant. Thus, there needs to be some mechanism for "creating" electrons above the tracking threshold on a regular basis to maintain a Maxwellian distribution in the absence of electron evaporation. Care needed to be taken to generate a proper Maxwellian distribution. This was accomplished in our simulation by periodically adding new tracked electrons to appear with one of a range of selected tracked velocities and at random locations. The distribution of added electrons across the range of selected velocities will be referred to as a production function, and the production function was formulated to lead to a Maxwellian steady-state distribution in model simulations.

The process for developing this production function was as follows. First, a number of simulations were run where electrons were not allowed to escape. Electrons were added into the system at regular intervals with random positions and random velocity directions, but a fixed velocity magnitude,  $v_i$ . As the simulation ran, some electrons were removed as they went below the threshold velocity. Eventually, the rate of electrons being added and removed balanced out and the system reached a steady state number of electrons, resulting in the  $i$ th

simulation having the electron velocity distribution,  $f_i(v_e)$  for tracked electrons. A linear combination of the velocity distributions,  $f_i(v_e)$ , were then fit to a Maxwellian distribution in the following manner:

$$f(v_e) = A \sum_i a_i f_i(v_e) \quad (6.16)$$

where  $f(v_e)$  is the appropriate Maxwellian distribution, the  $a_i$ 's are normalized fit coefficients, and  $A$  is a proportionality constant. Once the values of the  $a_i$ 's were found, the production function,  $F(v_e)$  could be described:

$$F(v_e) = \sum_i a_i \delta(v_e - v_i) dv \quad (6.17)$$

where the best fit  $a_i$  values represent the probability that an electron with the  $i$ th velocity magnitude will be generated. Once these values were determined, the production function could be utilized in simulations in which evaporation was included.

## 6.4 Results

A number of model simulations were run to determine the impact of different plasma parameters on the electron evaporation rate,  $\dot{N}$ . Specifically, evaporation rates were calculated as a function of plasma spatial size, plasma depth, and electron temperature. For all simulations, ion and electron densities were both held constant at  $1.35 \times 10^{13} m^{-3}$ , corresponding to a set of low-density UCP experimental parameters. In most calculations, this density corresponded to  $1.1 \times 10^5$  ions and electrons and a plasma radius of 1.248mm. Timesteps were 10 ns long, and each simulation lasted 300 time steps. Evaporation rates were extracted from averaging the number of escapes occurring in the last 200 timesteps, during which the plasma was in a steady state.

Table 6.1: Table of the resultant best fit parameters from Eq. 6.18 from a fit to model calculations. Each column in the table represents a series of model calculations of the evaporation rate as a function of  $W$  for a given constant  $T$ . The table gives the best fit parameter,  $\alpha$ , that corresponds to each of the constant  $T$  curves.

T	2K	3K	5K	7K	10K
$\alpha$	0.629	0.514	0.294	0.290	0.141

### 6.4.1 Parameter Scaling

Simulation results were used to test the veracity of scaling rules presented above. For instance, if  $T$  is held constant, Eq. 6.6 can be expressed as a functional form:

$$\dot{N} = -AW^\alpha e^{-W} \quad (6.18)$$

where  $A$  and  $\alpha$  are constants. The additional dependence on  $W$  in Eq. (13) is typically present in expression for evaporation in ultracold atomic gases. To see if this form was reasonable,  $\alpha$  and  $A$  can be treated as a fit parameters. From Eq. 6.6,  $\alpha$  would be expected to be 0.

To test whether Eq. 6.18 expresses the proper functional form, evaporation rates were calculated for a series of different  $W$  values at a constant  $T$ , and the resultant curve was fit to the functional form in Eq. 6.18. This process was then repeated for a series of different electron temperatures. The resultant  $\alpha$  parameters from these fits can be seen in Table 6.1:

These results show that  $\alpha$  is consistently greater than 0, indicating that evaporation is less strongly dependent on  $W$  than Eq. 6.6 would predict. Additionally, since  $\alpha$  increases with decreasing temperature, evaporation becomes more weakly dependent on  $W$  as  $T$  decreases. These results show that the functional form of the electron evaporation rate from Eq. 6.18 is itself incorrect, and thus evaporation cannot be modeled in such a manner. This point is further illustrated in Figure 6.2.

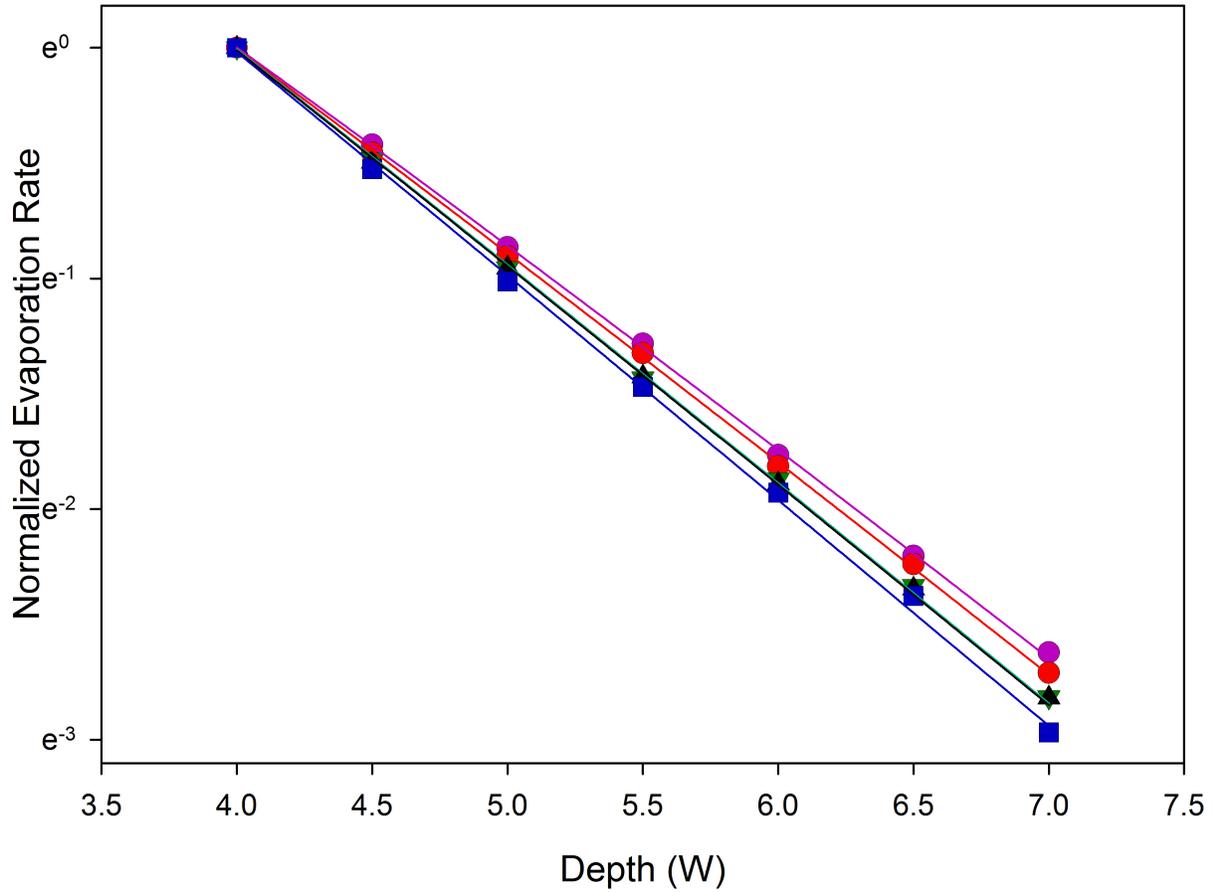


Figure 6.2: Plot of the resultant best fit curves from Eq. 6.18 being fit to model calculations. Each curve is fit to a series of model calculations which vary in  $W$  but are held at a constant  $T$ . The purple points correspond to a  $T=2\text{K}$  curve, the red points are  $T=3\text{K}$ , the green points are  $T=5\text{K}$ , the black points are  $T=7\text{K}$ , and the blue points are  $T=10\text{K}$ . The data points are model calculations of  $\dot{N}(W)/\dot{N}(W = 4)$ , and fit curves are represented by the corresponding curves shown. The plot shows that the curves scale differently. This suggests that Eq. 6.18 does not accurately describe  $\dot{N}$ . Note that the y-axis is on a log scale.

Table 6.2: Table of the resultant best fit parameters from Eq. 6.19 being fit to model calculations. Each  $W$  in the table represents a series of model calculations which vary in  $T$  but are held at a constant  $W$ , and  $k_1$   $k_2$  and  $k_3$  are the best fit parameters corresponding to this constant  $W$  curve.

$W$	4	4.5	5	5.5	6	6.5	7
$k_1$	0.522	0.512	0.538	0.484	0.499	0.505	0.590
$k_2$	0.866	0.911	0.943	0.941	0.932	0.947	1.055
$k_3$	4.041	4.823	4.395	6.646	5.653	5.690	4.130

Simulation results were also used to test the temperature dependence of evaporation at constant depth to temperature ratio (i.e. constant  $W$ ). At a constant  $W$ , Eq. 6.18 reduces to the following functional form:

$$\dot{N} = k_1 T^{-k_2} \ln(-k_3 T^{\frac{3}{2}}) \quad (6.19)$$

where  $k_1, k_2$ , and  $k_3$  are constants. Since I am only testing scaling laws at this point, these constants can be treated as fitting parameters. By using a method analogous to the constant  $T$  case, the  $T^{-\frac{3}{2}}$  scaling, implied by Eqs. 6.6 and 6.19, can again be tested. This leads to the results shown in Table 6.2:

These results are inconsistent with a  $T^{-\frac{3}{2}}$  scaling given the value of  $k_2$ . Additionally, Eq. (14) suggests that  $k_2$  and  $k_3$  should stay constant as a function of  $W$ , which is also contradicted by the our calculated results. These results show that the functional form of the electron evaporation rate from Eq. 6.19 is incorrect for the given experimental conditions. Examples of these fits can be seen in Figure 6.3.

The failure of Eq. 6.6 to properly predict the proper parameter scaling of the evaporation rate suggests that at least one of the two underlying assumptions of the theory is incorrect for typical UCP experimental conditions. The first of these assumptions states that when a collision excites an electron above the barrier, it is immediately considered to have escaped. However, if the plasma size is larger than the effective mean free path, it becomes likely

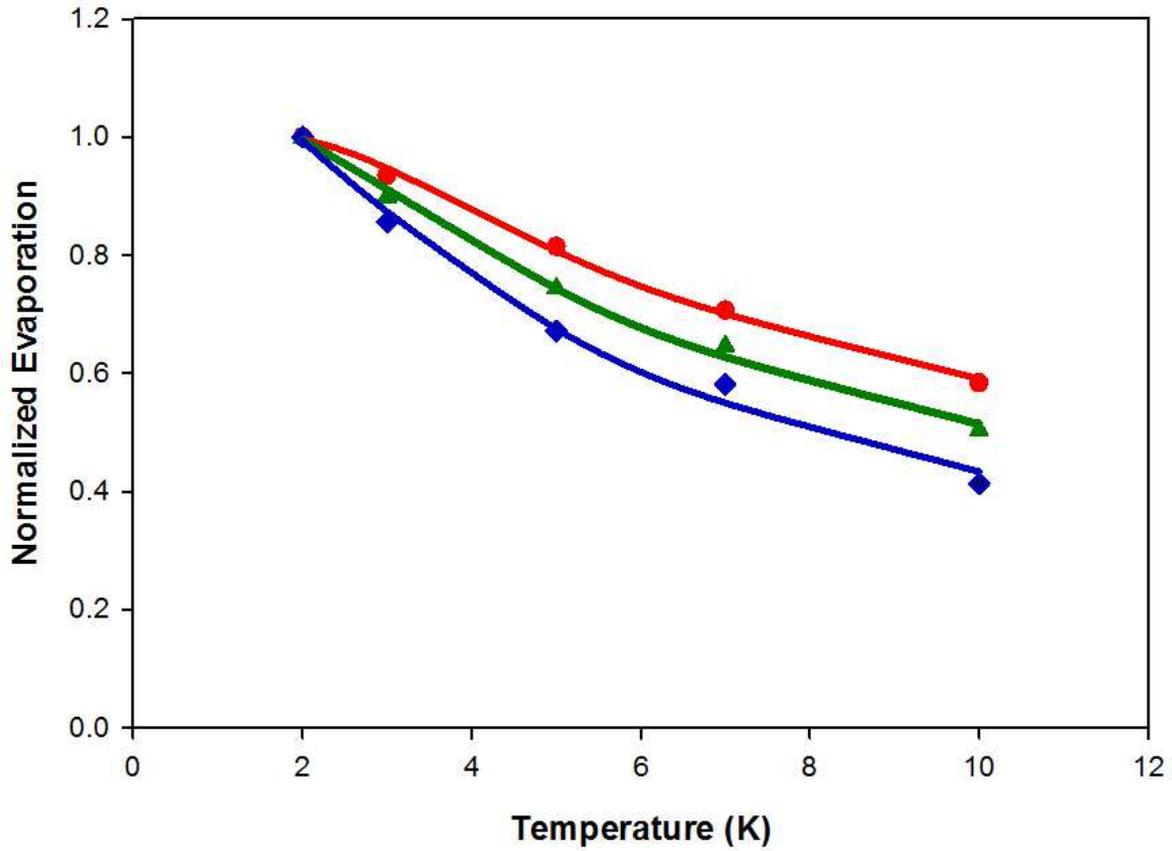


Figure 6.3: Plot of the resultant best fit curves from Eq. 6.19 being fit to model calculations. Each curve is fit to a series of model calculations which vary in  $T$  but are held at a constant  $W$ . The red points correspond to a  $W=4$  curve, the green points correspond to a  $W=5$  curve, and the blue points correspond to a  $W=7$  curve. The data points represent model calculation of  $\dot{N}/\dot{N}(T=2)$ , and fit curves are represented by the corresponding curve. The plot shows that not all of the curves scale identically. This suggests that Eq. 6.19 does not accurately describe  $\dot{N}$ .

that a secondary collision de-excites that same electron back below the barrier, preventing the electron from escaping. The only exception is if the electron is at the edge of the UCP. Since it is unclear what the effective mean free path is in the context of evaporation, it is questionable whether or not this assumption is valid for typical UCP collisions.

### 6.4.2 Testing Evaporation Assumptions

To characterize the impact that the mean free path will have on evaporation, it was useful to consider two limiting cases. In the case where the mean free path is much larger than the size of the plasma, the absolute evaporation should scale linearly with the electron number without any dependence on the plasma radius,  $R$ , assuming a constant electron density. Conversely, if the plasma is much larger than the mean free path, electrons in close proximity to the plasma edge should contribute much more heavily to the evaporation rate than other electrons. Presumably, in the limit of an infinitely large UCP, evaporation should scale with UCP surface area. However, if the electron density is held constant, the electron number scales with the UCP volume, which leads to the evaporation rate per electron,  $\dot{N}/N$ , scaling as  $1/R$  in this surface-dominated limit.

To determine where plasmas with the parameters used in the simulation were between these two limits, I investigated how  $\dot{N}/N$  varied with plasma spatial size,  $R$ . Simulations were run varying the UCP size and electron number in a manner consistent with a constant density, and a per electron evaporation rate vs UCP size curve was generated. The results of these calculations, for one set of conditions, can be seen in Figure 6.4.

