Collisional methods with applications to charged particle beams

Afnan Al Marzouk
ABSTRACT

COLLISIONAL METHODS WITH APPLICATIONS TO CHARGED PARTICLE BEAMS

Afnan Al Marzouk, Ph.D.
Department of Physics
Northern Illinois University, 2021
Professor Béla Erdélyi, Director

Accurate and efficient computational methods are essential to study the dynamics of charged particle beams. Most of the available numerical methods in beam physics are based on the collisionless model which is sufficient for applications that does not depend on Coulomb collisions of the simultaneously interacting particles. However, the development of collisional numerical methods has become greatly important in recent years for applications where collisions play an important role such as the electron cooling of high energy hadron beams. While collisional methods can provide very accurate simulations, their algorithms are very complex, and they face efficiency challenges.

In this work, we present our development of new collisional numerical methods which are the first methods that can provide an accurate microscopic description of beam dynamics with high computational efficiency. The first method is the Simò integrator which solves the N-body problem of beams by direct integration of the particles’ equations of motion in the presence of external electromagnetic fields. Its development included very unique techniques to obtain accuracy while resolving all the efficiency challenges known to N-body integrators. Consequently, the Simò integrator is the first large-scale collisional numerical method in beam physics that is accurate up to machine precision with a relatively high efficiency. Then, we
incorporate the Simò integrator to model collisions into our other collisional method referred to as the Particles’ High-Order Adaptive Dynamics (PHAD). PHAD employs an advanced version of the fast multipole method (FMM) along with Strang splitting method, and the addition of the Simò integrator makes PHAD the first most efficient, numerically symplectic, collisional method in beam physics. For an enhanced performance, the algorithms of both the Simò integrator and PHAD were fully parallelized on a large-scale high-performance hybrid cluster. We present simulations performed by our codes of three complicated beam dynamics problems. One application is for the electron cooling of ion beams to which our simulations demonstrate and give the first insight of the microscopic description of electron cooling with accurate prediction of cooling time. The other application illustrates density modulations of electron beams due to ions from a collisional picture of the dynamics and provide conditions to obtain a strong modulation signal necessary for variants of coherent electron cooling systems. The last application considers microscopic simulations of the relaxation of certain beam perturbations which illustrates finite $N$ effects in contrast to the kinetic limit of the collisionless methods, and the resulted relaxation times are important for applications like the beam echo.
COLLISIONAL METHODS WITH APPLICATIONS TO CHARGED
PARTICLE BEAMS

BY

AFNAN AL MARZOUK
© 2021 Afnan Al Marzouk

A DISSERTATION SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
DOCTOR OF PHILOSOPHY

DEPARTMENT OF PHYSICS

Dissertation Director:
Professor Béla Erdélyi
ACKNOWLEDGEMENTS

As I arrive at my destination of receiving a Ph.D. degree, I remember the saying: “It’s not about the destination, it's about the journey!” While I am glad to have made it, I have enjoyed the whole journey with all its ups and downs. Of course, the journey would not have been possible, nor enjoyable, without many people whom I would like to thank. The first person I would like to express my sincere appreciation and thanks to is my dissertation advisor Professor Béla Erdélyi for giving me the chance to start this journey and for his continuous guidance, support, and understanding until the end. I also would like to thank Dr. Philippe Piot and Dr. Yasuo Ito for being on my dissertation committee. I extend my appreciation to the Physics department staff at NIU for their assistance and hard work. A special Thanks goes to my friends and colleagues: Herman Schaumburg, Anthony Gee, Aliaksei Halavanau, and Alister Tencate who helped ease this journey and made it more enjoyable. I would like to send my appreciation to family and friends overseas who kept me in their thoughts and prayers. Last, I am very grateful to my family, especially my dear husband Ali for always believing in me and encouraging me, and my wonderful kids Hussain, Abdullah, and Alaa for the love and joy they bring into my life!
DEDICATION

She once told me about pursuing my Ph.D. as a mother: “It’s going to be difficult, but not impossible.” Dr. Vivian F. Incera, 2008. Indeed, it was difficult, but not impossible! To you Dr. Incera!
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>List of Appendices</td>
<td>xvi</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 The $N$-Body Problem</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Overview of the $N$-Body Problem</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Analytical Approaches</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Numerical Methods of the $N$-Body Problem</td>
<td>12</td>
</tr>
<tr>
<td>2.3.1 The Gravitational $N$-body Problem in Astrophysics</td>
<td>14</td>
</tr>
<tr>
<td>2.3.2 The Electrostatic $N$-Body Problem</td>
<td>15</td>
</tr>
<tr>
<td>2.3.2.1 Plasma Physics</td>
<td>15</td>
</tr>
<tr>
<td>2.3.2.2 Charged Particle Beams</td>
<td>16</td>
</tr>
<tr>
<td>2.4 Numerical Methods Based on Collisionality of Particle Beams</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1 Collisionless Methods</td>
<td>18</td>
</tr>
<tr>
<td>2.4.2 Collisional Methods</td>
<td>19</td>
</tr>
<tr>
<td>3 The Simò integrator†</td>
<td>21</td>
</tr>
<tr>
<td>3.1 Challenges of Collisional $N$-body Numerical Integrators</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Theoretical Basis of the Simò Integrator</td>
<td>24</td>
</tr>
<tr>
<td>3.2.1 Picard Iteration-Based Integrator</td>
<td>25</td>
</tr>
<tr>
<td>Chapter</td>
<td>Section</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Selection of Optimal Order and Optimal Time Stepsize</td>
</tr>
<tr>
<td>3.3</td>
<td>N-body Problem of Charged Particle Beams</td>
</tr>
<tr>
<td>3.3.1</td>
<td>System of Equations</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Radius of Convergence</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Implementation of the Simò Integrator</td>
</tr>
<tr>
<td>3.3.3.1</td>
<td>Computation of the Optimal Order and Optimal Time Step-size</td>
</tr>
<tr>
<td>3.3.3.2</td>
<td>Binning and Time Stepping</td>
</tr>
<tr>
<td>3.4</td>
<td>Simò Integrator Simulations of Charged Particle Beams</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Efficiency and Accuracy</td>
</tr>
<tr>
<td>3.4.1.1</td>
<td>Time Step-size Computation at Higher Orders</td>
</tr>
<tr>
<td>3.5</td>
<td>Parallel Implementation of the Simò Integrator Code</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Parallelization of Serial Loops</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Parallel Performance Tests</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Time Binning and the Parallelization</td>
</tr>
<tr>
<td>3.5.4</td>
<td>Memory Usage</td>
</tr>
<tr>
<td>3.6</td>
<td>The Window Feature</td>
</tr>
<tr>
<td>4</td>
<td>Particles’ High-Order Adaptive Dynamics (PHAD)</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Fast Multipole Method</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Differential Algebraic FMM</td>
</tr>
<tr>
<td>4.3</td>
<td>Strang Splitting</td>
</tr>
<tr>
<td>4.4</td>
<td>Accurate Time Stepping</td>
</tr>
<tr>
<td>4.5</td>
<td>PHAD Performance with the Simò Integrator</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>5</td>
<td>78</td>
</tr>
<tr>
<td>5.1</td>
<td>78</td>
</tr>
<tr>
<td>5.1.1</td>
<td>84</td>
</tr>
<tr>
<td>5.1.1.1</td>
<td>90</td>
</tr>
<tr>
<td>5.1.1.2</td>
<td>92</td>
</tr>
<tr>
<td>5.1.2</td>
<td>104</td>
</tr>
<tr>
<td>5.1.2.1</td>
<td>108</td>
</tr>
<tr>
<td>5.1.2.2</td>
<td>115</td>
</tr>
<tr>
<td>5.1.2.3</td>
<td>121</td>
</tr>
<tr>
<td>5.1.2.4</td>
<td>122</td>
</tr>
<tr>
<td>5.2</td>
<td>123</td>
</tr>
<tr>
<td>5.2.1</td>
<td>127</td>
</tr>
<tr>
<td>5.2.1.1</td>
<td>127</td>
</tr>
<tr>
<td>5.2.1.2</td>
<td>130</td>
</tr>
<tr>
<td>5.2.1.3</td>
<td>130</td>
</tr>
<tr>
<td>5.2.2</td>
<td>132</td>
</tr>
<tr>
<td>5.2.2.1</td>
<td>133</td>
</tr>
<tr>
<td>5.2.2.2</td>
<td>134</td>
</tr>
<tr>
<td>5.2.2.3</td>
<td>134</td>
</tr>
<tr>
<td>5.2.3</td>
<td>136</td>
</tr>
<tr>
<td>5.2.3.1</td>
<td>138</td>
</tr>
<tr>
<td>6</td>
<td>141</td>
</tr>
<tr>
<td>6</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>163</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>A comparison of the CPU times of the Simò integrator of different algorithms: adaptive with variable order, adaptive with fixed order, fixed stepsize with variable order, and fixed stepsize and order. The algorithms were compared for three beams at the same initial conditions and the same simulation time, but each beam is of a specific species of particles.</td>
<td>52</td>
</tr>
<tr>
<td>5.1</td>
<td>The parameters used in the bunched electron cooling simulations to benchmark PHAD with the IMP Experiments.</td>
<td>95</td>
</tr>
<tr>
<td>5.2</td>
<td>The parameters of each simulation of the bunched electron cooling.</td>
<td>95</td>
</tr>
<tr>
<td>5.3</td>
<td>Longitudinal cooling time of Sim.1 and Sim.2 from the analytical formulas, the experiment, and the simulations.</td>
<td>97</td>
</tr>
<tr>
<td>5.4</td>
<td>Longitudinal cooling time of Sim.3, Sim.4, and Sim.5 from the simulations and the analytical formulas.</td>
<td>98</td>
</tr>
<tr>
<td>5.5</td>
<td>Parameters used for the simulations of the modulator section of the CeC.</td>
<td>108</td>
</tr>
<tr>
<td>C.1</td>
<td>Summary of the number of jobs launched and the computational time used for our work, which were performed on the computing cluster Gaea.</td>
<td>172</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Lower bounds and the smallest absolute value of the actual roots at different orders for the collision of a proton of 1 MeV kinetic energy incident at a rest proton.</td>
</tr>
<tr>
<td>3.2</td>
<td>The lower bounds and the smallest absolute value of the actual roots at different orders for the collision of a 7 TeV proton incident at a rest proton.</td>
</tr>
<tr>
<td>3.3</td>
<td>Lower bounds and the smallest absolute value of the actual roots at different orders for the collision of a 1 MeV proton incident at a rest lead ion in the presence of an external magnetic quadrupole field.</td>
</tr>
<tr>
<td>3.4</td>
<td>A Flowchart of the implementation of the Simò integrator.</td>
</tr>
<tr>
<td>3.5</td>
<td>The CPU time of the serial Simò integrator showing the quadratic dependence on the number of particles with a clear efficiency gain when the time bins are used.</td>
</tr>
<tr>
<td>3.6</td>
<td>The change of the mutual optimal time stepsizes of an electron orbiting a proton moving along the ( z )-axis. The optimal stepsizes change according to the mutual distance and the change on the momenta.</td>
</tr>
<tr>
<td>3.7</td>
<td>Plots of the individual optimal time stepsizes of two different protons in the simulation of a uniform beam of 10,000 relativistic protons, demonstrating the adaptivity of the Simò integrator.</td>
</tr>
<tr>
<td>3.8</td>
<td>The change of the optimal time stepsizes of a proton in the simulation of a uniform beam of non-relativistic protons. The optimal stepsizes increase as a result of the protons moving away from each other.</td>
</tr>
<tr>
<td>3.9</td>
<td>Histograms to compare the optimal time stepsizes of low energy beams of 100 particles (with the same initial conditions) of: (a) protons, (b) lead ions, (c) electrons. The large masses have larger stepsizes than lighter masses.</td>
</tr>
</tbody>
</table>
3.10 Comparison between the optimal time stepsizes of a non-relativistic and a relativistic proton beams. The optimal stepsizes of the relativistic beam are smaller for some time and increase faster than the stepsizes of the non-relativistic beam.

3.11 A histogram of the optimal time stepsizes of a Gaussian beam of non-relativistic beam of 100 protons at one of the time steps. There are more particles in the core of the beam with relatively smaller stepsizes than the halo particles.

3.12 The change of the optimal time stepsizes of a uniform beam of non-relativistic protons subjected to a magnetic dipole field in the $y$-direction. The optimal stepsizes oscillate with an overall increase as the particles rotate and spread in the $y$-direction.

3.13 Histograms to compare the calculated optimal orders (in about 400 time steps) at high accuracy in the simulation of two beams (100 protons per beam) at different energies: (a) non-relativistic, (b) relativistic. The relativistic protons required lower optimal orders than the non-relativistic protons.

3.14 Histograms to compare the required optimal orders to achieve an accuracy of $\delta = 10^{-16}$ of low energy beams of: (a) lead ions, (b) electrons. The massive lead ions required higher optimal orders than the light mass electrons.

3.15 Conservation of the energy and momentum using the Simò integrator, showing the relative errors of: (a) the total momentum, (b) the total energy. The relative errors are shown for three beams each is of a specific species of particles, but all have the same initial conditions and with a required accuracy of $\delta = 10^{-16}$.

3.16 An example where the estimation of the radius of convergence, and hence the optimal time stepsize, can significantly change for different orders: (a) The estimated radius of convergence computed at variable orders, (b) The estimated errors associated with the optimal time stepsize when the stepsize is computed at the second order, and when its computed at variable orders.

3.17 The speedup resulted from the parallelization of the first loop of the force computation for a different number of processors and nodes, which indicates a low performance.

3.18 The speedup resulted from the parallelization both loops of the force computation, varying the number of processors for each parallel loop.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.19</td>
<td>59</td>
</tr>
<tr>
<td>3.20</td>
<td>60</td>
</tr>
<tr>
<td>3.21</td>
<td>62</td>
</tr>
<tr>
<td>3.22</td>
<td>63</td>
</tr>
<tr>
<td>3.23</td>
<td>64</td>
</tr>
<tr>
<td>3.24</td>
<td>65</td>
</tr>
<tr>
<td>4.1</td>
<td>76</td>
</tr>
<tr>
<td>4.2</td>
<td>77</td>
</tr>
<tr>
<td>5.1</td>
<td>79</td>
</tr>
<tr>
<td>5.2</td>
<td>82</td>
</tr>
</tbody>
</table>

3.19 The performance of the Simò code resulted from the parallelization of the second loop of the force computation: (a) The speedup of the strong scaling for a different number of processors and nodes which indicates a high performance, (b) The efficiency of the weak scaling showing high efficiency as the initial problem size per processor increases.

3.20 The performance of the parallel Simò integrator when only the second loop of the force computation is parallelized is illustrated by the decrease of the CPU time as a function of the number of processors when simulating $10^5$ particles.

3.21 The CPU time of the parallel Simò code as a function of the number of time bins for two types of particles distributions: (a) Uniform distribution, (b) Gaussian distribution. The time decreases as the number of time bins increases until it reaches a minimum, and then it starts increasing again.

3.22 The CPU time of the parallel Simò code as a function of the number of time bins using different number of processors, for both types of time bins and both uniform and Gaussian distributions.

3.23 The estimated memory usage of the parallel Simò integrator code almost independent of the number of processors, and mainly dependent on the number of particles.

3.24 An example of eliminating particles using the window feature of the Simò integrator code: (a) Initial distribution of the particles before applying the window, (b) Distribution of particles after applying the window.

4.1 A flow chart that illustrates PHAD algorithm.

4.2 The performance of PHAD and its main components, the FMM and the Simò integrator, as a function of the clustering parameter $q$. The simulations were performed for $10^4$ electrons of a Gaussian distribution.

5.1 Ideal circular orbit of a reference particle and the coordinate system that rotates with it.

5.2 Single particle trajectory tracing an ellipse where its maximum position and angle are determined by the beta function $\beta(s)$. 
5.3 The initial spatial distributions of the bunched electron cooling simulations: (a) Sim.1 of short pulse and (b) Sim.2 of long pulse. The blue points represent the protons and the orange points are the electrons.

5.4 The initial spatial distributions of the bunched electron cooling simulations: (a) Sim.3 of short pulse and $I = 15$ mA, (b) Sim.4 of long pulse and $I = 15$ mA, and (c) Sim.5 of long pulse and $I = 1.5$ mA. The blue points represent the protons and the orange points are the electrons.

5.5 Eigen emittances change with time that resulted from the electron cooling simulation Sim.1: (a) longitudinal with a linear fit (red) and (b) transverse.

5.6 Eigen emittances change with time that resulted from the electron cooling simulation Sim.2: (a) longitudinal with a linear fit (red) and (b) transverse.

5.7 Eigen emittances change with time that resulted from the electron cooling simulation Sim.3: (a) longitudinal with a linear fit (red) and (b) transverse.

5.8 Eigen emittances change with time that resulted from the electron cooling simulation Sim.4: (a) longitudinal with a linear fit (red) and (b) transverse.

5.9 Eigen emittances change with time that resulted from the electron cooling simulation Sim.5: (a) longitudinal with a linear fit (red) and (b) transverse.

5.10 Comparison between the resulted longitudinal eigen emittance of the ion beam propagated through the ring elements including the cooling section and when the cooler is not included. Clearly, the reduction of the eigen emittance when including the cooler is due to the electron cooling.

5.11 The invariant of the 4D emittance when the nonlinear electron-ion interaction forces were relatively weak in the electron cooling simulations: (a) Sim.1 and (b) Sim.2.

5.12 The 4D emittance of the electron cooling simulations: (a) Sim.3, (b) Sim.4, and (c) Sim.5. The 4D emittance was not invariant when the nonlinear electron-ion interaction forces were strong in Sim.3 and Sim.4, but was relatively invariant for Sim.5 when the nonlinear forces were relatively weak.

