Improving Molecular Dynamics Simulations of Ion Coulomb Crystals in Linear Paul Traps

by

Kevin Loeffler

A thesis submitted to the

Faculty of the University of Colorado in

partial fulfillment of the requirements for the degree of

Bachelor of Science in Engineering Physics

Department of Physics

2017

Committee Members:
Thesis Advisor: Professor Heather Lewandowski, Physics
Honors Council Representative: Professor Paul Beale, Physics
Professor Tiziana Vistarini, Philosophy

This Honors Thesis defended on April 7th, 2017

This thesis entitled:
Improving Molecular Dynamics Simulations of Ion Coulomb Crystals in Linear Paul Traps
written by Kevin Loeffler
has been approved for the Department of Physics

_____
Prof. Heather Lewandowski, Physics, Thesis Advisor

_____
Prof. Paul Beale, Physics, Honors Council Representative

_____
Prof. Tiziana Vistarini, Philosophy

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Loeffler, Kevin (B.S., Engineering Physics)

Improving Molecular Dynamics Simulations of Ion Coulomb Crystals in Linear Paul Traps

Thesis directed by Prof. Heather Lewandowski, Physics, Thesis Advisor

Molecular dynamics simulations of ion Coulomb crystals in linear Paul traps allow for the extraction of properties such as ion temperature and ion counts enabling the investigation of ion chemistry in the ultra-cold regime. These simulations reproduce experimental CCD images using density plots of simulated ions allowing for comparisons to be drawn between simulated and experimental results and providing information inaccessible to the researcher, such as the locations of clouds of non-fluorescing species. In this thesis, a method for creating these simulated CCD images is presented, as well as the demonstration of a fourth order integrator for producing more accurate calculations of the ion energy. The simulation of a single frequency Paul trap is presented and the ability to match simulations with experimental parameters as inputs to experimental results is demonstrated in order to show that the simulations that have been constructed can be used to analyze experimental data and work as a predictive lab tool for future explorations of ultra-cold chemistry.

Acknowledgements

Firstly, I would like to thank my thesis advisor Professor Heather Lewandowski for the chance to work and to learn to perform research in a lab that is on the cutting edge of ultra cold chemistry. The skills I have developed over the last four years will be invaluable in my career.

I would also like to thank my fellow lab members. Dr. Philipp Schmid was incredibly helpful in helping me to understand the physics of ion trapping and laser cooling and frequently met with me and reviewed my code throughout the development of this project. James Greenberg and Kyle Miller were always available for me to ask questions and for support and were both helpful in coming to understand the physics of ion trapping and ultra cold chemistry themselves. Cam Straatsma helped me to debug some of the stranger and more subtle problems in the simulations I present and contributed his extensive knowledge of molecular dynamics simulations to helping me succeed in this project.

Thank you to my committee members, all of whom have touched my undergraduate career in meaningful ways and have taken time out of their busy careers to help me become a better scientist.

I would also like to thank my parents for their support throughout my undergraduate career.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

The reactions of ions and molecular radicals in the interstellar medium (ISM) is of great interest to the astrophysics and chemistry community as these reactions make up the foundations of understanding the processes that occur in the ISM. These objects, which have been observed using spectroscopy in the ISM, are difficult to study under conditions similar to those found in the ISM, namely high vacuum and low temperatures, and as a result reactions between these objects have not yet been studied in great detail [27]. This extreme environment can be reproduced in the lab using ion traps and laser cooling techniques, and have been demonstrated previously for reactions of ions and neutrals [14, 15]. These investigations utilized molecular dynamics simulations as components of their studies in order to extract ion counts as well as reveal the positions of ions that could not be directly imaged using CCD cameras. Sympathetic cooling has allowed for the chemistry between ions and molecules to be viable as laser cooled ions can cool co-trapped species to temperatures of a few Kelvin or lower [15].

In order to begin to study the chemistry between molecules and ions in the ISM computational tools must be developed and improved in order to be able to characterize reaction rates and extract useful data about trapped and reacting species in ion traps. Charged particles in ion traps form ion Coulomb crystals, which are one component plasmas that are governed by classical electromagnetic interactions between charged particles that are confined, for example in an ion trap or occurring naturally in exotic objects such as neutron stars [7, 25]. These crystals are advantageous for studies of chemistry that may occur in the ISM because they are clean sources of ions and molecules that can

be reacted at very low temperatures. Single component ion Coulomb crystals have temperatures of less than 10mK [7, 25]. These crystals range in shape from spherical to cigar-shaped with *bcc* and *fcc* character appearing in their structures [7]. Experimentally crystals of a single laser cooled species, often a species such as calcium that has advantageous electronic transitions that make them simple to laser cool, are allowed to develop while experiencing laser induced florescence which allows them to be observed using a CCD camera. The other reactant can then be leaked in and the reaction rate can be characterized by watching the fluorescing species disappear, as well as through spectroscopy of the contents of the trap. Molecular dynamics is then an important tool for producing images of the positions of the non-fluorescing species, as well as for characterizing properties of the crystal such as temperature and ion counts that are critical for determining reaction rates.

The study of Coulomb crystals of laser-cooled ions using molecular dynamics simulations has been established as a powerful method for extracting properties such as ion temperatures and counts from experimentally produced images of Coulomb crystals in linear Paul traps [15, 31, 18]. Molecular dynamics simulations designed to solve Newton's equations of motion for both sympathetically cooled and laser cooled ions can be used to produce simulated CCD images of ion Coulomb crystals perpendicular to the trap axis. These images can then be compared directly to experimental CCD images in order to extract information about the ion energies and also reveal the probable locations of the sympathetically cooled species that are not undergoing laser induced florescence (LIF) [31]. Further, simulations can be used as a predictive tool for experimental conditions for special types of Coulomb crystals [16]. In previous studies, simulations have been used to predict experimental parameters for which planar or spherical Coulomb crystals can be produced. In this thesis, we will briefly consider predictions of experimental parameters for which sympathetic cooling is observed between two species co-trapped in a linear Paul trap.

Ion Coulomb crystals are valuable structures for studies in a variety of research areas, such as quantum information, geophysics, exotic stellar objects, and ultra-cold chemistry [7]. For our purposes, we utilize ion Coulomb crystals to study ultra cold chemistry, defined as the chemistry that occurs between 1mK to 1K [3]. Ion Coulomb crystals allow for species that are not easily cooled

using lasers to be cooled sympathetically, which is where two species are trapped simultaneously, one of which is laser cooled and will cool the other. From simulations and experimental observations it is possible to extract a reaction rate between the two species [14, 7]. This is possible because the simulations will tell you the locations and numbers of the sympathetically cooled species which will not fluoresce (and therefore cannot be characterized directly from CCD images like the laser cooled species can be), and experimentally you can record the reduction in size of the crystal as the fluorescent species reacts away to characterize the reaction rate. This experimental method made possible by ion Coulomb crystals and a robust molecular dynamics simulation enables investigations of ion chemistry in a new regime, that of ultracold chemistry [7].

This thesis describes the enhancement to an existing molecular dynamics suite developed by the Softley group at Oxford University called Coulomb Crystal Molecular Dynamics, or CCMD. The work described includes the replacement of the existing second order Verlet integrator with a fourth order Position Extended Forest-Ruth Like (PEFRL) Integrator, the development of an imaging scheme that directly corresponds with CCD imaging, the ability to match experimental images of crystals to simulated images, and a discussion of the implications of the data collected from these simulations for future experiments. The remainder of Chapter 1 provides background on the experimental apparatus used in the lab and that our version of CCMD simulates, as well as providing an introduction to the theory behind the simulations and of ion Coulomb crystals. Chapter 2 and Chapter 3 provide details about the new integrator and imaging algorithms added to CCMD and characterize their benefits. Chapter 4 details changes to the existing simulation model that enhanced the accuracy of the simulations, such as changes to the heating model and the addition of features to improve efficiency. Chapter 5 presents conclusions that CCMD has been improved in terms of physical accuracy as well as produces useful predictions of experimental parameters and replicating experimental results. The latter half of Chapter 5 presents conclusions about this work and presents a description of future work to improve the existing model and improve the agreement between the simulations and experimental data.

1.1 Experimental Apparatus

The experimental apparatus consists of a vacuum chamber that has been pumped down to ultra-high vacuum pressures (about 10^{-9} Torr) which contains a linear Paul trap.

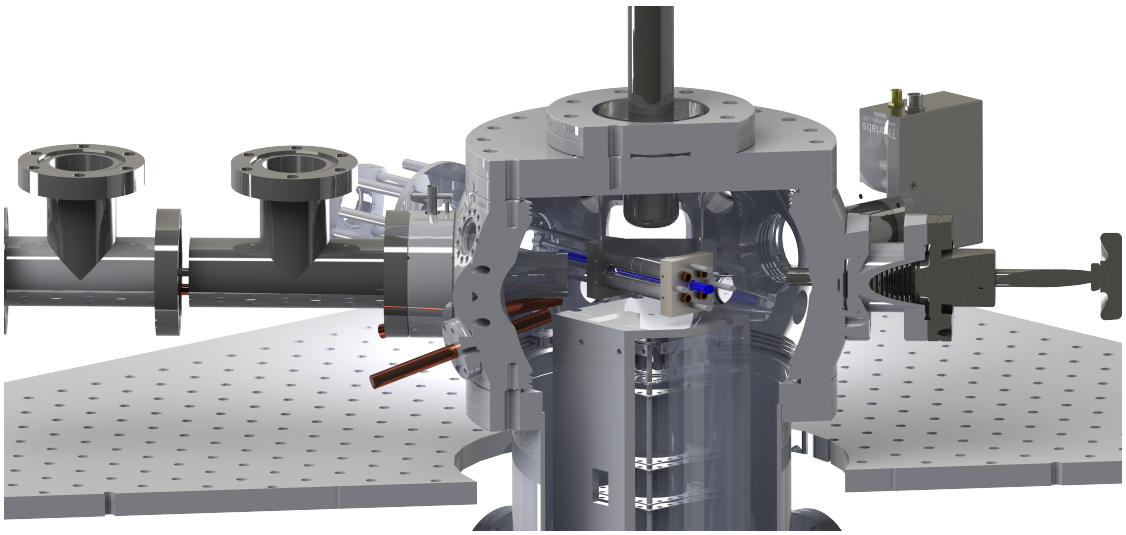

Figure 1.1: This SolidWorks cutout of the interior of the vacuum chamber shows the location of the linear Paul trap CCMD models, held at about 10^{-9} Torr. Shown in blue is a depiction of the beam path from which the cooling lasers propagate into the trap and interact with the ions held in the center of the ion trap. Also depicted is a time of flight mass spectrometer (TOFMS) which is used to determine the contents of crystals and clouds destructively.

The apparatus also contains a calcium oven, which creates a flux of calcium into the chamber by heating solid calcium, and a leak valve for the introduction of gaseous species for reactions. In Figure 1.1 the blue cylinder passing through the center of the trap represents the beam path for the set of cooling lasers used to Doppler cool calcium, which is further discussed in the Laser Cooling section. The primary cooling laser propagates through the trap in both directions, preventing the ions from being pushed to one side of the trap due to radiation pressure. A second cooling laser propagates in only one direction, but is of lower intensity and is there fore a more rare transition for calcium as detailed in the laser cooling section of this chapter. There is also a third laser used for ionizing the calcium. Shown below the trap in Figure 1.1 is a time of flight mass spectrometer (TOFMS), which can be used to determine the contents of the trap. This is a destructive process

that requires putting a high voltage on the top two rods of the trap, forcing the contents of the trap down the tube and causing them to collide with and be detected by MicroChannel Plates (MCPs).The tube entering the chamber from above represents the optics for the Charge Coupled Device (CCD) camera that sits in a recessed glass window, allowing us to image the ions in the center of the trap that are fluorescing due to Laser Induced Florescence.

The general procedure for using this apparatus is to load a cloud of calcium into the chamber, ionize them with the ionizing laser, and then cool them using the set of laser cooling lasers. Reactants can be bled in via the leak valve and the resulting crystal and reaction can be imaged in real time by the CCD. The resulting object can be dumped into the TOFMS for analysis. Images from the CCD can be analyzed using comparisons to molecular dynamics simulations, as described in this thesis, allowing for more information to be collected such as the ion temperatures.

In figure 1.2 we can see the actual beampaths through the chamber for the three lasers we utilize. In order to laser cool calcium two lasers are utilized- 866nm and 397nm- that correspond to transition wavelengths for Calcium 40. These lasers propagate along the trap axis while the powerful ionization laser crosses through the trap center to ionize calcium that has entered the trap from the attached calcium oven.

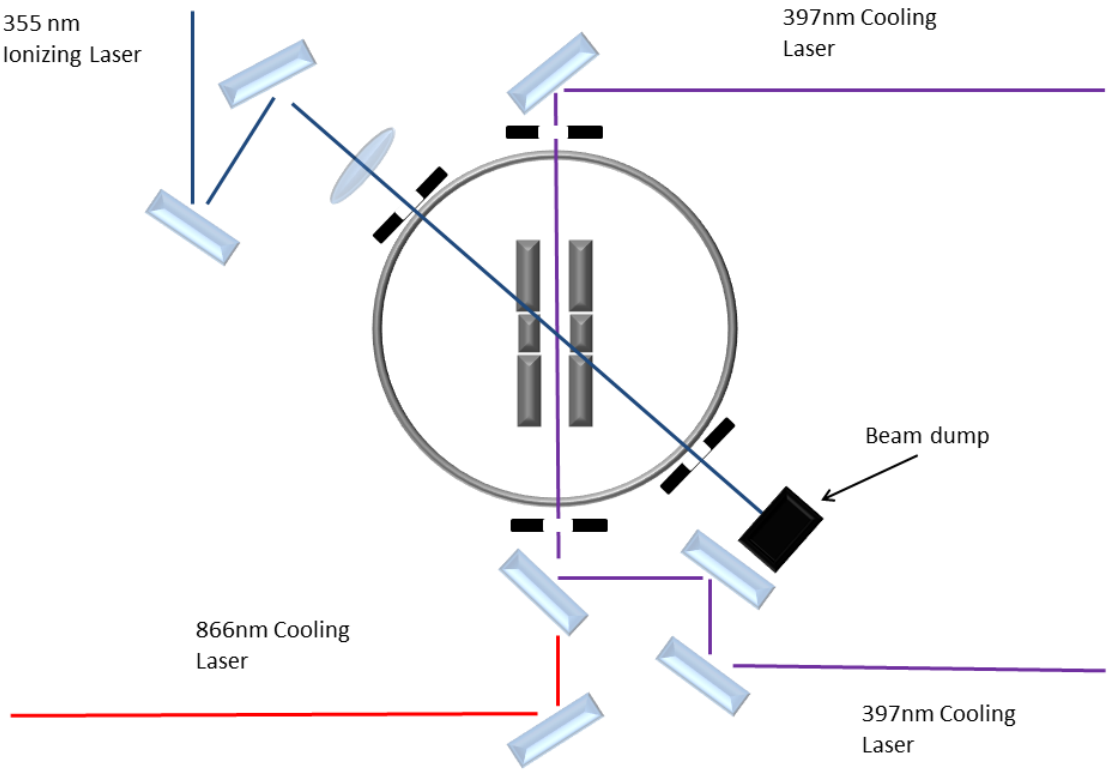

Figure 1.2: Topdown view of the apparatus, showing the path of the ionization laser, cooling lasers, and the placement of optics on the optical table.

1.2 Theory

1.2.1 Ion Coulomb Crystals

Ion Coulomb crystals are spatially-ordered structures that trapped ions form when cooled sufficiently [2].

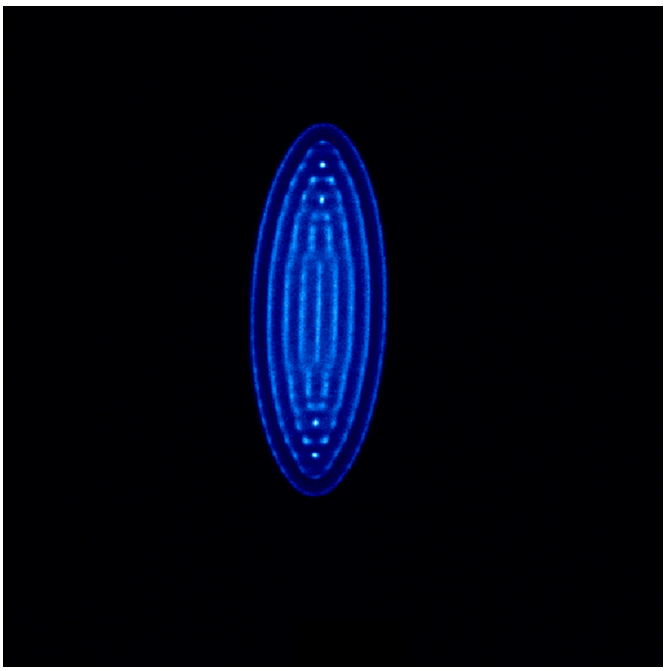

Figure 1.3: Experimental CCD image of a large calcium ion Coulomb crystal.