The figure shows that per particle evaporation scaled roughly as  $1/R$  at larger values of  $R$ , which is consistent with the typical mean free path being much smaller than the plasma spatial size. At smaller plasma sizes, this scaling became shallower. Constant evaporation scaling with  $R$ , the scaling implicitly assumed in Eq. 6.6, was not observed over the investigated range of parameters. These results show that the underlying assumption in Eq 6.6 about the mean free path is incorrect. Furthermore, for typical simulation plasma param-

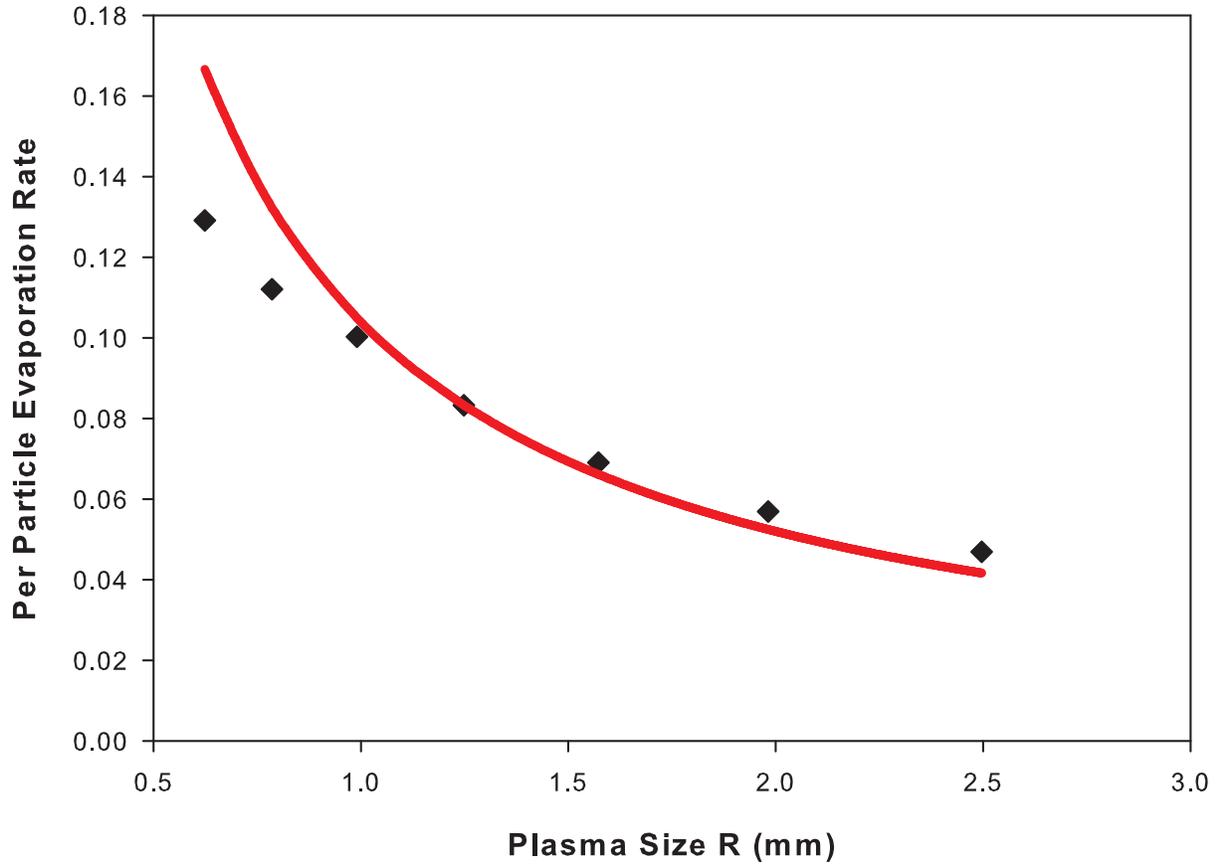


Figure 6.4: Plot of  $\dot{N}/N$  as a function of plasma size,  $R$ , for a  $T=10\text{K}$  and  $W=4$  plasma. The black data points are the model calculations, and the red line is a  $1/R$  curve. All calculations are made at a constant electron and ion density of  $1.35 \times 10^{13} m^{-3}$  which corresponds to electron and ions numbers ranging from  $1.375 \times 10^4$  to  $8.8 \times 10^5$ . For larger values of  $R$ , model calculations scale roughly as  $1/R$  indicating that the plasma size is much greater than the typical mean free path for these conditions.

eters, the evaporation rate is consistent with mean free path effects being dominant. It is therefore desirable to quantify the magnitude of these effects.

To do this, it is necessary to introduce the concept of an effective electron evaporation source density,  $n_{eff}$ . In general, electrons that are closer to the edge are more likely to escape the plasma. The purpose of the effective density is to weight these more-likely-to-leave electrons more highly than their counterparts near the plasma center, and to quantify how this weighting changed as a function of  $R$ . To do this the following simple approximate model was utilized :

$$n_{eff} = ne^{-\frac{R-r}{c}} \quad (6.20)$$

where  $n$  is the electron density,  $R$  is the plasma size,  $r$  is the standard radial coordinate, and  $c$  is an effective evaporation skin depth quantifying the degree of which mean free path effects impact evaporation. Utilizing this approximate description, it is assumed that  $\dot{N}$  takes the following form:

$$\dot{N} = -\kappa N_{eff} = -4\pi\kappa \int_0^R r^2 n_{eff} dr \quad (6.21)$$

where  $\kappa$  is a proportionality constant, and  $N_{eff}$  is the effective number derived from  $n_{eff}$ . Combining Eq. (15) and Eq. (16) and evaluating the integral yields the following expression:

$$\dot{N} = -4\pi n \kappa c (R^2 - 2Rc + 2c^2 - 2c^2 e^{-R/c}) \quad (6.22)$$

By treating  $\kappa$  and  $c$  as fit parameters, Eq. 6.22 was fit to results similar to those in Figure 6.4, and an effective evaporation skin depth was extracted.

The resultant fit parameters,  $\kappa$  and  $c$ , can give insight about the applicability of the assumed analytical evaporation functional form, even in the absence of mean free path considerations. In the limit of  $R \ll c$ , the model predicts an evaporation rate of  $\kappa$ . If the functional form of Eq. 6.6 is correct, its predicted evaporation rate should match our

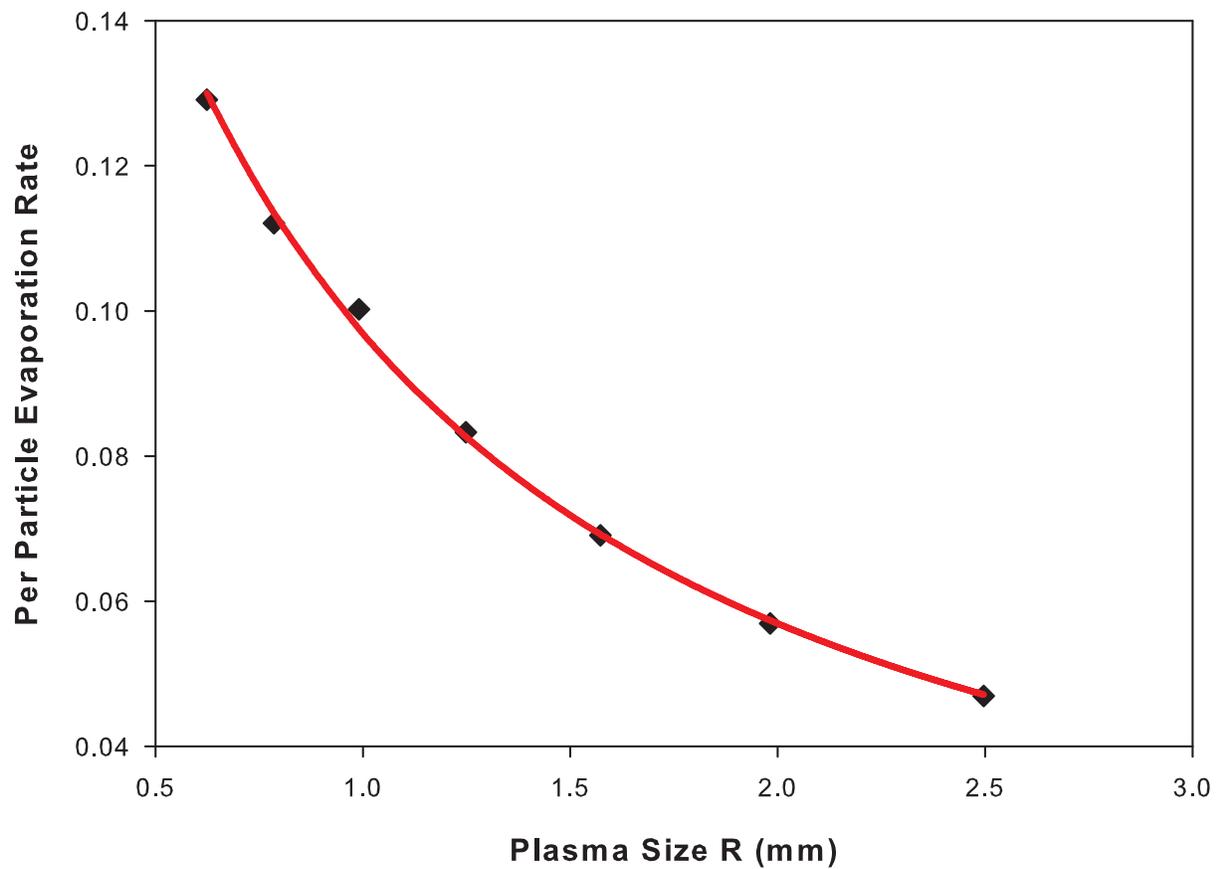


Figure 6.5: The plot shows Eq. 6.22 being fit to model calculations of  $\dot{N}/N$  as a function of  $R$ . The black data points are the model calculations for a  $T=10\text{K}$  and  $W=4$  plasma, and the red line is the resultant fit function. The plot shows that Eq. 6.22 describes the model results well.

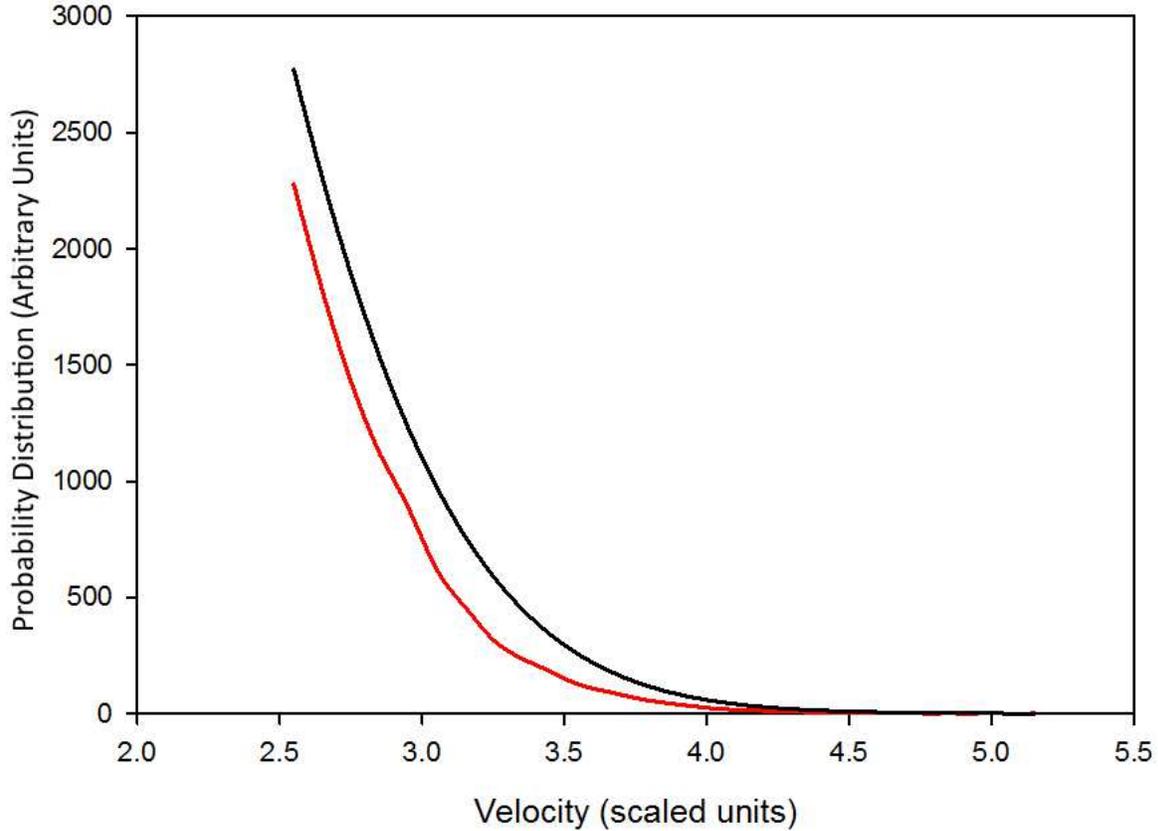


Figure 6.6: The plot shows the discrepancy between a Maxwellian velocity distribution and the equilibrium velocity distribution for electrons in a finite potential well. The black curve represents an ideal Maxwellian, and the red curve is the model calculated velocity distribution for a  $W=5$ ,  $T=10\text{K}$  plasma.

calculated  $\kappa$ . This was not the case, however, as I observed  $\kappa$  to be 3-7 times smaller than would be implied by Eq. 6.6. This was, at least in part, due to the electron velocity distribution. Eq. 6.6 assumes a Maxwellian, but a collection of electrons in a potential well will not form a Maxwellian distribution in steady state. At the low electron energies, these distributions are roughly the same, but the Maxwellian will significantly overestimate the electrons near the barrier. This leads to an overestimation of  $dN/dt$  in Eq. 6.6, and subsequently an overestimation in the evaporation rate.

The mean free path fit parameters were also used to test of the scaling of the  $dv/dt$  component of the evaporation functional form in the limit of  $R \ll c$ . We compared our

calculated values of  $c$  to an estimate involving the form of  $dv/dt$  in Eq. 6.5 above. In the limit of small velocity changes, for a given initial velocity,  $v_0$ , above the barrier, the characteristic time,  $\tau$ , it takes to decay to the barrier velocity,  $v_U$ , can be defined:

$$\tau(v_0) = \frac{v_0 - v_U}{(dv/dt)|_{v_0}} \quad (6.23)$$

This implies an approximate velocity dependent mean free path of  $c(v) = v\tau(v)$ . By integrating over the velocities of all of the electrons above the barrier an average  $c$  can be calculated. For typical experimental conditions, such a calculation resulted in a  $c$  on the order of about a mm, an order of magnitude larger than suggested by the results from the prior mean free path fit. The results of this calculation suggests that the average electron velocity slowing,  $dv/dt$ , is not the relevant rate with regards to evaporation. This is likely due to  $dv/dt$  being an average quantity. The evaporation rate is the sum of discrete electron escape events, and it is not immediately obvious that an average rate would properly account for the relevant physics. In addition, the average escape path for model electrons can often be much longer than  $R$ . A typical escaping electron will undergo a number of deflecting collisions over the course of its escape, which presumably effectively lengthen the electron's escape path.

Unfortunately, the previously developed analytical expressions seem to not be applicable to ultracold plasmas, at least under the conditions studied. Eq. 6.6 incorrectly predicted electron evaporation rates and did not scale correctly. While analytical expressions that accurately predict the electron evaporation rate could presumably be developed, such expressions do not currently exist. Thus, a numerical model, such as the one developed in this chapter, will be needed to properly calculate the electron evaporation rate in ultracold plasma systems.

## 6.5 Conclusion

I have developed a model that calculates the rate of evaporation from an ultracold plasma. Model results were compared to previously developed analytical expressions for evaporation. These expressions proved inconsistent with our results as the model scaled with plasma parameters differently than the simple evaporation expressions predicted. Furthermore, it was demonstrated that this discrepancy can, at least partially, be explained by the finite size of the plasmas examined in this work, and that absence of size considerations is a limitation of these previously developed expressions. This work demonstrates that such simple scaling rules are not accurate when used to calculate evaporation in UCPs, and that a model like the one developed in this work is needed.