5.13 Transverse eigen emittances change with time from the electron cooling simulation Sim.1 with and without the cooling section. Both $\epsilon_1$ and $\epsilon_2$ oscillate when there is a cooler while they remain unchanged when there is no cooler.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.14 The transverse projected emittances from the electron cooling</td>
<td>103</td>
</tr>
<tr>
<td>simulation Sim.4 showing oscillations due to the solenoid field and</td>
<td></td>
</tr>
<tr>
<td>growth to the strong electron-ion interactions.</td>
<td></td>
</tr>
<tr>
<td>5.15 Density plots from PHAD simulations demonstrating the change of</td>
<td>109</td>
</tr>
<tr>
<td>the electron beam transverse size as the beam propagates through the</td>
<td></td>
</tr>
<tr>
<td>lattice of the modulator section of the CeC.</td>
<td></td>
</tr>
<tr>
<td>5.16 PHAD simulations of density modulations of the electron beam</td>
<td>110</td>
</tr>
<tr>
<td>due to a single centered ion through the modulator section of the</td>
<td></td>
</tr>
<tr>
<td>PoP CeC at RHIC. The longitudinal signal increases with the</td>
<td></td>
</tr>
<tr>
<td>propagation distance while the transverse density varies according</td>
<td></td>
</tr>
<tr>
<td>to the modulator lattice.</td>
<td></td>
</tr>
<tr>
<td>5.17 Comparison of the resulted longitudinal density modulations from</td>
<td>111</td>
</tr>
<tr>
<td>two simulations of the same density, but differ in size and the</td>
<td></td>
</tr>
<tr>
<td>number of electrons. One simulation included $10^5$ electrons within</td>
<td></td>
</tr>
<tr>
<td>$\sigma_z$, and the other included $2 \times 10^5$ electrons within</td>
<td></td>
</tr>
<tr>
<td>$2\sigma_z$ of the same density.</td>
<td></td>
</tr>
<tr>
<td>5.18 PHAD simulations of the longitudinal density modulations of</td>
<td>112</td>
</tr>
<tr>
<td>electron beams through the modulator section of the PoP CeC at</td>
<td></td>
</tr>
<tr>
<td>RHIC with a single ion located at $(x_{ion}, 0, 0)$ for different</td>
<td></td>
</tr>
<tr>
<td>$x_{ion}$. The signal is reduced by the displacement from the center</td>
<td></td>
</tr>
<tr>
<td>of the $x$-axis compared to the centered ion.</td>
<td></td>
</tr>
<tr>
<td>5.19 PHAD simulations of the longitudinal density modulations of the</td>
<td>113</td>
</tr>
<tr>
<td>electron beams through the modulator section of the PoP CeC at</td>
<td></td>
</tr>
<tr>
<td>RHIC with a single ion located at $(0, y_{ion}, 0)$ for: (a) $y_{ion}$</td>
<td></td>
</tr>
<tr>
<td>= $\sigma_y$ and $y_{ion} = 2\sigma_y$. The signal decreases as the</td>
<td></td>
</tr>
<tr>
<td>ion displacement from the center increases.</td>
<td></td>
</tr>
<tr>
<td>5.20 Comparison between the longitudinal density modulations of $10^5$</td>
<td>114</td>
</tr>
<tr>
<td>electrons for an ion located at $(0, 0, z_{ion})$ when $z_{ion} = 0$,</td>
<td></td>
</tr>
<tr>
<td>$0.5\sigma_z$, and $\sigma_z$. The reduction of the signal due to</td>
<td></td>
</tr>
<tr>
<td>off-centered ion is a result of the decrease of the number of</td>
<td></td>
</tr>
<tr>
<td>electrons.</td>
<td></td>
</tr>
<tr>
<td>5.21 PHAD simulations of the longitudinal density modulations of the</td>
<td>115</td>
</tr>
<tr>
<td>electron beam through the modulator section of the PoP CeC at RHIC</td>
<td></td>
</tr>
<tr>
<td>with a single ion located at $(0, 0, z_{ion})$ for $z_{ion} = \pm \sigma_z$. The signals are displaced to the location of the ions and are of the same strength.</td>
<td></td>
</tr>
<tr>
<td>5.22 PHAD simulations of the longitudinal density modulations of the</td>
<td>116</td>
</tr>
<tr>
<td>electron beam due to a centered ion, moving with respect to the</td>
<td></td>
</tr>
<tr>
<td>electron beam through the modulator section of the PoP CeC at RHIC.</td>
<td></td>
</tr>
<tr>
<td>The signal due to the moving ion is compared to that of the stationary</td>
<td></td>
</tr>
<tr>
<td>ion for ion’s velocities: (a) $\sigma_{v_z}$, (b) $3\sigma_{v_z}$,</td>
<td></td>
</tr>
<tr>
<td>and (c) $10\sigma_{v_z}$.</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.23</td>
<td>PHAD simulations of density modulations of the electron beam through the modulator section of the PoP CeC at RHIC due to two ions, one is at $\sigma_z$ and the other is at $-\sigma_z$. There are two peaks around the two ions longitudinally, and the signal is about twice that of a single ion transversely.</td>
</tr>
<tr>
<td>5.24</td>
<td>PHAD simulations of density modulations of the electron beam through the modulator section of the PoP CeC at RHIC due to three ions placed at $-\sigma_z$, $0$, and $\sigma_z$. There are three peaks around the three ions longitudinally, and the signal is about three times that of a single ion transversely.</td>
</tr>
<tr>
<td>5.25</td>
<td>The longitudinal modulation signals due to three ions in a uniform longitudinal electron distribution showing that the strengths of the signals is the same compared to the case of the Gaussian distribution where the strengths vary.</td>
</tr>
<tr>
<td>5.26</td>
<td>PHAD simulation results at the end of the modulator section showing density modulations of the electron beam due to three ions when the distance between the ions is: (a) $0.2\mu$m, (b) $0.1\mu$m, and (c) $0.05\mu$m. The modulation is improved when the ions are placed more than $0.1\mu$m apart.</td>
</tr>
<tr>
<td>5.27</td>
<td>Comparison between the modulation signals when the signal is a result of one simulation of multiple ions (orange), and when the signal is a superposition of signals due to individual ions (blue). The density modulations are due to: (a) two ions, (b) three ions, and (c) five ions.</td>
</tr>
<tr>
<td>5.28</td>
<td>PHAD simulations of density modulations of the electron beam due to a centered ion through the modulator section of the PoP CeC at RHIC using signal averaging for 20 runs at different propagation distances: (a) $0.6\text{m}$, (b) $1.8\text{m}$, and (c) $3\text{m}$ (end of the modulator section). The inset of each plot shows the region where the signal should be, represented by the small bump in (b) and (c).</td>
</tr>
<tr>
<td>5.29</td>
<td>The phase space distribution of a beam of $10^4$ electrons of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed initially with $n_b = 10%n_0$. Simulated by the Simò integrator, the momentum distribution relaxes towards a Gaussian after about 200 ns.</td>
</tr>
<tr>
<td>5.30</td>
<td>The Simò integrator simulation of a beam of $10^4$ electrons of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed initially with $n_b = 10%n_0$: (a) the evolution of the momentum distribution and (b) the initial and final longitudinal spatial distribution.</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.31</td>
<td>129</td>
</tr>
<tr>
<td>The Simò integrator simulation of two electron beams, $4 \times 10^4$ and $10^5$ electrons, both of initial uniform spatial distributions and Gaussian momentum distributions of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that are slightly perturbed with $n_b = 10% n_0$. The relaxation time decreases with increasing beam density, and the spatial longitudinal density was not affected here.</td>
<td></td>
</tr>
<tr>
<td>5.32</td>
<td>131</td>
</tr>
<tr>
<td>The Simò integrator simulation of a beam of $10^4$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10% n_0$. The momentum distribution did not relax to a Gaussian and the longitudinal spatial distribution became similar to the momentum distribution.</td>
<td></td>
</tr>
<tr>
<td>5.33</td>
<td>132</td>
</tr>
<tr>
<td>The Simò integrator simulation of a beam of $4 \times 10^4$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10% n_0$. The momentum distribution relaxed to a Gaussian and the longitudinal spatial density changed from uniform to a Gaussian.</td>
<td></td>
</tr>
<tr>
<td>5.34</td>
<td>133</td>
</tr>
<tr>
<td>The Simò integrator simulations of a beam of $10^5$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with different $n_b$. Increasing $n_b$ moderately increases the relaxation time and slightly shifts the center of the Gaussian distribution towards the perturbation. The Longitudinal density was not affected here.</td>
<td></td>
</tr>
<tr>
<td>5.35</td>
<td>135</td>
</tr>
<tr>
<td>The Simò integrator simulations of two beams consisting of $4 \times 10^4$ and $10^5$ protons, which were initially of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10% n_0$. The momentum distributions relaxed to Gaussian and the longitudinal spatial densities did not change, and the relaxation time is faster for the higher density beam.</td>
<td></td>
</tr>
<tr>
<td>5.36</td>
<td>136</td>
</tr>
<tr>
<td>The Simò integrator simulation of a beam of $4 \times 10^4$ protons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10% n_0$. The momentum distribution did not relax to a Gaussian and the longitudinal spatial distribution became similar to the momentum distribution.</td>
<td></td>
</tr>
</tbody>
</table>
5.37 The Simò integrator simulations of a beam of $10^5$ protons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with different $n_b$. Increasing $n_b$ moderately increases the relaxation time and slightly shifts the center of the Gaussian distribution towards the perturbation. The Longitudinal density was not affected here.

5.38 The Simò integrator simulations of a relativistic proton beam consisting of $4 \times 10^4$ protons, which were initially of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distributions did not relax to a Gaussian during 800 ns.

5.39 The Simò integrator simulation of a beam of $4 \times 10^4$ relativistic protons initially with a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10\% n_0$. The initial beam density was increased by decreasing the cylinder’s dimensions in which the protons are uniformly distributed. The momentum distribution started to relax toward a Gaussian and the longitudinal spatial density changed from uniform to a Gaussian.

C.1 Gaea cluster assembly at NIU’s computing facility.
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A  The Differential Algebra Techniques</td>
<td>163</td>
</tr>
<tr>
<td>B  The Concept of Eigen Emittance</td>
<td>167</td>
</tr>
<tr>
<td>C  Gaea Cluster Information and Statistics of Utilization</td>
<td>170</td>
</tr>
</tbody>
</table>
A charged particle beam consists of a group of close-by particles, electrons or ions, that move in about the same direction with approximately the same kinetic energy. Beam physics considers a detailed study of the motion of the charged particle beams through the electromagnetic fields and through a particle accelerator complex. In accelerators, particle beams are generated with specific properties that can be manipulated by the electromagnetic elements through the accelerator. Particle accelerators are well-known research tools of scientific discovery especially for fundamental physics. Nonetheless, the ability to control the properties of particle beams made particle accelerators span a wide range of applications in different areas such as industry [1], security [2], energy [3], and medical purposes [4]. Of the tens of thousands of accelerators worldwide, about 5% of them are utilized for applied research and only less than 1% are designated for basic research [5].

The numerical simulation of charged particle beam dynamics is an essential step towards understanding beam dynamics and supporting the accelerators design and optimization. As the hardware and software of computers advances, along with the development of supercomputers, the capabilities of beam dynamics simulation tools increase and more complicated phenomena in beam physics can be explored. Taking full advantage of the available computational power entails that architectural and algorithmic improvements multiply to greatly advance the reach of the simulations such that they are of high accuracy and efficiency.

Because a charged particle beam is a kind of plasma, common simulation methods of plasma were employed for the multi-particle beam dynamics simulations. Depending on the particular application, modeling plasma is performed by the macroscopic models, the
microscopic models, or the mesoscopic models. The macroscopic model is also known as the fluid model where the plasma is regarded as a conducting fluid acted upon by external electromagnetic forces. While this model is the simplest, it is the least accurate and it is not appropriate for simulating beam dynamics since beams usually consists of bunches [6]. On the other hand, the microscopic model is the most accurate and also is the most complex as it entails solving the six differential equations of motion of each charged particle. Consisting of three equations for the position components and three equations for the momentum components, there are $6N$ equations to be solved for a beam of $N$ particles. The mesoscopic model is in between the macroscopic model and the microscopic model both in terms of accuracy and complexity, and therefore it is the most commonly implemented for the simulation of charged particles dynamics. This model is known as the kinetic model where the evolution of the phase space distribution function representing the particles interactions with electromagnetic fields is described by Vlasov equation (or the collisionless Boltzmann equation).

There are many algorithms and codes based on the collisionless model that have been extremely useful for the simulation of various beam dynamics that does not depend on the pair-wise Coulomb collisions of the simultaneously interacting particles. However, the collisionless methods are not suited to describe beam dynamics phenomena where the collisional effects are important such as in the high-intensity beams which are of current interest for future accelerator applications. In such applications, Coulomb interactions can lead to growth of the beam, instabilities, and beam loss. Hence, it is important to understand these effects to gain a good control of the beam for a successful accelerator design and application.

Current and future research in high energy physics and nuclear physics requires accelerators with high-energy and/or high-intensity beams in order to discover new fundamental particles and to study the fundamental structure of matter. Therefore, colliding beam facilities or storage rings are constructed where two counter rotating accelerated particle beams
collide head-on. An important aspect of modern colliders is achieving high luminosity at the collision energy as the luminosity is a measure of the ability of the collider to produce the required number of interactions for the production of particles. Since the luminosity is inversely proportional to the beam size, it has become increasingly important for modern colliders to decrease the beam size in order to maximize their luminosity, especially at the collision energy. This can be accomplished by applying a cooling technique that reduces the six-dimensional phase space volume of the beam (the beam emittance).

In early 2020, the U.S. Department of Energy has started the construction of the electron-ion collider (EIC) at Brookhaven National Laboratory (BNL) that will rely on high-intensity, high-energy hadron beams for the advances of nuclear physics. One of the methods that is proposed to increase the luminosity of the ion beam is the novel electron cooling method. In conventional electron cooling, a DC cold electron beam is accelerated by an electrostatic high voltage and then co-propagated with a hot ion beam in a small straight section of the collider. While the conventional electron cooling method is a mature technique, it is suitable to cool intense ion beams at low energies [7].

Because DC coolers cannot accelerate electron beams to very high energies due to the technical limitations of the high power and high voltage, other methods are considered for cooling high energy ion beams. One promising technique to accelerate electron beams to higher energies is by using radiofrequency (RF) fields in an energy-recovering linac system in which the accelerated electron beam will be bunched. To explore the bunched electron cooling method, low energy experiments were carried out by a collaboration team from the Jefferson Lab (JLab) in USA and the Institute of Modern Physics (IMP) in China at the IMP facility where the first bunched electron cooling was demonstrated [8]. Another experimental demonstration of bunched electron cooling at low energy was also performed at the Low Energy RHIC electron Cooler (LEReC) at BNL [9].
During the electron cooling, ions experience a dynamical friction due to their Coulomb interactions with the surrounding electrons. A classical friction force formula is derived based on the relaxation of two components plasma where one component is the hot ions and the other one is the cold electrons. Then, the cooling time is estimated by dividing the friction force by the ion’s momentum. Estimating the cooling time is crucial for the design of the colliders and currently this is usually done by numerically calculating the friction force. All of the available tools are based on the collisionless model with some tunable parameters and can provide a good estimate of the cooling time especially for low energy DC coolers (e.g. BETACOOL [10] and JSPEC [11]). However, it has been shown that any numerical simulations based on the analytical friction force do not provide an accurate description of the cooling process [12]. That is because the derivation of the friction force is based on some strong assumptions such as ignoring finite time effects and close encounters, in addition to the difficulty to include finite magnetic field strengths. Although close encounters do not occur so often, they result in strong collisions that significantly contribute to the friction force [13]. Therefore, a good estimation of the cooling time requires very accurate collisional numerical simulations that are based on first principles with a minimum set of assumptions.

An alternative cooling method that is proposed to cool high-intensity, high-energy hadron beams is the strong electron cooling such as the coherent electron cooling (CeC) [14]. A typical CeC system consists of a modulator where the ions are co-propagated with the electrons, imprinting a density modulations on the electrons distribution; an amplifier where the density modulations are amplified; and a kicker where the amplified electron density is overlapped with the ions and kick their energy toward their central energy.

In the CeC system, collisions also play an essential role from the start to the end. The modulator is a crucial component, and the density modulations can affect the final cooling. Therefore, it is also important to accurately simulate the electrons density modulations in order to provide a precise estimate of the cooling time. Density modulation is a result of
Debye shielding (screening) of the ion by the surrounding electrons. While some collisionless simulations were performed for the modulator section of the proof-of-principle (PoP) CeC experiments in RHIC at BNL [15, 16, 17, 18], collisional methods could provide additional insights and more accurate simulations.

Collisions are also important to study the relaxation process of certain beam perturbations. In accelerators, the longitudinal momentum of a charged particle beam generally follows a Gaussian distribution with a specific momentum spread. This momentum distribution can deviate from a Gaussian through a brief interaction between the beam and an external field. The relaxation of the perturbed momentum distribution is also usually described through the collisionless model in which the Vlasov equation can have an infinite number of equilibrium solutions in a time scale that is shorter than the collision time. Because of the kinetic limit $N \rightarrow \infty$, the collisionless model does not always provide an accurate description of the physical phenomena. Specifically, the dynamics in the limit $N \rightarrow \infty$ are valid on a finite time interval and it differs from the $N$-body description for a finite $N$ or for a very long-time interval $t \rightarrow \infty$ [19]. In addition, Coulomb potential is smoothed in the collisionless approach such that it does not represent the dynamics in time scales shorter than the collision time. It has been suggested that actual, finite $N$ physical behaviors may be overlooked by employing the kinetic model and that collisional (microscopic) model from the $N$-body classical mechanics can give the most fundamental explanation of the underlying physics [20, 21, 22, 19]. The study of the relaxation of the momentum distribution towards equilibrium is an initial important step to study the beam echoes phenomena in accelerators, and also relevant to the relaxation of the Bump-on-Tail (BoT) problem in plasma physics.