These structures are not limited to trapped ions however. They were first discussed by Wigner in the 1930s, who considered the formation of regularly ordered structures of electrons in a theoretical experiment where the density of the electrons became so low that their individual wavefunctions did not overlap [7, 28]. Ion Coulomb crystals are thought to make up parts of exotic stellar objects, such as White Dwarfs and Neutron Stars. One such idea of how they may be present in these objects is in the crusts of neutron stars, which are thought to be made of fully stripped iron ions, or Fe^{26+}, the last product from which the neutron star will release energy through fusion [7, 10]. In the laboratory they can be created in Penning or Paul traps. This limits the highest practical ion density to where the ions are separated by 10 microns. The condition for Coulomb

crystallization is given by the plasma coupling parameter, Γ.

$$\Gamma = \frac{E_{Coulomb}}{k_B T} \tag{1.1}$$

Here k_B is Boltzmann's constant and T is temperature. The Coulomb energy is defined as it is typically for classical systems:

$$E_{Coulomb} = \frac{Q^2}{4\pi\epsilon_0 a} \tag{1.2}$$

Here Q is the charge of your particle and a is defined as the Wigner-Seitz radius. The Wigner-Seitz radius corresponds with the radius of the largest sphere containing a single ion [5]. Thus, the plasma coupling parameter can be thought of as the ratio between the Coulomb and kinetic energies of your particle [7]. This ratio determines whether or not your particles will form a Coulomb crystal, and it has been proven through extensive Monte-Carlo investigations of one-component plasmas like Coulomb crystals that the boundary between melting and crystallization is set at $\Gamma \geq 170$ [20].

Because the density is fixed by the use of ion traps in laboratory experiments, a minimum temperature is required to satisfy this criterion for Coulomb crystallization:

$$T \leq\sim 10mK \tag{1.3}$$

This condition comes directly from solving for temperature with the 10 micron spacing limit imposed by the trap using equation 1.1. This requires laser cooling to achieve, which is why ion Coulomb crystals in the laboratory were realized only after the development of laser cooling techniques. These crystals were first demonstrated by Dr. Wineland, NIST, Boulder, USA and by Walther at the Max-Planck Institute for Quantum Optics, Germany [10, 6, 7]. For standard laser Doppler cooling the minimum temperature is given by a range between 0.1 and 1 mK, much lower than could be achieved previously by other techniques such as using liquid helium [7].

Ion Coulomb crystals have been shown via numerical simulations to form BCC structures in their lowest possible energy states. This requires an infinite ion Coulomb crystal however, so

realistically in the laboratory they form spherical and cylindrical shells as can be observed in 3.2.2 [25]. In the lab it is also possible for crystallization to occur at smaller values of Γ, where central parts of the crystal will crystallize but not the outer parts [25].

1.2.1.1 Laser Cooling

Laser cooling is the method by which the ions in the trap are brought to low enough temperature such that the thermalize and form ion Coulomb crystals. It was originally proposed by David Wineland and simultaneously by Hans Dehmelt in 1975, and utilizes photons from lasers to excite electrons and decrease the momentum of atoms [5]. In our experiment calcium is utilized, however laser cooling is a common laboratory technique for cooling any system of two or more levels, such as barium ions [5, 23]. These systems are typically chosen because of the simple laser cooling scheme they make possible, for example with calcium just two cooling lasers are needed in order to cool the ions in the trap to millikelvin temperatures. The three energy levels utilized are shown in Figure 1.4.

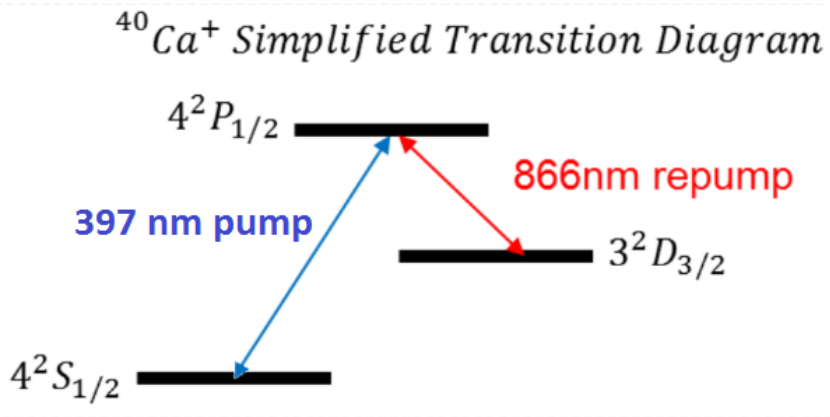

Figure 1.4: Shown are the Doppler cooling transitions for the Calcium 40 ion. Lasers corresponding with these wavelengths are used to cool the calcium ions in our Paul trap.

The idea behind Doppler laser cooling is to use light slightly detuned to the red, or lower energy, side of the electronic transition of the atom. For calcium the laser wavelengths that work best for the energy levels shown in Figure 1.4 are a 397nm laser which brings the electrons to the

P state, and an 866nm laser which is used to bring electrons back to the P state if they fall into the D state, which they will 7.5% of the time [23, 24]. While this probability is small, it is required because if the electrons fall into the D state and there is no 866nm laser present, cooling will cease for all ions for which the electrons fall into the D state, and since the idea is to excite the atoms often this eventually would prevent you from laser cooling your cloud. Another consequence of the small probability of the electron falling into the D state is that we only need to pass the 866nm laser through the apparatus in one direction, since it will have a very small effect on the atoms in comparison to the 397nm cooling laser. With the laser detuned to the red the ions in the trap will absorb more photons as they move towards the source, and since the laser propagates in both directions this means they will absorb more photons from the beam counter to the direction they are moving. Thus, with the counterpropagating lasers the ions in the center of the trap will absorb photons and then spontaneously emit a photon in a random direction, decreasing their momentum by $\frac{h}{\lambda}$. This photon recoil is modeled in our simulations as small kick in a random direction. As this absorption and recoil process will be repeated many times, the ion will eventually decrease in kinetic energy and therefore cool. This allows a temperature sufficient for ion Coulomb crystallization to be met [5, 23, 24].

1.2.1.2 Sympathetic Cooling

For species that do not have a simple level structure that can be exploited easily for laser cooling, or to decrease the complexity of the experiment, molecules and atoms that are not laser cooled can be cooled by a laser cooled species sympathetically. This is achieved through their mutual Coulomb interactions [31, 5, 11]. The linear Paul trap will separate species based on their charge to mass ratios, confining more massive particles to the outside of the clouds of smaller particles on the interior of the trap. Thus, species of similar charge to mass ratio have better energy transfer than those of highly disparate charge to mass ratios as they will interact more inside the trap and their clouds will overlap [31]. Eventually, after equilibrating, the ions will reach a stable species-dependent temperature [11]. In CCMD, the ions initially reach the same

temperature before being heated to their equilibration temperature since at the beginning of the simulation their are forced into their lowest-energy configuration for stating positions [8, 22].

1.2.2 Linear Paul Trap

A linear Paul trap generates an oscillating electric field to trap and confine charged particles [21]. It was originally developed by Wolfgang Paul, who won the Nobel Prize in 1989 for his work on RF ion traps [19]. The frequency of the oscillations is at radio frequencies, hence the alternate name for these traps being RF ion traps. The average force on a charged particle, such as an ion, in these oscillating fields will either be convergent, bringing the particle into the area of lowest potential at the center of the trap, or divergent causing the particle to leave the trap [21]. A DC potential from a pair of endcaps, situated on the ends of the trap, allow for a third dimension of trapping. The potential inside the trap can be thought of as a saddle, where the edges of the saddle are constantly rotating. If the frequency of these oscillations is tuned properly for the object sitting in the middle of the saddle, it will essentially see four walls around it, confining it inside the trap [19]. A cartoon of this potential is shown in Figure 1.5.

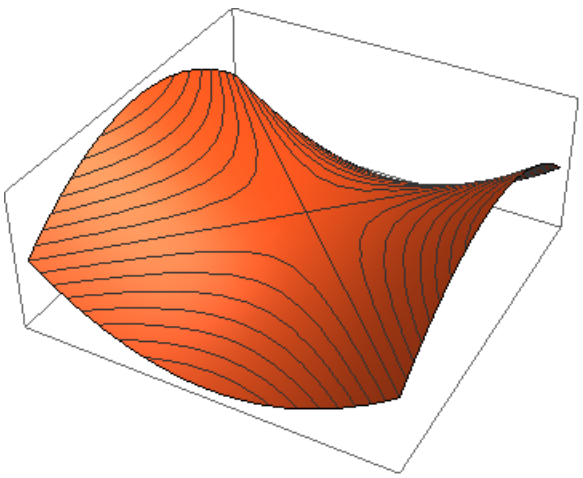

Figure 1.5: Cartoon of the trapping potential inside the trap at a single instant in time, demonstrating how there is a saddle point in the trap center where the ions sit. The radio frequency voltage applied to the trap rods causes this saddle potential to rotate, such that on average the ions are surrounded by a trapping potential that keeps them confined.

For our specific trap, we have a quadrupole potential formed by four trapping rods. At the end of these rods we have two endcaps which apply the DC field. This design allows for us to understand the potential inside the trap analytically. The rods are multi-segmented, and each will carry a given potential that will change in time such that on average the ions inside are confined. A diagram of how these potentials are distributed at a given point in time is shown in Figure 1.6.

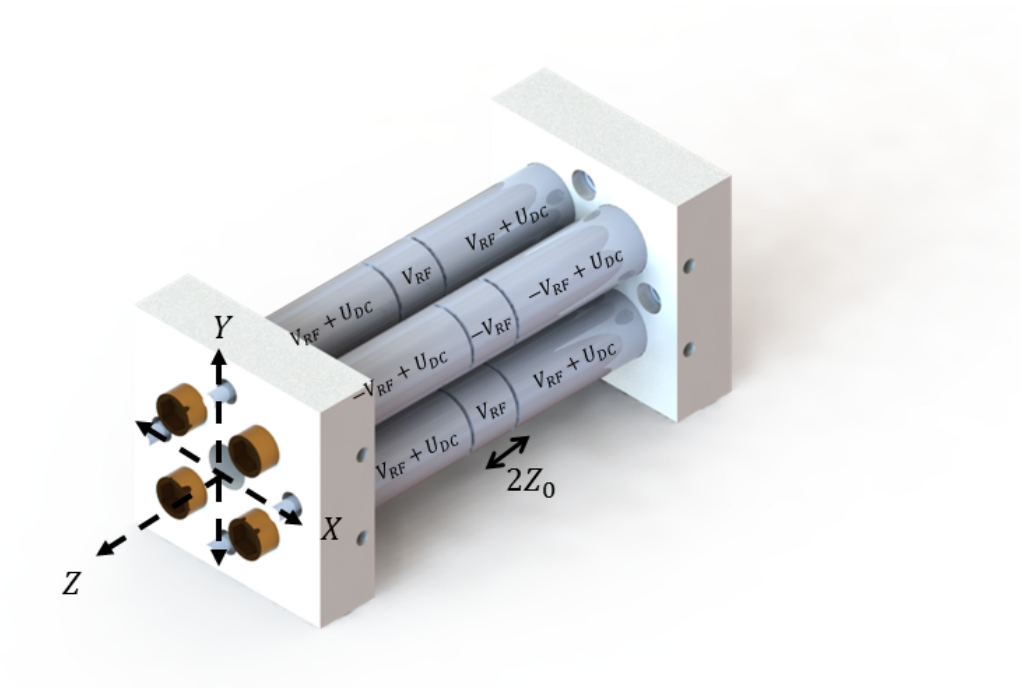

Figure 1.6: Diagram of a linear Paul trap displaying the voltages applied to each segment of the trap rods, as well as the endcaps. Also displayed is the coordinate system for which the equations describing the potential created by the trap is displayed.

Also shown in 1.6 is the coordinate system chosen for analyzing the potential inside this trap. We define two geometric quantities that describe the spacings of the rods which we will use in our

mathematical model of the trapping potential and later in the equations of motion for the ions inside this trap. These quantities are R_0 and Z_0, which tell us about the spacing between the rods and the depth of the rods in the trap center where the ions will be confined. These quantities are defined as shown in Figure 1.7.

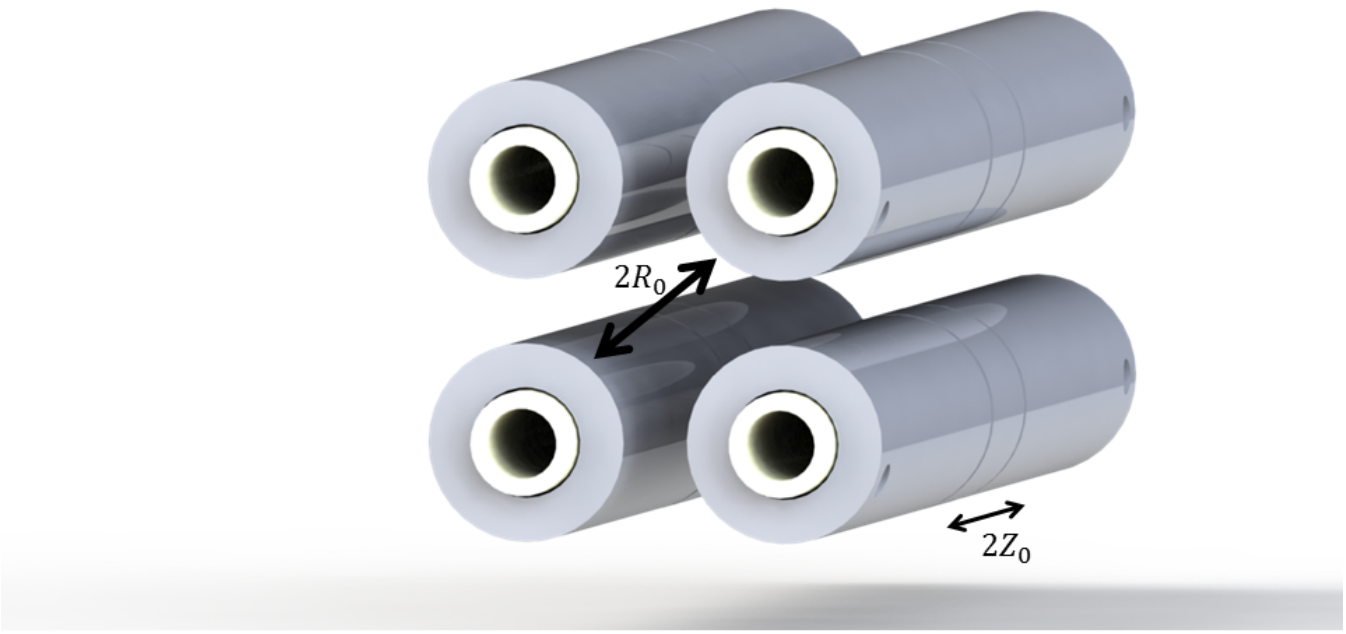

Figure 1.7: Diagram displaying just the trap rods in the linear Paul trap, displaying how the segments are connected as well as how the dimensions of the trap rods are defined.

For our linear Paul trap, $R_0 = 3.92mm$ and $Z_0 = 3.5mm$. With these quantities defined we can give an analytic description of the potential in the center of the trap as a function of position and time. With an RF voltage applied to the rods $V_0 \cos(\Omega t)$ we have the potential shown in

Equation 1.4 which is a superposition of the potential from both the rods and the endcaps, which are set at a DC voltage $V_{Endcaps}$ [21, 31]. The RF frequency is defined as Ω and a geometric constant determined by the arrangements of the trap electrodes is given as η.

$$V(x,y,z,t) = \frac{2V_0}{r_0^2}(x^2 - y^2)\cos(\Omega t) + \frac{\eta V_{Endcaps}}{2z_0^2}(2z^2 - x^2 - y^2) \tag{1.4}$$

From this superposition of potentials we can define our equations of motion in terms of the trap quantities as well as the properties of the charged particles. The RF fields provide only radial confinement, so for the equation of motion along the z-axis we have only the effects of the DC fields from the endcaps, as shown in Equation 1.5. In this equation, we define m as the mass of the particle and Q as that particle's charge [21, 5].

$$\frac{d^2z}{dt^2}m = -\frac{2QV_{End}}{z_0^2}z \tag{1.5}$$

In the x and y directions, we have the contribution from the RF fields as well as that from the DC fields. The result of this superposition is shown in the equation for the x direction shown in equation 1.6.

$$\frac{d^2x}{dt^2}m = -Q\left(\frac{V_0}{r_0^2}\cos(\Omega t) - \frac{V_{End}}{z_0^2}\right)x \tag{1.6}$$

In the next section these equations of motion will be rearranged such that they can be utilized directly in the simulations to evaluate Newton's laws of motion for each ion in the trap.