# References

- [1] W. Ketterle and N. V. Druten, *Advances in Atomic, Molecular, Optical Physics*. **37**, 181 (1996).
- [2] G. B. Andresen et al., *Phys. Rev. Lett.* **105**, 013003 (2010).
- [3] T. Wilson, W.-T. Chen, and J. Roberts, *Phys. Plasmas*. **20**, 073503 (2013).
- [4] C. Witte and J. L. Roberts, *Physics of Plasmas* **21**, 103513 (2014).
- [5] F. Robicheaux and J. D. Hanson, *Phys. Rev. Lett.* **88**, 055002 (2002).
- [6] T. C. Killian et al., *J. Phys. B: At. Mol. Opt. Phys.* **38**, S351 (2005).
- [7] P. Gupta et al., *Phys. Rev. Lett.* **99**, 075005 (2007).
- [8] J. L. Roberts, C. D. Fertig, M. J. Lim, and S. L. Rolston, *Phys. Rev. Lett.* **92**, 253003 (2004).
- [9] R. S. Fletcher, X. L. Zhang, and S. L. Rolston, *Phys. Rev. Lett.* **99**, 145001 (2007).
- [10] T. C. Killian et al., *Phys. Rev. Lett.* **86**, 3759 (2001).
- [11] J. C. Stewart and K. D. Pyatt, *Astrophys. J.* **144**, 1203 (1966).
- [12] M. Lyon and S. D. Bergeson, *J. Phys. B: At. Mol. Opt. Phys.* **44**, 184014 (2011).
- [13] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell, *Science* **269**, 198 (1995).
- [14] K. B. Davis et al., *Phys. Rev. Lett.* **75**, 3969 (1995).
- [15] M. D. Barrett, J. A. Sauer, and M. S. Chapman, *Phys. Rev. Lett.* **87** (2001).
- [16] P. Bellan, *Fundamentals of Plasma Physics*, Cambridge University Press, 2006.

- [17] G. B. Andresen, *Evaporative Cooling of Antiprotons and Efforts to Trap Antihydrogen*, PhD thesis, Aarhus University, 2010.

# Chapter 7

## Non-Neutral Expansion

Over the course of their lifetimes, ultracold plasma systems undergo two distinct phases. These are the formation phase, where the UCP cloud evolves toward an equilibrium distribution shortly after it is created, and the expansion phase, where the net positive space charge in the plasma and the electron thermal pressure leads to the expansion of the UCP ions. Typically, the expansion phase has been modeled by assuming an equal number of electrons and ions in the system[1, 2, 3, 4, 5]. While UCP systems are uniformly non-neutral, many of these systems have sufficiently high densities where  $N_i - N_e \ll N_i$ [5, 6]. Thus, for these high density systems, the neutral plasma approximation is often sufficient to accurately describe the UCP expansion phase. However, for lower density UCPs, like the UCPs in our lab, the number of electrons in the system can be significantly lower than the number of ions. Furthermore, low density UCP systems typically have higher rates of evaporation, which in turn also leads to higher levels of non-neutrality. Thus, in the low density case, it is unclear whether the neutrality approximation breaks down.

At first glance, it would seem that any increase in charge imbalance would result in an increase in the total UCP expansion rate due to an increase in electrostatic forces. While this is almost certainly true for ions on the UCP's exterior, it remains unclear whether the UCP interior would be subjected to additional Coulomb forces. Simple approximations of the UCP charge density distribution suggest that the interior charge density of the UCP should remain unaltered by moderate non-neutrality. Gauss' Law and the spherical symmetry of UCPs imply no additional electric field in the interior of the UCP from an increase in the number of unpaired ions in the exterior region. Moreover, even in the absence of symmetry, any electric fields from the exterior ions would be screened by the electrons in the UCP.

However, there are two possible ways that a change in charge neutrality could affect not only the total UCP expansion rate but the expansion rate of the interior region of the UCP as well. Changes in the exterior charge distribution will alter the expansion rate of that part of the UCP and that in turn will influence the expansion rate at the boundary of the interior region. A change in the rate of expansion at the interior region boundary could lead to an overall change in the interior region expansion rate as a whole. Furthermore, a change in charge imbalance will alter the UCP potential depth. This in turn influences the confined electron spatial density distribution, possibly affecting interior expansion.

Given the fact that the UCPs in our lab are low-density, quantifying non-neutrality effects on UCP expansion is necessary to determine if such effects need to be included in for an accurate description of UCP parameters. To quantify these effects, I developed a numerical model to calculate this expansion. This chapter will discuss the various components of this model, and how they all fit together. This chapter will also discuss the results from model simulations, and will discuss the implications of these results. Additionally, the chapter will give an overview of the traditional neutral expansion model, and compare some the predictions that it makes with results obtained from the non-neutral model.

## 7.1 Simple Model for Neutral Plasma Expansion

The simplest description of ultracold plasma ion dynamics comes from the neutral expansion model seen in reference [1]. In this model the UCP ion density is treated as a single fluid, and the total number of UCP ions,  $N_i$ , is assumed to be equal with the total number of electrons,  $N_e$ . While the model is quite simple the it is able to provide a number of key insights into the expansion dynamics of UCP systems. This section will give an overview of the formulation of this model, and discuss some of the insights that it gives into dynamics of UCPs.

The model is composed of three equations. The first of these equations is the mass continuity equation for the ions

$$\frac{\partial n_i(r, t)}{\partial t} + \frac{1}{r^2} \frac{\partial(r^2 v_i(r, t) n_i(r, t))}{\partial r} = 0 \quad (7.1)$$

where  $v_i(r, t)$  is the ion velocity, and  $n_i(r, t)$  is the ion density, which is assumed to be spherically symmetric. The second equation is the momentum continuity equation in the limit of zero ion pressure

$$\frac{\partial v_i(r, t)}{\partial t} + v_i(r, t) \frac{\partial(v_i(r, t))}{\partial r} = a(r, t) \quad (7.2)$$

where  $a(r, t)$  is the ion acceleration, and the ion velocity  $\vec{v}_i = v_i(r, t)\hat{r}$ . Finally, the third equation defines the acceleration

$$a(r, t) = \frac{q^2}{\epsilon_0 m_i r^2} \int_0^r (n_i(r'), t) - n_e(r', t) r'^2 dr' \quad (7.3)$$

where  $\epsilon_0$  is the permittivity of free space,  $m_i$  is the ion mass,  $q$  is the charge of the ions,  $n_e$  is the electron density.

By making a number of assumptions and variable changes, these three model equations can be rewritten. First, it is assumed that both  $n_i$  and  $n_e$  are Gaussian distributions. Next, the electrons are considered to be in thermal equilibrium, i.e. the electron velocity distributions are Maxwellian. Finally, near neutrality is assumed, i.e.  $N_i \gg N_i - N_e$ . From these assumptions, the acceleration of UCP ions can be calculated (see section 3.3.3). This expression for acceleration is as follows

$$a(r, t) = 2\beta(t)k_b T_e r / m_i \quad (7.4)$$

where  $k_b$  is the Boltzmann constant,  $T_e$  is the electron temperature, and  $\beta(t)$  is defined as  $1/2\sigma^2$  where  $\sigma$  is the standard Gaussian spatial size. In other words, the ion density distribution as a function of the distance  $r$  from the center of the UCP is  $n = n_0 \exp(-\beta r^2)$  where  $n_0$  is the peak density. With the following change of variables,  $v_i(r, t) = \gamma(t)r$ , Eq. 7.1 can be rewritten as,

$$\frac{d\beta}{dt} = -2\gamma(t)\beta(t) \quad (7.5)$$

and Eq. 7.2 can be rewritten as the following.

$$\frac{d\gamma}{dt} + \gamma^2 = 2k_b T_e(t)\beta(t)/m_i \quad (7.6)$$

By assuming that the ions are initially at rest, and by using conservation of energy, a third equation can be developed. This equation is as follows

$$\frac{3}{2}k_b T_e(0) = \frac{3}{2}k_b T_e(t) + \frac{3}{4}m_i \frac{\gamma^2(t)}{\beta} \quad (7.7)$$

where the left hand side of the equation corresponds to the initial thermal energy of a UCP electron, the first term on the right hand of the equation is the time dependent thermal energy of a UCP electron, and the second term on the right hand side is the time dependent average kinetic energy of a UCP ion.

Simultaneously solving equations 7.5-7.7 yields two main insights into UCP dynamics. The first of these is that the functional form of the ion expansion, which is the described by the following expression

$$\sigma(t) = \sqrt{\sigma_0^2 + \frac{k_b T_e(0)}{m_i} t^2} \quad (7.8)$$

where  $\sigma_0$  is defined as  $\sigma_0 = \sigma(t = 0)$ . As can be seen from Eq. 7.8, the spatial extent of the ion cloud expands over time. Plots of these expansion curves for multiple different temperature conditions can be seen in figure 7.1. The other main insight of this model is quantifying temperature evolution of the UCP electron distribution. This evolution is

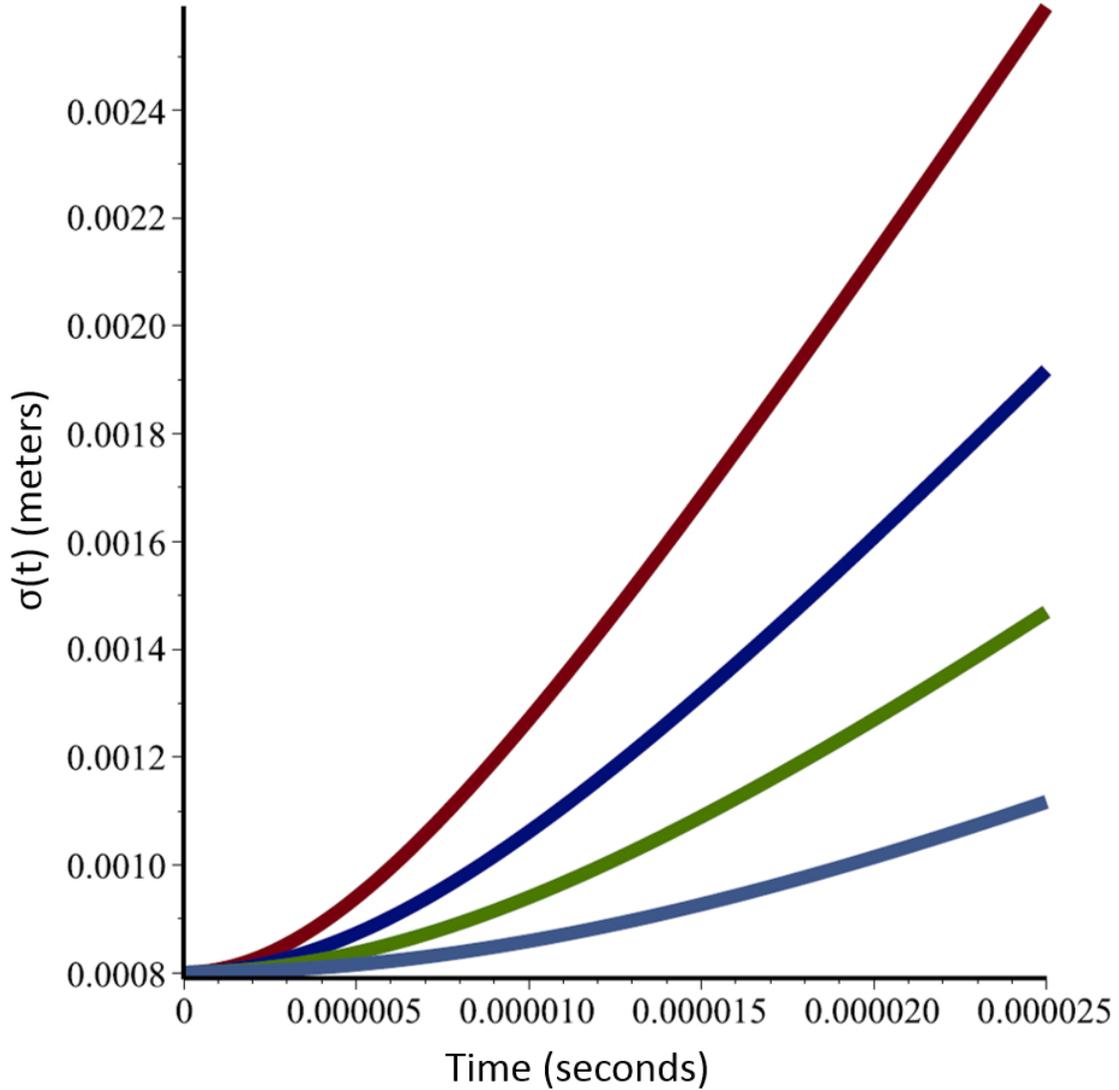


Figure 7.1: Graph of ion expansion curves for a number of different initial temperatures,  $T_0$ . The light blue curve has a initial temperature of  $T_0 = 10k$ , the green curve has  $T_0 = 25k$ , the dark blue curve has  $T_0 = 50k$ , and the red curve has  $T_0 = 100k$ . All curves have an size of  $\sigma_0 = 800\mu ms$ , and the ion mass of a  $^{85}\text{Rb}$  ion. As can be seen in the figure, the ion expansion is strongly dependent of the electron temperature. Note that the y-axis does not begin at zero.