Although collisional models are important to describe the electron cooling along with some other beam dynamics phenomena where collisions are important (e.g. the ultra-cold high-brightness charged particle sources [23, 24]), none of the available algorithms currently used by the beam physics community is collisional. Collisional methods based on first princi-
ples give very complex algorithms and it is challenging to efficiently implement them. Main challenges come from: the very large number of particles in a beam bunch, the long-range pair-wise interaction, the presence of close encounters, the difficulty of including external electromagnetic fields and relativistic effects, the vast spatial and time scales, and the importance of maintaining symplecticity.

In a beam bunch, the number of particles ranges from 1 to $10^{13}$ which makes collisional simulations of very large $N$ extremely expensive when solving the $6N$ equations of motion along with an $O(N^2)$ computational cost of Coulomb forces. The long-range pair-wise interaction comes from the fact that Coulomb force, repulsive or attractive, is inversely proportional to the square of the distance between a pair of particles. Hence, all particles in the beam simultaneously interact with each other. While some particles can be far enough that their Coulomb interaction forces are smooth, Coulomb force varies quickly for close encounters as the distance between a pair of particles becomes very small. Inaccurate modeling of close encounters can lead to unphysical results and an accurate modeling can lead to inefficiencies. For a group of relativistically interacting particles, the Hamiltonian formulation of the single particle motion in external electromagnetic field that satisfies Maxwell equations is invalid. The time scale of the dynamics can be long which makes it difficult to preserve the geometric features of the time-continuous system for a discrete particles system, mainly the symplectic feature.

Motivated by the importance of accurate modeling of collisional effects in beam dynamics, we have developed two novel collisional methods for efficient and accurate simulations of charged particle beams. Collisional effects arise from the discrete nature of the charged particles and the pair-wise Coulomb collisions, which are best described as an $N$-body problem. The basic concept of the $N$-body problem involves $N$ point-like particles interacting simultaneously via pair-wise forces which are generally long-range and proportional to the inverse square of the distance between a particle pair. This view of charged particle beams as an
N-body problem is new and has not been pursued before in beam physics. In our pioneering
N-body approach to beams, we consider direct integration of the equations of motion to
which we develop a numerical N-body time integrator. However, this approach entails very
complex algorithms that is usually considered inefficient. Most of the previous work was
performed for the astrophysical N-body problem where it is clear that it is challenging to
obtain accuracy while maintaining efficiency. Main challenges come from modeling close en-
counters that imposes difficulties on the choice of the timestep size, the order of integration,
and maintaining error tolerance. The simplest technique for the time stepping is to use a
fixed time step sizes. The drawbacks of this technique is that it can overstep a close encounter
and result in unphysical behaviors if a large stepsize is used. On the other hand, while a
very small stepsize can give accurate results, it is very inefficient. Great advancements were
achieved by the development of adaptive integrators where stepsizes are varied adaptively as
needed by the dynamics. Similar to the time stepping, a low order of integration results in
inaccuracies while a high order results in inefficiencies. A few integrators employ a variable
order scheme to increase the efficiency without losing accuracy. Yet, there are no integrators
that employ an optimal selection of the stepsizes and orders such that the truncation errors
are controlled and the computational time is minimized. Hence, in this work, we address
the accuracy and efficiency challenges and our approaches to deal with them. We describe
the development of our first collisional method, the Simò numerical integrator, that we have
devised with unique tools to overcome the challenges, and we demonstrate its high perfor-
mance for beam dynamics (first published in [25]). Consequently, our Simò integrator is the
first large-scale collisional numerical method in beam physics which provides relatively fast
simulations that are accurate up to machine precision.

Despite its high efficiency, the computational complexity of the Simò integrator is of
$O(N^2)$, and thus its efficiency starts to decline for larger $N$. In addition, the Simò integrator
is more suitable for short time scale applications as its accuracy decreases when modeling
long-time dynamics. Therefore, the Simò integrator was incorporated in another collisional method called PHAD (Particles’ High-order Adaptive Dynamics), developed by our research group [26], to be able to model large $N$ with higher efficiency and to deal with larger time scales accurately. As a result, PHAD is the first efficient collisional method with a computational cost that scales with $\mathcal{O}(N)$ and that is symplectic to machine precision. Both the algorithms of the Simò integrator and PHAD were fully parallelized on the large-scale high-performance hybrid cluster “Gaea” at Northern Illinois University.

Using our collisional numerical methods, we performed simulations to some important applications in beam physics that were mentioned earlier. We provide the first microscopic electron cooling simulations with accurate estimated cooling times that are benchmarked with experimental data. We also performed simulations of the modulator section of the CeC system and extracted the density modulation signal from the particle description. The accuracy of these simulations supports the possibility of density modulations and relates the strength of the modulation signals to different ions configurations with respect to the electron beam. At last, we provide a microscopic description of the relaxation of the perturbed momentum distribution and how this relaxation can be affected by different factors such as the beam density, the momentum spread, and the size of the perturbation. These simulations take into account all collisions within the short collision time and include all the finite $N$ effects unlike the collisionless model.

The dissertation is organized as follow: a background of the $N$-body problem in physics with an overview of its numerical methods, especially the methods based on the collisionality of the system are provided in Chapter 2; we introduce our novel Simò $N$-body integrator including our approaches to different accuracy and efficiency challenges in collisional methods with some applications, and we also describe the parallelization of the algorithm and its performance in Chapter 3; our collisional method PHAD and its main components where the Simò integrator is incorporated for a highly efficient algorithm that is symplectic to
machine precision is presented in Chapter 4; practical applications of complicated beam physics phenomena of current interest such as the electron cooling and the relaxation of certain beam perturbations are presented in Chapter 5; the dissertation is concluded in Chapter 6.
CHAPTER 2

THE $N$-BODY PROBLEM

2.1 Overview of the $N$-Body Problem

The $N$-body problem has emerged from celestial mechanics a few centuries ago. Since then, it has maintained a unique place at the heart of classical physics, and has motivated astronomers, physicists, and mathematicians with its historical and practical interest. Consequently, it has influenced a significant part of the theoretical work in astronomy, solid state physics, plasma physics, differential equations, and potential theory.

In his attempts to solve the problems of celestial mechanics, Sir Isaac Newton published the basic ideas of the $N$-body problem in his Principia in 1687. The $N$-body problem consists of $N$ bodies (particles) that interact simultaneously via pair-wise forces. In general, those pair-wise forces are long-range and proportional to the inverse square of the distance between the particles. The gravitational and the electrostatic $N$-body problems are very distinguished examples for this case. The dynamics of the $N$-body problems involve the equations of motion of a system of $N$ particles which are described by a set of $N$ ordinary differential equations (ODEs). Each equation of motion is a straightforward application of Newton’s second law of motion on an individual particle plus, in case, the external potential fields.

The classical gravitational $N$-body problem considers an inertial reference frame in three dimensional space where $N$ point masses $m_i, i = 1, 2, \ldots, N$ move under the influence of
the mutual gravitational attraction. Applying Newton’s laws of motion and of gravity, the equations of motion are

\[ F_i = m_i \ddot{r}_i = -G \sum_{j=1}^{N} \sum_{j \neq i} m_i m_j \frac{r_i - r_j}{|r_i - r_j|^3}, \] (2.1)

where \( r_i \) is a three-dimensional position vector function of time \( t \) of the \( i \)th particle, and \( G \) is the universal constant of gravitation.

Given the initial conditions, finding a general solution to the initial value problem (IVP) of the ODEs of the \( N \)-body problem is very challenging. In his Principia, Newton implies the impossibility of solving the \( N \)-body problem due to the mutual gravitational interaction forces. In 1710, the two-body problem was completely solved analytically Johann Bernoulli [27, 28]. Despite his illustration of solving the equations of motion by integration, the cases where \( N > 2 \) have no complete analytic solution in closed form and remain one of the well-known unsolved dynamical problems. While it is deemed impossible to solve the \( N \)-body problem via the method of first integrals, numerous theoretical and numerical approaches have been developed to the problem when \( N > 2 \).

### 2.2 Analytical Approaches

The standard technique for solving the \( N \)-body problem was quantitative as Bernoulli derived the two-body problem solution by integration. Many of the early efforts to find a quantitative solution of the \( N \)-body problem has failed, leading to question the possibility of integrating the problem where \( N > 2 \). Since the three-body problem is the simplest complication of the two-body problem, it was one of the most studied special cases. The quantitative methods were successful only for special forms of the general three-body problem. Among
those solutions are the particular solutions obtained by Euler (1767) and Lagrange (1772) [29].

Concerning the three-body problem of celestial mechanics, Poincaré developed qualitative methods to study the $N$-body problem which are described in his book that he published in 1892 ([29] and the references therein). His initial work was published in a paper that won a prize celebrating the 60th birthday of King Oscar II of Sweden and Norway. Poincaré’s approach has laid the foundations for qualitative analysis of non-linear differential equations (Hamiltonian systems) and led him to discover a new phenomenon traditionally known as chaos ([27] and the references therein).

In the early 1900’s, a series solution in powers of $t^{1/3}$ for the three-body problem was obtained by Sundman [30, 29]. A global analytical solution of the $N$-body problem in the form of a convergent power series was provided by Wang in 1991 [31], excluding the cases that lead to singularities. However, Wang commented in his paper that his series solution of the $N$-body problem is not useful for practical purposes mainly due to the very slow rate of convergence [31]. As a result, various numerical methods that integrate and simulate the $N$-body problem have been developed to play the critical role of describing the $N$-body systems.

### 2.3 Numerical Methods of the $N$-Body Problem

Historically, the $N$-body simulations started by Holmberg in 1941, who followed the evolution of two model galaxies interacting gravitationally by calculating the intensity of 37 light bulbs at many intervals [32, 33]. Computer simulations of the classical $N$-body problem is said to have been established by von Hoerner’s work published in 1960 [34] as he told the story in his paper “How it All Started” [35]. Since then, the substantial
advancements of hardware and software have greatly influenced the development of various numerical methods.

The typical quantitative methods are the direct numerical methods (also called the particle–particle methods) that solve the \( N \)-body problem by integrating the differential equations of motion numerically. The main difficulties of these methods are the singularities, and the exponential growth and accumulation of numerical errors as the integration time increases. In addition, the force evaluation over all pairs of particles have a time complexity of \( \mathcal{O}(N^2) \) which makes large-scale calculations extremely time consuming. One of the very well-known direct integration methods is the Taylor method. For evolutions that takes place in a short time scale, the common Runge-Kutta methods with automatic time stepsize are usually used. Close encounters are often dealt with by using some regularization based on the Kustaanheimo-Stiefel (KS) transformation [36] which uses a perturbative method over the analytical two-body solution to apply transformations of coordinates. The Aarseth methods are predictor-corrector methods that were introduced in 1985 [37], and they were utilized by many \( N \)-body algorithms since then.

Because of the drawbacks of quantitative methods, the qualitative methods have acquired more attention and were more advanced since they are valid for a very long time, and usually for all times. Examples of the qualitative approaches are the variational methods and perturbation theory in which the numerical integration is a correction to the analytic trajectories. These symplectic methods consider the geometrical properties of the equations of motion and obey Hamilton’s equations to a high degree of accuracy.

Since the \( N \)-body problem was originated in celestial mechanics, most of the work performed for the \( N \)-body problem was from the gravitational \( N \)-body problem in astrophysics. The electrostatic \( N \)-body problem, where the Coulomb potential has a similar form to the gravitational potential, has received a lot of consideration due to their importance in dif-
ferent areas such as plasma physics, beam physics, and atomic physics. Some examples of applications of the numerical $N$-body simulations are given below.

### 2.3.1 The Gravitational $N$-body Problem in Astrophysics

Many problems in astrophysics concern the dynamical behavior of systems which consist of $N$ particles interacting gravitationally at the same time, and these $N$-body systems can only be approached by computer simulations. Because these systems usually involve a large number of bodies and/or long time intervals, more efficient methods were always needed. As a result, most of the early work of the $N$-body numerical methods have emerged from the field of astrophysics. In addition, the wide range of different $N$-body astrophysical problems has influenced the development of various numerical techniques depending on the particular context. The choice of the appropriate method for a specific problem mainly depends on the time scale and collisionality of the problem.

The astrophysical $N$-body problem can be divided into different domains, each requires a specific $N$-body technique to achieve the best performance and accuracy. One example of these domains is celestial mechanics where a single massive body dominates the gravitational field, and it is usually treated by approximate methods based on perturbation theory. Another example is the dense stellar systems which consist of a large number of roughly equal masses. These systems are collisional with multiple close encounters and usually modeled by methods based on the Boltzmann equation and the Fokker-Planck equation. In the galaxy dynamics and cosmology domain, there is a very large number of particles where close encounters are not important. Thus, this domain is described in terms of a mean field and softening is usually employed to avoid the unphysical formation of binaries.
2.3.2 The Electrostatic N-Body Problem

Most of the numerical methods used for the electrostatic N-body problem are similar to the methods used for the gravitational N-body problem. Examples of areas in physics that involve the electrostatic N-body problem are plasma physics and beam physics described below.

2.3.2.1 Plasma Physics

Plasma physics is another rich source of the N-body problems, where the bodies are charged particles (ions and electrons). One of the most fundamental phenomena in plasma physics arises from the wave-particle interactions. Problems including the wave-particle interactions are extensively studied, such as Langmuir waves and their Landau damping or growth. Besides the experimental observations in natural or laboratory plasmas, numerical methods of these electrostatic N-body problems are essential in understanding the dynamics of charged particles in a plasma. The numerical analysis of plasmas is based on models that vary in complexity. The least complex model is the macroscopic model where the plasma is represented by its density, pressure, and velocity. This model lacks accuracy, but it is not computationally expensive. On the other hand, the most complex model is the microscopic model which includes solving the equations of motion of all particles in their self-consistent electromagnetic fields. Although this model is the most accurate, it is extremely expensive. In between those two models in complexity and accuracy is the mesoscopic model where the plasma is characterized by a position-velocity distribution function in phase space. An example of the mesoscopic model is the Vlasov-Poisson (VP) model that is widely used for plasma studies.
In different areas of application, the number of real particles could be too large to simulate all of them, and macro-particles are used where each of them stands for many real particles. The particle-based simulation methods have been originally proposed by Hockney [38]. These methods were significantly improved later [39] and led to the development of the methods now known as Particle-In-Cell (PIC) methods which are the most common numerical methods used to study charged particles. PIC methods employ particle-mesh methods [40] where space charge is considered to obtain the electric field of the charged particles. Macro-particles are used to approximate the plasma and are advanced in time with the electromagnetic fields which are computed on a grid. The first PIC algorithms ignored collisions, but techniques based on the Monte Carlo method were utilized to include collisions later [41].

A main drawback of the standard PIC algorithms is that they do not conserve total energy exactly, and thus energy conserving algorithms were developed based on a variational principle [42, 43] and are considered as a good alternative to PIC methods. In variational methods, the Vlasov equation is discretized on a grid of the phase space, such as in the semi-Lagrangian methods [44, 45].

2.3.2.2 Charged Particle Beams

Similar to plasma, charged particle beams can be described as an electrostatic $N$-body problem in the beam rest frame. In addition, the charged particles motion is affected by the applied external electromagnetic fields that accelerate and guide the beam in particle accelerators. In the laboratory (lab) frame, the particles are seen to move with a longitudinal velocity, while they are considered stationary in the beam rest frame. The magnetic field is effectively zero in the beam rest frame, and thus the problem is electrostatic in this frame. Therefore, Coulomb interactions between the particles are usually calculated in the beam
rest frame and then transformed to the lab frame using the Lorentz transformation [46]. In the lab frame, the problem is electromagnetic, and the external electric field and the external magnetic field are given.

Precise beam dynamics simulation is a very important research topic to understand the dynamics and to support the design and optimization of modern accelerators. The dynamics of the charged particles in a beam is similar to that of a plasma. In both systems, an accurate $N$-body model of the large number of the simulation particles subjected to complicated external electromagnetic fields is a challenging problem. The general approach to follow the charged particles through the accelerator is by using multi-particle tracking codes where the majority of them employ the PIC method to include the effect of particles fields (examples: [47, 48, 49, 50]).

2.4 Numerical Methods Based on Collisionality of Particle Beams

Depending on the type of the dynamics that needs to be modeled, two main methods are employed to study collisionless and collisional dynamics. For long time-scale dynamics and very large number of particles, collisionless methods are commonly applied. On the other hand, collisional methods are used for a relatively short time-scale dynamics and a moderate number of particles. Both methods include different techniques in order to treat the particular $N$-body system optimally.
2.4.1 Collisionless Methods

For $N$ particles in phase space, the positions and velocities of the particles are described by a $6N + 1$ dimensional phase space distribution function at a specific time. Since the potential generated by the particles varies significantly on a microscopic level, the mesoscopic domain is usually considered were the average potentials gives a coarse-grained distribution of the particles [21]. The dimensionality of the phase space is then reduced to $6 + 1$, and the collisionless dynamics follow from the collisionless Boltzmann equation (derived from the Liouville theorem [51]). This is called the mean field approximation where the time evolution of the single particle distribution describes the properties of the system [52].