1.2.3 Coulomb Crystal Molecular Dynamics (CCMD) Overview

In Figure 1.8 a flowchart of how input data is taken into the simulations and turned into an output image and collection of data files is presented. CCMD was originally developed by the Softley Group at Oxford University for the purposes of creating simulated CCD images of ion Coulomb crystals and modeling various types of traps, primarily linear Paul traps and digital ion traps [4]. CCMD considers ion positions and velocities in response to various forces, such as the trapping

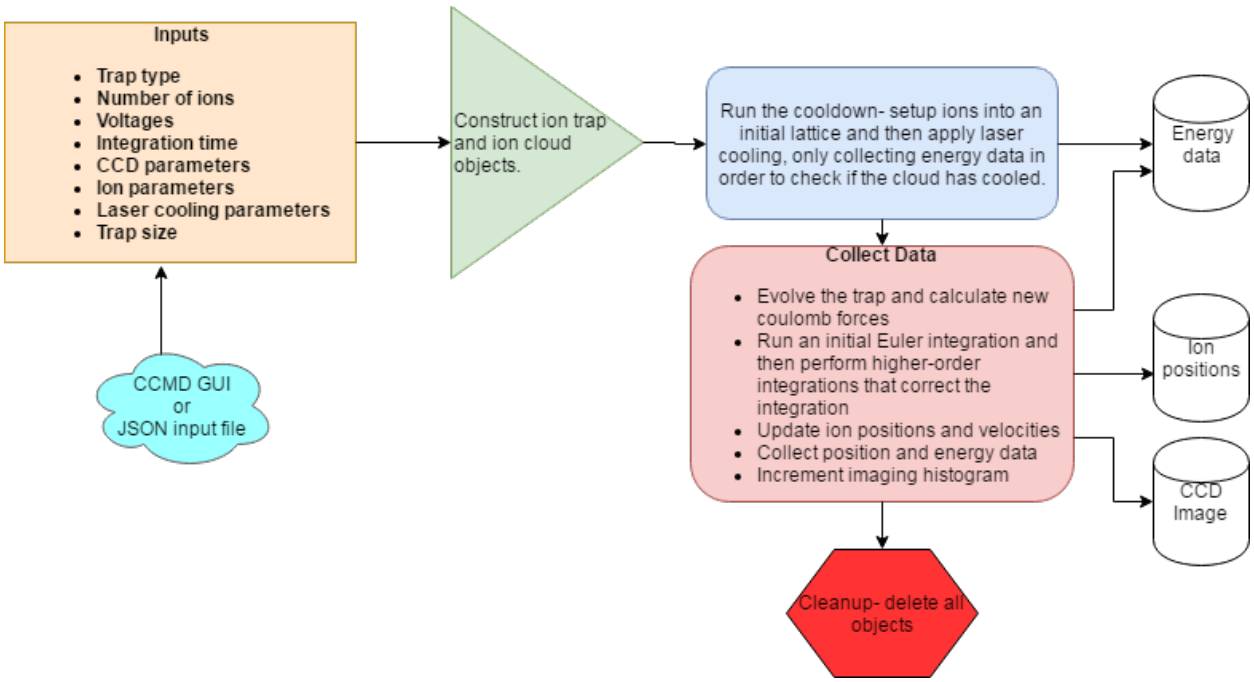

Figure 1.8: Flowchart of how CCMD is used in a laboratory setting, showing what inputs it takes in, operations it takes, and data outputs.

force and the Coulomb interaction with neighboring ions, and originally evolved the simulation using a second order Velocity Verlet integrator. The input parameters used in the experiment, such as voltages on the endcaps, amplitudes for the RF voltage on the trap, the speculated number of ions, depth of field of the camera, and others are input into the program and the effects of all forces on the ions are applied and integrated in time. At the end of the simulation a simulated CCD image is provided along with data files containing ion positions, energies, and velocities in SI units.

The forces applied to the ions are as follows:

$$M_i\ddot{r}_i = F_i, Total(r_1, ..., r_N) = F_{Trap} + F_{Coulomb} + F_{PR} + F_{Laser} + F_{RP} \tag{1.7}$$

In Equation 1.7 M_i is the mass of the i-th ion and r_i is that ion's position. N is the total number of ions or molecules in the simulation. The force applied to each ion consists of that of the force applied by the trap F_{Trap}, the Coulomb interaction with the other ions $F_{Coulomb}$, the photon recoil if the species is laser cooled F_{PR}, the laser cooling force F_{Laser}, and a radiation pressure from the lasers propagating through the trap F_{RP}.

For the trapping force, a set of equations of motion are solved to apply the contribution from the radial trapping and the force from the DC potential of the endcaps. By rearranging the equations of motion from the previous section, Equations 1.5 and 1.6, we can extract two dimensionless parameters that are used to determine the stability of the parameters for the particles in the trap. We also define a new timing constant, τ, to turn the equations into the canonical forms of the Mathieu equations [9, 21].

$$\tau = \frac{\Omega t}{2}, a = -\frac{4QV_{End}}{\Omega^2 m z_0^2}, q = -\frac{2QV_0}{\Omega^2 m r_0^2} \tag{1.8}$$

The rearranged equations of motion are as follows, with an equation for the y direction found using symmetry arguments from the equation for the x direction.

$$\frac{d^2x}{d\tau^2} = (2q\cos(2\tau) - a)x \tag{1.9}$$

For the y direction, by symmetry we have $a_x = a_y, q_x = -q_y$ such that the sign changes. Since the a term is associated with the end caps which act axially, the sign stays the same for both equations of motion [21, 15].

$$\frac{d^2y}{d\tau^2} = (-2q\cos(2\tau) - a)y \tag{1.10}$$

$$\frac{d^2z}{d\tau^2} = 2az \tag{1.11}$$

The z equation of motion is the simplest and is nearly identical to its previous form before parameterization.

There are a number of standard solutions to the Mathieu equations, and only certain pairings of a and q provide stable trapping. The stable regions can be found using Floquet's theorem, however in general in the experimental case the idea is to keep the parameters such that $a^2 < q < 1 \ll 1$ in order to provide stable trapping along the x and y directions for the charged particles in the center of the trap [21, 12].

The Coulomb force is calculated with regard to each other charged particle in the simulation. It is of the classical vector form of Coulomb's law:

$$F_{Coulomb} = \frac{Q_i}{4\pi\epsilon_0} \sum_{j=1, j\neq i}^{N} \frac{Q_j \mathbf{r}_{ij}}{r_{ij}} \tag{1.12}$$

The laser cooling force is modeled as a resistive force as an approximation for calculating the Einstein A and B coefficients and the resultant cooling using quantum mechanics. It is only applied to the laser cooled species in the simulation. This saves computation time and provides an excellent approximation for this process [15, 31].

$$F_{Laser} = -\beta \mathbf{v}_z \tag{1.13}$$

Here, β is set to match the experiment and has units of kg/s. This force is applied only along the z direction to match the configuration of the lasers in the experiment. Cooling lasers along

only one direction provide sufficient cooling due to the mutual Coulomb interaction between all of the ions, leading to a fast thermalization for the ensemble of ions in the simulation [31]. In order to avoid the ions ending in local minima of energy rather than the lowest energy configuration as they do experimentally the simulation is setup to initially cool all of the ions without any heating due to the photon recoil in order to force the ions into initial positions that correspond closely with the minimal energy configuration [8, 22]. For calcium a $\beta = 0.8$ was typically used as it allowed the ions to thermalize quickly. In figure 1.9 the minimization of a 100 Ca ion crystal is demonstrated.

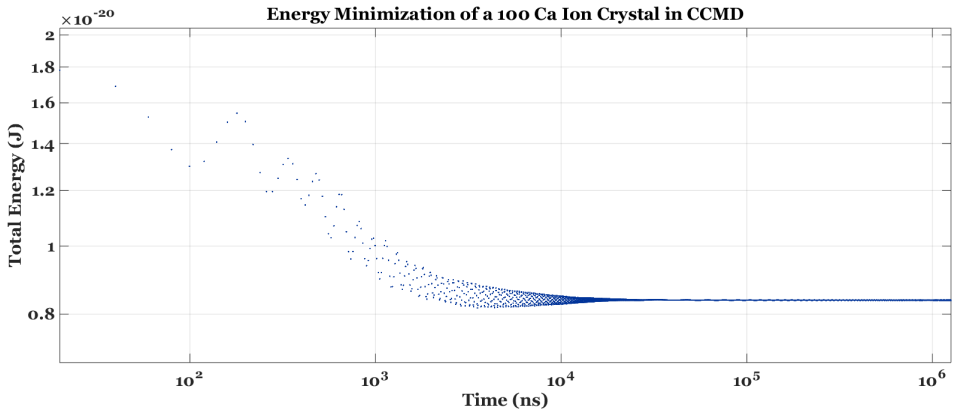

Figure 1.9: Energy minimization for a 100 Ca ion crystal using a laser cooling parameter of $\beta = 0.8$.

The photon recoil force is a random kick that corresponds with the ejection of a photon by the laser cooled species due to Doppler cooling. A small kick of a size set by the user in order to match the temperatures of the ions in the simulation to that of the experimental images is applied to each laser cooled ion each integration step. The kick is given a random direction mapped onto a unit sphere, ensuring that the direction is truly random and corresponding with the actual probability for a photon to be ejected in a given direction by an ion in the experiment. This force effectively raises the temperature and gives a way to increase or decrease the temperature of the observed ions until the images and therefore energies of the simulated ions match those of the experiment [15].

The laser pressure force is also experimentally set and is a small force applied along the z axis to model the radiation pressure on each ion due to the photons from the cooling lasers. This

force is applied in both directions both in the experiment and in the simulation as to not push the ions to the side of the trap [23, 5].

Input parameters also consist of the frequency for which RF voltage applied to the ions, the time they wish to evolve the simulation, an exposure time, an amount of time spent minimizing the ion energy before collecting data, and information about each species in the simulation such as mass, charge, name, and dimensionless parameters that allow the laser cooling and photon recoil to be tuned for each species to match experimental images. These are input into JSON input datafiles and can be collected into folders to be run in bulk to scan entire parameter spaces using a GUI developed for the work in this thesis that allows for increased productivity in running and scheduling multiple simulations.

In this thesis the existing CCMD simulation is extended in a number of key ways. CCMD has been corrected in terms of its model for the photon recoil heating of the laser cooled ions, which originally preferentially caused velocity kicks of low magnitudes that meant that the crystal would become unevenly heated. In the original simulations the energy would not be distributed evenly radially as would be expected; this has been corrected. The images also reflected a great deal of displacement of ions outside where the Coulomb force could explain their positions, which also prompted this work to produce a more robust imaging algorithm such that a single plane of ions could be imaged to rule out displaced ions from other planes appearing in our images as well as prompting a look at the integrator to ensure that the results of numerous force calculations on the ions were accurate over time. The imaging algorithm has been completely rewritten such that the bins in the imaging histogram (and therefore the pixels in the output image) now correspond exactly in terms of size and depth to that of the actual CCD being used to image the crystals, allowing for a higher degree of intuition when comparing simulated and physical CCD images. In order to improve the accuracy of the energy calculation key to collecting the output temperature data of the ions, a fourth order integrator is introduced that allows for better conservation of energy and greater accuracy than the original second order integrator, namely a fourth-order PEFRL integrator.

Chapter 2

Position Extended Forest-Ruth Like Integration

2.1 Design

The original CCMD codebase featured a second-order Velocity Verlet integrator which was used to evolve the simulation in time. Velocity Verlet works by taking an initial half step of Euler integration, wherein the ions started on an initial grid and were given an initial kick, and then were evolved in half steps in which the velocity was calculated for each ion due to the accelerations from the various forces acting on the ion, per Equation 1.7. The position would then be updated once per step [26]. These steps in the simulation corresponded to a fraction of a single RF cycle for the trap, which is on the order of MHz meaning that each step was typically nanoseconds to a few microseconds in length. The original algorithm can be summarized as follows:

$$v_{n+\frac{1}{2}} = v_n + \frac{1}{2}hF(x_n) \tag{2.1}$$

$$x_{n+1} = x_n + hv_{n+\frac{1}{2}} \tag{2.2}$$

$$v_{n+1} = v_{n+\frac{1}{2}} + \frac{1}{2}hF(x_n + 1) \tag{2.3}$$

where h is the time interval between steps, x is position, v is velocity, F is the net force being applied to the ion, and n is your current step [26]. Thus, at any given step we have access to both position and velocity for all of our ions, which is what we need in order to succeed in our

goal of extracting useful data about ion Coulomb crystals such as temperature as having both of these quantities makes the calculation of the total ion energy possible. The Verlet algorithms have the advantage for molecular dynamics simulations in that they are symplectic integrators, which means they are area-preserving, conserve angular momentum, and are time-reversal invariant. This means that the error for our real-time simulations are slow eroding, and should oscillate around the actual value of the ion energy due to being symplectic. Velocity Verlet is well suited for solving the equations of motion for our ions, but it is also simple.

While the second-order Verlet algorithm is simple to implement there are more sophisticated and accurate higher-order integrators that can be used to improve the accuracy of our simulations. Since the crucial result we wish to retrieve from these simulations is the energy of the ions, and because matching ion positions to actual experimental images are the method by which we plan on determining which simulations correspond with the experimental images that were performed, it is necessary that the results be as accurate as reasonably possible. It was observed with the Velocity Verlet algorithm that the fluctuations in the energy around the desired value were quite large, which means that at certain times if the simulation were to be stopped the ions would appear noticeably warmer or colder than they should be in the experiment. This means that the ion positions, too, were occasionally highly disordered and would fail to converge to the same positions as the experimental image. In order to correct this, a higher-order symplectic and time reversible integrator was designed and integrated into the CCMD simulations.

The higher order integrator that was implemented was a modification of the fourth-order Forest-Ruth integrator, which is a symplectic integration algorithm that also allows simulations to know simultaneously the position and velocity of the particles, but has the added benefit that it is more accurate. The integrator's error is proportional to h^4, rather than h^2 for the Velocity Verlet algorithm [17]. However, the accuracy has been further improved by the work of Omelyan et al. who developed a Position Extended Forest Ruth Like (PEFRL) integrator, which uses the same overall method as the Forest Ruth integrator, but adds a fourth force evaluation to the algorithm, which avoids a large time step present in the Forest Ruth approach that causes its error to be higher than

it can be. By scaling from three force evaluations per time step to four to avoid a long time step as part of the algorithm, PEFRL has been shown to be 26 times more accurate than the Forest Ruth approach for the same time step [30]. In the PEFRL approach, the position is first calculated using a fraction of a timestep set by a numerical constant ξ, while the velocity is then calculated with a time step proportional to the numerical constant λ. These constants are summarized in Table 2.1. A position is then calculated using a time step proportional to χ. Midway through the evaluation a step is added to the Forest Ruth approach in which another force evaluation is taken and the position is evaluated using a time step proportional to the sum of ξ and χ. The remaining steps retrace the calculations of position and velocity using the same pattern of numerical constants such that they sum to one full time constant h. In all, there are nine total steps that are summarized below in equations 2.4 through 2.12. The constants in Table 2.1 were determined numerically by Omelyan et al. who developed the PEFRL integrator for molecular dynamics simulations [17].

Table 2.1: PEFRL Constants

ξ	0.178617895844809
λ	0.212341831062605
ξ	0.6626458266981849^{-1}

$$x = x + \xi h v \tag{2.4}$$

$$v = v + (1 - 2\lambda)\frac{h}{2}F(x) \tag{2.5}$$

$$x = x + \chi h v \tag{2.6}$$

$$v = v + \lambda h F(x) \tag{2.7}$$

$$x = x + (1 - 2(\chi + \xi))h v \tag{2.8}$$

$$v = v + \lambda h F(x) \tag{2.9}$$

$$x = x + \chi h v \tag{2.10}$$

$$v = v + (1 - 2\lambda)\frac{h}{2}F(x) \tag{2.11}$$

$$x = x + \xi h v \tag{2.12}$$

The integration method is summarized graphically in Figure 2.1. This flow chart shows how the algorithm is implemented in CCMD and also explains in words what each step is doing.

A detailed listing of the C++ code used to implement the PEFRL integrator can be found in the PEFRL Integrator section of the Appendix.