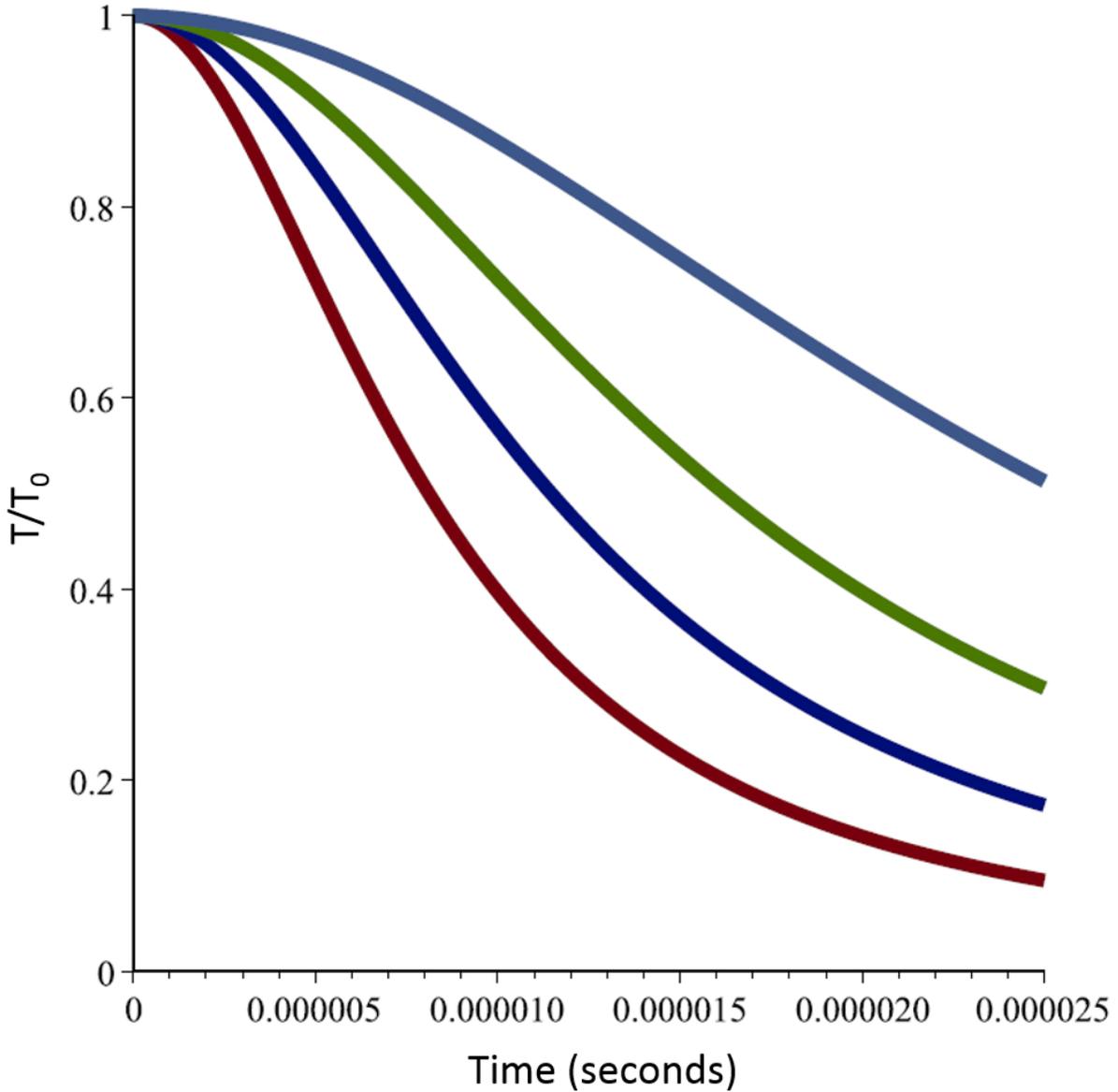


Figure 7.2: Electron cooling curves for a number of different initial temperatures,  $T_0$ . The light blue curve has a initial temperature of  $T_0 = 10k$ , the green curve has  $T_0 = 25k$ , the dark blue curve has  $T_0 = 50k$ , and the red curve has  $T_0 = 100k$ . All curves have an initial size of  $\sigma_0 = 800\mu ms$ , and the ion mass of a  $^{85}\text{Rb}$  ion. As can be seen in the figure, the electron cooling rate is strongly dependent of the initial electron temperature.

described as by the following expression.

$$T_e(t) = \frac{T_0\sigma_0^2}{\sigma(t)^2} = \frac{T_0\sigma_0^2}{\sigma_0^2 + \frac{k_b T_e(0)}{m_i} t^2} \quad (7.9)$$

As can be seen from Eq. 7.9, and can be discerned from simple conservation of energy considerations, the model shows that the electron cloud will cool over time. Furthermore, eq 7.9 shows that the quantity  $T_e\sigma^2$  remains invariant with respect to time. A graph of electron cooling curves, for different initial electron temperatures, can be seen in figure 7.2.

## 7.2 Model Components

Unlike the neutral description of the ion expansion, the ion expansion in the non-neutral expansion model is not expressed as a series of analytical equations. Instead, the non-neutral model treats the ions as particles, and integrates their motion over time, in a manner similar to molecular dynamics simulations. To integrate the motion of the ions, a number of different components have to be included in the model. This section will give a brief overview of how the model works, and will describe how the different model components operate and discuss how they fit together.

### 7.2.1 Overview of the model

By treating UCP ions as macro-particles, the non-neutral expansion model integrates the motion of the ions over the course of their expansion. Additionally, the model also calculates the time evolution of the electron spatial distribution as well as the electron temperature. The model is limited to single dimension as spherical symmetry is assumed for both the ion and the electron spatial distribution. Energy is assumed to remain constant in the system, and the total number of particles remains fixed (i.e. evaporation is not included in the model). Different UCP initial parameters can be altered (e.g. electron temperature, UCP

size and density, neutrality) to determine their influence on the UCP expansion. All of the calculations that were made with this model were done in the absence of any external fields.

At the heart of the non-neutral expansion lies the King model of the electron distribution. This model distribution is described in more detail in the subsection below. In general, this model is used to describe the equilibrium electron density distribution in the presence of an arbitrary ion potential. In the non-neutral model, the King model is used in conjunction with the ion distribution function and the electron temperature to self consistently solve for the electron density distribution, as well as the electrostatic potential of the system. Once this potential is known, it is trivial to compute local electric fields. The expansion of the ions can thus be modeled via computing these electric fields and their effect on UCP ions.

Finally, energy conservation is used to calculate the time evolution of the electron temperature. As the ions expand, they pick up additional kinetic energy, and the potential energy present in the system's electric field decreases. However, these two quantities do not completely offset. Thus, by assuming energy is conserved, the change in the electron temperature can be calculated by computing the difference between these two quantities.

### **7.2.2 King Model of the Electron**

The dynamics of ultracold plasmas bear a striking resemblance to the dynamics of globular cluster star systems. In globular clusters, stars collide via the interaction of a  $1/r$  potential, binary star formation occurs in a process analogous to three-body recombination, and high energy stars escape from the system in an evaporative process. Analytical expressions for the equilibrium distributions of stars in a globular clusters were originally developed in the 60's[7]. These expressions, henceforth known as the King model, have more recently been adapted to ultracold plasma systems[8]. From this adaptation a solution for the electron equilibrium phase space distribution can be found. These solutions have been validated by more sophisticated Monte Carlo simulations.

The King model of the electron offers a number of advantages as compared to other methods of modeling electrons in ultracold plasmas. By solving a single second-order differential equation the electron equilibrium distribution is acquired, skipping over equilibration dynamics required in molecular dynamics simulations at the price of assuming that the electrons are in thermal equilibrium with themselves at all times. The model also has the flexibility to incorporate arbitrary ion distributions. While a Gaussian ion distribution has been traditionally used, minimal straightforward changes can enable the use of an ion distribution of choice. Finally, the model also allows for arbitrary amounts of charge imbalance, which is clearly important to any investigation into the effects of non-neutrality. However, there are some downsides to the model as well. The most notable of these is that the model does not lend itself to incorporating external electric fields, which are present in all of the experiments in our lab.

Perhaps the simplest way to think about the King model of the electron is a Maxwellian with a energy cutoff. This energy cutoff is introduced into the model in part by truncating the electron spatial extent at a value of  $r_0$ . The potential energy of an electron at  $r_0$ , is defined as  $E_0 = -q\phi(r_0)$ , where  $q$  is the fundamental charge constant and  $\phi$  is the electrostatic potential. This leads to an energy cutoff, characterized by  $E_0$ , being introduced into the electron charge density in the following expression:

$$\rho_e(r) = A \int_0^{v_e} v^2 dv (e^{-(E(r,v)-E_0)/k_b T} - 1) \quad (7.10)$$

where  $k_b$  is the Boltzmann constant,  $v_e$  is the cutoff velocity,  $E(r, v)$  is the energy of an electron as a function of  $v$  and  $r$ ,  $A$  is a normalization constant, and  $T$  is a parameter that is not, but is related to, the electron temperature. Figure 7.3 illustrates the impact of  $E_0$  on the electron energy distribution.

From Gauss' law, we know that:

$$\nabla^2 \phi(r) = \rho_i / \epsilon_0 + \rho_e / \epsilon_0 \quad (7.11)$$

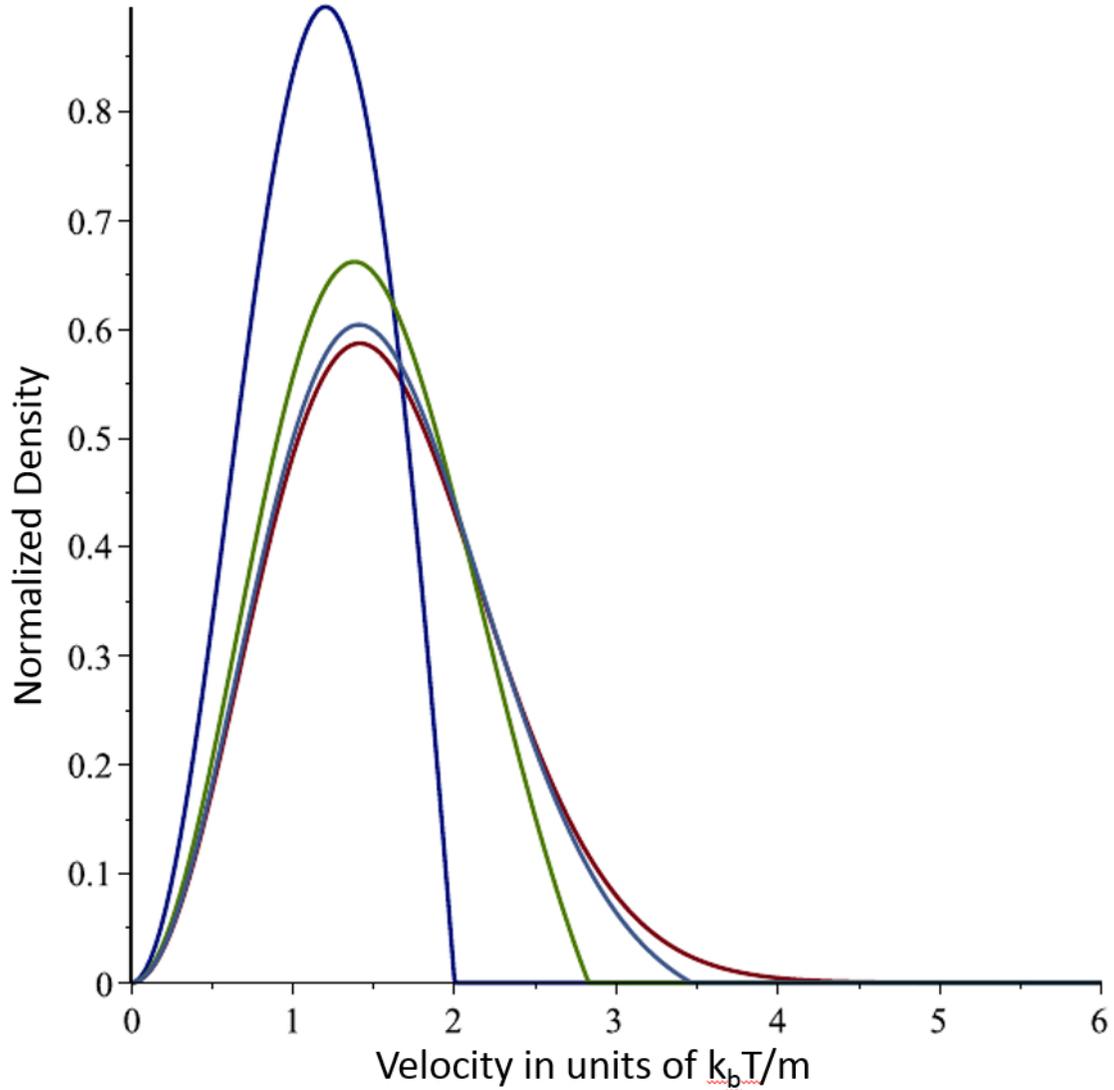


Figure 7.3: Graph of normalized electron velocity distributions of a Maxwellian and King distributions with different energy cutoffs. The red curve corresponds a Maxwellian, the dark blue curve represents a King distribution with an energy cutoff of  $E_0 = 2k_b T_e$ , the green curve has a cutoff of  $E_0 = 4k_b T_e$ , and the dark blue curve has a cutoff of  $E_0 = 6k_b T_e$ . As can be seen in graph, as the energy cutoff  $E_0$  increases, the King distribution begins to better approximate a Maxwellian.

where  $\rho_i$  is the ion charge density and  $\epsilon_0$  is the permittivity fo free space. To simplify this equation, the following scaled potential can be used:

$$W(R) = \frac{q(\phi(R) - \phi(R_0))}{k_B T} \quad (7.12)$$

where  $R$  is defined as  $r/\sigma_0$  and  $\sigma_0$  is the initial Gaussian spatial extent of the ion distribution. This allows for Eq. 7.11 to be rewritten as:

$$\frac{1}{R} \frac{d^2}{dR^2} RW(R) = -\frac{N_i}{N_0} (\Sigma_i(R) - \alpha \Sigma(W(R))) \quad (7.13)$$

where  $N_i$  is the number of ions, and the scaling constant,  $N_0$ , is defined as  $N_0 = 4\pi\epsilon_0 k_B T \sigma_0 / q^2$ . The  $\Sigma$  and  $\Sigma_i(R)$  terms are proportional to the electron and ion charge density, respectively.  $\Sigma_i$  is defined by the following the equation:

$$\Sigma_i(R) = 4\pi\sigma_0^3 n_i(R) / N_i \quad (7.14)$$

where  $n_i(R)$  is the unscaled ion density and  $\Sigma(W)$  is defined as:

$$\Sigma(W) = \begin{cases} \sqrt{\frac{\pi}{2}} e^W \text{erf}(\sqrt{W}) - (\frac{2}{3}W + 1)\sqrt{2W} & : W > 0 \\ 0 & : W \leq 0 \end{cases} \quad (7.15)$$

The  $\alpha$  in eq 7.13 is a normalization constant defined as:

$$\alpha = \frac{N_e}{N_i} \frac{1}{\int_0^{R_0} R^2 \Sigma(W(R)) dR} \quad (7.16)$$

where  $N_e$  is the total number of electrons.

Since the King distribution has an energy minimum at  $R = 0$ ,  $W'(0) = 0$  is the first boundary condition for the system. The second boundary condition,  $W(R_0) = 0$ , is true by definition, as illustrated in eq 7.12. By applying these boundary conditions, and by

simultaneously solving equations eq 7.13 - eq 7.16, it is possible to obtain a solution for  $W(R)$  for all  $R \leq R_0$ .

### 7.2.3 Two Point Boundary Value Problems

The King model of the electron does not have an analytical solution, necessitating the use of numerical integration techniques. As previously stated, the King model has two distinct boundary conditions,  $W(R_0) = 0$  and  $W'(0) = 0$ . This prevents the use of many simple ODE integrators, such as Runge Kutta, which are only designed for a single initial boundary point. Thus a more sophisticated method is required. For a two point boundary value problem, such as this one, typically either the shooting method or the relaxation method are used[9].

To understand how the shooting method works consider a simple 2nd order differential equation:

$$\frac{d^2y}{dx^2} = f(x, y, y') \quad (7.17)$$

with boundary conditions  $y(a) = Ya$  and  $y(b) = Yb$ . The shooting method starts off by solving an initial value problem at  $y(a)$  and  $y'(a)$ . However,  $y'(a)$  is not known, and thus is randomly guessed. Clearly, this does not lead to the correct value for  $y(b)$ , and thus a new value of  $y'(a)$  is guessed, but this time the previous value of  $y(b)$  is factored into the new guess. This process continues to be iterated until the value of  $y(b)$  is within some specified range of  $Yb$ .

The most common relaxation method is the finite difference method (FDM). This method reduces a system of differential equations into a system of algebraic equations. To understand how this works, lets again consider the simple 2nd order differential equation in Eq. 7.17. First, FDM establishes a mesh of  $N$  evenly spaced mesh points over the range of interest. In the case of our example, this range is  $x = a \rightarrow b$ , which leads to the mesh points have a spacing of  $\Delta x = \frac{b-a}{N-1}$ . Next, derivative quantities are rewritten as finite differences. One

possible implementation of this is as follows:

$$y'_i = \frac{y_{i+1} - y_i}{\Delta x} \quad (7.18)$$

$$y''_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} \quad (7.19)$$

where  $i$  represents the  $i$ th mesh point. This allows for eq 7.17 to be rewritten as:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} = f(x_i, y_i, y_{i+1}) \quad (7.20)$$

This equation can be applied to all interior mesh points, yielding  $N - 2$  equations. Since  $y_0$  and  $y_{N-1}$  are known from boundary conditions, there are also  $N - 2$  unknown parameters. This makes the system of equations solvable, and thus allowing for a solution to eq 7.17.