For the mean field approximation, there exist different numerical methods in which the complexity of both the force computation and the softened interaction potential between the particles are reduced, and hence these methods are less expensive than the collisional methods. In addition to the increased efficiency, softening the close encounters prevent the artificial formation of binary systems [53]. Some examples of the collisionless methods are the particle-mesh methods that solves Poisson equation on a mesh such as the known PIC methods. Other methods increase the computational efficiency by applying an Adaptive Mesh Refinement (AMR) [54] or by coupling a softened direct treatment of particles’ interactions with a large scale mean field description which is called the Particle-Particle, Particle-Mesh (P3M) method [55] [52]. In addition, there are methods that solve the Vlasov equation [56] to model the time evolution of the single particle distribution such as in [57].
2.4.2 Collisional Methods

In collisional methods, the dynamics of the $N$-body system are followed by numerically solving a system of ODEs of the individual particles. The most accurate and popular $N$-body methods are the direct methods (or the particle–particle methods) due to the minimum number of simplifying assumptions involved in these methods. Yet, the direct $N$-body integration is very expensive as it entails an $O(N^2)$ of force computation, and thus the number of simulated particles is limited. Because of its complexity and the exponential growth of integration errors with time, these methods are usually used for relatively small number of particles and short time scales simulations.

Main challenges in collisional methods come from the computation of mutual forces between all particles, accurate time stepping schemes, and the ability to deal with long simulation time scales. The difficulties in time stepping or time integration are mostly due to close encounters, where the distance between two particles is too small resulting in strong forces. The performance of time integrators in astrophysics has advanced over the past decade mostly in terms of accuracy (details in Chapter 3). The introduction of hierarchical techniques (tree codes) pioneered by Barnes and Hut [58] has significantly reduced the computational time to $O(N \log N)$, or it can be further reduced to $O(N)$ using the Fast Multipole Method (FMM) developed by Greengard and Rokhlin [59]. The main concept of tree methods is to divide particles into groups such that the potential of a group of particles is calculated using multipole expansions and apply that potential to another distant group of particles [58]. Although the accuracy of these methods is relatively reduced, a major advantage of these methods is that they can be applied to different spatial distributions.

In some applications, the mean field approximation can be used to describe collisional systems where the collisionless Boltzmann equation is modified to include a collision operator.
Examples of these methods are methods that solve Fokker-Planck equation [51] such as Monte Carlo methods that were introduced by Hénon [60]. Variants of PIC methods capture some collisional effects like in [61, 62].

The $N$-body approach to the dynamics of charged particle beams was deemed impossible in the past due to the inefficient algorithms and the limited computer power. The current advanced computer power, along with super computers, contributed to the development of more efficient algorithms, making the $N$-body approach to beam dynamics feasible. However, this approach was not pursued before our work in this dissertation, except for a preliminary work by our research group in [26]. Our $N$-body approach is the first in beam physics and will provide important insights of beam dynamics.
CHAPTER 3
THE SIMÒ INTEGRATOR*

3.1 Challenges of Collisional $N$-body Numerical Integrators

As mentioned in the previous chapter, Chapter 2, the $N$-body problem has to be solved numerically. While different numerical algorithms were developed to compute the time evolution of the particles’ positions and momenta since the early sixties, especially for the gravitational $N$-body problem [63, 34, 33, 64], the non-linear nature of the $N$-body problem and the necessity to study systems with large $N$ still give rise to efficiency challenges for numerical integrators. The main difficulties for an efficient $N$-body numerical integrator come from close encounters and the dependence of each particle’s force calculations upon the positions of all particles. Direct methods are the most accurate in which $6N$ ODEs have to be integrated, and the usual approach is to find an approximate solution where an initial condition is given. One of the oldest numerical procedures to compute approximate solutions of ODEs is the well-known Taylor method. However, this approach was considered too complicated and expensive as it requires the computation of a function’s derivatives and the evaluation of them to obtain the coefficients of the function’s Taylor series.

The development of the algorithmic (automatic, computational) differentiation [65] has made Taylor approach feasible. Automatic differentiation provides an approach to compute derivatives with only round-off errors, which makes computing Taylor series approximations

*Part of this chapter was First Published in SIAM Journal on Scientific Computing, Collisional N-body numerical integrator with applications to charged particle dynamics, in Vol.40, No.6, published by the Society for Industrial and Applied Mathematics (SIAM). Copyright © 2018 Society for Industrial and Applied Mathematics.
of the solutions practical and of high accuracy. Consequently, various packages were developed to compute and evaluate derivatives of smooth functions to an arbitrary precision [66]. Thus, numerical integrators were able to deploy the Taylor method as in [67]. Based on the Taylor method, Le Guyader [68] proposed a numerical solution to the solar N-body problem that computed the solution’s Taylor polynomial of arbitrary high order using recurrence formulas, and he showed that the method gives remarkably accurate results despite its relative inefficiency. Later, other numerical algorithms based on the Taylor method used Picard iteration to generate the Taylor series of the solution to polynomial ODEs [69], [70].

The efficiency limitations in any numerical integration method dealing with close encounters come from the choice of the time stepsize, the choice of the order, and controlling the truncation error. The simplest algorithms consist of a fixed time stepsize and a fixed low order. Methods that use a constant time stepsize cannot achieve a given accuracy, and lead to unphysical results if close encounters are involved. On the other hand, a shared adaptive time stepsize scheme in which all the particles advance by the same time stepsize at each time step can precisely model a close encounter. However, such a method demands a long CPU time as the two closest particles dictate the time stepsize for the rest of the particles [34].

A significant advance that resulted in a great gain in efficiency came from employing variable and individual time stepsizes for each particle (introduced by Aarseth [63]). This idea is essential for efficient integrators that deal with particles in a close encounter (where the force changes rapidly) because it greatly reduces the number of force calculations on every particle outside the encounter [71]. Since computing the force on a particular particle at a specific time requires the positions of all other particles at the same time, the variable and individual (adaptive) time stepsize scheme requires an integration method in which particles’ positions are extrapolatable within some time interval. Thus, integration methods that only determine the numerical solution at discrete points in time face difficulties in utilizing
adaptivity because they are unable to obtain the positions of the particles at an arbitrary time [72]. Standard symplectic methods lose their desirable properties if variable time stepsizes are applied, hence they cannot handle close encounters. Nevertheless, there are some techniques used to develop variable time step symplectic integrators such as [73, 74, 75, 76, 77, 78].

Although there are different methods adaptive integrators employ to determine the appropriate time stepsize, none are optimal. An optimal time stepsize should minimize the number of arithmetic operations. The initial choice of the time stepsize for each time step is especially problematic. In general, algorithms will start with an initial assumption of the time stepsize and the integrator will perform the force computations and test this assumed stepsize. If the stepsize is rejected, then the force computations are discarded, and the process is repeated with a smaller stepsize. If the initial time stepsize was correct, no force calculations would be wasted.

Another critical aspect of the N-body integrators is the choice of order. While fixed low orders are preferred and usually used because they save computation time, Makino noticed that high orders are more efficient for close encounters [72]. Consequently, adaptive algorithms of fixed low orders are not suitable for close encounters [70]. However, high order algorithms do not always have lower truncation error than low order algorithms. Moreover, algorithms of a fixed high order can be inefficient if the required accuracy can be achieved by a lower-order integrator. This suggests the existence of an optimal order that depends on the required accuracy and the configuration of the system at any time [70]. The available integrators employ the same order for all particles in different ways: some have a fixed order at all time steps (examples are [79] and [80]), and some have a varied optimal order per time step (an example is in [70]).

Our interest in studying close encounters is due to their crucial overall effect in capturing the dynamics of charged particle beams. This case is more complex than the gravitational case due to the possibility of the interaction force being repulsive or attractive (for same or
opposite charges, respectively). In addition, the motion of the particles could be affected by the external electromagnetic fields that accelerate and guide the beam in an accelerator. While the values of the external fields acting on a particle are independent of the other particles, the positions of all particles are needed to calculate the value of the Coulomb field experienced by the particle due to the other particles. The general approach to deal with this case is to compute the electric Coulomb fields from the charged particles in the beam frame where the problem is electrostatic, and then transform these fields to the laboratory frame through the Lorentz transformation [46] (an example is in [26]).

Despite the complexity of the direct $N$-body integrators, these integrators deliver high accuracy when dealing with close encounters in collisional systems. Although the accurate and efficient modeling of the charged particles trajectories is very critical, only a few integrators exist for plasmas such as [81, 82, 83, 84, 85]. Thus, a high-accuracy numerical integrator capable of efficiently solving systems of large $N$ remains a goal for the developers of $N$-body numerical integrators.

In order to model close encounters and resolve the efficiency challenges while achieving a given prescribed accuracy up to machine precision, we have developed our Simò $N$-body numerical integrator designed to model the $N$-body problem of charged particle beams [25]. The Simò integrator is adaptive and variable order with dense output. In addition, the integrator employs an optimal particle-by-particle selection of the time stepsizes and orders.

### 3.2 Theoretical Basis of the Simò Integrator

Our efficient, highly accurate Simò integrator has two components: a readily adaptive, variable order integrator with dense output, and a strategy to select the particle-by-particle optimal time stepsizes and optimal orders. For these components, we choose a Picard
An accurate numerical solution depends on the choice of the numerical method used to solve the initial value problems (IVPs) of the ODEs. For the first component of the Simò integrator, we found that the Taylor method can be a very competitive, variable order and variable time stepsize numerical method for the high precision solutions of the ODEs, and it might be the only practical possible method in some cases [87], [88], [89]. Indeed, an effective numerical method can be developed using Picard iteration to generate Taylor series of the solution around a specific point up to an arbitrary order [90]. Therefore, we adapted a Picard iteration-based integrator that uses Theorem 1 below, combined with Differential Algebra (DA) as described in [91]. In Theorem 1 and throughout, we denote the $n$th Taylor polynomial of $g$ in $t$ centered at zero by $T_n[g(t)]$.

**Theorem 1** Consider the initial value problem $Y'(t) = f(Y(t), t)$ with $Y(0) = Y_0$. Suppose that $f$ has a Taylor series in $t$ centered at zero with non-zero radius of convergence $\rho$ and $t < \rho$, then

$$T_n[Y(t)] = Y_0 + \int_0^t T_{n-1} [f (T_{n-1}[Y(s)])] \, ds.$$  

The main reason of our choice of this Picard iteration-based integrator is that it can be computed using the DA techniques of the COSY INFINITY software [92]. The basic idea of the DA techniques is to make the treatment of functions and operations on them in a computer environment similar to the treatment of real numbers (see Appendix A). The DA contains the mathematical structure needed for efficient algorithmic differentiation.
that allows propagation of derivatives from the identity function to much more complicated functions using elementary operations along with the derivative and anti-derivative [93]. This implementation of the DA provided effective computational tools to determine high order derivatives of complicated multivariable functions with high accuracy. Employing DA with Picard iteration in the Simò integrator is essential since evaluating a function in DA provides its truncated Taylor series. Moreover, [91] shows that this technique can be used for any smooth ODEs and not just polynomial ODEs.

There are several advantages of combining DA with Picard iteration to employ the Taylor method. Unlike classical numerical methods that provide the numerical solutions just in a discrete set of points, the Taylor method gives the numerical solution as a power series. As a result, one main advantage of using the DA with the Taylor method is that it directly generates a high order dense output, a Taylor polynomial that gives the value of the trajectory at any time within the polynomial’s radius of convergence.

The dense output is essential to any numerical solution of the $N$-body problem which entails changing the time stepsize very often and requires many evaluations of the Taylor polynomials. Moreover, the use of the DA allows fast computation of Taylor polynomials to sufficiently high order such that the radius of convergence and the truncation errors can be estimated, hence we can choose the time stepsize based entirely on the initial conditions of the step and provide accurate results efficiently. For more details, we refer the reader to [91].

Another great feature of using the DA with the Taylor method is that it is straightforward to implement an adaptive time stepsize and a variable order scheme. The adaptive stepsize scheme is critical for an automatic control of the truncation error and to reduce the number of force computations. The ability to employ a variable order is crucial to avoid extremely small stepsizes, especially when dealing with close encounters. As a result, we can implement the best strategy to achieve accuracy and efficiency which is to use an adaptive stepsize as
large as possible while varying the order (the order is reduced if the stepsize is small or increased if the stepsize is large) [88], [86].

In order to achieve high accuracy without losing efficiency, the Simò integrator utilizes all these properties and employs a particle-by-particle variability and adaptivity in each time step. In addition, the Simò integrator’s choice of the order and time stepsize is optimal as described in the following Section 3.2.2.

### 3.2.2 Selection of Optimal Order and Optimal Time Stepsize

For an efficient and accurate integration algorithm, the time stepsize $h$ and the order $p$ need to be well chosen. In the Simò integrator, the following two conditions are imposed for optimally selecting $h$ and $p$: a tolerance of the error due to the truncation of the Taylor series, and minimizing the total number of arithmetic operations [88], [86]. For an IVP as in Theorem 1, approximating $Y(t)$ with $T_n[Y(t)]$ results in an absolute error $\varepsilon = |Y(t) - T_n[Y(t)]|$, where the order $n$ is clear from the context. The two conditions for an optimal $h$ and $p$ can be satisfied using Simò’s theorem, Theorem 2, or the theorem of Jorba and Zou, Theorem 3.

**Theorem 2 (Simò [86])** Suppose that the function $f(Y, t)$ has a Taylor series in $t$ centered at $t_c$ with a non-zero radius of convergence $\rho = \rho(t_c)$, where the Taylor coefficients of the solution $Y^{[j]}$ satisfy $A_1 \rho^{-j} < |Y^{[j]}| < A_2 \rho^{-j}$ for some $0 < A_1 < A_2$. When the relative error $\varepsilon_r$ tends to zero, the optimal time stepsize $h$ that minimizes the number of operations tends to

$$h = \frac{\rho}{\exp(2)}.$$  \hspace{1cm} (3.2)

Then, the optimal order that achieves a desired relative error $\varepsilon_r$ is $p = -\frac{1}{2} \ln \varepsilon_r$.  

Theorem 3 (Jorba and Zou [88]) Suppose that the function $f(Y, t)$ has a Taylor series in $t$ centered at $t_c$ with a non-zero radius of convergence $\rho = \rho(t_c)$, where the Taylor coefficients of the solution $Y^{[j]}$ satisfy $|Y^{[j]}| \approx M \rho^{-j}$ for a positive constant $M$. Then, the optimal time stepsize that minimizes the computational cost when the absolute error tends to zero is $h = \rho / \exp(2)$. For an absolute error $\varepsilon$, the optimal order is

$$p = -\frac{1}{2} \ln \left( \frac{\varepsilon}{M} \right) - 1. \quad (3.3)$$

Theorem 2 and Theorem 3 are basically equivalent, except that Theorem 2 speaks of the relative error $\varepsilon_r$, while the error $\varepsilon$ in Theorem 3 is the absolute error. Because the condition $0 < A_1 < A_2$ in Theorem 2 might not be satisfied in some cases, and since we are interested in the absolute error $\varepsilon$, Theorem 3 is more suitable for our implementation of the optimal order $p$ in Eq. 3.3. However, there is a fundamental difficulty in employing Eq. 3.3 because it is not straightforward to know the value of $M$ [88]. On the other hand, the method of 3 is analogous to the basic strategy of selecting a stepsize $h$ in such a way that the last term in the Taylor series becomes of the order of a prescribed error tolerance [88]. In other words, 3 implies that the absolute error $\varepsilon$ that results from approximating $Y(t)$ with $T_p[Y(t)]$ is equivalent to an estimation of the remainder of the Taylor series within its convergence interval.

According to the Taylor Remainder Theorem, if a function $Y(t)$ is $n + 1$ times differentiable at $t = a$, then there exists $\zeta$ in $[a, t]$ such that the remainder term $R_n$ of its Taylor polynomial of degree $n$ centered at $a$ is given by

$$R_n = \frac{Y^{(n+1)}(\zeta)}{(n + 1)!} (t - a)^{n+1}.$$
Consider that the IVP in 1 requires an optimal order $p$ to achieve an accuracy $\delta$, then the Taylor polynomial of the solution at time $t = h$ can be written as

$$Y(h) = Y_0 + Y'(0)h + \frac{Y''(0)}{2!}h^2 + \cdots + \frac{Y^{(p)}(0)}{p!}h^p.$$  

(3.4)

When the required accuracy $\delta$ is small, the contribution of the last term of the right-hand side of Eq. 3.4 to the Taylor polynomial will be small as well. Obtaining an absolute error $\varepsilon$ of the order of $\delta$ means that, roughly speaking, any terms with contributions below $\delta$ are not necessary. Thus, estimating $R_n$ for Eq. 3.4 when $\zeta \to 0$ gives an acceptable approximation of the absolute error $\varepsilon$ which can be compared with the requested accuracy $\delta$ to determine the optimal order $p$. Denoting the estimated $R_n$ when $\zeta \to 0$, $t \to h$, and $a = 0$ by $\tilde{R}_n$, we can calculate $\tilde{R}_n$ by

$$\tilde{R}_n = \frac{Y^{(n+1)}(0)}{(n+1)!}h^{n+1}.$$  

(3.5)

Therefore, we utilized Eq. 3.5 in the Simò integrator such that $\tilde{R}_n$ is calculated for any computed Taylor polynomial at each order $n$ until $\tilde{R}_n$ is equal to (or smaller than) the requested accuracy $\delta$. Hence, the optimal order $p$ of our Simò integrator is the smallest integer $n \geq 0$ that satisfies: $\tilde{R}_n \leq \delta$.