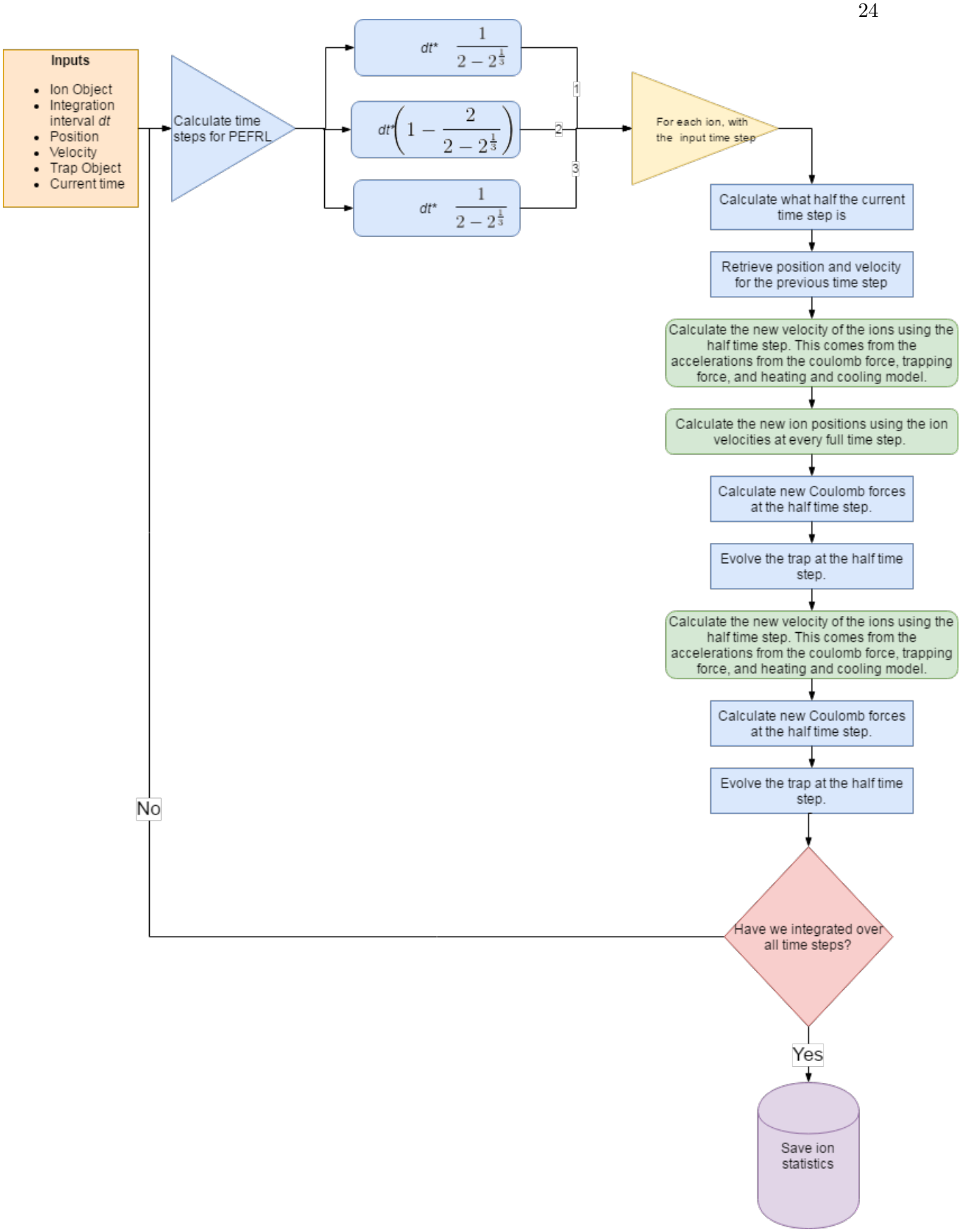

Figure 2.1: Flowchart of the PEFRL integrator written for CCMD, showing the exact flow of data through the integrator and the varying time step used for the integrations it performs.

2.2 Methods

In order to evaluate the effectiveness of the new PEFRL integrator versus the existing Verlet integrator as well as the non-symplectic Fourth Order Runge-Kutta integrator, an objective test using the CCMD simulation program was devised. A Fourth-Order Runge-Kutta integrator was written and briefly considered for use in CCMD due to its speed since it requires fewer force evaluations, however since it is non-symplectic it is not suited to accurately letting us determine ion energies and also has a known problem with energy drift that could cause errors for long-duration simulations [30]. The classical test of whether or not an integrator is conserving energy and behaving properly is to simulate a simple harmonic oscillator. The analog in the world of ion trapping is a single ion in the center of the trap subject to no damping forces. This means that for this test, a single calcium ion was placed in the center of the trap and allowed to evolve with no heating or laser cooling such that energy remained constant. Under these conditions, the ion behaves like a simple harmonic oscillator and will oscillate around the trap center at the secular frequency of the trap, or its macromotion. This is observed in Figure 2.2 in which the ion is oscillating back and forth on the order of nanometers.

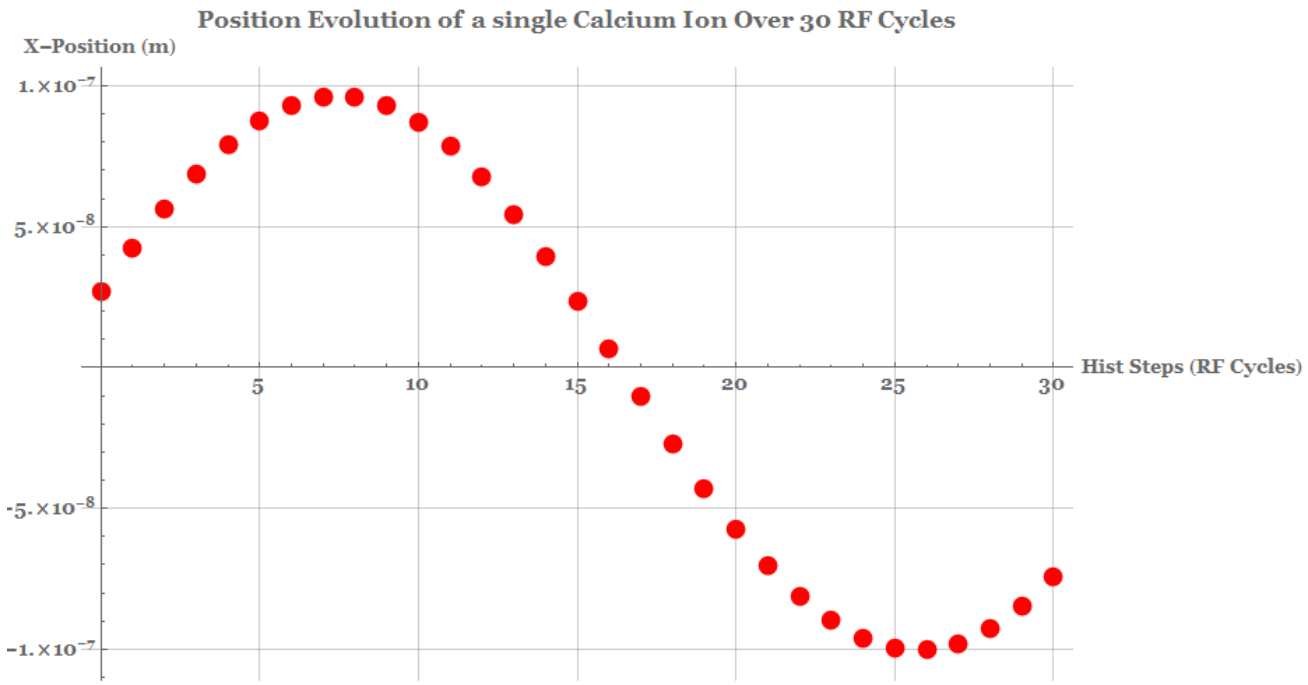

Figure 2.2: For a single calcium ion the simulation was evolved for 30 RF cycles with only the forcing from the trap applied, which had a frequency of 3.14MHz. This created a situation where the ion behaved as a simple harmonic oscillator in the trap. The oscillation frequency is that of the secular frequency, or macromotion, of the ions [13].

2.3 Comparisons with Other Integrators

Subject to this test, identical simulations were performed using the following parameters for each integrator:

Table 2.2: Experimental Parameters for Characterizing Integrators

Experimental Parameters	**Values**
Minimization RF Cycles	2000
Data Collection RF Cycles	2000
Integrations Per RF Cycle	16
RF Frequency	3.14MHz

First, the Velocity Verlet integrator and the PEFRL integrators were compared. As seen in Figure 2.3 the energy fluctuations (taken as the standard deviation of kinetic energy for the ion in the trap) is orders of magnitude larger than that for the PEFRL integrator, implying that the accuracy of the PEFRL integrator has significantly improved the accuracy of the energy calculation in CCMD by decreasing the overall fluctuations in the energy. This means that regardless of when the simulation is to stop, it is unlikely that the simulation will be caught at a local energy minimum or maximum that would cause significant displacement in the ions, making our ability to match simulated ion Coulomb crystals and physical ion Coulomb crystals more accurate.

This result corresponds well with the findings of other research groups that investigated various integrators. For example the MOLDY molecular dynamics suite, used to simulate condensed matter, likewise found that the use of the Omelyan integrator decreased the energy fluctuations significantly resulting in more accurate simulations [1].

In order to demonstrate the need for both a higher order integrator as well as a symplectic integrator, the non-symplectic fourth-order Runge-Kutta integrator was also tested. While this algorithm runs faster due to not needing as many force evaluations per time step, as can be observed from Figure 2.4 the error is still many orders of magnitude larger than that for the PEFRL integrator. While the error in the Runge-Kutta integrator is smaller than that of the Velocity Verlet integrator by two orders of magnitude, it is further unsuitable for use for simulating

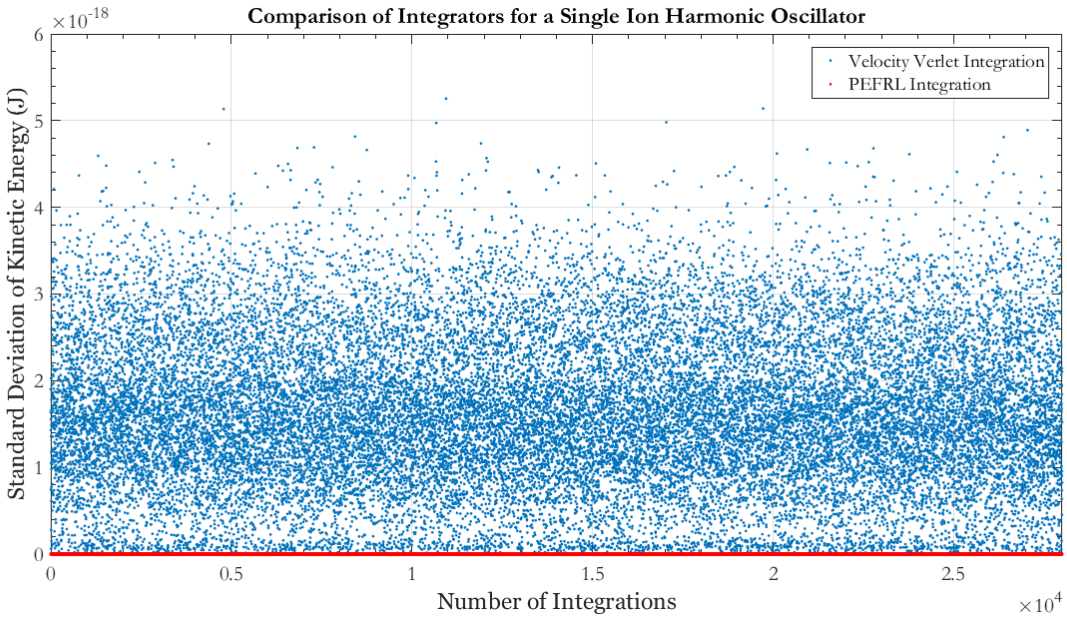

Figure 2.3: For a single calcium ion the simulation was evolved using both the PEFRL integrator and the second-order Velocity-Verlet. The energy fluctuations for this ion, essentially behaving as a simple harmonic oscillator, were measured in the form of the standard deviation in the kinetic energy.

ion Coulomb crystals because it is non-symplectic, and since simulations of ion Coulomb crystals are long-duration real time simulations there would be a high risk of energy drift causing errors in the output data, making matching crystals with this algorithm unreliable. Despite this, this result demonstrates that going to higher-order integrators will decrease the energy fluctuations as desired, albeit not as dramatically as with the PEFRL algorithm.

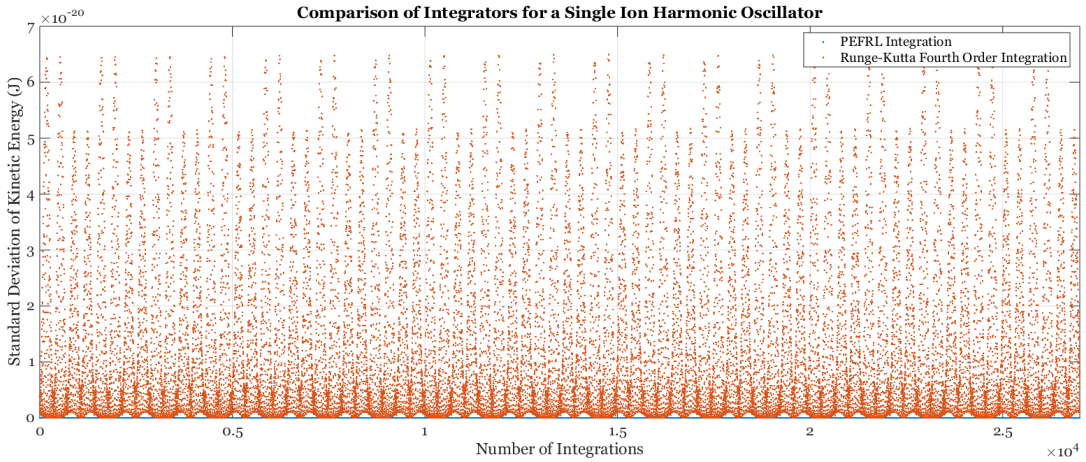

Figure 2.4: For a single calcium ion the simulation was evolved using both the PEFRL integrator and the fourth-order Runge-Kutta integrator. The energy fluctuations for this ion, essentially behaving as a simple harmonic oscillator, were measured in the form of the standard deviation in the kinetic energy.

Chapter 3

Simulated CCD Imaging Algorithm

3.1 Design

In order to match the simulated CCD images of ion Coulomb crystals with their experimental counterparts, a robust imaging scheme is needed to turn the output ion positions from the simulation into physically realistic images. In order to do this, a new imaging algorithm was written for CCMD that allows the imager to be tuned to the physical parameters of the experimental CCD camera.

The imaging algorithm is fundamentally a 2D histogram where each bin is a pixel and has its brightness increased each RF cycle if an ion was in that particular bin during that time. In this way, there is an initial blurring of pixels because the ions will move around — small distances if cold, and large distances if warm. The bin size is set by the pixel size of the CCD. For our camera, we have 16 micron pixels behind a 10x objective lens, meaning that we input into the simulator to have bins that correspond with 1.6 micron pixels. The bins have a "depth" for which the program will screen out any ions that fall in the bin but outside the depth of field of the camera specified by the user. In reality the CCD cannot see ions that are outside its depth of field and is set to look at the central plane of whatever ion Coulomb crystal has been produced. To match this, any ion outside the depth of field envelope, which takes the depth of field and cuts it in half to screen for any ions half the depth of field above and half the depth of field below the y=0-plane normal to the camera, will not increment any bins of the image during the data collection period since they would be effectively invisible in the experiment. For our experiment, the depth of field is 3 microns. This means that the simulated images CCMD produces will always be of ions in the central plane of the

crystal, just like in the experiment. This also means that the images essentially correspond with probability density plots, rather than images of actual particles [31]. This is physically realistic, as the ion sites observed in the experimental crystals also correspond with probability densities, with ions swapping in and out of these ion sites constantly during the observation of the physical crystals. A summary of this algorithm is shown graphically in Figure 3.1.

The main goal of this imaging algorithm is to allow the images taken from CCMD to be compared to the experimental CCD images directly. Since they are the same size and the pixels correspond with the same locations, images from CCMD can effectively be overlayed on top of experimental images to analyze whether or not the simulation matches what was observed experimentally. This makes the process more accurate and less qualitative as now individual ion sites should correspond in simulation in position and observed "size" due to blurring due to heating, allowing for the number of ions present in the simulation to be extracted as well as temperature.

In order to get the blur of the ions physically accurate, a Gaussian blur is applied to the density plot of pixels. In the actual experiment we are limited by the resolution of our camera, so the blur acts as a way to model this resolution by smearing out the ion positions in addition to the natural blurring that occurs because of their motion in the trap. This ensures the images are realistic, and objects are the same size in the simulated image as they would be in reality. The imager features a Gaussian blur which has a width set by the resolution of the CCD camera which is input by the user. This repurposes an existing Gaussian blur function from the original imaging algorithm, which was supplanted by the imaging algorithm used in this thesis in order to provide a better procedure for matching experimental and simulated images. For our particular setup, the resolution is given by $4.5 \pm 0.5 \mu m$. A 1D unnormalized Gaussian function is convolved with each row and column of the image to provide a realistic blurring of the images such that they match the experimental images as much as possible. The form of this 1D Gaussian function is shown in Equation 3.1.

$$f(x) = e^{\frac{x^2}{2\sigma^2}} \tag{3.1}$$

In the imaging algorithm, x would correspond with each pixel in the row or column currently being blurred. At the end of the convolution the blurred pixel brightness is normalized by the kernel sum, which is collected while the Gaussian function is being applied to each pixel. This scales the brightness such that the pixel brightness in the image corresponds well with that of the experimental images by conserving the brightness of the ions such that pixels that would appear bright in the experiment are also bright in the simulated images and dim pixels are also dim in both images. This allows for the images to be interpreted in terms of estimating how long an ion is staying in a given location, or if the ions had not yet crystallized because they were still moving into position.

The blur of these images makes it possible to compare the simulated CCD images directly with experimental images in terms of temperature. The initial blur due to the ion motion reflects the ion energy, and thus by heating or cooling the ions more or less in the simulation images can be matched to their experimental counterparts. If there is a high degree of blurring in the images, the ions moved around more during the exposure period and as such are warmer. If we get small dots representing the ions, they barely moved and correspond with very cold ions. The second blur allows for the images to be compared to the experimental images directly, since they simulate the resolution of the camera.