For the non-neutral expansion model I opted to utilize the finite difference method. By using this method, calculating the initial electrostatic potential function was relatively computationally expensive. However, a single timestep would only result in a small amount of ion expansion. This means that the ion density would remain relatively constant, and thus the previous electrostatic potential of the system would be an excellent guess for the new potential. For this reason, subsequent calls to the finite difference method would be relatively computationally inexpensive. For this reason, it was assumed that this method would be faster than the shooting method in the limit of many timesteps.

## 7.2.4 Ion Expansion

The ions in the UCP are treated as zero temperature particles and assumed to be initially distributed with a Gaussian density distribution described by

$$f(r) = \frac{N_i}{(2\pi\sigma_0^2)^{3/2}} e^{-r^2/2\sigma_0^2} \quad (7.21)$$

where  $N_i$  is the number of ions,  $\sigma_0$  is the Gaussian spatial scale of the ions at time  $t = 0$ , and  $r$  is the distance from the UCP center. To track the change in the ion density distribution with time, the ions are assigned to a series of thin concentric spheres called shells. Since Gaussian distributions have an infinite spatial extent, a truncation point for the distribution had to be established. For this reason, the shells were extended over a distance of  $5\sigma_0$ , after which the distribution is truncated. A sufficiently large number of shells was needed to avoid sharp discontinuities in the ion distribution. For this reason, the model would typically use 10,000 shells. This, combined with the  $5\sigma_0$  spatial extent of the ions leads to shells having a width of  $\Delta r = \sigma_0/2000$ . This in turn leads to the following expression for the total charge contained in the  $j$ th ion shell

$$q_j = 4\pi q \int_{(j-1)\Delta r}^{j\Delta r} r^2 f(r) dr \quad (7.22)$$

where the center most ion shell is defined as  $j = 1$ . Finally, the model defined the position of the  $j$ th shell,  $r_j$ , as the midpoint of the shell, such that, at initialization,  $r_j = (j - 1/2)\Delta r$ .

By self consistently solving eq 7.13, the effective forces present on the ion shells can be determined, allowing for the ion expansion to be calculated. However, for eq 7.13 to be solved, the ion density at each of the mesh points must be known. To calculate these densities, the ion shells were smoothed into a continuous density function by utilizing a series of interpolating polynomials. This density function was then utilized to evaluate the ion density at each of the different mesh points, allowing for eq 7.13 to be fully defined.

Once the ion density is known, eq 7.13 could be solved, allowing for the ion expansion to be computed. To do this, the ion expansion was broken up into a series of timesteps. At each time step, the electric field was calculated as

$$E = -\frac{k_B T}{q\sigma_0} \frac{dW(R)}{dR} \quad (7.23)$$

which leads to the acceleration of the  $j$ th shell being expressed as follows

$$a_j = \frac{d^2 r_j}{dt^2} = - \frac{k_B T}{m_i \sigma_0} \left. \frac{dW(R)}{dR} \right|_{R=r_j/\sigma_0} \quad (7.24)$$

where  $m_i$  is the mass of an ion, and  $r_j$  is the position of the  $j^{th}$  ion shell. Each shell was accelerated for the duration of the time step in accordance with the force it experienced and its position was advanced by the amount determined by its velocity. New shell positions and velocities were then recorded for use in the next timestep. Ion-ion collisions were not included in the model as they were not expected to play a significant role over the expansion timescales of the UCP systems of interest[10]. For this reason ion shells were allowed to pass through other ion shells unaffected.

## 7.2.5 Temperature Evolution

As the ions in the UCP expand, the temperature of the electrons begins to cool. In addition to providing insight into the dynamics of UCPs systems, quantifying the rate of electron cooling is necessary to evaluate Eq. 7.13. Since  $W$  is a function of  $T$ , it is necessary to know the value of  $T$  to calculate the ion expansion. By utilizing conservation of energy the rate of electron cooling can be calculated. This process is complicated by  $T$  not being the actual electron temperature. Instead,  $T$  is a global parameter that is related to the electron temperature. The effective electron temperature,  $T_{eff}$ , varies as a function  $W$  and thus varies as a function of  $R$ . This quantity can be expressed as the following.

$$T_{eff} = \frac{\sqrt{\frac{2}{\pi}} e^W \operatorname{erf}(\sqrt{W}) - (4/15W^2 + 2/3W + 1)\sqrt{2W}}{\sqrt{\frac{2}{\pi}} e^W \operatorname{erf}(\sqrt{W}) - (2/3W + 1)\sqrt{2W}} T \quad (7.25)$$

The total energy of the UCP system can be described as a sum of the electrostatic energy of the total charge distribution, the total ion kinetic energy, and the electron kinetic energy

(as specified locally by  $T_{eff}$ ). This leads to the following expression for the total energy,  $U$ ,

$$U = 2\pi\sigma_0^3\epsilon_0 \int_0^{R_0} R^2 E(R)^2 dR + \frac{1}{2}m_i \sum_{j=1}^{10000} N_j \left(\frac{dr_j}{dt}\right)^2 + \frac{3}{2}k_B\alpha N_i \int_0^{R_0} R^2 \Sigma(R) T_{eff}(R) dR \quad (7.26)$$

where  $E(R)$  is the electric field, and  $N_j$  is the number of ions in the  $j$ th shell. In all of our calculations, the number of electrons were held constant, and thus  $U$  is held fixed. As the UCP expands, ions are accelerated and their kinetic energy increases. Expansion leads to a decrease in charge density, which causes a decrease in the electrostatic energy. Finally, electrons transfer thermal energy to ion kinetic energy, leading to cooler electrons. Changes in ion kinetic energy and electrostatic energy were calculated explicitly. Decreases in electron thermal energy followed from energy conservation.  $T$  was adjusted throughout the modeled expansion to produce the appropriate decrease in electron thermal energy required to maintain energy conservation.

Since the model fixes the total energy of the system, and the number of particles is fixed as well, evaporation cannot be incorporated into the model as it is currently constructed. However, with some simple tweaks, it should be relatively straightforward for evaporation to be implemented into the model as well. Perhaps the simplest method for implementing evaporation would be to fix the evaporation rate based on the depth of the potential. In this formulation, evaporation would be described a reduction in the number of electrons and a reduction in the electron temperature. Implementing evaporation into the non-neutral model is the subject of future work.

### 7.3 Model Results

The effect of charge imbalance on the rate of UCP expansion can be illustrated by comparing the results of a pair of UCP simulations. The degree of charge imbalance is quantified

by the parameter  $\delta = (N_i - N_e)/N_i$  where  $N_i$  is the number of ions and  $N_e$  is the number of electrons. For the simulations serving as an illustration of typical results, charge imbalances of  $\delta=0.1$  and  $\delta=0.4$  are used. Both simulations had initial parameters of  $\sigma_0 = 3.75 \times 10^{-4}m$ ,  $N = 3.0 \times 10^5$  and  $T = 85K$ . The ion mass used in these calculations was the mass  $^{85}\text{Rb}$ . Time steps were set to be 50ns, and the simulations ran for 85 steps ( $4.25\mu s$ ).

The first noticeable differences between the two cases was visible in their electric field vs. position at time  $t=0$ . In both simulation runs there was a center region where the electric field matches that computed by using a simple model that assumes a neutral plasma[1]. Far away from the UCP, the electric field behaves as expected and matches that of a point charge at the origin whose magnitude is equal to the charge imbalance. However, the field in between these two regions is strongly influenced by the UCP charge imbalance. For the  $\delta=0.4$  simulation, this middle region has a pronounced electric field peak whose magnitude is much larger than the electric field associated with the neutral plasma approximation. In contrast, the electric field in the more neutral case follows the approximate field with only slight positive deviations until the "edge" of the UCP is reached. At that point, the electric field in the  $\delta = 0.1$  case slowly declines to its  $1/r$  asymptotic behavior. A figure illustrating this electric field behavior can be seen in figure 7.4.

This difference in electric field structure had a significant impact on the expansion of the UCP. The  $\delta=0.1$  simulation's expansion was approximately uniform, maintaining an approximately Gaussian shape throughout the simulation. The  $\delta=0.4$  simulation, however, showed a much larger degree of exterior expansion, as can be seen in figure 7.5. The ions in the aforementioned middle region undergo a much larger acceleration than the rest of the ions outside the middle region, and so the ions initially residing farther out in the UCP exterior ultimately had lower velocity than the ions that started in the region where the electric field is the largest. The difference in velocities led to interior ions eventually catching up to exterior ions and forming a large ion spike at the edge of the UCP, as originally theoretically predicted in Ref. [11] and also observed theoretically in Ref. [12]. The extra expansion

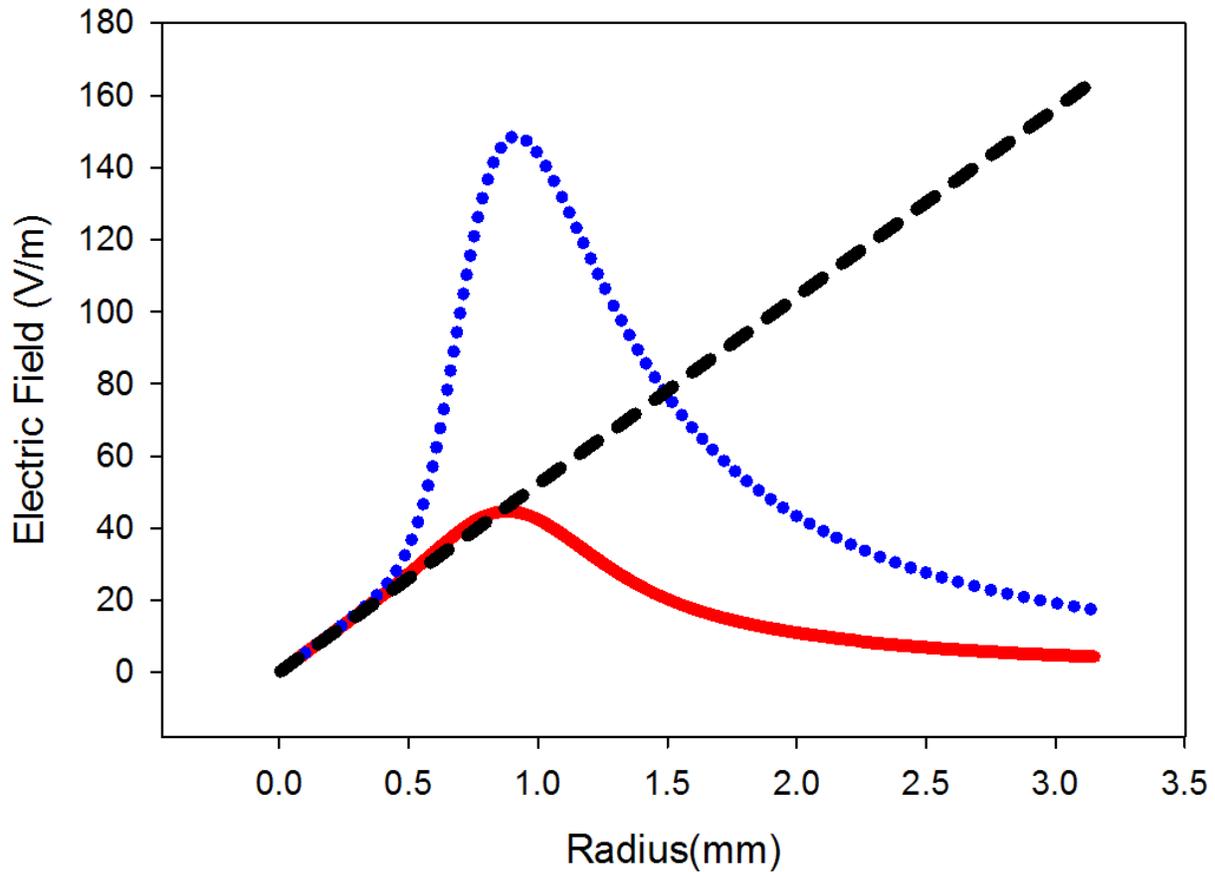


Figure 7.4: Initial electric field vs. position for two  $^{85}\text{Rb}$  UCPs with different charge imbalances. The solid line is a  $\delta = 0.1$  plasma, and the dotted line is a  $\delta = 0.4$  plasma. Both UCPs have initial parameters  $T=85\text{K}$ ,  $\sigma_0=3.75\times 10^{-4}\text{m}$ , and  $N=3\times 10^5$ . The dashed line is from a simple model for a 85K neutral plasma as described in the main text. The electric fields at the center of the plasmas mirror the predictions of the simple model well. However, once outside the center, the  $\delta=0.4$  field increases substantially while the  $\delta=0.1$  field increases only slightly.

also led to a larger overall UCP spatial size as compared to the  $\delta = 0.1$  case. Our predictions confirmed the naive expectation that larger charge imbalances lead to higher rates of overall UCP expansion despite electron screening in the interior of the UCP.

While the charge imbalance has a large and visible impact on the exterior of the plasma, the variation of the expansion of the interior is shielded to some extent from effects associated with changes in the charge imbalance. This is reasonable given the screening properties of a plasma. The radius of a sphere that encloses 90% of the electrons in a  $\sigma_0=3.75 \times 10^{-4} \text{m}$ ,  $T=85 \text{K}$ ,  $N=3.0 \times 10^5$ , and  $\delta = 0.1$  UCP is approximately  $900 \mu\text{m}$ . Electron densities at the edge of this sphere are approximately  $2.5 \times 10^{13} \text{m}^{-3}$  which corresponds to a Debye length of  $127 \mu\text{m}$ . Since the UCP is larger in spatial extent than the Debye length, it is reasonable to expect that edge perturbations would have little effect on the center. The UCP electric field reinforces this view. At small values of  $R$ , the electric field does not change as a function of charge imbalance.

The center region of a UCP was not shielded from all the effects of changes in charge imbalance, however. As the charge imbalance increased, the outer ions expanded faster. This resulted in the overall volume of the electron gas in the interior increasing faster in turn, which then resulted in a larger amount of electron cooling via expansion. An example of this can be clearly seen in figure 7.6, where the high charge imbalance case cooled much more quickly than the low charge imbalance case.