The optimal time stepsize $h$ in Eq. 3.2 is determined by the radius of convergence $\rho$, which can be estimated from the coefficients of the Taylor polynomial (explained in Section 3.3.2). The independence of $h$ on the a priori user-defined accuracy implies that increasing the order entails less effort than decreasing the stepsize. Thus, the variable order Taylor method is one of the most efficient schemes for the high accuracy regime [88].
3.3 **N-body Problem of Charged Particle Beams**

The charged particles in a beam experience an internal force due to the Coulomb interactions between the particles, and an applied external electromagnetic force. In the lab frame, the particles are seen to move with a velocity $v_z$, while they are considered stationary in the beam rest frame. The magnetic field $B$ is effectively zero in the beam rest frame, and thus the problem is electrostatic in this frame. Therefore, we can calculate Coulomb interactions between the particles in the beam rest frame and then transform them to the lab frame using the Lorentz transformation. In the lab frame, the problem is electromagnetic where the external electric field $E$ and the external magnetic field $B$ are given.

### 3.3.1 System of Equations

The dynamics of the charged particles in the beam are described by a set of ODEs that indicates the change in the particles’ positions and momenta with respect to time. Using a Cartesian coordinate system, the particle $i$’s position and momentum are denoted by $(x_i, y_i, z_i)$ and $(p_{x_i}, p_{y_i}, p_{z_i})$, respectively. Thus, there are six ODEs for each particle: three for the position components derivatives, and three for the momentum components derivatives. For a system of $N$ particles, we need to solve $6N$ ODEs. For a particle $i$, we define the array $\mathbf{Y}_i(t) = (x_i, y_i, z_i, p_{x_i}, p_{y_i}, p_{z_i})$.

Because of the small quantities of the mass and the charge of the charged particles in a beam (such as protons and electrons), along with the small distances and/or large relative speeds or accelerations, the resulting required time stepsizes are very small. Therefore, for numerical stability reasons, we scale the time variable $t$ by multiplying it by the speed of light...
c and we get $\dot{t} = tc$ ($t$ unit is meters). The momentum is also scaled as $\mathbf{p} = \frac{\mathbf{p}}{mc}$ where $m$ is the mass of the user’s default particle. The resulting scaled array is $\hat{\mathbf{Y}}_i(\dot{t}) = (x_i, y_i, z_i, \hat{p}_{x_i}, \hat{p}_{y_i}, \hat{p}_{z_i})$.

The mass $m_i$ of a particle $i$ is described as a factor $f_i = \frac{m_i}{m}$ of the mass of the default particle. Similarly, the charge $q_i$ is given by a factor $n_i = \frac{q_i}{q}$ of the charge $q$ of the default particle. In our Simò integrator, we chose the proton as the default particle with the mass $m$ and the charge $q$. The velocity of the charged particle $i$ is the vector $\hat{\mathbf{v}}_i = \frac{\mathbf{v}_i}{c} = \frac{\mathbf{p}_i}{\sqrt{f_i^2 + \hat{p}_{x,i}^2 + \hat{p}_{y,i}^2 + \hat{p}_{z,i}^2}}$, and its system of ODEs is

$$
\frac{d\hat{\mathbf{Y}}_i}{dt} = \begin{bmatrix}
\frac{q n_i}{m c^2} \left[ \sum_{j=1}^{N} \frac{\gamma (x_i - x_j) n_j}{4\pi\epsilon_0 \left( (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right)^{3/2}} + E_{x_i} + c (\hat{v}_{x_i} B_{y_i} - \hat{v}_{y_i} B_{x_i}) \right] \\
\frac{q n_i}{m c^2} \left[ \sum_{j=1}^{N} \frac{\gamma (y_i - y_j) n_j}{4\pi\epsilon_0 \left( (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right)^{3/2}} + E_{y_i} + c (\hat{v}_{x_i} B_{y_i} - \hat{v}_{y_i} B_{x_i}) \right] \\
\frac{q n_i}{m c^2} \left[ \sum_{j=1}^{N} \frac{\gamma (z_i - z_j) n_j}{4\pi\epsilon_0 \left( (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right)^{3/2}} + E_{z_i} + c (\hat{v}_{x_i} B_{y_i} - \hat{v}_{y_i} B_{x_i}) \right]
\end{bmatrix}.
$$  (3.6)

For a detailed derivation of Eq. 3.6, we refer the reader to [26]. We define $\alpha_i(\dot{t}) = f_i^2 + \hat{p}_{x,i}^2 + \hat{p}_{y,i}^2 + \hat{p}_{z,i}^2$ and $\beta_{i,j}(\dot{t}) = (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)$. Note that $\alpha_i(\dot{t})$ and $\beta_{i,j}(\dot{t})$ are functions of time because the particles’ positions and momenta are functions of time. The components of the right-hand side of Eq. 3.6 are singular when $\alpha_i(\dot{t}) = 0$ or $\beta_{i,j}(\dot{t}) = 0$ ($\alpha_i(\dot{t})$ can be singular in the complex plane). The Simò integrator approximates each component of the right-hand side of Eq. 3.6 by a truncated Taylor series (polynomial)
up to an arbitrary order $n$ using Picard iterations, as described in [91]. Applying Theorem 3 requires finding the interval $[0, \rho]$ over which the approximations converge to the solutions.

### 3.3.2 Radius of Convergence

According to Simò’s theorem (2), calculating the optimal time stepsize $h$ entails obtaining the radius of convergence $\rho$. For a Taylor polynomial, the radius of convergence represents the distance in the complex plane from the expansion center to the nearest singularity of the expanded function. Thus, we need to locate the zeros of $\alpha(\hat{t})$ and $\beta(\hat{t})$ (we dropped the subscripts for simplicity of notation) to determine the singularities of the right-hand side of Eq. 3.6. Because these functions are analytic, we can use the Hurwitz’s theorem to pass from the zeros of the corresponding functions to the roots of their Taylor polynomials [94]. By Hurwitz’s theorem, the roots of $T_n[\alpha(\hat{t})]$ and $T_n[\beta(\hat{t})]$ tend to the zeros of $\alpha(\hat{t})$ and $\beta(\hat{t})$ (respectively), and, hence, also to the singularities of the right-hand side of Eq. 3.6. However, it is impractical to calculate the roots of Taylor polynomials of high orders for a large number of particles over many time steps. Therefore, we used the Taylor polynomials’ coefficients to estimate the radii of convergence in just a few floating-point operations. This way, the slight loss in performance due to underestimation of the radius of convergence is more than compensated for by the fast estimation of it.

The problem concerning the location of the roots of complex polynomials in analytic theory has been frequently studied over many decades. There are theorems that give an upper bound for the radius of the disk containing all the roots of the polynomial. Other theorems provide an annulus containing all the roots of the polynomial within an upper and lower bounds. The earliest contribution concerning the location of the roots of a polynomial was due to Gauss, which was improved later by Cauchy [95]. After that, many theorems
were proposed to improve the result of Cauchy. As we are concerned with the radius of convergence (the distance from the center of the Taylor polynomial to the nearest root), we need a good estimation of the lower bound. Therefore, we investigated many lower bounds to determine which bound provides the best estimation of \( \rho \) for our applications. We list in Theorem 4 below six of these lower bounds.

**Theorem 4**

If

\[
p(z) = \sum_{j=0}^{n} a_j z^j,
\]

is a non-constant polynomial with complex coefficients, then \( r_1, r_2, \ldots, r_6 \) defined below are lower bounds for the absolute value of the roots of \( p(z) \)

(i) (Affane-Aji, Biaz, and Govil [96]):

\[
r_1 = \min_{1 \leq j \leq n} \left\{ \frac{j \binom{n}{j} |a_0|}{n \cdot 2^{n-1} |a_j|} \right\}^{1/j}.
\]

(ii) (Bidkham and Shashahani [97]):

\[
r_2 = \min_{1 \leq j \leq n} \left\{ \frac{\lambda_j |a_0|}{|a_j|} \right\}^{1/j}, \text{ where } \lambda_j = \frac{5^n P_j^{(n)}}{(1 + \sqrt{2})^{2n} + (1 - \sqrt{2})^{2n}} P_{2n}
\]

and \( P_j \) is the \( j \)th Pell number.

(iii) (Kim [98]):

\[
r_3 = \min_{1 \leq j \leq n} \left\{ \frac{\binom{n}{j}}{2^n - 1} \left| \frac{a_0}{a_j} \right| \right\}^{1/j}.
\]

(iv) (Dalal and Govil [95]):

\[
r_4 = \min_{1 \leq j \leq n} \left\{ \frac{C_{j-1} C_{n-j}}{C_n} \left| \frac{a_0}{a_j} \right| \right\}^{1/j}.
\]
where \( C_j \) is the \( j \)th Catalan number.

(v) (Fujiwara [99],[100]):

\[
    r_5 = \frac{1}{2} \min_{1 \leq j \leq n} \left\{ \left| \frac{a_0}{a_j} \right| \right\}^{1/j}.
\]

(vi) (Lagrange [101],[100]): The lower bound \( r_6 \) is the inverse of the sum of the largest two numbers in the set

\[
\left\{ \left( \left| \frac{a_j}{a_0} \right| \right)^{1/j} : 1 \leq j \leq n \right\}.
\]

The lower bounds of Fujiwara and Lagrange (\( r_5 \) and \( r_6 \) here) were derived by combining the known upper bounds of Fujiwara [99] and Lagrange [101] with the fact that the roots of the reversal of a polynomial \( p(z) \) are reciprocals of the roots of \( p(z) \) [100].

For our tests of the lower bounds in 4, we conducted several numerical experiments of close encounters between two particles with different variations of distances, energies, applied external fields, and types of the particles. Three numerical experiments in which we compared the lower bounds \( r_1-r_6 \) of the roots of \( T_n[\alpha(\hat{t})] \) or \( T_n[\beta(\hat{t})] \) to the smallest absolute value of the actual roots of \( T_n[\alpha(\hat{t})] \) or \( T_n[\beta(\hat{t})] \) for \( 2 \leq n \leq 20 \) are shown in Fig. 3.1, Fig. 3.2, and Fig. 3.3.

Figure 3.1 represents the lower bounds of two colliding protons separated initially by a distance of order of \( 10^{-4} \) m, where one proton is incident with a kinetic energy of 1 MeV while the other is at rest. The lower bounds in Fig. 3.2 are for the case of a relativistic proton with a kinetic energy of 7 TeV incident at a rest proton where the initial distance between them is of the order of \( 10^{-9} \) m. The lower bounds in the last example in Fig. 3.3 are for the collision of a 1 MeV proton and a lead ion at rest (initial distance is of the order of \( 10^{-12} \) m) in the presence of an external magnetic quadrupole field. All our numerical experiments (including the examples presented in the three figures here, but many other tests not shown) indicate
Figure 3.1: Lower bounds and the smallest absolute value of the actual roots at different orders for the collision of a proton of 1 MeV kinetic energy incident at a rest proton.

Figure 3.2: The lower bounds and the smallest absolute value of the actual roots at different orders for the collision of a 7 TeV proton incident at a rest proton.

that Lagrange’s lower bound $r_6$ gives the best estimation of the radius of convergence for the types of the Taylor polynomials of interest to our applications.
The lowest order we can estimate the radius of convergence using Lagrange’s lower bound $r_6$ is the second order. Our numerical experiments (including Fig. 3.1, Fig. 3.2, and Fig. 3.3) indicate that the estimation of the radius at the second order is equal or larger than $r_6$ at higher orders. Yet, in all our tests, the ratio of $r_6$ at the second order to $r_6$ at higher orders is less than two. In addition, the smallest absolute value of the actual roots at all orders is always greater than $r_6$ at the second order. As a consequence of these tests, and for efficiency reasons, we chose to use Lagrange’s lower bound $r_6$ at the second order to estimate the radius of convergence of each particle’s position and momentum, and then use that to compute each particle’s optimal time stepsize.
3.3.3 Implementation of the Simò Integrator

Our Simò integrator is implemented using the COSY INFINITY general purpose non-linear dynamics code [92], in which Differential Algebra and their operations, including differentiation and integration are coded efficiently [93] (a summary is in Appendix A). COSY has a collection of advanced data types used for different aspects of modern scientific computing. One unique data type is the Differential Algebra vector (DA vector), which is used in high-order multivariate automatic differentiation and differential algebraic computations. The DA vector is an array that represents a truncated Taylor series of a multivariable function at an order \( n \) [92]. Differential Algebra plays an important role to fulfill the accuracy and efficiency requirements in the Simò integrator as it allows the deployment of Picard iteration to generate Taylor expansions of the ODE’s solution and easily apply operations to these expansions.

In our implementation of the Simò integrator, the time \( t \) is scaled to \( \hat{t} = tc \), but we will drop the hat from the scaled time notation in the rest of the paper for simplicity of notation. The Simò integrator performs simulations of a system of \( N \) particles within a period of time \( T \), starting from an initial time \( t_0 \) and ending at a final time \( t_f = t_0 + T \). At each time step, each particle \( i \) is at its own current time \( t_{ci} \). All the particles’ current times are synchronized at the beginning and at the end of the simulation. Hence, all the current times are initialized to \( t_0 \) (which is some constant, possibly non-zero, value) at the beginning of the first time step. We set the simulation time \( t_s \) at the current time step to \( t_s = \min\{t_{ci}\} \). Then, Picard iterations are performed for each particle’s six ODEs in Eq. 3.6 to generate Taylor expansions of their solutions up to their optimal orders that do not exceed a globally prescribed maximum order. After the optimal selection of the orders and stepsizes of all the particles (described in Section 3.3.3.1), the particles are distributed over a number of time
bins in order to be propagated (Section 3.3.3.2). Then, particles in the first bins will be propagated in each time step until all the particles current times reach the final time of the simulation (Section 3.3.3.2).

### 3.3.3.1 Computation of the Optimal Order and Optimal Time Stepsize

Once the Taylor expansion of any components of particle \( i \)'s position or momentum reaches order two, the Simò integrator estimates the radius of convergence of each of these Taylor polynomials according to Lagrange’s lower bound \( r_6 \). Since the estimation of \( \rho_i \) is done at the second order, the calculated optimal order is two or more. Each particle \( i \) has \( N \) radii of convergence: one resulting from the singularity of the position derivatives in Eq. 3.6 \( (\alpha_i(t) = 0) \) and \( N - 1 \) from the singularity of Coulomb force due to the interaction with the other particles of the system \( (\beta_{i,j}(t) = 0, \text{momentum derivatives in Eq. 3.6}) \). The minimum of these radii of convergence is the radius of convergence \( \rho_i \) of particle \( i \). Then, the integrator calculates the particle’s optimal time stepsize \( h_i \) using Eq. 3.2. Hence, each particle’s functions expansions will be valid in the interval \( [t_{c_i}, t_{c_i} + h_i] \).

At order two and above, the truncation errors are estimated from the remainder terms by calculating \( \tilde{R}_n \) of the Taylor polynomials of the position and momentum of the \( i \)th particle and compared to the required accuracy \( \delta \). If \( \tilde{R}_n \leq \delta \), then the current order is the optimal order \( p_i \) of particle \( i \). Otherwise, iteration of the order will continue until the optimal order is achieved or until the expansions’ orders reach a previously set maximum allowed order.
3.3.3.2 Binning and Time Stepping

After generating Taylor polynomials of all particles’ positions and momenta up to their optimal orders and computing all their optimal time stepsizes, each particle $i$’s new current time $new(t_{c_i}) = t_{c_i} + h_i$ determines which time bin the particle falls in. A collection of time bins $B_1, B_2, \ldots$ is a partition of the set of the particles in the system with the property that particle $i$ is in $B_k$, particle $j$ is in $B_\ell$, and $new(t_{c_i}) \leq new(t_{c_j})$ implies $k \leq \ell$. The first time bin starts at the minimum new current time ($\min\{new(t_{c_i})\}$), and the last bin ends at the maximum new current time ($\max\{new(t_{c_i})\}$).

Our Simò integrator includes two different types of time bins. One type is time bins of equal number of particles, in which the particles are distributed evenly over the desired number of time bins. The particles are ordered according to their new current times ($new(t_{c_i})$’s) from smallest to largest, and then divided into sets of equal number of particles where the first partition is the first bin and so on with the other partitions.

The other type of time bins is time bins of equal time widths. Here, the particles with new current times ($new(t_{c_i})$’s) that are within a bin’s boundaries will fall into that bin. The width of any bin equals: $(\max\{new(t_{c_i})\} - \min\{new(t_{c_i})\})$/number of bins. The number of particles will vary over the bins in this case, in contrast to bins of equal number of particles where the bins’ widths vary.

The number and type of time bins in the simulation is set by the user. After binning the particles, particles in the first bin will be propagated to the new simulation time $new(t_s) = \min\{new(t_{c_i})\}$. Thus, the Taylor polynomials associated with the particle $j$ in the first bin will be evaluated at a stepsize $\Delta t_j = new(t_s) - t_{c_j}$. These particles’ current times will be updated to $(t_{c_j} + \Delta t_j)$, and they will require new expansions of their positions and momenta in the next time step. The particles in the other bins will keep their expansions for the next
time step, but will need to shift the center of their expansions to $new(t_s)$ when calculating the forces on the propagated particles, which is easily done due to the integrator’s dense output and using DA in COSY. Note that since the $new(t_s)$ is the minimum of $\{new(t_{ci})\}$, then $0 \leq \Delta t_i \leq h_i$ and is within the convergence interval of the Taylor polynomials of the position and momentum of any particle $i$.