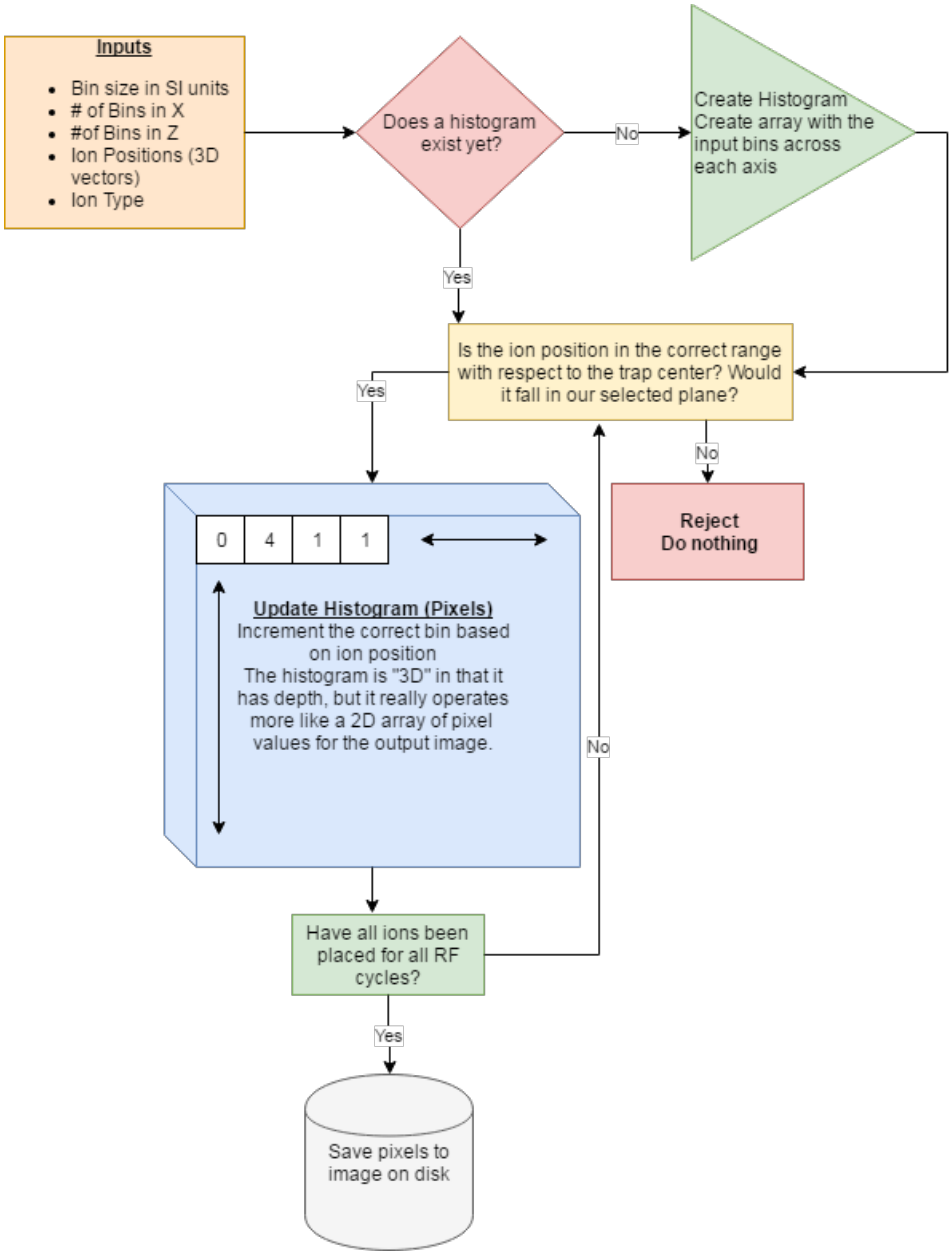

Figure 3.1: Flowchart of how the algorithm written for CCMD produces simulated CCD images that correspond with the physical parameters of the experimental CCD setup.

3.2 Comparing Experimental and Simulated CCD Images

3.2.1 Methods

In order to demonstrate the ability to match experimental and simulated CCD images in CCMD, experimental images were identified from tests of the experimental apparatus and the experimental parameters that were logged for that particular crystal were input into CCMD in order to determine if the simulation produced the same results as expected. For some crystals time-of-flight mass spectrometer data (TOFMS) data were available, which was able to provide ion numbers to be input into the simulation so that all quantities corresponded to their experimental counterparts. For those that did not have TOFMS spectra, a process of refining the number of ions was pursued where the same simulation was run multiple times and the results compared to the experimental image until a good match was identified. For all simulations, the blur was set to be the resolution of the CCD camera at 4.5 microns in radius, so in order to match the blur to the experimental images the photon recoil heating was varied instead in order to warm or cool the crystals sufficiently to the point where the amount the ions moved blurred the image similarly to the experimental image. A demonstration of this process is shown in Figure 3.2.

The quality of the match was judged qualitatively and quantitatively. Qualitatively was the judgment of the amount of blur and therefore the temperature of the ions in the simulation since an actual temperature for the physical crystal was not available for these comparisons due to the experimental difficulty of obtaining this information, and since it is a goal for the CCMD simulations to be a laboratory tool for extracting this information. Additionally, the simulations are not perfect at placing the ions in exactly the same place as the physical crystals. Physically patch potentials can move ions into positions the idealized simulation would not solve for the ion to be, and the simulations further are not perfect and preventing the ions from becoming slightly disordered due to higher order effects such as Coulomb shielding due to the shells of the crystals. Due to this, some leeway is needed in order to match the crystals to their experimental counterparts and physical intuition needs to be utilized. Quantitatively comparing the number

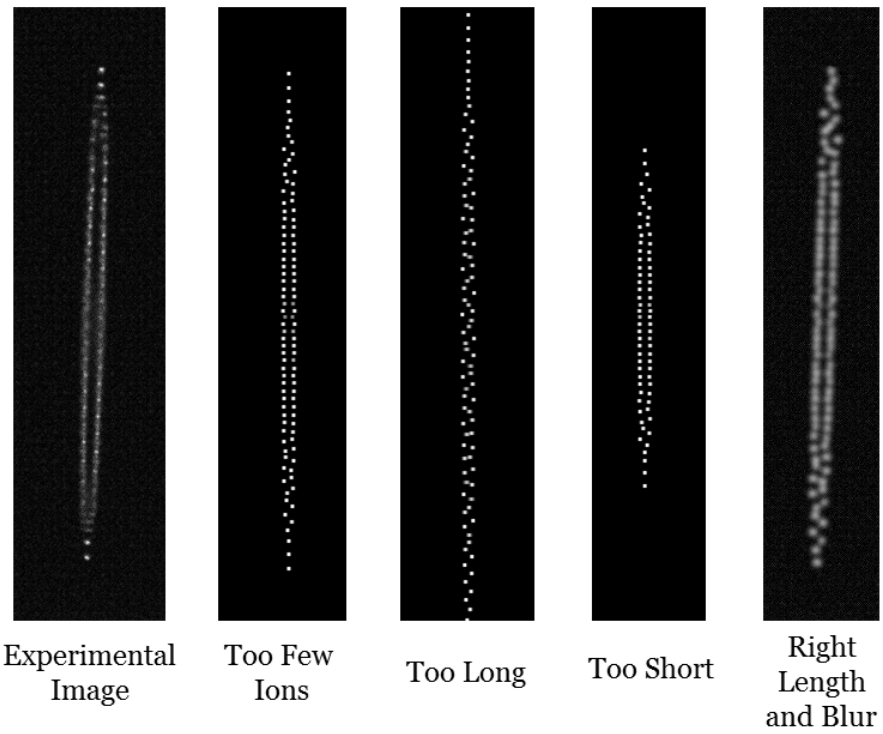

Figure 3.2: Demonstration of the process for matching experimental and simulated ion Coulomb crystals. If the number of ions is not known, the quantity is varied until the crystals can be matched. In order to match the crystals in terms of temperature, the heating of the crystal due to photon recoil is raised or lowered until the blur corresponds with the experimental image.

of ions and the experimental parameters was used as a check in ensuring that the simulations matched their experimental counterparts, as for the same experimental parameters and number of ions the same structures should appear. In this way, the ability for the CCMD simulations to match experimental parameters was demonstrated.

3.2.2 Results

The most simple ion Coulomb crystal observed is a string. A simple string was taken and simulated with the parameters shown in Table 3.1. With the exact input parameters from the experiment input, such as the RF frequency and the endcap voltages, the crystal was simulated. The ion number was known due to the small number of ions in the experimental image. From the simulation, the internal kinetic energy of the system was output and was used to extract a temperature, as shown in Table 3.1. The internal kinetic energy of the system corresponds with the temperature of the ion string, and since the blurring of the ions was exactly the same as the experimental counterpart image it was possible to draw the conclusion that the observed ions were close to the stated temperature [31].

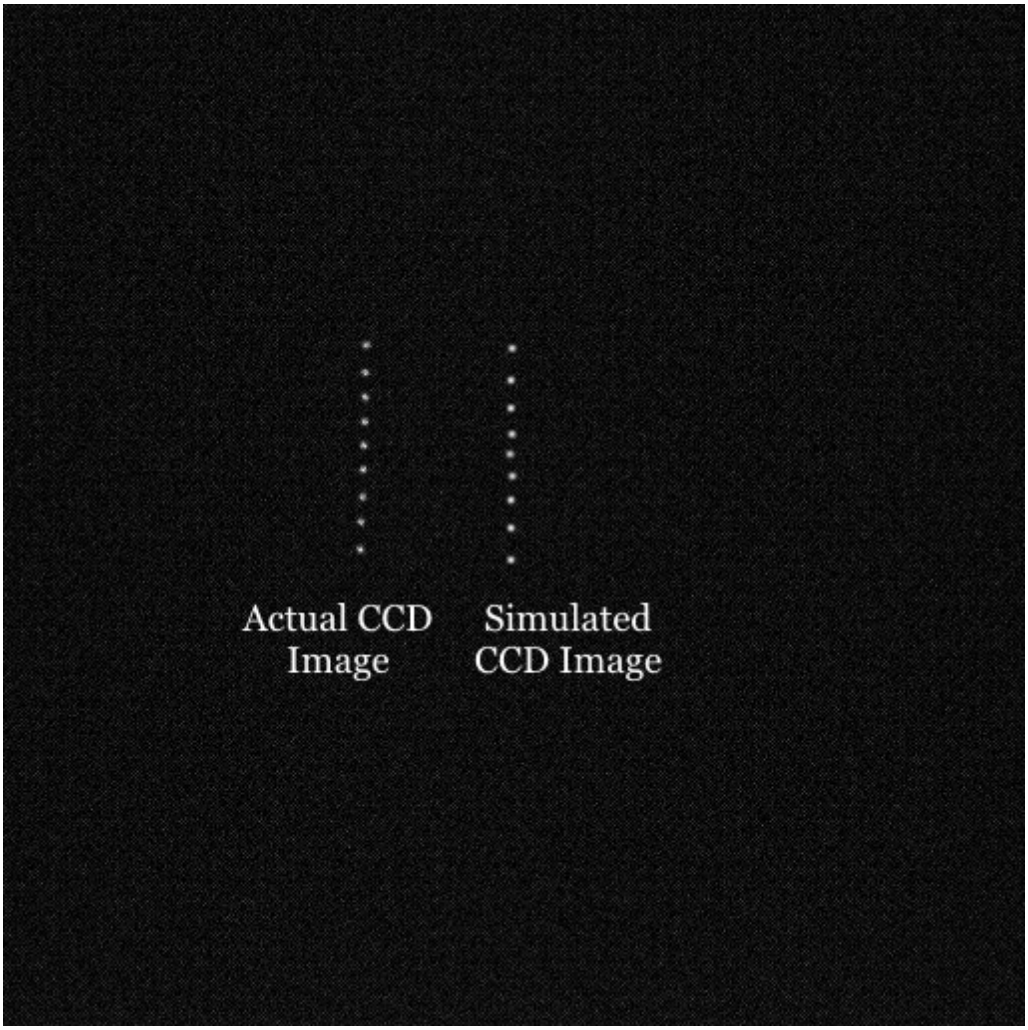

Figure 3.3: Direct comparison between a simulated image and an experimental image of a 9-ion calcium string, demonstrating how for the same trap parameters the simulator will produce accurate ion spacings and positions.

Table 3.1: Experimental and Simulation Parameters for a 9 Ion String

Experimental Parameters	**Values**
RF Voltage Amplitude (Peak to Peak)	250V
Endcap Voltage	1.7V
RF Frequency	6.2876 MHz
Number of Ions	9 Ca Ions
Resulting Temperature	3±1mK

Moving from a string up to a more complicated crystal is one that has a single ring of ions in a zig-zag pattern, as the pattern in Figure 3.2.2 is typically called [7]. For this particular crystal TOFMS spectra was not available as this was an intermediate crystal that eventually grew into a more complicated one, so at this intermediary step the power of CCMD to identify ion numbers was demonstrated. For the same experimental parameters as are summarized in Table **??** the number of ions was varied until the same structure became apparent in the simulated crystal as in the experimental one. From this experiment, it was determined that the experimental crystal must have contained around 90 ions. Note how the same zig-zag pattern is apparent in both the experimental and simulated crystal for the same parameters, demonstrating the ability for CCMD to capture the various structures ion Coulomb crystals tend to take on in their interiors.

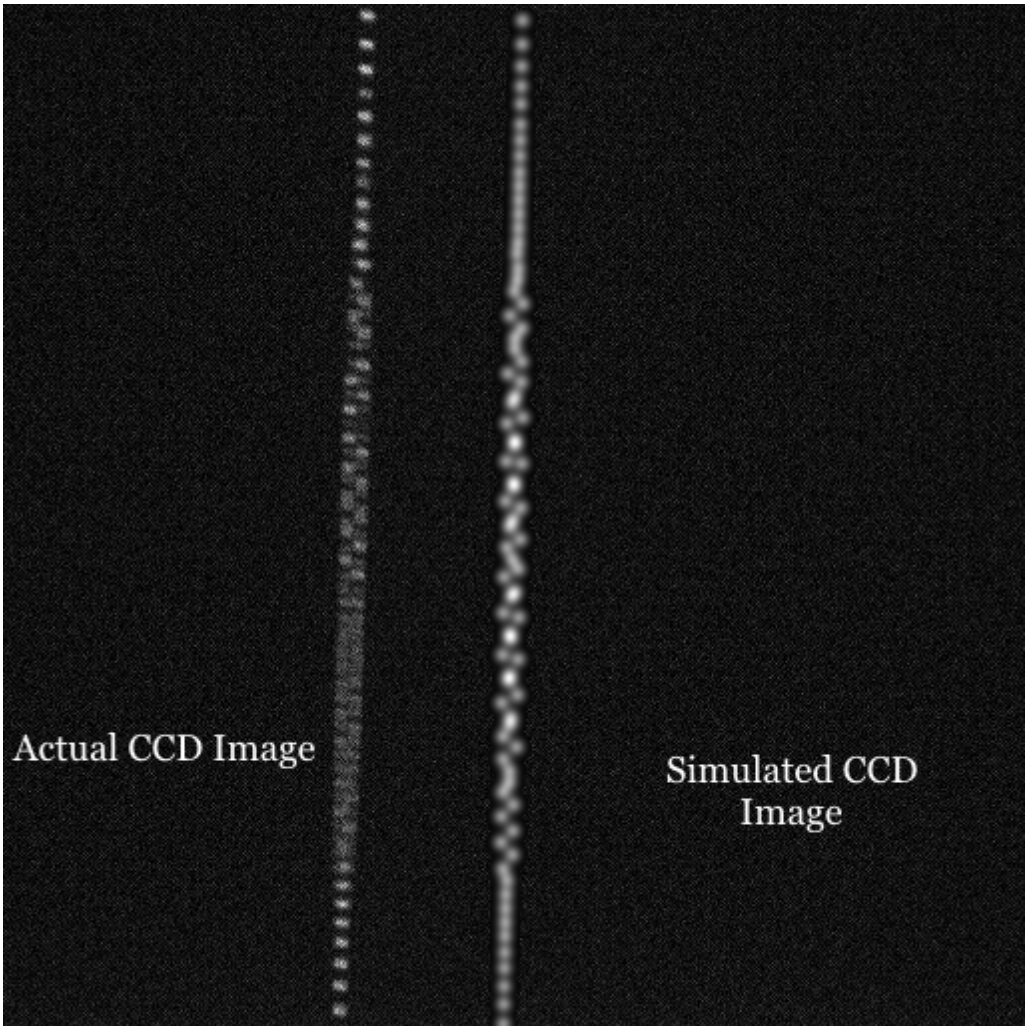

Figure 3.4: Direct comparison between an experimental and a simulated image of a 75 ± 5 ion Coulomb crystal consisting of calcium ions. Ion numbers were extracted from the simulated image to determine approximately how many ions were present in the experimental image.