The cooling of the electrons in figure 7.6 is quantified by the decrease in  $T$  as a function of time. It is worth reiterating that  $T$ , while similar, is not identical to the electron temperature. For the  $\delta = 0.4$  case, the total thermal energy of the UCP electrons is equal to  $3/2 N_e k_B T$  to an excellent approximation, as would naively be expected. In contrast, for the  $\delta = 0.1$  case, the total electron thermal energy is 11% less than this naive expectation. However, if both a  $\delta = 0.1$  and the  $\delta = 0.4$  UCP with an initial average temperature of  $\langle T_{eff} \rangle = 85 \text{K}$  are simulated, a plot of  $\langle T_{eff} \rangle$  vs time would look qualitatively similar to the above

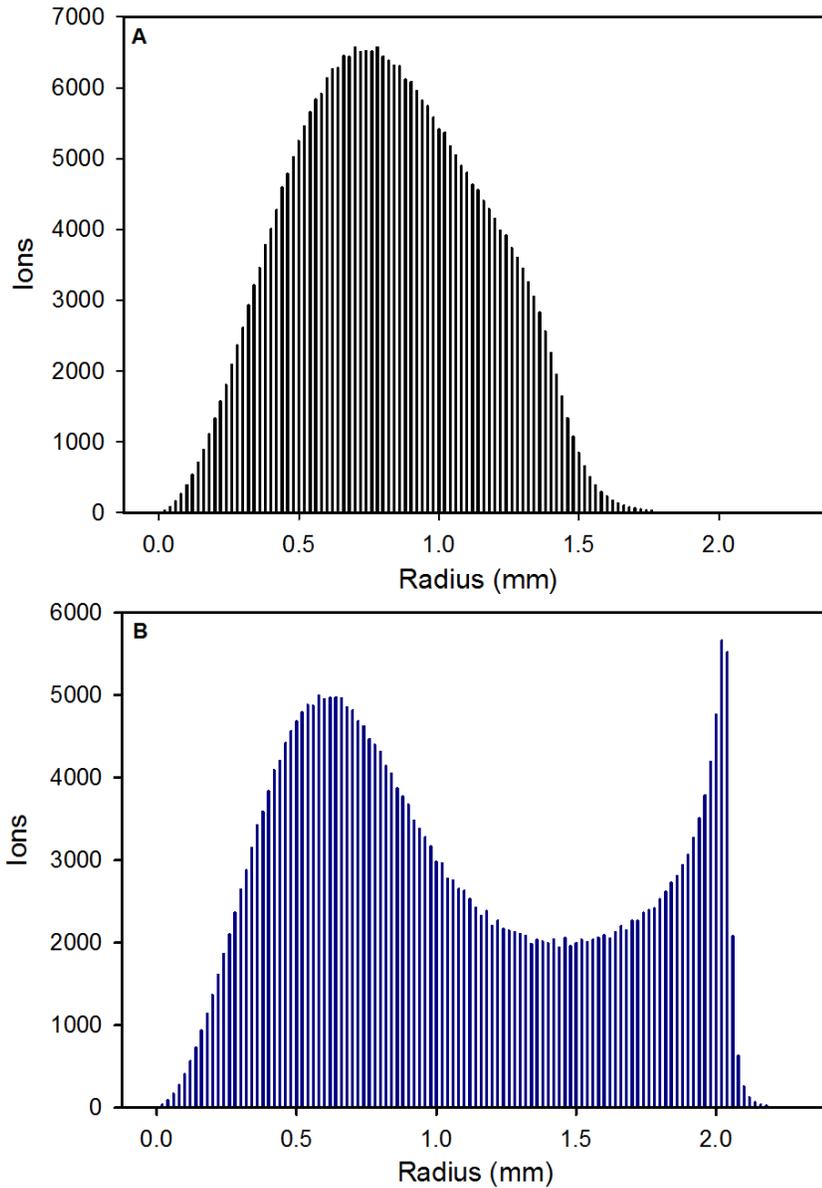


Figure 7.5: A) An example of the Ion distribution of a plasma after expanding for  $4\mu\text{s}$ . This particular plasma had initial parameters  $T=85\text{K}$ ,  $\sigma_0 = 3.75 \times 10^{-4}\text{m}$ ,  $N = 3 \times 10^5$ , and  $\delta = 0.1$ . The ion distributions remains mostly Gaussian, but a small bulge is detectable at the edge. B) A second plasma, identical to the plasma in (A), except that for this case  $\delta = 0.4$ . The increase in charge imbalance produced substantially more expansion on the exterior and resulted in the formation of a so-called ion spike on the edge.

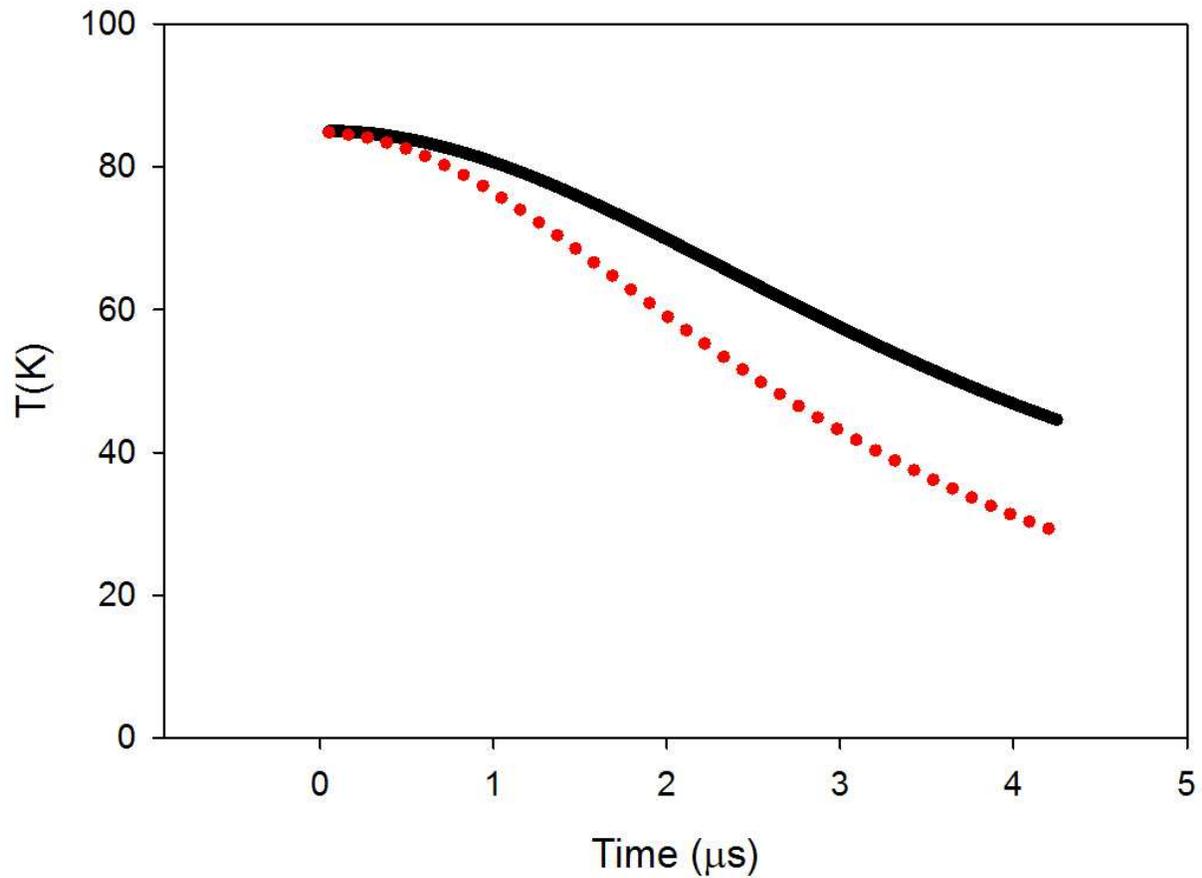


Figure 7.6: Plots of  $T$  vs time for two plasmas with different charge imbalances. The solid line is a  $\delta=0.1$  plasma, and the dotted line is a  $\delta=0.4$  plasma. Both plasmas have initial parameters  $T=85\text{K}$ ,  $\sigma_0=3.75\times 10^{-4}\text{m}$ , and  $N=3\times 10^5$ .

figure. The most noteworthy difference would be that the two curves would be slightly closer together, with  $\delta=0.1$  curve about 8% colder at the later times shown in figure 7.6.

The resulting increase in the rate of electron cooling resulted in a small decrease in the expansion rate of the *central portion* of the UCP over the early time evolution of the UCP that we studied. Generally, at the very center of the UCP, the ions' density distribution is well approximated by a Gaussian over the timescales of the simulation. To characterize the central expansion rate of the UCP, a scale length  $\sigma_c$  based on the center ion density was calculated during the expansion.  $\sigma_c$  was calculated at each timestep by integrating the ion density from the center outward until a radius within which the number of enclosed ions matched the number of ions expected to be enclosed within one  $\sigma$  in a standard Gaussian distribution.  $\sigma_c$  was defined through this radius.

The time evolution of  $\sigma_c$  can be seen in figure 7.7. The  $\delta=0.4$  simulation expanded at a slightly slower rate than the center of the  $\delta=0.1$  simulation. While the difference between the two expansion rates was small over the course of the simulations, this difference is large enough to surpass the few percent sensitivity of a two cycle rf technique and thus should be experimentally detectable[13]. The fact that increasing  $\delta$  leads to overall faster expansion while leading to a slower expansion at the center of the UCP indicates that a change in UCP shape from the initial Gaussian distribution is a fundamental feature associated with changes in charge neutrality.

The fact that there can be significant impact from the change in charge neutrality on  $T$  with only a mild change to the effective size vs. expansion time has experimental implications. In previous work, the electron temperature has been measured by measuring the rate of change of UCP density[2, 3, 14]. From this an electron temperature is extracted, often through a computation based on a simple neutral plasma expansion model like Ref[1] that explicitly assumes perfect neutrality in the interior region of the UCP. For a more non-neutral UCP, this technique would produce an incorrect temperature by not accounting for

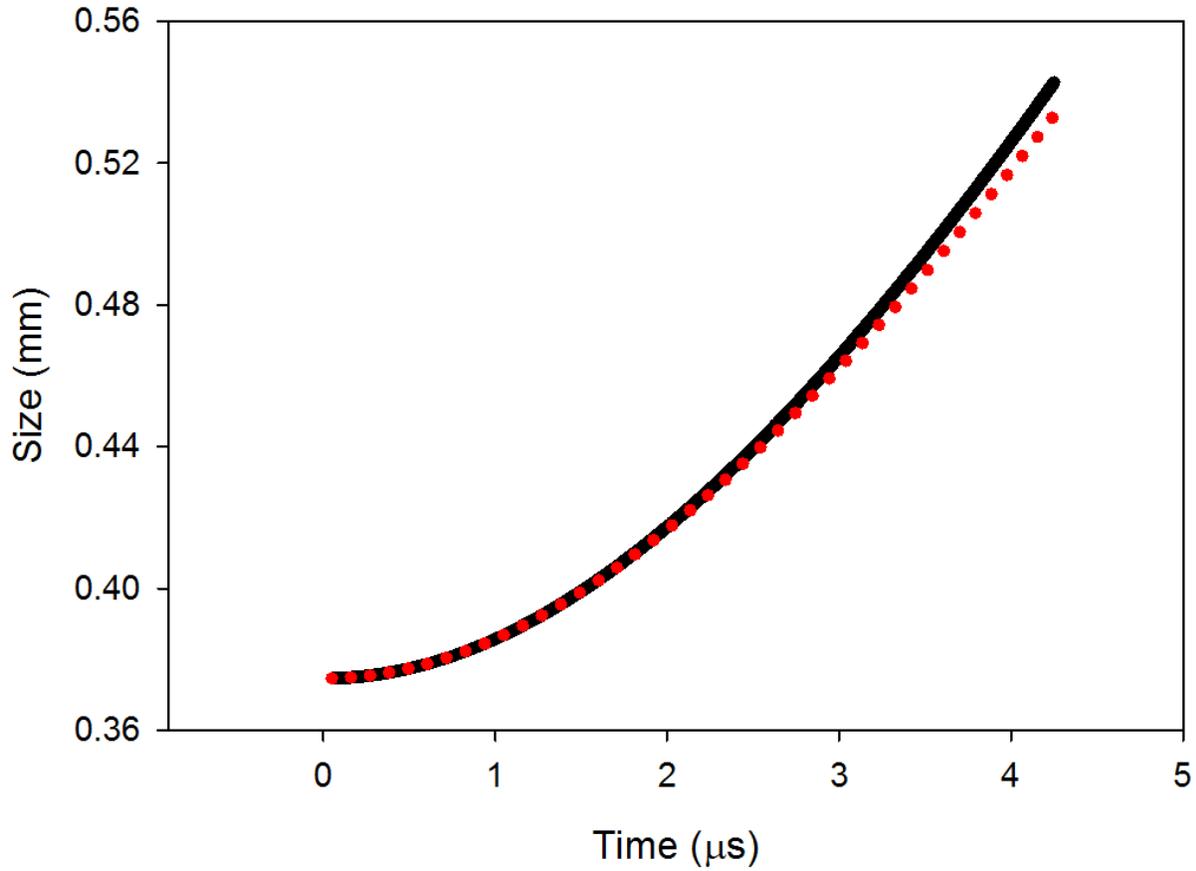


Figure 7.7: Plots of effective center size vs time for two plasmas with different charge imbalances. The solid line is a  $\delta=0.1$  plasma, and the dotted line is a  $\delta=0.4$  plasma. Both plasmas have initial parameters  $T=85\text{K}$ ,  $\sigma_0=3.75\times 10^{-4}\text{m}$ , and  $N=3\times 10^5$ . Since the UCP interior behaves like a neutral plasma, the interior ions can be described by a Gaussian distribution at all times.

the interplay between the outer ion expansion and the electron temperature. For a more accurate prediction, a model like the one outlined in this paper would have to be utilized.

The robustness of the model results were tested by additional simulation runs over a variety of experimentally achievable initial conditions. This included variations of initial ion number,  $T$ , and UCP size (i.e.  $\sigma_0$ ). For all tested sets of parameters, UCP expansion and cooling had a qualitatively similar dependence on charge imbalance as the example simulations described above. However, the degree of increased cooling with increasing  $\delta$  varied with all of these parameters. Conversely, though, these results indicate that non-neutral effects do not complicate the modeling of the interior ion expansion significantly.

Some care needs to be taken when quantifying how variation in initial parameters affected the additional cooling due to non-neutrality. For example, the initial  $T$  of a neutral UCP has a significant impact on the UCP's cooling rate. To quantify the additional cooling caused by non-neutrality as function of initial  $T$ , it is necessary to decouple the effect that the initial  $T$  has on a neutral UCP cooling rate. This was accomplished by generating cooling curves for  $\delta = 0.1$  UCPs with different initial values  $T$  to serve as a baseline. For each different initial  $T$  curve, the times over which the UCP cooled to 75% of the initial  $T$  were recorded. The amount of cooling over these associated time periods was then calculated for  $\delta = 0.4$  UCPs with the same initial value of  $T$ . This allowed for a direct comparison of the change in relative cooling rates for two different charge imbalances at a specific initial  $T$ .

It was found that high  $\delta$ 's led to more additional fractional cooling in UCPs with low initial values of  $T$  than UCPs with high initial values of  $T$ . This effect can be seen in figure 7.8. A similar method revealed that UCPs with higher densities (higher ion number, or smaller size) also experienced more additional charge imbalance induced fractional cooling than UCPs with low densities.