In the following time steps, the simulation time $t_s$ is updated. Picard iterations will be performed for the propagated particles (the first bin particles in the previous step), and their $(\rho_j, h_j, new(t_{cj}), p_j)$ will be calculated. After updating the new simulation time $new(t_s)$, all the particles are re-binned and the first bin particles are propagated as explained in the preceding paragraph. Once $new(t_s) \geq t_f$, all the particles will be propagated to the final time $t_f$ and the simulation ends. A flowchart of the Simò integrator is illustrated in Fig. 3.4.

The type and number of time bins used in any simulation determine how many particles will be propagated in each time step, and thus will need new expansions in the next step. This mainly affects the CPU time used by the Simò integrator, and hence can influence its efficiency. To show that influence, we compared the CPU times when time bins were turned on and when they were turned off while varying the number of particles. The results are shown in Fig 3.5 where we simulated protons of a uniform spatial distribution and used 10 equal widths time bins, and it is clear that CPU time decreased greatly when turning on the time bins. In both cases, the CPU time shows the quadratic dependence on the number of particles as expected.

Although, strictly speaking, binning is not a necessary part for the proper functioning of the algorithm, this feature is important for its parallel version. Currently, our serial Simò integrator can simulate a few hundreds of thousands of particles before it runs into memory issues on a typical workstation. Although we did not identify any problem for such simulations to be completed, these simulations entail an extremely long time when performed with the serial Simò integrator. Thus, the parallelization of the Simò integrator
Start time step loop while $t_s < t_f$

Iteration loop until optimal or maximum order

Loop over particles $i$

Calculate position derivatives

If order = 3

Calculate radius from $\alpha_i$

False

Loop over particles $j \neq i$

Calculate distance between particles $i$ and $j$

If order = 3

Calculate radius from $\beta_{i,j}$

False

Next particle $j$

If order = 3

Calculate $\rho_i = \min \{\text{radii}, h_i, \text{new}(t_{e_i})\}$

False

Calculate momentum derivatives

Integrate all derivatives

If order $> 2$

If order is optimal

True

Stop iterations for particle $i$

False

If not all orders optimal or maximum

True

If not all orders optimal or maximum

False

Set $\text{new}(t_s) = \min \{t_{e_i}\}$

False

If $\text{new}(t_s) < t_f$

True

Bin particles

False

If $\text{new}(t_s) = t_f$

Set $\text{new}(t_s) = t_f$

Propagate first bin particles, and shift point of expansions for other particles

False

Propagate all particles

Next time step

End time step loop

Figure 3.4: A Flowchart of the implementation of the Simò integrator.
Figure 3.5: The CPU time of the serial Simò integrator showing the quadratic dependence on the number of particles with a clear efficiency gain when the time bins are used.

is crucial to simulate a large number of particles in a reasonable amount of time (explained in Section 3.5).

In the collision of any two particles, the Simò integrator is not designed to account for quantum effects, decaying, or the production of particles. Therefore, we set a minimum limit for $h_i$ and $\Delta t_i$ to ensure only classical physics effects. We set an approximate limit that captures most cases as the time required for light to travel the distance of the diameter of a proton. For the scaled time used in the integrator $\hat{t}$, this limit is equal to the diameter of a proton.

### 3.4 Simò Integrator Simulations of Charged Particle Beams

We performed several simulations with our Simò $N$-body numerical integrator for different charged particle beams with variations of the initial kinetic energy, initial size of the
beam, particle types, number of particles, type of the position distribution, and the presence of external electric or magnetic fields. These simulations were conducted with the two types of time bins at different levels of accuracy up to machine precision. We present here some examples of the Simò integrator simulations. In these examples, we set the maximum allowed order to 20.

As an example of a close encounter simulation, we considered a two-dimensional collision of an incident proton (kinetic energy of order of KeV) at a rest electron where the initial separation between them is relatively small. Their mutual interaction resulted in both of them having the same optimal time stepsize at all time steps. As the proton moves along its direction of motion with very small oscillations, the electron rotates around the proton (due to the attractive Coulomb force). Hence, the distance between them changes as well as their momenta, which reflects on their optimal stepsizes depicted in Fig. 3.6. This shows the adaptivity of the Simò integrator as the optimal time stepsizes were proportional to the distances, taking into account the change in the momenta.

To show the adaptivity of the Simò integrator in a simulation with more than two particles, we simulated a beam of 10,000 relativistic protons with initial kinetic energy of about 7 TeV. The beam was initially within a cube of 2 cm edge length. Here, we used three time bins of equal number of particles and required an accuracy of $\delta = 10^{-12}$. Due to the repulsive Coulomb force between the protons, the protons move away from each other which should allow for larger optimal stepsizes. Yet, in such a high concentration of protons, a proton could get closer to other protons causing its optimal stepsizes to decrease. Therefore, the Simò integrator allows each proton’s optimal time stepsize to increase or decrease as permitted by its local configuration in the system. Figure 3.7 demonstrates the adaptivity of the optimal time stepsizes of two individual protons in all time steps in this simulation.

If the simulation time is long enough, the time stepsizes of the particles in a beam of same particles species increase in general. The reason for this is the repulsive Coulomb force
Figure 3.6: The change of the mutual optimal time stepsize of an electron orbiting a proton moving along the $z$-axis. The optimal stepsizes change according to the mutual distance and the change on the momenta.

Figure 3.7: Plots of the individual optimal time stepsizes of two different protons in the simulation of a uniform beam of 10,000 relativistic protons, demonstrating the adaptivity of the Simò integrator.

between the particles which causes the particles to scatter, and the increase of the relative distances between particles allows for larger time stepsizes. Fig. 3.8 shows the increase of
the optimal time stepsizes of a selected proton from a simulation of a spatially uniform beam of 100 non-relativistic protons with an average kinetic energy of 1 MeV. The initial size of the beam was within a cube of 2 mm edge length, and the simulation was performed with three bins of equal time widths for an accuracy of $\delta = 10^{-16}$. The optimal time stepsizes were small at the beginning of the simulation (of order of the initial size of the beam) and increased gradually thereafter.

Figure 3.8: The change of the optimal time stepsizes of a proton in the simulation of a uniform beam of non-relativistic protons. The optimal stepsizes increase as a result of the protons moving away from each other.

We tested how the optimal time stepsizes change if we replace the protons with heavier particles (such as lead ions), or with lighter particles (such as electrons). At fixed energy, the more massive the particle is, the slower it will be, and therefore we expect its time stepsize to be larger. Fig. 3.9 compares the optimal time stepsizes of beams of protons, lead ions, and electrons. Since the distances between the particles within a beam are about the same for all the three beams, and the average kinetic energy of each beam is also about the same,
the different masses alone are responsible for the difference in the time stepsizes. As the lead ions are the heaviest, they are the slowest and thus have larger stepsizes than the protons and the electrons.

Now, if we increase the beams’ initial kinetic energy such that the particles become relativistic and compare their optimal time stepsizes with the non-relativistic beams, we expect the stepsizes to be smaller because the particles are traveling much faster. In Fig. 3.10, the optimal time stepsizes of a relativistic (average kinetic energy is 7 TeV) and non-relativistic
proton beams are illustrated. With the initial distances between the particles being the same for both beams, we see in Fig. 3.10 that the stepsizes of the relativistic protons are smaller (or comparable) with those of the non-relativistic protons for some time, then they become larger. Due to their high energy, the relativistic protons scatter faster than the non-relativistic protons, allowing for larger optimal stepsizes to be taken sooner than the non-relativistic beam.

Figure 3.10: Comparison between the optimal time stepsizes of a non-relativistic and a relativistic proton beams. The optimal stepsizes of the relativistic beam are smaller for some time and increase faster than the stepsizes of the non-relativistic beam.

It is clear that the distances between the particles in a beam plays a major role in how large or small the time stepsizes will be. For instance, increasing the initial variance of the distribution of a uniformly distributed beam allows for larger time stepsizes to be taken by the particles. As another example, a Gaussian distribution of a beam will change how the optimal time stepsizes are distributed because there are more particles in the core of the beam with smaller distances between them compared to the distances from the halo.
particles. Therefore, there will be more particles with relatively smaller optimal stepsizes. Fig. 3.11 shows a histogram of the optimal stepsizes of a Gaussian non-relativistic proton beam at one of the time steps, which can be compared with the stepsizes of the uniform beam shown previously in Fig. 3.9a.

![Histogram of optimal time stepsizes](image)

Figure 3.11: A histogram of the optimal time stepsizes of a Gaussian beam of non-relativistic beam of 100 protons at one of the time steps. There are more particles in the core of the beam with relatively smaller stepsizes than the halo particles.

Applying external fields to a beam will affect the behavior of the optimal time stepsizes depending on how the fields will influence the distances between the particles and their momenta. As an example, we applied a magnetic dipole field of 0.2 T in the y-direction to a uniform beam of non-relativistic protons. As a result, the beam rotates in the negative x-direction while the protons drift away from each other along the y-direction. Consequently, the optimal stepsizes oscillate with some overall increase during the simulation as illustrated in Fig. 3.12.
Figure 3.12: The change of the optimal time stepsizes of a uniform beam of non-relativistic protons subjected to a magnetic dipole field in the $y$-direction. The optimal stepsizes oscillate with an overall increase as the particles rotate and spread in the $y$-direction.

As the Simò integrator decides the optimal time step sizes depending on the configuration of the system, the variable order is selected to achieve the required accuracy. For example, the calculated optimal orders of a uniform beam of non-relativistic protons when the required accuracy was low ($\delta = 10^{-5}$) did not exceed the second order, while higher orders were needed for the high accuracy of $\delta = 10^{-16}$ as shown in Fig. 3.13a. These optimal orders are compared with the ones of the relativistic beam in Fig. 3.13b. As explained previously, the optimal time stepsizes of a relativistic beam are smaller than the stepsizes of a non-relativistic beam (for beams of same size and particle types), and hence the relativistic protons required lower optimal orders. Fig. 3.13 also indicates that the Simò integrator allows low orders when possible, and requires high orders only when needed.

Since the optimal time stepsizes are influenced by the particles masses as we have seen earlier, the optimal orders will be affected too. The heavy particles (such as lead ions)
Figure 3.13: Histograms to compare the calculated optimal orders (in about 400 time steps) at high accuracy in the simulation of two beams (100 protons per beam) at different energies: (a) non-relativistic, (b) relativistic. The relativistic protons required lower optimal orders than the non-relativistic protons.

had optimal stepsizes that were large compared to the lighter particles (such as protons and electrons), and therefore the lead ions required higher optimal orders than the protons and electrons did. Fig. 3.14 shows that at a low average kinetic energy of 1 MeV, the optimal orders are higher for lead ion beams than electron beams when $\delta = 10^{-16}$. Those optimal orders can be compared to the optimal orders of the non-relativistic protons beam in Fig. 3.13a which are lower than the lead ions’ optimal orders and higher than the electrons’ optimal orders.

### 3.4.1 Efficiency and Accuracy

We demonstrate the performance of the Simó integrator using some of the examples presented in Section 3.4. First, we show the efficiency of the Simó integrator’s particle-by-particle adaptive, variable order algorithm by comparing its CPU time with the CPU time of the integrator when employing different algorithms: a fixed order and adaptive time stepping algorithm, a fixed time stepsize and variable order algorithm, and a fixed order and fixed
Figure 3.14: Histograms to compare the required optimal orders to achieve an accuracy of \( \delta = 10^{-16} \) of low energy beams of: (a) lead ions, (b) electrons. The massive lead ions required higher optimal orders than the light mass electrons.

The comparison was performed for a specific simulation time for three beams: proton beam, electron beam, and lead ion beam. The three beams have the same initial conditions: 100 particles uniformly distributed within a cube of 2 mm edge length, and a low initial average kinetic energy of 1 MeV. Table 3.1 includes the comparison of the CPU times, and shows that the Simò integrator’s adaptive, variable order algorithm is the most efficient. The algorithm of the fixed stepsize was usually the slowest, especially if combined with a fixed order. The adaptive algorithm with a fixed order could be less efficient if the order is very high as in the case of the lead ion beam.

Second, we present some examples for the conservation of the energy and momentum of the Simò integrator. We used the same three beams and required an accuracy of \( \delta = 10^{-16} \). Fig. 3.15 shows the relative errors of the total momentum and energy as functions of the simulation time, and indicates very good conservation of the energy and momentum.
### Table 3.1: A comparison of the CPU times of the Simò integrator of different algorithms: adaptive with variable order, adaptive with fixed order, fixed stepsize with variable order, and fixed stepsize and order. The algorithms were compared for three beams at the same initial conditions and the same simulation time, but each beam is of a specific species of particles.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Proton Beam</th>
<th>Electron Beam</th>
<th>Lead Ion Beam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive</td>
<td>93.24</td>
<td>78.72</td>
<td>154.03</td>
</tr>
<tr>
<td>Adaptive</td>
<td>160.22</td>
<td>119.77</td>
<td>798.86</td>
</tr>
<tr>
<td>Fixed</td>
<td>1777.35</td>
<td>15795.71</td>
<td>269.69</td>
</tr>
<tr>
<td>Fixed</td>
<td>6091.32</td>
<td>41960.59</td>
<td>4096.05</td>
</tr>
</tbody>
</table>

3.4.1.1 **Time Stepsize Computation at Higher Orders**

In Section 3.3.2, we have seen that estimating the radius of convergence at the second order is sufficient. However, we have noticed that this was not true in some later applications performed by the Simò integrator. In Fig 3.16a, we show an example of how the radius of convergence can vary notably when its computed at variable orders. As a result, the estimated $\tilde{R}_n$ with an optimal time stepsize computed at the second order $h_2$ became

![Graph showing conservation of energy and momentum](image-url)
significantly large as the order increases as shown in Fig 3.16b. On the other hand, using an optimal time stepsize computed as the order varies $h_{\text{order}}$ resulted in decreasing the estimated $\tilde{R}_n$ with increasing the order as it is expected (Fig 3.16b).

![Graph](image)

(a) The estimated radius of convergence. (b) The estimated errors.

Figure 3.16: An example where the estimation of the radius of convergence, and hence the optimal time stepsize, can significantly change for different orders: (a) The estimated radius of convergence computed at variable orders, (b) The estimated errors associated with the optimal time stepsize when the stepsize is computed at the second order, and when its computed at variable orders.

In order to improve the accuracy of the Simô integrator code, we have implemented a re-computation of the radius of convergence for the particles that reaches high orders while the condition $\tilde{R}_n < \delta$ is still unsatisfied. Moreover, this re-computation is only performed twice at most for each particle: at half the maximum Simô order, and at the maximum Simô order if needed. This is done to ensure that the efficiency of the Simô integrator is not degraded due to this re-computation.

### 3.5 Parallel Implementation of the Simô Integrator Code

Since the Simô integrator algorithm is a direct $N$-body method, the computational time of the serial Simô integrator code is proportional to $N^2$. This means that the efficiency of
the serial code will decrease as the runtime increases quadratically for large $N$. In order to enhance the performance of the Simò integrator code and to be able to deal with a large particle number, the parallel implementation of the Simò integrator code has been developed.

The performance of a parallel code depends on the distribution of the data/tasks among the processors/nodes, the synchronization and communication between processors/nodes, and gathering the final results. We take advantage of the fact that the Simò integrator code is written in COSY INFINITY 9.1 which supports MPI based parallelization. COSY INFINITY includes a set of the MPI commands which are integrated with the COSYScript, resulting in a straightforward parallel loop construction in a similar way to OpenMP. The main parallel processing construct in COSY INFINITY is the `PLOOP-ENDPLOOP`. In addition, the intrinsic procedure `PNPRO` allows to dynamically find the number of processors available for the application. The parallel implementation was performed using Gaea, a high performance computing cluster at Northern Illinois University (NIU) (described in Appendix C).

### 3.5.1 Parallelization of Serial Loops

The main loops that require long computational time are the ones needed for the computations of the mutual forces

$$ q_i \sum_{j=1, j \neq i}^{N} \frac{q_j (r_i - r_j)}{|r_i - r_j|^3}. \quad (3.7) $$

These computations are performed using two nested loops with $N(N - 1)$ operations

```plaintext
LOOP I 1 N;
LOOP J 1 N;
IF J \neq I;
```

;
Let the loop over the $i$ index be LOOP1, and the loop over the $j$ index be LOOP2. We employed three parallelization strategies of the force computations: parallelize only LOOP1, parallelize both LOOP1 and LOOP2 using a nested PLOOP, and parallelize only LOOP2. The performance of these strategies is examined in Section 3.5.2. We usually set the length of the PLOOP to be the same as the number of processors $P$ (always integer), and thus we have $P$ MPI processes. Parallelizing either serial loops means that each MPI process will perform computations for about $N/P$ particles depending on the used parallelization strategy.

### 3.5.2 Parallel Performance Tests

The performance of a parallel code is usually illustrated by its scalability, which shows how the runtime changes as the number of processors increase. There are two types of scaling: strong scaling and weak scaling. The strong scaling shows how the computational time varies with the increase of the number of processors for a fixed problem size. The weak scaling shows how the computational time changes as the problem size increases at the same rate as the number of processors while the problem size per processor is fixed.

In the strong scaling, one way to quantify the parallel performance is to measure the speedup $S(N, P)$ achieved by using $P$ processors. Assuming $T(N, P)$ is the amount of time it takes to complete a problem size of $N$ with $P$ processor, the speedup is typically linear with $P$, and is defined as

$$S(N, P) = \frac{T(N, 1)}{T(N, P)}.$$
However, an ideal linear speedup is rarely achieved because most codes include serial parts.