Table 3.2: Experimental Parameters for an Ion Crystal of 75 Calcium Ions

Experimental Parameters	**Values**
RF Voltage Amplitude (Peak to Peak)	300V
Endcap Voltage	300mV
RF Frequency	6.289MHz
Number of Ions	75 ± 5 Ca Ions

The next most complex crystal simulated is one that is a single shell. Like the previous crystal, this was an intermediary crystal that was imaged while producing a much larger crystal for testing, so the number of ions had to be matched. For this particular crystal, an ion count of 81 ± 5 calcium ions was extracted. The uncertainty in the number of ions comes from the fact that images of around 80 ions have similar structure and determining the number of ions around the edges of the crystals is an uncertain process. This uncertainty scales with the size of the crystal, which larger uncertainties in larger crystals where adding or subtracting ions cause more subtle effects to the crystal structure. Here more qualitative methods were needed to match the crystal, as to match the blurring disorder that began to appear in the top and bottom of the crystal. Overall, despite the disorder in these regions the crystals matched well in terms of size and brightness, demonstrating that the blur can be matched using tuning of the photon recoil parameter. These kinds of errors are common in these simplified simulations of ion Coulomb crystals, and it may be possible that adding some higher order effects could resolve these defect issues in the future [15, 31].

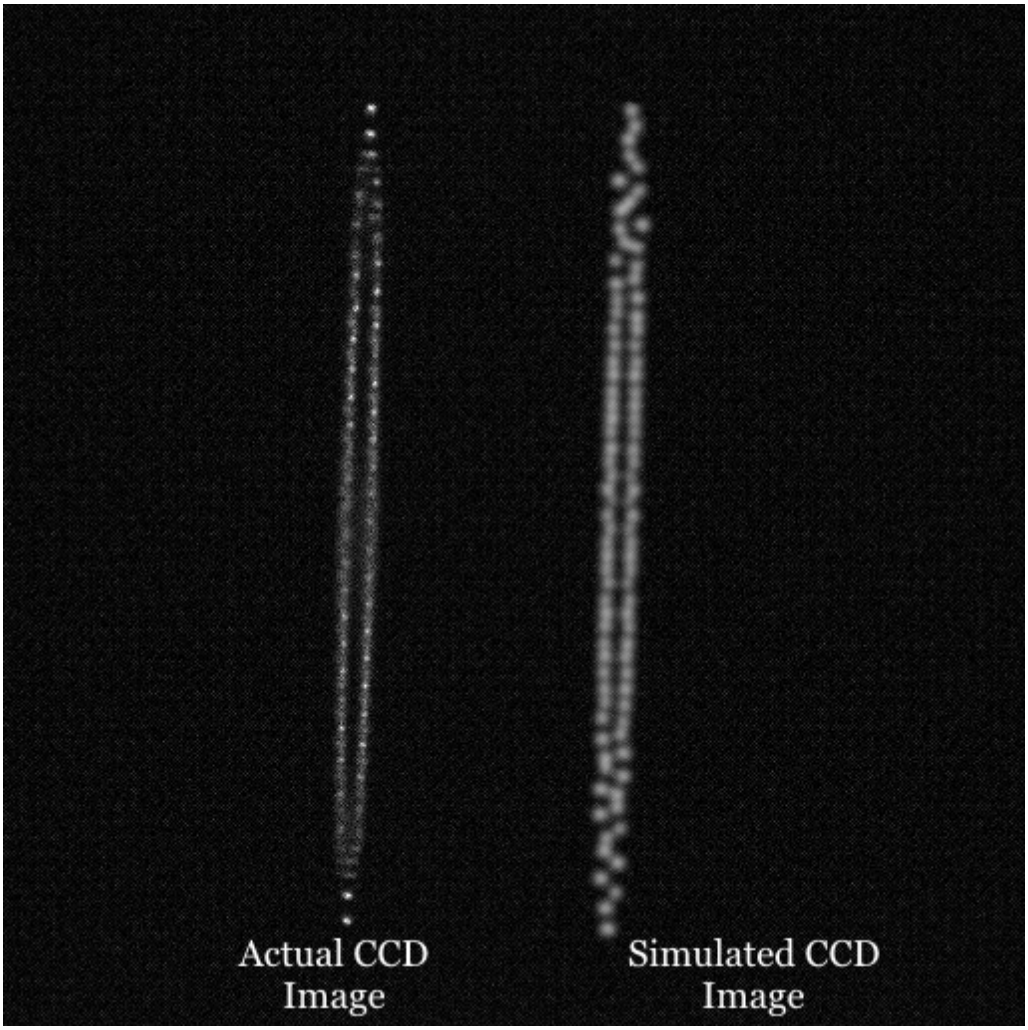

Figure 3.5: Direct comparison between an experimental and a simulated CCD image of an ion Coulomb crystal consisting of 81 calcium ions. Ion numbers were extracted from the simulated image to determine approximately how many ions were present in the experimental image.

Table 3.3: Experimental Parameters for an Ion Coulomb Crystal of 81 Calcium Ions

Experimental Parameters	**Values**
RF Voltage Amplitude (Peak to Peak)	300V
Endcap Voltage	780mV
RF Frequency	6.289MHz
Number of Ions	81 ± 5 Ca Ions

In order to determine how well CCMD handled being given an experimental quantity of ions in addition to the experimental parameters, a crystal with a given TOFMS spectra was identified and simulated. The TOFMS spectra for the crystal shown in 3.6 revealed that 54 Calcium ions were present in the trap when the crystal shown was dumped into the spectrometer. For this given input and experimental parameters, the same structure was revealed with a four ion core and an outer concentric ring of ions, demonstrating that CCMD can take in an experimental number of ions as well as experimental parameters and produce a simulated image that corresponds well with its experimental counterpart. This demonstrates that the simulations are effective at modeling the trap as it is experimentally setup.

Table 3.4: Experimental Parameters for an Ion Coulomb Crystal of 54 Calcium Ions

Experimental Parameters	**Values**
RF Voltage Amplitude (Peak to Peak)	215V
Endcap Voltage	5.42V
RF Frequency	6.289MHz
Number of Ions	54 Ca Ions

Finally, a crystal more typical for our experiment was reproduced, consisting of 255 Ca ions as seen in 3.7. This size of crystal and large crystals are more typical for conducting experiments for characterizing reaction rates. Given a TOFMS spectra that revealed that 255 calcium ions were present in the trap when this crystal was destroyed, a very similar crystal was produced for

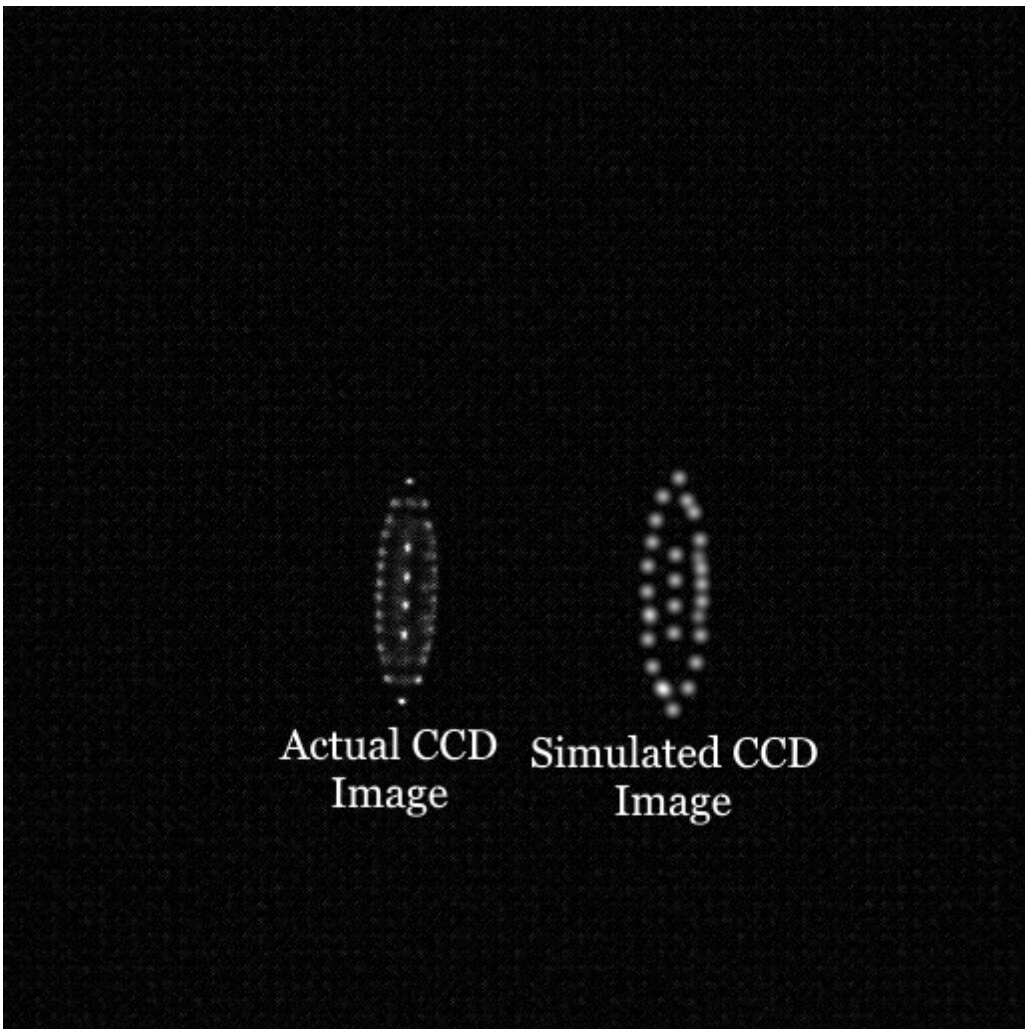

Figure 3.6: Direct comparison between an experimental and a simulated CCD image of an ion Coulomb crystal consisting of 54 calcium ions. The number of ions in the simulated image match the number extracted using a time-of-flight mass spectrometer from the experimental image, demonstrating that CCMD can reproduce crystals with good agreement when given experimentally collected data in addition to the experimental conditions.

the same experimental parameters. Notice that the blurring was able to be matched using the photon recoil parameter and the internal structure, such as the small internal concentric ring at the very center of the crystal was captured by the simulations. The ability to simulate large crystals demonstrates that CCMD can be an effective lab tool for crystals of all sizes and shapes, and is able to reproduce fine details even in more complex simulations such as those of hundreds of ions. It is expected that the simulation accuracy will scale to even larger crystals as demanded by future experiments.

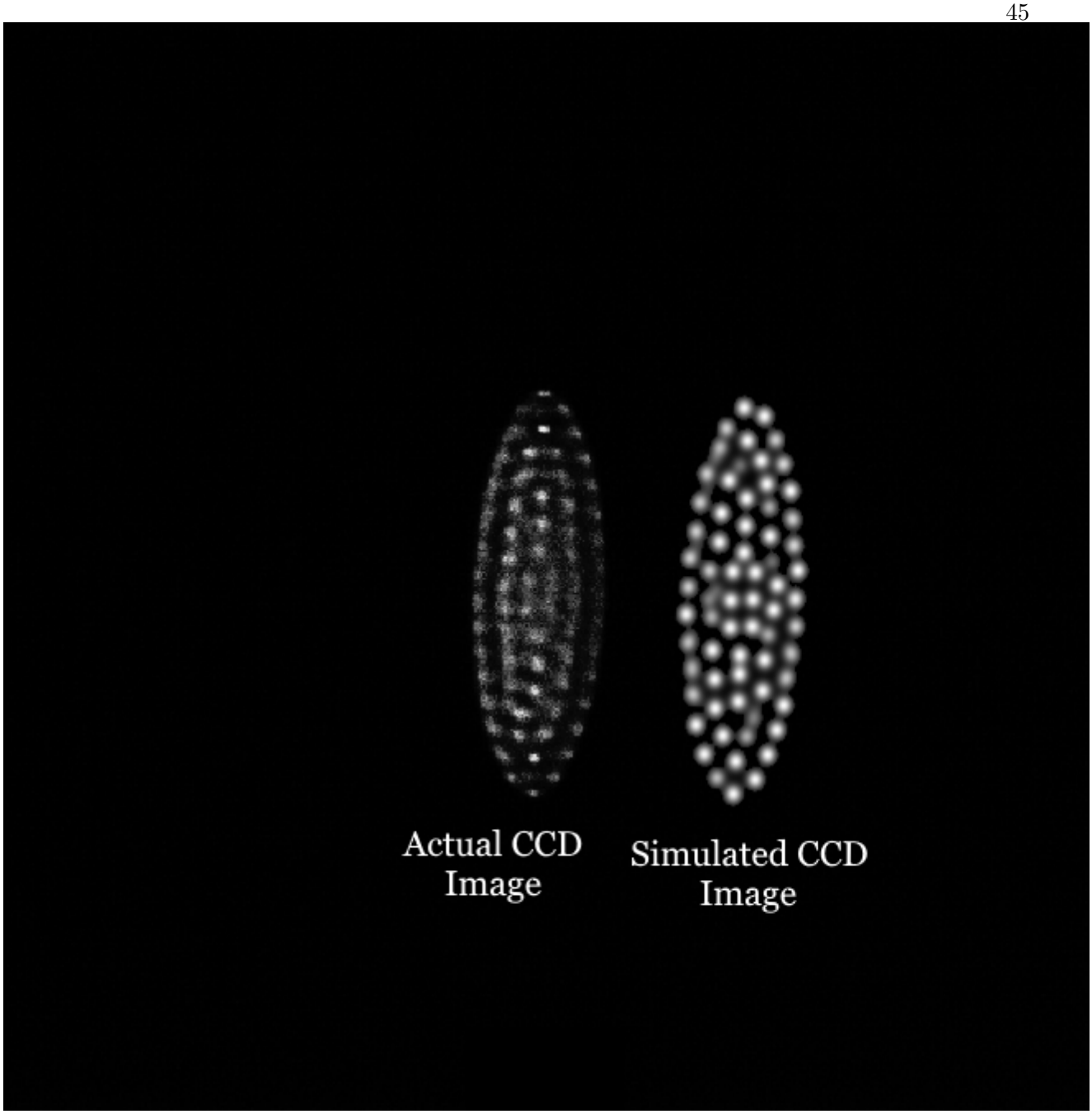

Figure 3.7: Direct comparison between an experimental and a simulated CCD image of an ion Coulomb crystal consisting of 255 calcium ions. The number of ions in the simulated image match the number extracted using a time-of-flight mass spectrometer from the experimental image, demonstrating that CCMD can reproduce crystals with good agreement when given experimentally collected data in addition to the experimental conditions even for large crystals.

Table 3.5: Experimental Parameters for an Ion Coulomb Crystal of 255 Calcium Ions

Experimental Parameters	**Values**
RF Voltage Amplitude (Peak to Peak)	150V
Endcap Voltage	5.45V
RF Frequency	6.2876MHz
Number of Ions	255 Ca Ions
Resulting Temperature	8±1mK

Chapter 4

Simulation Enhancements

This chapter focuses on changes to the simulation that were added that increase efficiency or accuracy outside of the integration algorithm as presented in Chapter 2. This section focuses on how the photon recoil model in the simulations was changed to be more physically accurate and quantifying its effects on the simulation, a demonstration of how updating the simulations to comply with best practices for scientific software increased the speed of the simulations, and finally a graphical user interface that was utilized to produce bulk simulations for matching experimental images is described.

4.1 Photon Recoil Heating Model

The original CCMD model featured a system for calculating random kicks for the photon recoil that did not correspond with the physical understanding of how photon recoil works. Initially the model featured the use of picking values from three normal distributions at random in order to form a 3D vector that was then applied to each ion. These vectors were scaled by the photon recoil heating factor from the simulation input parameters. The problem with this approach is that it was picking both a random direction and a random magnitude. When one considers a normal distribution centered at zero, the most commonly picked value if one were to pick a random number from this distribution would be around zero, which means that the photon recoil kicks rarely were large and unevenly heated the crystal. When an ion absorbs a photon, it will then emit a photon in a random direction with no real preference on which direction the photon will be emitted [23, 24].

Picking random directions and magnitudes produces a distribution like that seen in Figure 4.1. While spherically symmetric, the result of this distribution of kicks to the ions caused the kinetic energy of the ions in all three directions to be uneven, contrary to how energy is distributed in actual crystals The effect was that the kinetic energy of the system, which corresponds with the temperature, was not symmetric radially as would be expected if a sizable non-zero kick were given in a random direction, as they are in reality. The effects of the normal distributions favoring low-magnitude kicks can be seen in Figure 4.2 in which histograms of the kinetic energy in the x, y, and z directions were logged for various laser cooling parameters β. As can be observed the trend is the same for each laser cooling parameter, and the only real effect is the expected trend where increasing this parameter decreases the kinetic energy of the ions. Through this test it was discovered that the photon recoil model was to blame for the asymmetry, and a new model based on the physics of the photon recoil was devised.

A new algorithm was written for the photon recoil in which the direction was chosen at random on a unit sphere, always giving a random direction. The algorithm is summarized here, and the code is presented in the Appendix under the Random Kick using a Unit Sphere section.

An angle θ is chosen using a random number generator:

$$\theta = 2\pi x \tag{4.1}$$

where x is a random number generated using the Mersenne twister random number generator, which provides pseudo-random numbers given a seed provided by the user in the input parameters to the simulation or one chosen at random if no seed is provided.