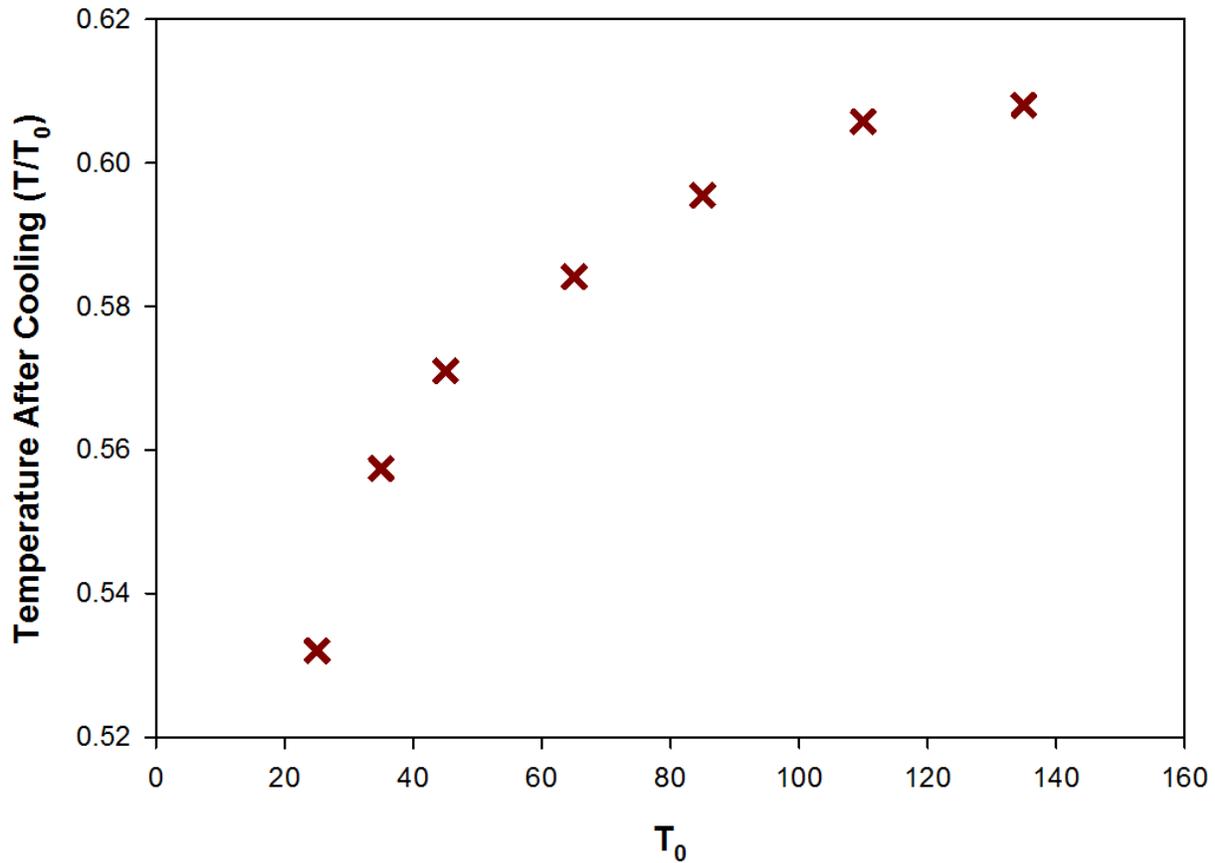


Figure 7.8: The above figure shows the "relative impact" that an increase in charge imbalance had on cooling as a function of initial  $T$  ( $T_0$ ). For a given  $T_0$ , a  $\delta=0.1$  UCP was simulated to expand until it had cooled to 75% of the value of  $T_0$ . An analogous  $\delta=0.4$  UCP was then simulated for the same amount of time as the  $\delta=0.1$  UCP. The data points in the figure are the final values of  $T$  for the  $\delta=0.4$  simulation divided by  $T_0$  as a function of  $T_0$ . The plot shows that the impact on UCP from larger  $\delta$ , becomes increasingly more important as the initial value of  $T$  decreases. All simulations had initial parameters  $\sigma_0=3.75 \times 10^{-4} \text{m}$  and  $N=3 \times 10^5$ . Note that the left axis of this plot does not start at zero.

## 7.4 Ion Doppler Profiles

In addition to investigating the influence of outer ions on the interior region of the UCP, our model can be used to predict the total fluorescence from Doppler-shift sensitive measurements of the ion velocity distribution. Many UCP experiments use ion absorption imaging to deduce UCP parameters[15, 16]. Such techniques use near resonant laser pulses to illuminate the UCP. Ion absorption is measured by a CCD camera behind the UCP. By altering the frequency of the laser, the Doppler broadened spectrum for the UCP can be obtained. Mapping the Doppler profiles of the entire UCP yields information on both the size and velocity distributions the UCP ions. For a more in depth discussion of this technique see Ref [15].

We investigated the effect that non-neutrality would have on such Doppler profiles for achievable experimental conditions where charge imbalance considerations are more important (e.g. at low density). Such measurements could be used to test the predictions of our model. Conversely, the considerations in this model predict  $\delta$ -dependent features in such UCP measurements.

To determine if the changes in  $\delta$  explored in this work would produce noticeable changes in the Doppler profile, it was necessary to extract theoretical Doppler profiles from model outputs. This was accomplished by calculating the scattering of an incoming uniform intensity laser by the ion shells. The  $z$  axis was taken to be the direction of laser propagation. Each ion shell is defined by a single velocity,  $v$ . However, since  $v$  is in the  $\hat{r}$  direction,  $v_z$  and subsequently the scattering rate changes as a function of  $x$  and  $y$  position. Thus, it was necessary to integrate across each shell to calculate that shell's scattering rate. The total absorption is calculated by adding up the scattering of each of the shells. Doppler absorption profiles were determined by calculating the total absorption as function of detuning.

Total absorption profiles were calculated at  $4.25 \mu\text{s}$  for both the  $\delta=0.1$  and the  $\delta=0.4$  test cases. The resulting absorption profiles, seen in figure 7.9, differed significantly. The

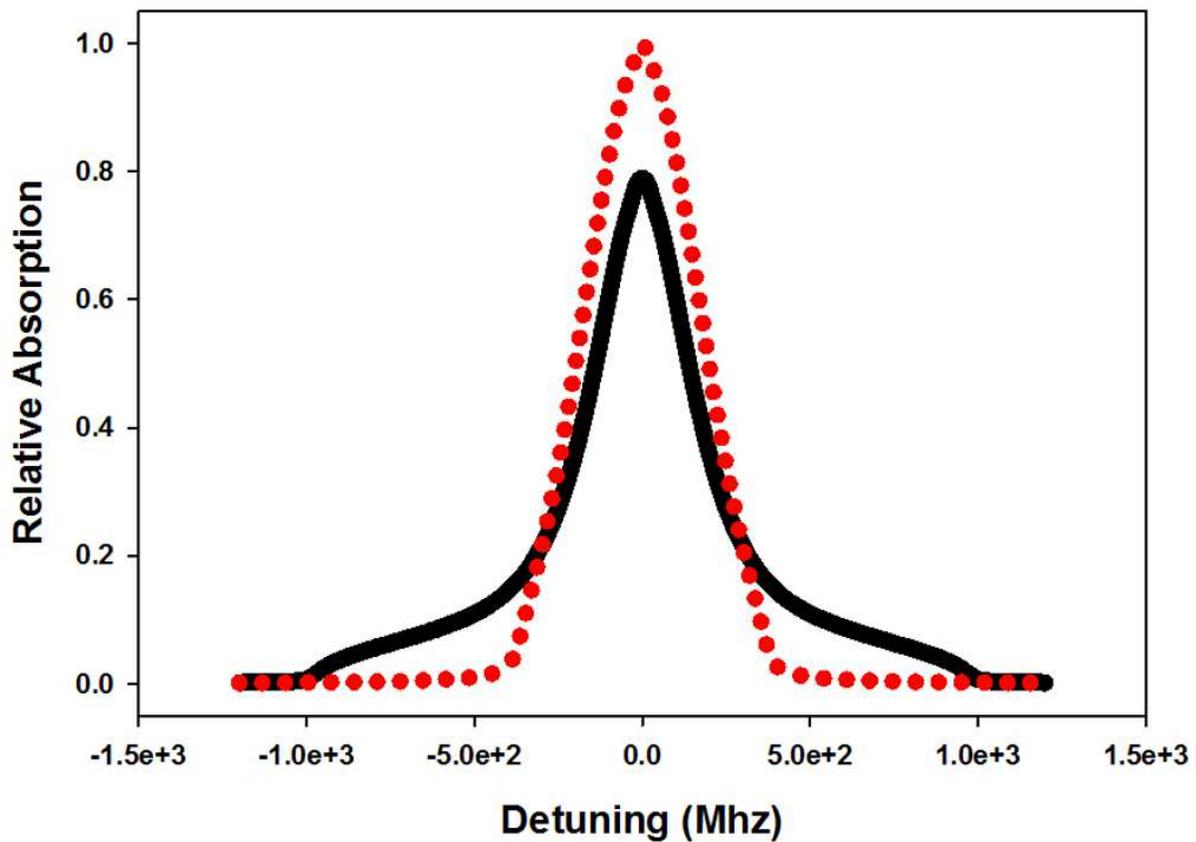


Figure 7.9: Doppler absorption profiles for two plasmas with different charge imbalances at the a  $t=4.25\mu\text{s}$ . The dotted line is a  $\delta=0.1$  plasma, and the solid line is a  $\delta=0.4$  plasma. Both plasmas have initial parameters  $T=85\text{K}$ ,  $\sigma_0=3.75\times 10^{-4}\text{m}$ , and  $N=3\times 10^5$ .

$\delta=0.4$  profile was broader, most notably in the wings of the distribution, than the  $\delta = 0.1$  profile. These results indicate that it is important to account for UCP charge imbalance when utilizing ion absorption imaging techniques in situations with reduced charge neutrality.

## 7.5 Conclusion

This chapter discussed the theoretical model I developed to investigate the influence of non-neutrality on the evolution of UCPs. It was found that while the expansion of the UCP interior, over initial expansion times, does not change significantly as a function of UCP neutrality, there is a non-negligible effect on the electron temperature. This effect is predicted to produce an increase in the electron cooling rate during UCP expansion at higher levels of charge imbalance. A modeling of the effects described in this work are necessary for a proper interpretation of electron cooling experiments in UCPs, including cooling due to forced and unforced evaporation. Beyond theoretically investigating the influence of charge imbalance on UCP expansion and electron temperature evolution, this model can be extended to include evaporation and three-body recombination induced heating as well to evaluate the likely effectiveness of forced evaporative cooling in achieving greater amounts of electron component strong coupling. This will be the subject of future work.

# References

- [1] F. Robicheaux and J. D. Hanson, Phys. Rev. Lett. **88**, 055002 (2002).
- [2] P. Gupta et al., Phys. Rev. Lett. **99**, 075005 (2007).
- [3] S. D. Bergeson, A. Denning, M. Lyon, and F. Robicheaux, Phys. Rev. A **83**, 023409 (2011).
- [4] R. S. Fletcher, X. L. Zhang, and S. L. Rolston, Phys. Rev. Lett. **99**, 145001 (2007).
- [5] P. McQuillen, T. Strickler, T. Langin, and T. C. Killian, Physics of Plasmas **22**, 033513 (2015).
- [6] M. Lyon, S. D. Bergeson, A. Diaw, and M. S. Murillo, PHYSICAL REVIEW E **91** (2015).
- [7] I. R. King, Astron J. **71** (1966).
- [8] D. Vrinceanu, G. S. Balaraman, and L. A. Collins, J. Phys. A: Math Theor. **41**, 425501 (2008).
- [9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran*, Cambridge University Press, 2nd edition, 1992.
- [10] L. Spitzer, *Physics of Fully Ionized Gases*, Dover Publications, Mineola, NY, 2006.
- [11] F. Robicheaux and J. D. Hanson, Phys. Plasmas **10**, 2217 (2003).
- [12] T. Pohl, T. Pattard, and J. Rost, Phys. Rev. A **70**, 033416 (2004).
- [13] T. Wilson, W.-T. Chen, and J. L. Roberts, Phys. Rev A **87**, 013410 (2013).
- [14] S. Kulin, T. C. Killian, S. D. Bergeson, and S. L. Rolston, Phys. Rev. Lett. **85**, 318 (2000).

[15] T. C. Killian et al., J. Phys. B: At. Mol. Opt. Phys. **38**, S351 (2005).

[16] M. Lyon and S. D. Bergeson, J. Phys. B: At. Mol. Opt. Phys. **44**, 184014 (2011).

# Chapter 8

## Future Work

The work outlined in this thesis represents just a small slice of potential investigations into ultracold plasma systems. While future research in the lab could progress in currently unexpected directions, there are a few specific areas the work that I have accomplished during my time at Colorado State University will likely contribute strongly to future research projects. My research into electron-ion collisions can be used to better inform future experimental studies into those collisions. Additionally, the computational machinery that I have developed (particularly the GPU-enhanced molecular dynamics modeling) can be adapted and applied to a wide variety of other interesting theoretical questions. This chapter will give a brief overview of some of the more immediate research directions that are currently planned to be taken in the lab.

There are three future research topics that I will discuss in this section. The first research thrust is studying electron-ion collisions in a highly magnetized environment. Coulomb collisions play an important role in the transport properties of many magnetized plasmas, and experimental measurements of the electron-ion collision rates in a magnetized UCP could provide insight into these more complex magnetized systems. The second area of future work is testing the implied assumptions in different strongly coupled plasma theories. Finally, The third area of future work will be to use the UCP formation model, which was discussed in chapter 4, that I have developed to theoretically calculate the heating resulting from continuum lowering in UCP systems. The following sections discuss these three areas of future research in greater detail.

## 8.1 Highly-Magnetized Electron-Ion Collisions

In highly magnetized plasmas the Larmor radius,  $r_L$ , for charged particles is much shorter than the Coulomb distance of closest approach,  $b_{90}$ , which is defined as  $b_{90} = q^2/4\pi\epsilon_0k_bT$ . In this limit charged particle motion is strongly confined to magnetic field lines[1]. In unmagnetized plasma systems the collisional cross-section for Coulomb collisions is on the order of  $b_{90}^2$ . However, in highly magnetized plasma systems, this cross-section is much smaller, since  $r_L \ll b_{90}$ . This leads to Coulomb collisions occurring with much less frequency than in their unmagnetized counterparts. This change in the Coulomb collision rate can have a profound effect on the transport properties of highly magnetized system. Since measuring Coulomb collision rates in many highly magnetized plasma systems is experimentally challenging, an experimental measurement of magnetized collision in UCPs could prove quite beneficial for characterizing such systems.

Whether a plasma system is considered magnetized or not is typically determined by the parameter  $\kappa$ [2]. If  $\kappa$  is greater than 1 a system is considered magnetized, and if  $\kappa$  is less than one a plasma is consider unmagnetized. For a given plasma system,  $\kappa$  is defined as follows

$$\kappa = \frac{2b_{90}}{r_L} \tag{8.1}$$

For a UCP with an electron temperature of 5K, a  $\kappa$  of 1 corresponds to an applied magnetic field of 74G. Since, the existing field coils in the experiment produce around a 10G field, the experiment will need to be altered before any highly magnetized experiment can begin.

The proposed experimental plan requires retrofitting the experimental apparatus with field coils which are capable of producing magnetic fields on the order of a few hundred Gauss. This should be able to produce  $\kappa$  values exceeding 1. Once these field coils are in place, the experiment will be operated in manner analogous to the electron-ion collision experiment outlined in chapter 3. Once this experiment is completed, the stronger field coils can be used to investigate other highly magnetized plasma physics. One intriguing possibility

is to quantify the effect that a strong magnetic field has on three-body recombination in UCPs. It is predicted that a strong magnetic field should drastically reduce the three-body recombination rate thus lowering the amount of electron heating generated by this process. However such an effect, while likely, has yet to be experimentally verified.

## 8.2 Testing Assumptions implied in Plasma Theories

In chapter 3, I discussed how my hybrid Monte Carlo molecular dynamics code could be used to test the validity of any strongly coupled plasma collision theory that contains a Coulomb logarithm. By utilizing a number of standard assumptions, any such theory can be expressed completely by a single value of the collisional cutoff,  $b_{max}$ . It was found, when applying the standard assumptions, that none of the more common strongly coupled plasma theories[3, 4, 5, 6] matched the molecular dynamics simulations for the electron center of mass damping at the conditions where experimental data was collected. In fact, the discrepancy was so large, that there was roughly a factor of three difference in the damping rate between the measured rate and strong coupling corrected electron ion collision predictions from Refs [3, 4, 5, 6]. However, it was found that by relaxing one of the standard assumptions, namely that the Coulomb logarithm is roughly constant and can be pulled outside of an averaging integral, roughly 75% of the original discrepancy vanished.

However, 25% of the original gap still remains, and finding the source of this remaining gap will be a subject of future work. The hypothesis is that by relaxing some of the other standard plasma assumptions the gap will eventually be closed. The work discussed in chapter 3 has already eliminated some of these assumptions as possibilities, but two assumptions remain untested. The first of these approximation is the Rutherford scattering approximation. This approximation introduces the concept of a cutoff impact parameter  $b_{max}$  where collisions with an impact parameter,  $b > b_{max}$  are assumed to not occur. Such an approximation is unphysical, since it leads to a discontinuous distribution of impact parameters, but it is never the less used in all plasma theories since the Coulomb logarithm would diverge

if a cutoff were not used. The second approximation the binary collision approximation. This approximation assumes that all collisions occurring in a plasma system can be accurately expressed as a the sum of many two body collisions occurring in a vacuum. Like the Rutherford scattering approximation, the binary collision approximation is included in all theoretical treatments of Coulomb collisions.