Since the computational time of the $N$-body problem is proportional to $N^2$, then

$$T(N, P) \propto \frac{N^2}{P}.$$  

For a fixed problem size $N$ and variable $P$, the resulted speedup is

$$S(N, P) = \frac{T(N, 1)}{T(N, P)} \propto \frac{N^2/1}{N^2/P} \propto P.$$  

In the weak scaling, the parallel efficiency $E(N(P), P)$ is considered where the total number of particles $N(P)$ is varied with the number of processors such that the problem size per processor starting with $N(1) = N$ remains constant. Here, the computational time $T(N(P), P)$ of $N(P)$ performed by $P$ processors should be the same as the computational time of the serial implementation $T(N(1), 1)$ with one processor. Then, the parallel efficiency can be written as

$$E(N(P), P) = \frac{T(N(1), 1)}{T(N(P), P)} \times 100\%.$$  

For the $N$-body problem, when the problem size is $N(1) = N$ using one processor, the workload of the processor is $N^2/1$. As $P$ varies, the total number of particles $N(P)$ vary as follow

$$\frac{N^2(P)}{P} = \frac{N^2}{1}$$

$$\Rightarrow N(P) = \sqrt{PN^2} = \sqrt{PN}.$$  

Thus, the parallel efficiency is

$$E(N(P), P) \propto \frac{N^2/1}{\left(\frac{\sqrt{PN}}{P}\right)^2} \times 100\% \approx 100\%.$$
This implies that a good weak scaling performance should have a constant efficiency that is about 100% as we increase $P$ and $N(P)$, but $E(N(P), P)$ is usually limited due to the communication overhead that grow with the number of processors.

We employed our first parallelization strategy of the force computations by parallelizing only LOOP1 and checked its scalability for a fixed problem size. Its strong scaling performance is shown in Fig. 3.17 using $N = 10^4$ particles and varying the number of processors per node and using 1, 2 and 4 nodes. Fig. 3.17 indicates poor scalability as the speedup was very low.

![Figure 3.17: The speedup resulted from the parallelization of the first loop of the force computation for a different number of processors and nodes, which indicates a low performance.](image)

The next parallelization strategy was to parallelize both LOOP1 and LOOP2. In this nested PLOOPs, the number of processors used by LOOP1 is $P_1$ and LOOP2 uses $P_2$ processors, such that the total number of processors $P = P_1 \times P_2$. We compared the CPU time when varying $P_1$ and $P_2$ for a fixed $P$ and $N$ to see which lengths of LOOP1 and LOOP2 will give the fastest run. An example of these comparisons is in Fig. 3.18 performed using 24 processors in 2 nodes. As $P_1$ increases ($P_2$ decreases), Fig. 3.18 shows that the runtime...
changes slightly until $P_2 = 2$ and it increases significantly when $P_2 = 1$ and $P_1 = P$. This confirms the low performance of our first parallelization strategy by parallelizing only LOOP1, and suggests that a better scalability can be achieved by using less processors for $P_1$ and more processors for $P_2$.

![Figure 3.18: The speedup resulted from the parallelization both loops of the force computation, varying the number of processors for each parallel loop.](image)

Lastly, we carried out scalability tests for our third parallelization strategy of parallelizing only LOOP2. The strong scaling was tested using $N = 5 \times 10^4$ particles, and its speedup, shown in Fig. 3.19a, was calculated for variable $P$ using 1 processor per node and 6 processors per node. The depicted speedup indicates a high performance of the parallel Simò code when only LOOP2 is parallelized, and the performance decreases when communication overhead grows for large $P$. Fig. 3.19a, along with other numerical experiments, shows that the collective communication calls among processors in one node is slower than the communication between processors on different nodes. Consequently, it is preferred to use more nodes and less processors per node. In terms of the weak scaling, we computed the parallel efficiency
varying the total problem size while fixing the problem size per processor for three cases: \( N = 10^3 \), \( N = 4 \times 10^3 \) and \( N = 10^4 \) particles. The resulted efficiency is illustrated in Fig. 3.19b which shows a good parallel efficiency as the total problem size and the number of processors vary until the communication overhead grows and decreases the efficiency. Fig. 3.19b also demonstrates a better efficiency as the problem size per processor increases which is expected as the computational time will be less than the communication overhead for the small problem size per processor.

![Graphs](image)

(a) The speedup of strong scaling.  
(b) The efficiency of the weak scaling.

Figure 3.19: The performance of the Simò code resulted from the parallelization of the second loop of the force computation: (a) The speedup of the strong scaling for a different number of processors and nodes which indicates a high performance, (b) The efficiency of the weak scaling showing high efficiency as the initial problem size per processor increases.

As a result of these parallel performance tests of the three parallelization strategies, we have decided to employ the last parallelization strategy of parallelizing only LOOP2 in the parallel Simò integrator code. An explanation of why this parallelization strategy gave better performance than the other two is in the following Section 3.5.3. In terms of CPU time, the simulation of \( 10^5 \) particles with the last parallelization strategy using different number of processors is shown in Fig. 3.20. In Fig. 3.20, the high parallel performance of the Simò integrator is demonstrated as the CPU time decreases with increasing the number
of processors, which changes from about 20 hours using 6 processors to about 2 hours using 36 processors.

Figure 3.20: The performance of the parallel Simò integrator when only the second loop of the force computation is parallelized is illustrated by the decrease of the CPU time as a function of the number of processors when simulating $10^5$ particles.

3.5.3 Time Binning and the Parallelization

Time bins play an important role in the efficiency of the Simò integrator code because they determine the number of particles that will be propagated in the current time step and will require generating new Taylor expansions of the solutions to their ODEs in the following time step. This number is usually just a fraction of the total number of particles, which greatly decreases the computational time as shown before. This is one main reason that the parallel performance of the Simò integrator is best when only the \texttt{LOOP2} of the force computation is parallelized. For the first parallelization strategy, the serial \texttt{LOOP2} takes along time to span the total number of particles while \texttt{LOOP1} spans the small fraction of the
number of particles due to time binning which increases the possibility of the communication overhead to grow fast. Therefore, the overall parallel performance is poor.

In case both LOOP1 and LOOP2 are parallelized, we saw that the performance depended on the number of processors assigned to each loop and that the parallel performance was better when LOOP2 runs with most of the processors. Therefore, we have concluded that it is best to only parallelize LOOP2 since it gave the best performance.

Now, we want to examine the effect of varying the number of time bins on the CPU time of both types of time bins. We used \( N = 5 \times 10^4 \) particles and two types of distributions: uniform and Gaussian, and we ran the simulations with the parallel Simò integrator code using 12 processors. In Fig 3.21a, both the two types of time bins have a similar behavior in terms of the CPU time. For a small number of time bins, the time is long because we have a lot of particles in the first bin. Then, the time decreases as the number of time bins is increasing which decreases the number of particles in the first bin. After reaching a minimum, the time starts increasing as we have very few particles in the first bin that are being propagated at each time step, which increases the number of time steps needed to finish the simulation time and thus increase the runtime. For a Gaussian beam in Fig 3.21b, the CPU time behavior is similar to that of the uniform beam when the time bins are of equal number of particles. The runtime does not show a sudden increase after reaching a minimum in the case of time bins of equal time widths, but it seems to be relatively constant. This could be explained by the fact that the mutual distances between the particles in the core are small, causing their optimal time steps to be small. Therefore, both the width of the first bin and the number of particles in it remain relatively fixed, and the CPU time does not change so much after reaching the minimum.

The relation between the number of processors and the time bins is shown in Fig 3.22 for the same previous configurations. In general, the runtime of the parallel Simò code decreases with increasing the number of processors regardless the number and type of time
(a) Uniform beam. (b) Gaussian beam.

Figure 3.21: The CPU time of the parallel Simò code as a function of the number of time bins for two types of particles distributions: (a) Uniform distribution, (b) Gaussian distribution. The time decreases as the number of time bins increases until it reaches a minimum, and then it starts increasing again.

However, the efficiency gained by increasing the number of processors could be lost when the number of time bins is large. This is a result of the growing communication overhead by the parallel loop used to propagate the very few particles in the first bin.

### 3.5.4 Memory Usage

An important aspect of the parallel implementation of the code is to keep the memory usage independent of the number of processors used by the code. In Fig 3.23, we show that this feature is satisfied by our parallel Simò integrator code, and that the memory usage is mainly dependent on the number of particles.
Figure 3.22: The CPU time of the parallel Simò code as a function of the number of time bins using different number of processors, for both types of time bins and both uniform and Gaussian distributions.

3.6 The Window Feature

The charged particle beams in accelerators are confined within the beamline pipe, and some particles escape the bunch and hit the pipe occasionally. Therefore, we have equipped the Simò integrator code with what we called the window feature that represent the beamline pipe boundaries. The limits of these boundaries can be set in the code for both $x$ and $y$ directions, and any particle that reaches those limits will be eliminated from the simulation.
Figure 3.23: The estimated memory usage of the parallel Simò integrator code almost independent of the number of processors, and mainly dependent on the number of particles. This feature is also important in the computational sense since any particle that goes far away from the bunch will induce numerical errors. In addition, the user has the choice to include the window in the simulation or not. As an example, Fig 3.24a shows an initial uniform distribution of particles within a radius of 1.5 mm and we applied a window with minimum limit of $-1$ mm, and a maximum limit of 1 mm in both $x$ and $y$ directions. The results are illustrated in Fig 3.24b where all the particles beyond the window limit were eliminated.
Figure 3.24: An example of eliminating particles using the window feature of the Simò integrator code: (a) Initial distribution of the particles before applying the window, (b) Distribution of particles after applying the window.
CHAPTER 4
PARTICLES’ HIGH-ORDER ADAPTIVE DYNAMICS (PHAD)

4.1 Introduction

The computational complexity of the Simò integrator is of $O(N^2)$, and thus the efficiency of the Simò integrator starts decreasing as $N$ increases. Also, the symplecticity feature can be lost when long time dynamics are involved due to the discretization. Consequently, the Simò integrator is not appropriate to model problems of very large $N$ or problems of long-time dynamics. Because both issues are important for some practical applications of charged particle beams, it is essential to develop an accurate collisional method that reduces the computational complexity to better than quadratic with $N$, and to maintain the symplecticity of the Hamiltonian system in the long-time scale. Hence, our research group have developed a novel collisional algorithm termed PHAD (Particles’ High-Order Adaptive Dynamics) that can reduce the computational complexity of the $N$-body problem to $O(N)$ and preserve the symplecticity feature of the long time dynamics. The main components of PHAD algorithm are Strang splitting which separates the forces to near and far forces and ensures symplecticity; the Fast Multipole Method (FMM) that reduces the computational complexity of the pair-wise forces in the far region; and an accurate time integrator that can resolve all collisions in the near region efficiently. An earlier version of PHAD appeared in [102, 26], where a Picard-Iteration based integrator [91] was used for the time stepping. Here, we replace the Picard-Iteration based integrator with the Simò integrator for an upgraded accuracy and efficiency of the PHAD algorithm. Furthermore, PHAD algorithm becomes
fully adaptive both in time and space without the loss of symplecticity, and the algorithm is fully parallelized.

4.2 Fast Multipole Method

In the direct collisional methods such as the Simò integrator, the $O(N^2)$ comes from the computations of the pair-wise Coulomb forces between the particles. To reduce this complexity, several techniques were developed such as PIC methods and the hierarchical methods. Among the hierarchical methods is the Fast Multipole Method (FMM) developed by Greengard and Rokhlin [59], which can reduce the complexity to $O(N)$.

For charged particle beams, the FMM is an efficient tool to calculate the Coulomb effect of a discrete set of charged particles $N$ on a test charge at a particular location. In Cartesian coordinates, the scalar electrostatic potential in infinite free space at a point $\mathbf{x}$ is given by

$$\phi(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{q_i}{\|\mathbf{x} - \mathbf{x}_i\|}. \quad (4.1)$$

In the FMM context, sources and targets are used to describe the set of charges that create the potential and the locations that the potential is evaluated at, respectively. Thus, in particle beams, the sources and targets are equivalent lists that include all the particles where $q_i$ and $\mathbf{x}_i$ in Eq. 4.1 refer to the sources.

The hierarchical subdivision of space applied by the FMM is based on the assumption that a group of particles are far away enough that their interaction forces are smooth. Therefore, this subdivision is performed without changing the original charge distribution, giving a solution close to that of the direct method. Equation 4.1 is approximated by the FMM by dividing the sum into a set of near and far evaluations. The near evaluations can be computed exactly, and the far evaluations are represented as a sum of multipole expansions.
level FMM employs two shifting operators which results in an algorithm with a computational complexity that scales asymptotically as $O(N)$. Furthermore, a novel adaptive multi-level FMM was introduced to treat different non-uniform spatial distributions [103, 26]. We give here an overview of the adaptive multi-level FMM, and the details can be found in [26].

In the adaptive multi-level FMM, the entire three-dimensional spatial distribution is first scaled into a unit cube called the root box. Then, recursive hierarchical subdivision takes place where each box is divided into congruent eight boxes. These subdivisions are referred to by a level $l$ at which they occur at. For instance, the level $l = 0$ is given to the root box where no subdivisions have been performed, and the first subdivision of this root box into the congruent eight boxes is called the first level subdivision $l = 1$ and so on for the next levels. If any two boxes are separated by a distance $b$, they are regarded as well-separated. Also, any set of boxes that share at least one side or a vertex with a specific box at the same level are called its neighbors. Any neighborhood consists of the box and its neighbors, and thus a neighborhood may contain up to 27 boxes. An interaction list is assigned to any box and it consists of the children boxes of the neighbors of its parent box without its own neighbors (up to 189 boxes).

The partitioning continues adaptively until a specific condition set by the clustering parameter $q$ is satisfied. For any specific target, the clustering parameter $q$ represents the maximum number of sources allowed in the neighborhood of that target. At the end, there will be $2^{6l}$ congruent boxes of side length $b/2^l$, where $b$ is the side length of the root box. This adaptive multi-level FMM ensures that there are no empty boxes and no over-populated boxes.

In the computational domain, the subdivision process is described by an octree type data structure where each box that is subdivided is called a parent box, and the eight boxes resulting from the division are regarded as child boxes. The tree structure is organized to include sets of source boxes and target boxes. Since the source and target boxes distributions
vary from each other in realistic applications, they are assigned to two different sets of data structures called trees. The set of target boxes which contain targets and satisfy the clustering parameter condition form the D-tree while the parent-child relationship traversing from the root box downwards is preserved. The boxes in the D-tree that contain sources are the interaction lists of the target boxes, and thus the C-tree is formed by traversing through the D-tree boxes bottom-up and obtaining their parent-child relationships. The process of selecting the D-boxes makes them appropriately organized into a single tree, while the process of selecting the C-boxes may result into a set of disconnected trees. The collection of trees of the same type is called a forest, and thus a C-forest is formed by the collection of C-trees while there is no D-forest.

After the subdivision and the trees structuring, the multipole expansion of the individual boxes is computed at the center of each box at the highest level. Then, the FMM applies two passes to translate the multipole expansions. The upward pass uses the C-forest and translates the expansions from the largest level to level 2 by applying the multipole-to-multipole (M2M) operator. The downward pass starts once level 2 is reached, and at each level, the expansion is translated to the boxes in the interaction list using the multipole-to-local (M2L) translation. As a result, the local expansion will be the sum of the translated local expansion and the existing local expansion of the box. Then, the local-to-local (L2L) translation is applied to shift the new local expansion to the center of the child boxes at the next larger level. The downward pass is completed by evaluating the local expansion at each target point. These evaluations are the far evaluations, and the near evaluations are performed by the point-to-point (P2P) calculations. At the end, the near and far interactions are added to give the total interactions.

In the non-adaptive multi-level FMM, the operators M2M and L2L are not employed while they mainly contribute to the advanced efficiency of the adaptive multi-level FMM. In addition, the adaptive multi-level FMM was implemented in the framework of Differential
Algebra (DA, described in Appendix A) where the translation operators can be computed efficiently and give rise to additional advantages as described in [104]. The three dimensional adaptive multilevel FMM in Cartesian DA was presented in [103, 26] and is outlined in the following Section 4.2.1.

### 4.2.1 Differential Algebraic FMM

Consider the electric potential at the center of a box \( \mathbf{x}_0 = (x_0, y_0, z_0) \) due to \( n \) sources of charges \( q_i \) at positions \( \mathbf{x}_i = (x_i, y_i, z_i) \), Eq. 4.1 can be written with arbitrary units where \( q_i/(4\pi\varepsilon_0) = 1 \) as

\[
\phi(x_0, y_0, z_0) = \sum_{i=1}^{n} \frac{1}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}},
\]

(4.2)

With \( r = \|\mathbf{x} - \mathbf{x}_i\| \), we define the DA variables \( d_r, d_x, d_y, \) and \( d_z \) as functions of the coordinates of the target (evaluation point):

\[
\begin{align*}
    d_r &= \frac{1}{r}, & d_x &= \frac{x - x_0}{r^2}, \\
    d_y &= \frac{y - y_0}{r^2}, & d_z &= \frac{z - z_0}{r^2}. 
\end{align*}
\]

(4.3)

The potential is expanded in terms of these DA variables and become

\[
\phi(x_0, y_0, z_0) = \sum_{i=1}^{n} \frac{d_r}{\sqrt{1 + \|\mathbf{x}_i - \mathbf{x}_0\|^2 d_r^2 - 2 (\mathbf{x}_i - \mathbf{x}_0) \cdot \mathbf{d}}} = d_r \cdot \phi_m,
\]

(4.4)

where \( \phi_m \) is the multipole expansion of the potential.

Applying the multipole expansion of the potential gives a series with different powers of the DA variables that converge only if \( r \gg b \). This means that a well-separated source
and target boxes is required which our data structure takes into account. High FMM order increases the accuracy of the computed potential since it includes higher order terms of the expansion. In our FMM algorithm, we use COSY INFINITY to employ the DA representation of the potential and calculate its multipole expansions at the center of each box at the largest level only once. These multipole expansions can be translated as needed using different translation operators described as maps of DA vectors in COSY INFINITY.