Next, an angle ϕ is calculated using another random number:

$$\phi = \arccos(1 - 2y) \tag{4.2}$$

where y is another random number generated using the Mersenne twister random number generator.

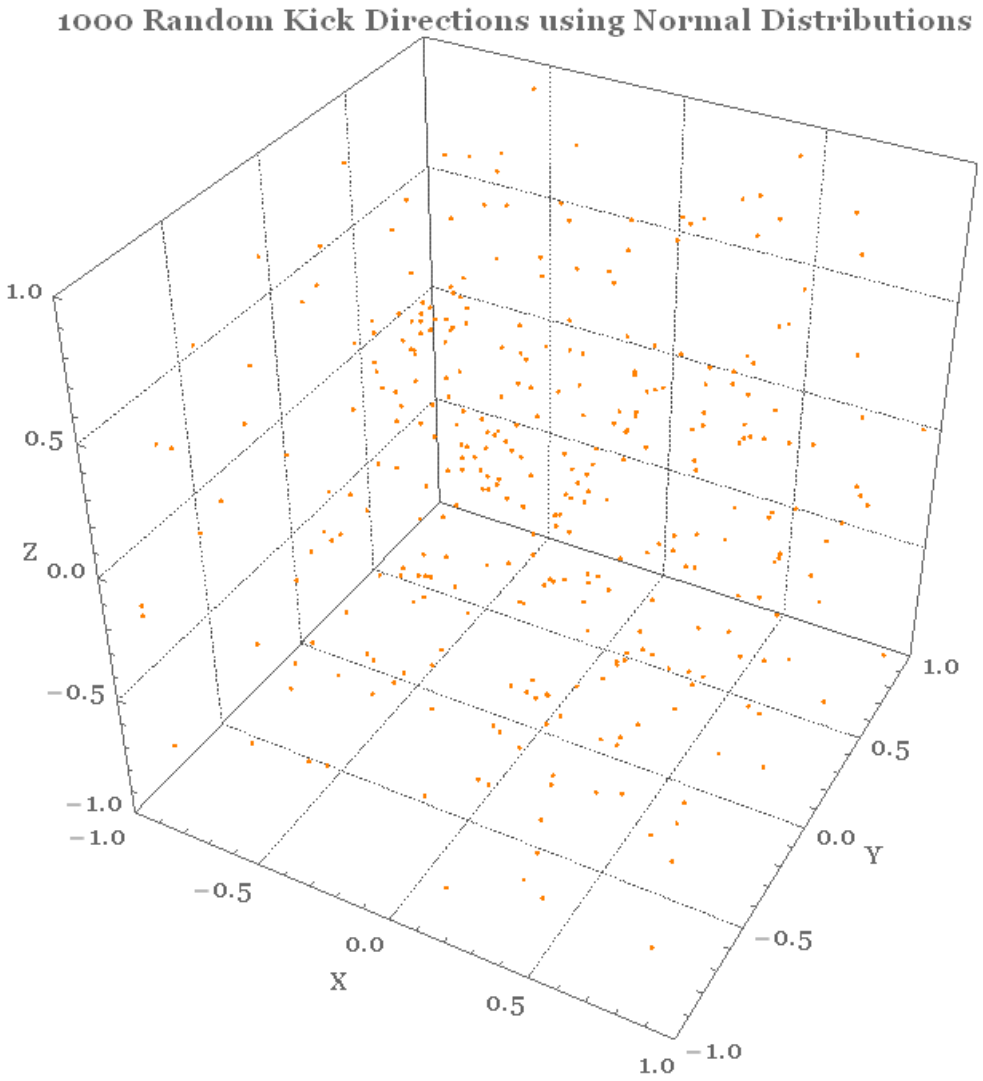

Figure 4.1: Results of 1000 random kicks using three random distributions to pick a value from 0 to 1 to build a 3D vector. This distribution of random kick magnitudes and directions results in an asymmetric heating of the ion Coulomb crystal.

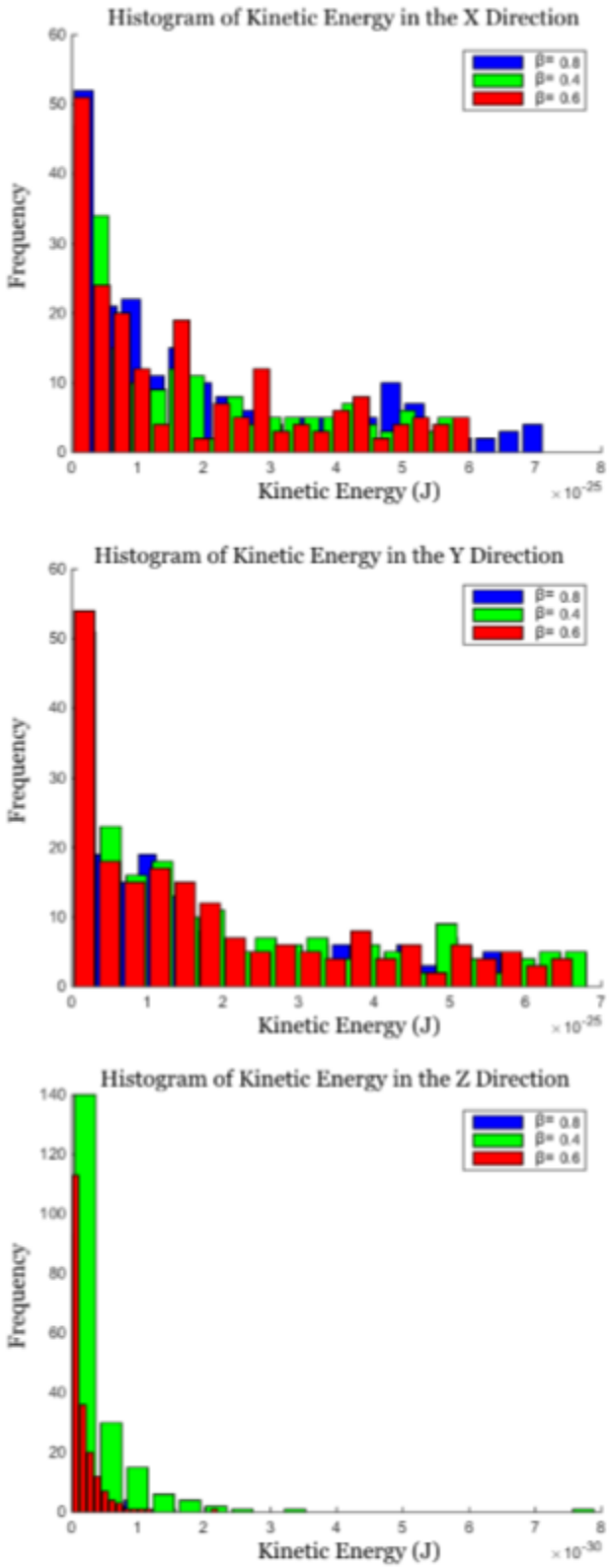

Figure 4.2: Histograms of kinetic energy values for different laser cooling parameters demonstrating that by favoring low-magnitude photon recoil kicks due to set of three normal distributions the resulting heating of the crystal is asymmetric, contrary to expectations.

A direction in Cartesian coordinates is then determined using the following equations:

$$x = \sin(\phi) * \cos(\theta) \tag{4.3}$$

$$y = \sin(\phi) * \sin(\theta) \tag{4.4}$$

$$z = \cos(\phi) \tag{4.5}$$

These values are turned into a 3D vector in Cartesian coordinates and normalized, unlike in the original algorithm. The direction given by this unit vector is then multiplied by the magnitude of the photon recoil parameter from the parameters input to the simulation, allowing the users to heat the ion Coulomb crystal symmetrically. The summary of 1000 random directions generated using this algorithm is shown in Figure 4.3, demonstrating that the new algorithm chooses random directions consistent with the physical understanding of how the photon recoil works.

The result for the heating of the crystals was that the kinetic energies are now symmetric as expected, revealing that the new algorithm fixed the issue with asymmetric heating as shown in Figure 4.4. This has the added benefit of reducing asymmetric dislocations in the ion positions due to heating the crystal unevenly.

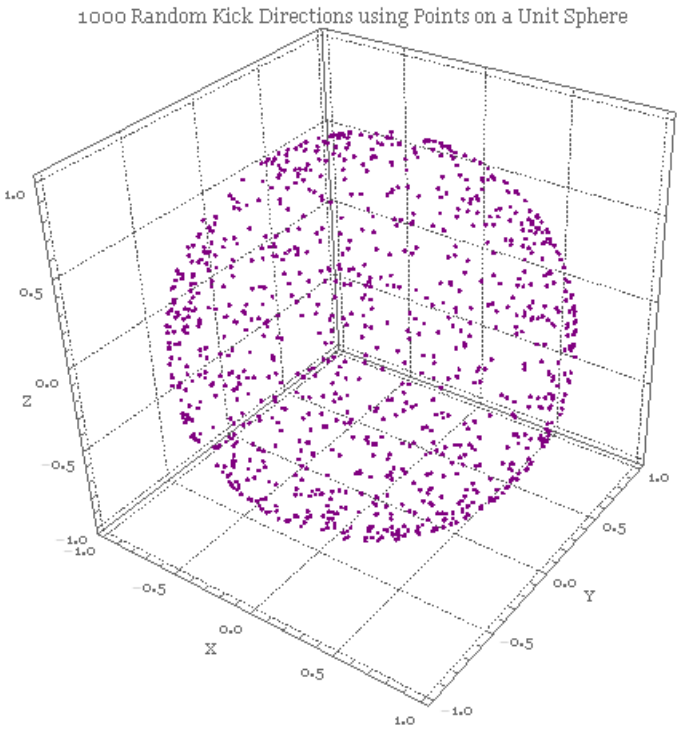

Figure 4.3: Results of 1000 random kick directions chosen by taking a random value on a unit sphere, revealing that the new algorithm for choosing directions for the random kick due to the photon recoil fixed the issue where the directions of the random kicks was not random as expected.

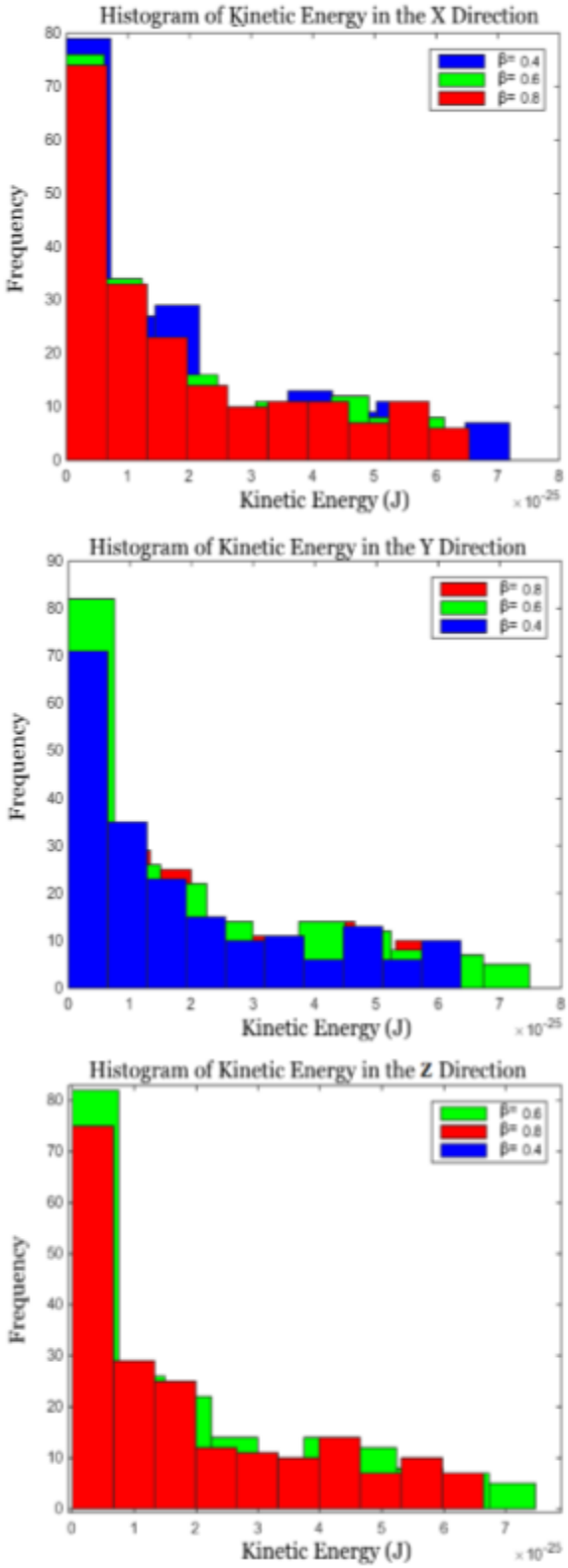

Figure 4.4: Histograms of kinetic energy values for different laser cooling parameters demonstrating that by generating unit vectors in random directions from a unit sphere and then giving equal kicks in those directions specified by the user the crystals are now heated symmetrically, as expected.

4.2 Simulation Speed Enhancement

The original CCMD program as well as the updated enhancements made to the simulations such as the new imaging algorithm and integrator were refactored and brought up to the best practices in scientific computing in order to speed up computations and allow for more simulations to be run in less time. This is critical to using CCMD in a laboratory setting as simulations of large crystals can take hours to days to evaluate, and numerous incremental changes may be necessary to bring the crystals into agreement with their physical counterparts. As a result, a number of improvements were made to the code that resulted in gains in computational speed.

First, the code was refactored such that code was not copy and pasted for calculations and where possible quantities were calculated only once and then retrieved from memory for use in future calculations rather than calculating them again. Not only did this make the code more readable, but it also allowed for the compiler to perform more complete optimizations of the simulation code, in addition to decreasing computation time on redundant calculations [29].

Another improvement made was a restructuring of the code to better optimize for the use of memory. The orignal version of the code had a great deal of jumps wherein small operations were performed by outside functions forcing the computer to branch to these locations to perform simple tasks like adding vectors or getting lengths of arrays. These branches add additional clock cycles to each computation and can make computationally expensive operations like calculating the Coulomb force on each ion even more expensive [29]. Thus, where possible operations were consolidated to minimize branching. Further, copying of objects was minimized. In some cases when a calculation or operation was being performed a copy of an object, like an ion, would be made and then copied back or a data structure would be built and expanded dynamically. Dynamic memory is expensive to use in terms of computation time, so where possible, pointers were utilized and dynamic allocations were minimized such that computation times could be improved [29]. This especially decreased the time spent outputting and collecting data during the simulation. A summary of the speed gains versus the unenhanced version of CCMD is presented in Figure 4.5.

This demonstrates a great deal of improvement in computation time for the simulations, saving as much as ten hours on simulations of 300 ions and helping to prevent the computation time from increasing too quickly for growing ion numbers in the region of reasonable ion numbers for our experiment. The CPU wall time was chosen as the time to evaluate how quickly the simulation ran for a certain number of ions as this is the real-world time elapsed between simulation initialization and when the simulation terminated.

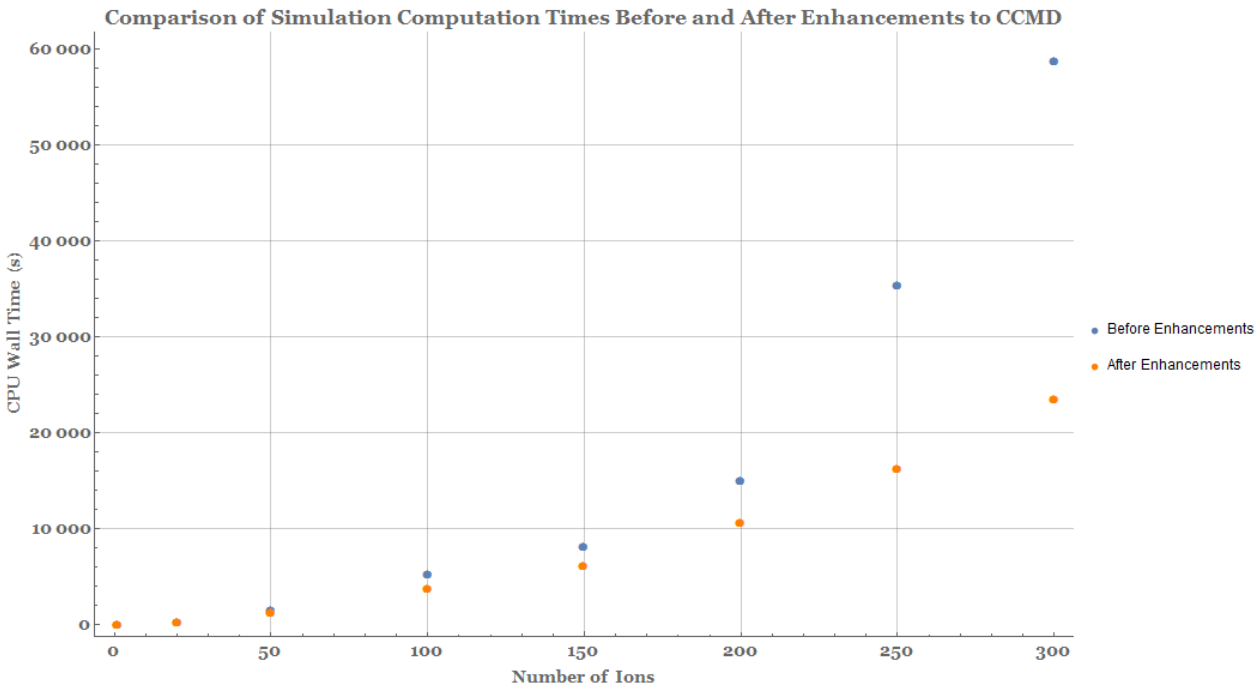

Figure 4.5: The computation speed per ion in terms of CPU Wall Time, or real world time elapsed in seconds between when the simulation was initialized and when it ended for a enhanced version of CCMD that ran faster due to optimized memory and refactored code and the original codebase.