Some thought must be given as to how to best test these assumptions. Unlike the assumptions tested in chapter 3, the remaining two assumptions lie very much at the core of how Coulomb collisions are treated in plasma systems. For this reason, it may be difficult to decouple these assumptions from the Monte Carlo treatment discussed in Chapter 3. However, the centrality of these assumptions is the very reason why an investigation into their limitations holds significant scientific interest. The potential breakdown of one of these assumptions could have an important impact on the field where strongly coupled plasmas are concerned.

### 8.3 Continuum Lowering and Formation Modeling

The current scientific understanding of continuum lowering is in a state of confusion. Two main theoretical models of continuum lowering(Stewart and Pyatt[7] and Ecker-Kröll[8]) exist, and they do not agree with each other. To make matters worse, there have been a few recent experiments which have measured continuum lowering, and the results of these experiments do not all agree with each other. One set of experiments[9, 10] has yielded results consistent with Ecker-Kröll, while another experiment[11] has yielded results that were consistent with Stewart and Pyatt. Given the current uncertainty surrounding continuum lowering, the field has generated significant interest in recent years.

Continuum lowering is predicted to be a significant electron heating mechanism in ultra-cold plasmas, making them a good system to study continuum lowering. Using the formation model that I developed in chapter 4 to calculate continuum lowering heating in UCPs will be another area of future work. Since the formation model uses a full molecular dynamics

treatment, a naive probing of the UCP's electron temperature will include the contributions from all of the different electron heating and cooling mechanisms. As such, the main challenge will be to decouple the different heating and cooling mechanisms from each other. This will have the added benefit of providing a quantitative prediction for all of the other heating and cooling mechanisms as well.

A secondary goal of this work is to probe the dynamical nature of the continuum lowering process in UCPs. Both Stewart and Pyatt and Ecker-Kröll are statistical mechanics based theories. However in UCP systems, continuum lowering occurs over a finite time. Atoms are ionized into a plasma background that has not had time to equilibrate. Certain atoms are ionized before others, and the entire process occurs over a timespan set by the pulse length of the ionizing laser pulse. To test the time-dependent nature of continuum lowering a number of different iterations of the formation model can be simulated, with each iteration simulating a different ionizing pulse length. This should allow for a prediction of continuum lowering heating as a function of ionization pulse length. If such an effect is found to exist, it could allow for experimental systems to choose pulse lengths that will minimize continuum lowering heating, and potentially make higher degrees of electron strong coupling accessible.

# References

- [1] T. Wilson, *Dynamics of Low-Density Ultracold Plasmas in Externally Applied Electric and Magnetic Fields*, PhD thesis, Colorado State University, 2013.
- [2] M. E. Glinsky, T. M. O'Neil, M. N. Rosenbluth, K. Tsuruta, and S. Ichimaru, *Phys. Fluids B.* **4**, 1156 (1992).
- [3] J. Daligault and D. Mozyrsky, *HIGH ENERGY DENSITY PHYSICS* **4**, 58 (2008).
- [4] P. E. Grabowski, M. P. Surh, D. F. Richards, F. R. Graziani, and M. S. Murillo, *Phys. Rev. Lett.* **111**, 215002 (2013).
- [5] S. D. Baalrud, *Phys. Plasmas* **19**, 030701 (2012).
- [6] L. G. Stanton and M. S. Murillo, *Phys. Rev. E* **93**, 043203 (2016).
- [7] J. C. Stewart and K. D. Pyatt, *Astrophys. J.* **144**, 1203 (1966).
- [8] G. Ecker and W. Kröll, *Phys. Fluids* **6**, 62 (1963).
- [9] O. Ciricosta et al., *Phys. Rev. Lett.* **109**, 065002 (2012).
- [10] T. Preston et al., *High Energy Dens. Phys.* **9**, 258 (2013).
- [11] D. J. Hoarty et al., *Phys. Rev. Lett.* **110**, 265003 (2013).

# Appendix A

## Sample Code

Chapter 2 discussed the general molecular dynamics model that I developed. Most of the code for this model is contained in libraries, and can be used without knowing the ins and outs of how it works. However, there is a bit of work that is needed to create an environment for which this model code can be run. The sample code shown in this chapter provides a template for setting up the model. As is, the code only initializes particles and model classes and then calculates their evolution. However, with fairly minor alterations the code should be capable of studying a wide variety of physics phenomena. Additionally, the code, as currently written, provides many adjustable parameters as well as number of binary switches, making this template a flexible starting point for developing UCP simulations in PyOpenCL. The sample code is as follows:

```
import numpy as np
import random
import pyopencl as cl
import pyopencl.clrandom as clrand
import pyopencl.array as cl_array
import uci.CoulombAcc as C_Acc
import time
import math
import UCP
import mylib
import Energy_lib
import matplotlib.pyplot as plt
import os
```

```
#Fundamental Constants
```

```

fund_charge = 1.602176565e-19
kb=1.3806488e-23
ion_mass = 84.911789738 * 1.673e-27
electron_mass=9.10938291e-31

#Initial Parameters
Ne=10000
Ni=10000
Te=2.
Ti=1.
sigma=3.5e-4
alpha=5.e-7
n=2.5e12

#Set simulation length and parameter save frequency
steps=25000
resolution=250
dt=1.e-11

#Determine how the ions will be treated
Ion_Force=True
Discrete_Ions=False
if Discrete_Ions:
    N=Ni+Ne
    Ion_Force=False
else:
    N=Ne

#Turns on the collision operator
ei_collisions=False
if Discrete_Ions:
    #shuts off collisions automatically if discrete ions are used
    ei_collisions=False

```

```

#External Fields
Efield_on=False
Bfield_on=False
Bfield_is_a_Force=True
linear_on=False
linear=-297.
DC=-4.8
B=9.e-4

#Initial Placement
density="gaussian"
read=False
vspreadE=math.sqrt(kb*Te/electron_mass)
vspreadI=math.sqrt(kb*Ti/ion_mass)
offset=0. #shift in z direction of the electron cloud
#sets the maximum radius for initial electron placement
cutoff=None

#####Create host arrays buffer#####

#Random Positions generated if necessary
if not read:
    if density=="uniform":
        R=(3.*float(N)/(4.*n*math.pi)**(1./3.))
        print "R is",R
        r = np.random.rand(N)**(1./3.)*R
        theta = np.arccos(2.*np.random.rand(N)-1)
        phi = 2.*np.pi*np.random.rand(N)
        x=(r*np.sin(theta)*np.cos(phi)).astype(np.float32)
        y=(r*np.sin(theta)*np.sin(phi)).astype(np.float32)

```

```

z=(r*np.cos(theta)).astype(np.float32)
elif density=="gaussian":
x=np.random.randn(N).astype(np.float32)*sigma
y=np.random.randn(N).astype(np.float32)*sigma
z=np.random.randn(N).astype(np.float32)*sigma
if (cutoff != None):
    for i in range(Ne):
        while abs(math.sqrt(x[i]**2.+y[i]**2.+z[i]**2.))>cutoff:
            x[i]=random.gauss(0.,sigma)
            y[i]=random.gauss(0.,sigma)
            z[i]=random.gauss(0.,sigma)
            z[0:Ne]+=offset
else:
    print "not a supported density"

#create velocity arrays
if not read:
vx=np.ndarray(shape=[N], dtype=np.float32)
vy=np.ndarray(shape=[N], dtype=np.float32)
vz=np.ndarray(shape=[N], dtype=np.float32)
for i in range(Ne):
    vx[i]=random.gauss(0,vspreadE)
    vy[i]=random.gauss(0,vspreadE)
    vz[i]=random.gauss(0,vspreadE)
for i in range(Ne,N):
    vx[i]=random.gauss(0,vspreadI)
    vy[i]=random.gauss(0,vspreadI)
    vz[i]=random.gauss(0,vspreadI)

#load parameters from file if necessary
if read:
    source=open("input.txt","r")

```

```

temp = np.loadtxt(source)
for i in range(N):
    x[i]=temp[i][0]
    y[i]=temp[i][1]
    z[i]=temp[i][2]
    vx[i]=temp[i][3]
    vy[i]=temp[i][4]
    vz[i]=temp[i][5]

# mass and charge arrays
m=np.ndarray(shape=[N], dtype=np.float32)
q=np.ndarray(shape=[N], dtype=np.float32)
for i in range(N):
    if i<Ne:
        m[i]=electron_mass
        q[i]=-fund_charge
    else:
        m[i]=ion_mass
        q[i]=fund_charge

#Constructs a log file of the basic parameters of the system
model_parameters=open("model_parameters.txt", "w")
model_parameters.write("Ne\t%d\nNi\t%d\nTe\t%e\nTi\t%e\ndt
\t%e\nsigma\t%e\nalpha\t%e\n"
%(Ne, Ni, Te, Ti, dt, sigma, alpha))
model_parameters.write("steps\t%d\nresolution\t%d\ndensity\t%s\noffset\t%e\n"
%(steps, resolution, density, offset))
if (cutoff != None):
    model_parameters.write("cutoff\t%e\n" %cutoff)
else:
    model_parameters.write("cutoff\tNone\n")
if Efield_on:
    model_parameters.write("E\t%e\n" %DC)

```

```

if Bfield_on:
    model_parameters.write("B\t%e\n" %B)
if B_is_a_Force:
    model_parameters.write("B is a force\n")
if read:
    model_parameters.write("data read from a file \n")
if linear_on:
    model_parameters.write("linear field %e\n" %linear)
if Discrete_Ions:
    model_parameters.write("discrete ions\n")
else:
    if ei_collisions:
        model_parameters.write("ion dist with collisions\n")
    else:
        model_parameters.write("ion dist without collisions\n")
model_parameters.close()

#make files and directories
if not os.path.exists("data"):
    os.makedirs("data")
energy_output=open("energy_output.txt","w")

#Set Up the OpenCL hardware context
ctx=cl.create_some_context()
queue=cl.CommandQueue(ctx)

#create device arrays
xd=cl_array.to_device(queue,x)
yd=cl_array.to_device(queue,y)
zd=cl_array.to_device(queue,z)

vxd=cl_array.to_device(queue,vx)

```

```

vyd=cl_array.to_device(queue,vy)
vzd=cl_array.to_device(queue,vz)

md=cl_array.to_device(queue,m)
qd=cl_array.to_device(queue,q)

#Updater and force classes
updater = UCP.Updater(ctx, queue,N)
coulomb=C_Acc.CoulombAcc(ctx,queue)
coulomb.impactFact=alpha**2.
acc=[coulomb]
trans=[]
if Efield_on:
    if linear_on:
        field=mylib.DC_field(N,field=DC,linear_field=linear,ctx=ctx,queue=queue,
        linear=linear_on)
    else:
        field=mylib.DC_field(N,field=DC,ctx=ctx,queue=queue)
    acc.append(field)
if Bfield_on:
    if Bfield_is_a_Force:
        B_field=mylib.Bz_field(ctx,queue,N,B)
        acc.append(B_field)
    else:
        #uses a rotation
        #timing of this rotation not optimized
        #see section 2.5.3 for how to optimize timing
        Bfield=UCP.B_field(ctx,queue,N)
        trans.append(Bfield)

if Ion_Force:
    ion=mylib.ions(N,sigma,Ni,ctx=ctx,queue=queue)
    acc.append(ion)

```

```

if ei_collisions:
    ###sets up GPU random number generator###
    seed=int(time.time())
    #datatype for GPU rand number generator
    #that needs to be passed collision operator
    rand=clrand.RanluxGenerator(queue,N,use_legacy_init=False,
                                seed=seed)
    collisions=UCP.collision_operator(ctx,queue,rand,Te,Ni,Ne,sigma)
    trans.append(collisions)

#energy class compiles energy calculating kernels
energy=Energy_lib.Energy(ctx,queue,N,Ni,N,sigma,alpha,Compile_Pair=False)
energy.Compile_Pair_Energy(double=False)

#Calculates the initial system energies
ke_E=[energy.kinetic_energy(vx[0:Ne],vy[0:Ne],vz[0:Ne],m[0:Ne])]
if Discrete_Ions:
    ke_I=[energy.kinetic_energy(vx[Ne:N],vy[Ne:N],vz[Ne:N],m[Ne:N])]
    ke=[ke_I[0]+ke_E[0]]
else:
    ke=[ke_E[0]]
    ke_I=[0.]

U1=[energy.compute_pair_potential(xd,yd,zd,qd)]
if Efield_on:
    U2=[energy.field_energy(x,y,z,q,[0.,0.,DC])]
else:
    U2=[0.]
if Ion_Force:
    U3=[energy.compute_ion_potential(xd,yd,zd,qd)]
else:
    U3=[0.]

```

```

E=[ke[0]+U1[0]+U2[0]+U3[0]]
energy_output.write("%e\t%e\t%e\t%e\t%e\t%e\t%e\t%e\n"
                    %(0.,ke_E[0],ke_I[0],ke[0],U1[0],U2[0],U3[0],E[0]))

#Main Computational Loop that Calculates the Evolution of the System
start=time.time()
for i in range(steps/resolution):
    #Complete Integration of Newtons Laws for all particle over a time
    #period resolution*dt
    updater.update(xd,yd,zd,vxd,vyd,vzd,qd,md,acc,trans,0.,dt,resolution)
    # Retrieves Data From Device Memory
    cl.enqueue_copy(queue,x,xd.data)
    cl.enqueue_copy(queue,y,yd.data)
    cl.enqueue_copy(queue,z,zd.data)
    cl.enqueue_copy(queue,vx,vxd.data)
    cl.enqueue_copy(queue,vy,vyd.data)
    cl.enqueue_copy(queue,vz,vzd.data)
    # Pauses the Program Until all Items in the Queue have been Completed.
    queue.finish()
    #Calculates the Current System Energies
    ke_E.append(energy.kinetic_energy(vx[0:Ne],vy[0:Ne],vz[0:Ne],m[0:Ne]))
    if Discrete_Ions:
        ke_I.append(energy.kinetic_energy(vx[Ne:N],vy[Ne:N],vz[Ne:N],m[Ne:N]))
        ke.append(ke_I[i+1]+ke_E[i+1])
    else:
        ke.append(ke_E[i+1])
        ke_I.append(0.)

    U1.append(energy.compute_pair_potential(xd,yd,zd,qd))
    if Efield_on:
        U2.append(energy.field_energy(x,y,z,q,[0.,0.,DC]))
    else:
        U2.append(0.)

```

```

if Ion_Force:
    U3.append(energy.compute_ion_potential(xd,yd,zd,qd))
else:
    U3.append(0.)
E.append(ke[i+1]+U1[i+1]+U2[i+1]+U3[i+1])
#Writes Current Energies to a File
energy_output.write("%e\t%e\t%e\t%e\t%e\t%e\t%e\n"
                    %((i+1)*dt*resolution,ke_E[i+1],ke_I[i+1],ke[i+1],
                    U1[i+1],U2[i+1],U3[i+1],E[i+1]))
#writes a file with the positions and velocities of all particles
dataf=open("data/data file%d.txt" %(i+1),"w")
for j in range(N):
    dataf.write("%e\t%e\t%e\t%e\t%e\t%e\n"
               %(x[j],y[j],z[j],vx[j],vy[j],vz[j]))
dataf.close()
#To keep track of simulation progress
print i, time.time()-start
#To make sure Energy is Being Conserved as the Simulation Progresses
print E[i+1]-E[i]

```