The parameters utilized for each simulation in this test to ensure that each simulation was identical except for the number of ions in the simulation is presented in Table 4.1.

Table 4.1: Experimental Parameters for Characterizing Simulation Speeds

Experimental Parameters	**Values**
Minimization RF Cycles	2000
Data Collection RF Cycles	2000
Integrations Per RF Cycle	16
RF Frequency	3.14MHz

4.3 Graphical User Interface

In order to match experimental and simulated crystal images incremental changes to the simulation parameters are required, as observed in Chapter 3. In order to facilitate these incremental changes, a graphical user interface (GUI) that allows multiple simulations to be queued in sequence on a lab computer or cluster was developed. This GUI allows the user to place a series of simulations into a folder and specify a simulation version to have run these simulations in sequence. The program will return the complete status of the simulation to the status bar of the GUI, as shown in Figure 4.6. This GUI allows for scanning entire parameter spaces with minimal human input to start the next simulation in sequence and also allows simulations to be run overnight and over weekends to minimize the lab time spent configuring and managing simulations. This GUI was used extensively in the collection of data for this thesis and also is used routinely in the lab to produce large sets of simulations for analyzing the effects of different experimental parameters.

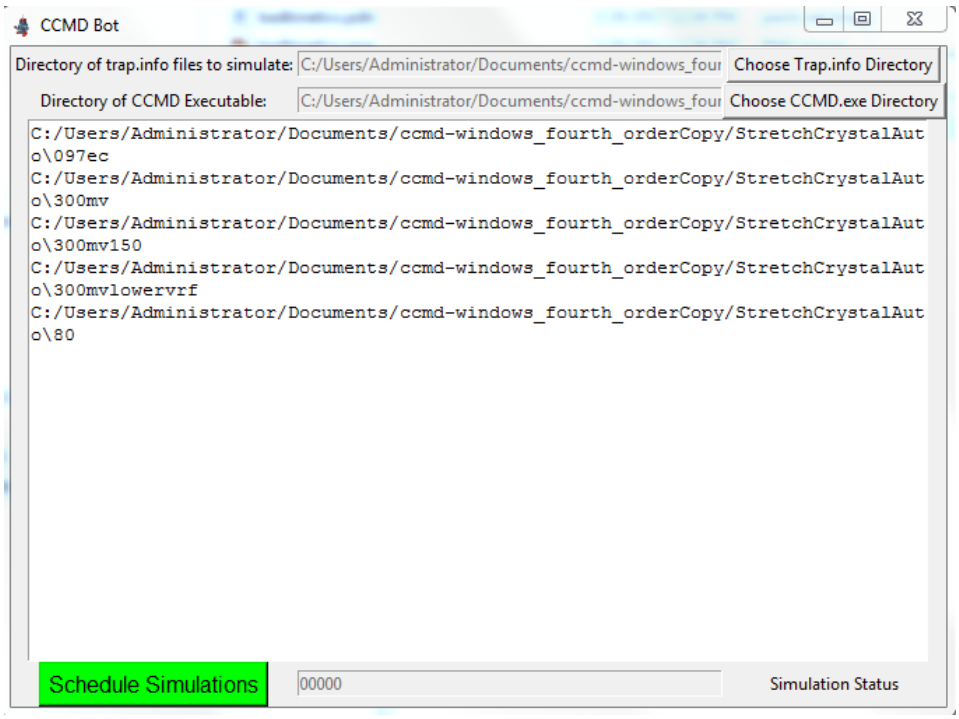

Figure 4.6: Screenshot of the graphical user interface developed to make scheduling multiple CCMD simulations easier, called CCMD Bot.

Chapter 5

Conclusions and Future Work

Through the addition of a higher-order integrator the accuracy of Coulomb Crystal Molecular Dynamics (CCMD) has been improved and previously observed issues with ions being found in dislocated positions or at energies higher or lower than expected for given experimental parameters have been minimized. The addition of a new imaging algorithm allows for simulated CCD images of ion Coulomb crystals to be compared directly to experimental CCD images since the new imaging algorithm takes in the physical properties of the CCD camera used in the experiment exactly, such as camera resolution and pixel size. The photon recoil model has been replaced with one that more accurately reflects the physics of photon recoil due to absorbing a photon during laser cooling. As a result the random kicks that heat the ions now heat the crystals symmetrically, giving direct control to users as to how the crystals are heated and making matching crystal temperatures more accurate. The simulation performance was enhanced through code refactoring and optimization of the memory the simulation uses, saving hours for large simulations. A graphical user interface was developed that allows for numerous simulations with incremental changes to simulation parameters to be run in sequence, allowing parameter spaces to be scanned automatically. These enhancements and features were utilized to match simulated CCD images to a set of experimental crystals of various sizes and shapes demonstrating the flexibility of the enhanced simulations. Further, it was demonstrated that CCMD can use complete experimental data such as apparatus voltages and frequency settings as well as ion counts to produce images with matching structures, or the simulations can be configured to extract ion counts for intermediary images that were not destroyed

in order to get an ion count from time of flight mass spectrometry. This was shown to be possible even for large crystals (> 250 ions) such as those used experimentally for determining reaction rates, demonstrating that CCMD is an effective laboratory tool for extracting the properties of ion Coulomb crystals.

Higher order effects could be added to the CCMD simulation suite to improve the accuracy of the positions of the ions in the simulated CCD images and to better match experimental images. Examples of higher order effects that could be simulated include is the effect of patch potentials on the trap rods, which are a naturally occurring phenomenon that could be diagnosed using simulations to explain disordered patterns in experimental images. Adding a two-frequency Paul trap model to the simulation would be a good predictive tool for future experiments and allow for the experimental parameters for which two species of highly disparate charge to mass ratios could be confined in the trap could be explored before building such an experiment. This would allow CCMD to be a predictive tool for reactions and studies with sympathetically cooled species of larger on the order of a megadalton [12]. CCMD could also be extended to produce a sequence of images automatically for producing simulated movies of crystal evolution to correspond with experimental movies of chemical reactions occurring in ion Coulomb crystals, which would also require the addition of functionality to add species at a later point in time after the simulation has initialized.

Bibliography

[1] Moldy algorithm development, Jul 2010.

[2] P. Bowe, L. Hornekær, C. Brodersen, M. Drewsen, J. S. Hangst, and J. P. Schiffer. Sympathetic crystallization of trapped ions. Phys. Rev. Lett., 82:2071–2074, Mar 1999.

[3] Lincoln D. Carr, David DeMille, Roman V. Krems, and Jun Ye. Cold and ultracold molecules: science, technology and applications. New Journal of Physics, 11(5):055049, 2009.

[4] Nabanita Deb, Laura L. Pollum, Alexander D. Smith, Matthias Keller, Christopher J. Rennick, Brianna R. Heazlewood, and Timothy P. Softley. Coulomb crystal mass spectrometry in a digital ion trap. Phys. Rev. A, 91:033408, Mar 2015.

[5] Michael V. DePalatis. Production of Cold Barium Monohalide Ions. PhD thesis, Georgia Institute of Technology, dec 2013.

[6] F. Diedrich, E. Peik, J. M. Chen, W. Quint, and H. Walther. Observation of a phase transition of stored laser-cooled ions. Phys. Rev. Lett., 59:2931–2934, Dec 1987.

[7] M. Drewsen. Ion Coulomb crystals. Physica B Condensed Matter, 460:105–113, March 2015.

[8] Daniel H. E. Dubin. Minimum energy state of the one-dimensional coulomb chain. Phys. Rev. E, 55:4017–4028, Apr 1997.

[9] Julio C Gutiérrez-Vega, RM Rodrıguez-Dagnino, MA Meneses-Nava, and S Chávez-Cerda. Mathieu functions, a visual approach. American Journal of Physics, 71(3):233–242, 2003.

[10] W. M. Itano, J. J. Bollinger, J. N. Tan, B. Jelenković, X.-P. Huang, and D. J. Wineland. Bragg diffraction from crystallized ion plasmas. Science, 279(5351):686–689, 1998.

[11] D. J. Larson, J. C. Bergquist, J. J. Bollinger, Wayne M. Itano, and D. J. Wineland. Sympathetic cooling of trapped ions: A laser-cooled two-species nonneutral ion plasma. Phys. Rev. Lett., 57:70–73, Jul 1986.

[12] Nathan Leefer, Kai Krimmel, William Bertsche, Dmitry Budker, Joel Fajans, Ron Folman, Hartmut Häffner, and Ferdinand Schmidt-Kaler. Investigation of two-frequency paul traps for antihydrogen production. Hyperfine Interactions, 238(1):12, 2016.

[13] Raymond E March. Quadrupole ion traps. Mass spectrometry reviews, 28(6):961–989, 2009.

[14] K. Mølhave and M. Drewsen. Formation of translationally cold mgh^+ and mgd^+ molecules in an ion trap. Phys. Rev. A, 62:011401, Jun 2000.

[15] K. Okada, M. Wada, T. Takayanagi, S. Ohtani, and H. A. Schuessler. Characterization of ion coulomb crystals in a linear paul trap. Phys. Rev. A, 81:013420, Jan 2010.

[16] Kunihiro Okada, Kazuhiro Yasuda, Toshinobu Takayanagi, Michiharu Wada, Hans A. Schuessler, and Shunsuke Ohtani. Crystallization of ca^+ ions in a linear rf octupole ion trap. Phys. Rev. A, 75:033409, Mar 2007.

[17] I.P. Omelyan, I.M. Mryglod, and R. Folk. Optimized forest–ruth- and suzuki-like algorithms for integration of motion in many-body systems. Computer Physics Communications, 146(2):188–202, jul 2002.

[18] A. Ostendorf, C. B. Zhang, M. A. Wilson, D. Offenberg, B. Roth, and S. Schiller. Sympathetic cooling of complex molecular ions to millikelvin temperatures. Phys. Rev. Lett., 97:243005, Dec 2006.

[19] Wolfgang Paul. Electromagnetic traps for charged and neutral particles. Reviews of modern physics, 62(3):531, 1990.

[20] E. L. Pollock and J. P. Hansen. Statistical mechanics of dense ionized matter. ii. equilibrium properties and melting transition of the crystallized one-component plasma. Phys. Rev. A, 8:3110–3122, Dec 1973.

[21] Shenglan Qiao. Constructing a Linear Paul Trap System for Measuring Time-variation of the Electron-Proton Mass Ratio. PhD thesis, Amherst College, 2013.

[22] R Rafac, J P Schiffer, J S Hangst, D H Dubin, and D J Wales. Stable configurations of confined cold ionic systems. Proceedings of the National Academy of Sciences, 88(2):483–486, 1991.

[23] Peter Staanum. Quantum optics with trapped calcium ions. PhD thesis, 2004.

[24] Hiroki Takahashi, Alex Wilson, Andrew Riley-Watson, Fedja Oručević, Nicolas Seymour-Smith, Matthias Keller, and Wolfgang Lange. An integrated fiber trap for single-ion photonics. New Journal of Physics, 15(5):053011, may 2013.

[25] Richard C Thompson. Ion coulomb crystals. Contemporary Physics, 56(1):63–79, 2015.

[26] MBBJM Tuckerman, Bruce J Berne, and Glenn J Martyna. Reversible multiple time scale molecular dynamics. The Journal of chemical physics, 97(3):1990–2001, 1992.

[27] Wakelam et al. Kida database. `http://kida.obs.u-bordeaux1.fr`.

[28] E. Wigner. On the interaction of electrons in metals. Phys. Rev., 46:1002–1011, Dec 1934.

[29] Greg Wilson, DA Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, Kathryn D Huff, Ian M Mitchell, Mark D Plumbley, et al. Best practices for scientific computing. PLoS Biol, 12(1):e1001745, 2014.

[30] Peter Young. The leapfrog method and other symplectic algorithms for i ntegrating newtons laws of motion, apr 2014.

[31] C. B. Zhang, D. Offenberg, B. Roth, M. A. Wilson, and S. Schiller. Molecular-dynamics simulations of cold single-species and multispecies ion ensembles in a linear paul trap. Phys. Rev. A, 76:012719, Jul 2007.

Chapter 6

Appendix

6.1 PEFRL Integrator

```
/**
 * @file PEFRL.cpp
 * @brief Fourth-order PEFRL integrator
 *
 * @author Kevin Loeffler
 * @copyright Copyright 2017 JILA
 */

#include <vector>

#include "include/integrator.h"
#include "include/ion.h"
#include "include/ioncloud.h"
#include "include/logger.h"
#include "include/vector3D.h"
#include <math.h>
```

```
PEFRLIntegrator::PEFRLIntegrator(const IonTrap_ptr it, const IonCloud_ptr ic,
                                 const IntegrationParams& integrationParams,
                                 const SimParams& sp)
    : Integrator(it, ic, integrationParams, sp) {
        Logger& log = Logger::getInstance();
        log.info("PEFRL integration.");
        n_iter_ = 0;
}

void PEFRLIntegrator::evolve(double dt, bool cooling) {
    float frCoefficient = 1.0f / (2.0f - pow(2.0, 1.0 / 3.0));
    float frComplement = (1.0f - 2 * frCoefficient);
    float xi = 0.1786178958448091;
    float lambda = -0.2123418310626054;
    float chi = -0.6626458266981849e-1;

    firstCoefficient(frCoefficient*dt,cooling);
    secondCoefficient(frComplement*dt,cooling);
    firstCoefficient(frCoefficient*dt,cooling);
    // Tell everyone we're done
    notifyListeners(n_iter_++);
}

void PEFRLIntegrator::firstCoefficient(double dt, bool cooling) {
    double half_dt = dt/2.0;
    std::vector<Vector3D> coulomb_force = coulomb_.get_force();
    int i = 0;
```

```
for (auto ion : ions_->get_ions() ) {
    // Calculate velocity at half time-step, uses Coulomb force from
    // previous time step.
    ion->kick(half_dt, coulomb_force[i++]);
    if(cooling==false){
        ion->heat(half_dt);    // Heating
    }
    ion->kick(half_dt);    // Trap, plus heating if LaserCooled.

    // Update positions by full time step
    ion->drift(dt);
}

// Calculate new acceleration
coulomb_.update();
trap_->evolve(half_dt);

coulomb_force = coulomb_.get_force();
i = 0;
for (auto ion : ions_->get_ions() ) {
    // Update velocity over second half time-step
    ion->kick(half_dt, coulomb_force[i++] );
    if(cooling==false){
        ion->heat(half_dt);    // Heating
    }
    ion->kick(half_dt);    // Trap, plus heating if LaserCooled.
}
```

```
    // Update trap again.
    coulomb_.update();
    trap_->evolve(half_dt);
}

void PEFRLIntegrator::secondCoefficient(double dt, bool cooling) {
    double half_dt = dt/2.0;
    std::vector<Vector3D> coulomb_force = coulomb_.get_force();
    int i = 0;
    for (auto ion : ions_->get_ions() ) {
        // Calculate velocity at half time-step, uses Coulomb force from
        // previous time step.
        ion->kick(half_dt, coulomb_force[i++]);
        if(cooling==false){
              ion->heat(half_dt);    // Heating
        }
        ion->kick(half_dt);    // Trap, plus heating if LaserCooled.

        // Update positions by full time step
        ion->drift(dt);
    }

    // Calculate new acceleration
    coulomb_.update();
    trap_->evolve(half_dt);
```

```
    coulomb_force = coulomb_.get_force();
    i = 0;
    for (auto ion : ions_->get_ions() ) {
        // Update velocity over second half time-step
        ion->kick(half_dt, coulomb_force[i++] );
        if(cooling==false){
              ion->heat(half_dt);    // Heating
        }
        ion->kick(half_dt);    // Trap, plus heating if LaserCooled.
    }

    // Update trap again.
    coulomb_.update();
    trap_->evolve(half_dt);
}
```

6.2 Random Kick using a Unit Sphere

```
Vector3D random_kick() {
       //{ return Vector3D( normal(), normal(), normal())*kick_size; }
           double pi = 3.14159265358979323846264338327950288;
           double theta = 2 * pi * uniform01(generator);
           double phi = acos(1 - 2 * uniform01(generator));

           double x = sin(phi) * cos(theta);
           double y = sin(phi) * sin(theta);
           double z = cos(phi);
```

```
        return Vector3D(x, y, z).normalise()*kick_size;
    }
```