First, the multipole expansion $\phi_m$ of a specific box at a level $l + 1$ is computed by evaluating Eq. 4.4 in DA. This multipole expansion is translated to the center of another well-separated box at a level $l$ using the M2M operator $T_{mm}$ resulting into the multipole expansion $\phi_{m2m}$. Thus, the translation is from a child box to a parent box and the translation process is expressed mathematically as a composition of the two maps $\phi_{m2m} = \phi_m \circ T_{mm}$. The $T_{mm}$ is performed from the largest level until level 2.

Second, the M2L operator $T_{ml}$ translates the multipole expansion of a parent box $\phi_{m2m}$ to a well-separated box at the same level. This operator is performed by any occupied interaction list, and the local multipole expansion $\phi_l$ of this box is computed by the composition $\phi_l = \phi_{m2m} \circ T_{ml}$. This expansion is valid within a radius that contains the targets meaning all the children of that box. In the FMM algorithm, the M2L operator is identified as the most computationally expensive operator [105, 106], and hence it is optimized in our implementation using rotation [107, 26].

Next, the after rotation local multipole expansion $\phi'_l$ at the center of the parent box is translated to the centers of its child boxes using the L2L operator. The new local multipole expansion $\phi_{l2l}$ is calculated by the composition: $\phi_{l2l} = \phi'_l \circ T_{ll}$. The operators $T_{ml}$ and $T_{ll}$ are performed from level 2 to the largest level.

Last, the potential is evaluated at the location of each target point inside any child box using the L2P operator. The final potential at any target point is the sum of the evaluated potential at that target and the interaction potential between the target and
its neighborhood. This interaction potential is computed directly using the P2P operator (Coulomb potential).

### 4.3 Strang Splitting

The dynamics of the charged particle beams usually need to be followed for a long time. While our novel adaptive multi-level FMM gives an optimal solution of force computation in the high accuracy regime, it would require applying this FMM with very small time stepsizes. In fact, this time stepsize is the smallest stepsize that a particle requires at each simulation time step. In addition to this being inefficient in simulating long time dynamics, it is essential to preserve the geometrical properties of the time-continuous system in the discretization [108]. Because the motion of the charged particles in electromagnetic fields is Hamiltonian, it is essential for the numerical simulations to preserve the symplecticity feature [109]. The problem of the very small time stepsizes can be solved using adaptive integrators, but that causes them to lose the symplectic structure.

We approach this problem by employing splitting where the Hamiltonian can be split into two parts as $H = H_1 + H_2$, and hence the differential equations can be solved by applying a symplectic operator splitting method [108, 110]. Specifically, we apply Strang splitting, a second order accurate operator splitting method [111, 112]. In Chapter 3, we have derived the system of differential equations of a particle $i$ in Eq 3.6 and their solutions are given in the array $\hat{Y}_i$. The IVP can be written as

\[
\begin{align*}
\dot{\hat{Y}} &= F(\hat{Y}) \\
\hat{Y}(0) &= \hat{Y}_0.
\end{align*}
\]
The function $F$ represents the right hand side of Eq 3.6, which we rewrite here in a compact form as

$$\dot{x}_i = \frac{\hat{p}_x i + \hat{p}_y j + \hat{p}_z k}{(f_i^2 + \hat{p}^2_{\tau_i} + \hat{p}^2_{\tau_i} + \hat{p}^2_{\tau_i})^{1/2}}, \quad (4.5)$$

and

$$\frac{d\hat{p}_i}{dt} = \frac{qn_i}{mc^2} \left[ \frac{q}{4\pi\epsilon_0} \sum_{j=1}^{N} n_j \left( (x_i - x_j) i + (y_i - y_j) j + \gamma^2 (z_i - z_j) k \right) + \gamma \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right]^{3/2} + E_i + c\hat{v}_i \times B_i \right]. \quad (4.6)$$

According to Strang splitting, the function $F$ is split into two simpler parts as $F = F^{[1]} + F^{[2]}$, and now we have two IVPs

$$\begin{cases} \dot{\hat{Y}} = F^{[1]}(\hat{Y}) \quad \text{and} \quad \dot{\hat{Y}} = F^{[2]}(\hat{Y}) \\ \hat{Y}(0) = \hat{Y}_0 \quad \text{and} \quad \hat{Y}(0) = \hat{Y}_0 \end{cases}. \quad (4.7)$$

If these two IVPs have the exact solutions $\phi^{[1]}_{\tau/2} (\hat{Y}_0)$ and $\phi^{[2]}_{\tau/2} (\hat{Y}_0)$ at a time step $\tau$, then the full solution is a composition of these two solutions such that

$$\phi_{\tau} = \phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2} \circ \phi^{[2]}_{\tau/2} (\hat{Y}_0). \quad (4.8)$$

In Section 4.2, we saw that the FMM is applied to our charged particle beams based on the fact that the forces acting on a particular particle come from two regions: near and far regions. The forces due to the particles in the near region are fast varying and require frequent evaluations at small time stepsizes. On the other hand, the forces from the far region are slow varying and considered smooth and do not change within a relatively large step. Based on the FMM partitioning of the spatial distribution, the function $F$ is split such
that $\mathbf{F}^{[1]}$ is the near region equation and $\mathbf{F}^{[2]}$ is the far region equation. Therefore, we split Eq. 4.5 to

$$
\dot{\mathbf{x}}^{[1]}_i = \frac{\dot{p}_x_i \mathbf{i} + \dot{p}_y_i \mathbf{j} + \dot{p}_z_i \mathbf{k}}{(f^2_i + \hat{p}^2_x + \hat{p}^2_y + \hat{p}^2_z)^{1/2}},
$$

(4.9)

and

$$
\dot{\mathbf{x}}^{[2]}_i = 0.
$$

(4.10)

Similarly, Eq. 4.6 can be expressed as the sum of

$$
\frac{d\mathbf{p}^{[1]}_i}{dt} = \frac{qn_i}{mc^2} \left[ \frac{q}{4\pi\epsilon_0} \sum_{j=1 \atop j \neq i}^N n_j \left( (x_i - x_j) \mathbf{i} + (y_i - y_j) \mathbf{j} + \gamma^2 (z_i - z_j) \mathbf{k} \right) \gamma \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right]^{3/2} + \mathbf{E}_i + e\mathbf{v}_i \times \mathbf{B}_i \right],
$$

(4.11)

and

$$
\frac{d\mathbf{p}^{[2]}_i}{dt} = \frac{qn_i}{mc^2} \frac{q}{4\pi\epsilon_0} \sum_{j \in N^c_i} n_j ((x_i - x_j) \mathbf{i} + (y_i - y_j) \mathbf{j} + \gamma^2 (z_i - z_j) \mathbf{k}) \gamma \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + \gamma^2 (z_i - z_j)^2 \right]^{3/2}.
$$

(4.12)

To get the solution over a long simulation time $t = k\tau$, we iterate solving the near and far equations at fixed time steps $\tau$ where the final solution is given as a composition of all solutions.

$$
\underbrace{\phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2} \circ \phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2} \circ \cdots \circ \phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2} \circ \phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2} \circ \phi^{[2]}_{\tau/2}}_{k \text{ compositions of } \phi^{[2]}_{\tau/2} \circ \phi^{[1]}_{\tau/2}} (Y_0).
$$

(4.13)

The near equations are solved numerically using the Simò integrator which takes different subsequent time steps adaptively to accurately model collisions and resolve all close encounters, giving a solution that is accurate to machine precision. In the far region, there are no close encounters, and the interaction forces are smooth and thus they can be solved exactly using the FMM. Therefore, we can significantly reduce the computational cost by calling the FMM with relatively large, fixed time steps. While combining the numerical and exact
solutions do not generally produce symplectic solution [108], a numerical solution that is accurate to machine precision allows to preserve symplecticity to machine precision.

4.4 Accurate Time Stepping

In the first implementation of PHAD algorithm [26], collisions in the near region were modeled by a Picard-iteration based integrator [91]. While this integrator works well for the time stepping in PHAD, its drawbacks were that it was not adaptive and the time stepsize have to be chosen carefully to ensure both efficiency and accuracy. As we described in Chapter 3, we have developed the Simò integrator, a collisional method that is variable order, adaptive, and with dense output. It is also devised with automatic selection of the particle-by-particle optimal order and time stepsize based on a theorem of Simò [86], and utilizes the Differential Algebraic methods [93]. In addition, the Simò integrator can achieve high accuracy levels up to machine precision without losing efficiency. Therefore, the Simò integrator is the most suitable choice to achieve optimal time stepping in combination of the FMM for force computation in PHAD.

4.5 PHAD Performance with the Simò Integrator

The integration of the Simò integrator in PHAD gives the first efficient and accurate collisional simulation of charged particle beams in external electromagnetic fields. As the Simò integrator is fully adaptive in time and our multi-level FMM is fully adaptive in space, PHAD algorithm is fully adaptive both in space and time and it is numerically symplectic over a long run. The algorithm of PHAD is illustrated by the flow chart in Fig. 4.1.
Figure 4.1: A flow chart that illustrates PHAD algorithm.

The current parallel algorithm of PHAD uses the serial Simò integrator to solve the near equations of each neighborhood in parallel by each MPI process. The number of particles assigned to the Simò integrator within PHAD is equal to the clustering parameter $q$, and thus the performance of PHAD is going to be affected by varying $q$ value. As noted in [103], a small $q$ results in increasing the number of the most expensive operation in the adaptive multi-level FMM algorithm, the M2L translations. In contrast, a large $q$ increases the direct P2P computations in the adaptive multi-level FMM (the time integration in PHAD) and the performance of the algorithm goes back to $O(N^2)$ when $q = N$. Both limits can decrease the performance of PHAD, and hence it is important to choose the $q$ that gives PHAD a computational efficiency of $O(N)$ provided that other PHAD parameters are also appropriately selected.

A description of selecting an optimum $q$ for the adaptive multi-level FMM was presented in [103], but selecting the $q$ value for PHAD algorithm can be different. As an example, we performed PHAD simulations of $10^4$ electrons of a Gaussian distribution to see how the computational time of PHAD varies with different $q$ values. The results are illustrated in Fig. 4.2 where it can be seen that the computational time of PHAD is mainly composed of
Figure 4.2: The performance of PHAD and its main components, the FMM and the Simò integrator, as a function of the clustering parameter $q$. The simulations were performed for $10^4$ electrons of a Gaussian distribution.

As the $q$ value increases, the computational time of the FMM decreases while the that of the Simò integrator increases. Once the computational time of the Simò integrator becomes larger than that of the FMM, the performance of PHAD starts to decrease. In this example, this occurs for $q > 100$. Since the computational time of the Simò integrator depends on the exact value of $q$, we expect similar behavior for any $N$ and the $q$ value should not be much larger than 100.
CHAPTER 5
APPLICATIONS AND RESULTS

5.1 Electron Cooling Simulations

An important area of research in particle and nuclear physics is the production of particles, known and unknown, by colliding two counter-rotating high energy beams in collider facilities. To produce new effects, it is essential to accelerate the beams to very high energies. In addition, one of the basic concepts in colliders is the number of useful interactions (events) which become critical in the case of a small production cross section $\sigma$ of rare events [113]. This leads to another crucial parameter called the luminosity $L$ that determines if a collider is able to produce the required number of interactions. The luminosity relates the event rate per second $dR/dt$ to the collision cross section $\sigma$ as

$$
\frac{dR}{dt} = L\sigma.
$$

Assuming that the two colliding beams have the same cross section, Gaussian, and both consists of $n_b$ bunches, then the luminosity is given by:

$$
L = \frac{N_1 N_2 f_{\text{rev}} n_b}{4\pi \sigma_x \sigma_y}, \quad (5.1)
$$

where $f_{\text{rev}}$ is the frequency of revolution [113]. The horizontal and vertical beam sizes are $\sigma_x$ and $\sigma_y$, respectively.
Modern particle colliders aim to increase their productivity by maximizing the luminosity at the collision energy. Equation 5.1 shows that high luminosity can be achieved by increasing the number of particles and the number of bunches in each beam, and with high bunch repetition rate. The high luminosity can also be achieved by decreasing the transverse beam size $\sigma_x$ and $\sigma_y$ which is necessary to increases the probability of collisions.

It is typical in beam physics to describe the dimensions of the beam by the six-dimensional phase space volume occupied by the beam particles known as the beam emittance $\varepsilon$ [114]. There are three two-dimensional emittances: one longitudinal (the direction of the beam propagation) and two transverse (perpendicular to the direction of the beam propagation). Each particle is described by the 6D phase space coordinates: $(x, p_x, y, p_y, z, p_z)$. In circular accelerator, describing the motion of the particles is usually done in a coordinate system moving with an ideal particle called the reference particle, which has the ideal energy $E_0$ and follows an ideal circular orbit of radius $\rho$. In this system shown in Fig 5.1, the arc length $s$ coordinate is pointing into the direction of the longitudinal motion and that is moving with the particle around the ring. The transverse coordinates describe the displacement of a particle with respect to the ideal orbit in the horizontal $x$ coordinate and the vertical coordinate $y$. The transverse momenta are replaced by the slope of the trajectories $p_x \rightarrow x' = p_x/p_0$ and $p_y \rightarrow y' = p_y/p_0$.

![Figure 5.1: Ideal circular orbit of a reference particle and the coordinate system that rotates with it.](image)
Because in reality the particle trajectory deviates from the ideal circular orbit, its longitudinal momentum is described by the momentum deviation from the ideal momentum \( \Delta p = p - p_0 \) or the relative momentum deviation \( \Delta p/p \) (known as the momentum spread). The relative energy deviation from the ideal energy \( \delta = \Delta E/E_0 = (E - E_0)/E_0 \) is also used sometimes to describe the particle coordinates and we get the coordinates \((x, x', y, y', s, \delta)\).

In the periodic lattice of a storage ring, a single particle moves under the influence of the quadrupole and dipole fields and its dynamics is determined by a differential equation known as the Hill’s equation \[114\]. Using \( u \) to denote \( x \) or \( y \), the transverse coordinate of the particle with respect to the design orbit, Hill’s equation when \( \delta = 0 \) is written as

\[
\begin{align*}
    u''(s) + K(s)u(s) &= 0, \\
    \psi(s) &= \int_0^s \frac{ds}{\beta(s)},
\end{align*}
\]  

(5.2)

where the derivative is taken with respect to the orbit coordinate \( s \). The solution of this equation is similar to that of the simple harmonic motion, but with an amplitude and phase that depend on the position \( (s) \) in the ring. The general solution and its first derivatives are \[114\]

\[
\begin{align*}
    u(s) &= \sqrt{\varepsilon} \cdot \sqrt{\beta(s)} \cdot \cos(\psi(s) + \delta) \\
    u'(s) &= \frac{-\sqrt{\varepsilon}}{\sqrt{\beta(s)}} \cdot \sin(\psi(s) + \delta) + \alpha(s) \cdot \cos(\psi(s) + \delta),
\end{align*}
\]  

(5.3)

where \( \varepsilon \) and \( \delta \) are constants of the particular trajectory, and \( \alpha(s) = -\frac{1}{2} \beta'(s) \). \( \beta(s) \) is a periodic function known as the beta function and it depends on the overall focusing properties of the lattice and follows the periodicity of the storage ring. Substituting the solution \( u(s) \) from Eq. 5.3 into the Hill equation Eq. 5.2 and rearranging, we get the phase advance of the oscillation

\[
\psi(s) = \int_0^s \frac{ds}{\beta(s)}.
\]
From the solution \( u(s) \) in Eq. 5.3

\[
\cos(\psi(s) + \delta) = \frac{u(s)}{\sqrt{\varepsilon} \sqrt{\beta(s)}},
\]

which is used in \( u'(s) \) to give the following expression of the integration constant \( \varepsilon \)

\[
\varepsilon = \gamma(s) u^2(s) + 2\alpha(s) u(s) u'(s) + \beta(s) u'^2(s),
\]

(5.4)

where

\[
\gamma(s) = \frac{1 + \alpha(s)^2}{\beta(s)}.
\]

Thus, the solution to Hill’s equation given by Eq. 5.4 represents a particle tracing out an ellipse in the transverse phase space \( u - u' \). According to Liouville’s theorem, the area of the ellipse \( \pi \varepsilon \) is a constant of motion, known as Courant-Snyder invariant, as long as conservative forces are considered. It also shows that the shape and the orientation of the phase space ellipse depends on the Twiss parameters \( \beta, \alpha, \) and \( \gamma \).

The maximum position \( u \) can be obtained from Eq. 5.3 as \( u_{\max} = \sqrt{\varepsilon \beta} \), and the maximum angle \( u' \) can be derived by replacing \( \alpha(s) \) and \( \gamma(s) \) in Eq. 5.4 by their expressions to get \( u'_{\max} = \sqrt{\varepsilon \gamma} \) (see Fig. 5.2). Hence, the maximum amplitude that a single particle trajectory can reach at a given position in the ring is determined by the beta function \( \beta(s) \).

In the case of a charged particle beam, the phase space area occupied by the ensemble of many particles \( N \) is characterized by the beam emittance \( \varepsilon \). The beam emittance is an intrinsic beam parameter that cannot change by the focusing properties of a storage ring. In a ring, the particle with the highest emittance \( \varepsilon_{\max} \) will trace the largest ellipse while the other particles in the beam will trace smaller ellipses of the same shape inside the largest ellipse. Hence, the whole beam can be represented by the largest ellipse and the beam size is defined by \( u_{\max} \). Since the beta function determines \( u_{\max} \), it is considered as the envelope of
Figure 5.2: Single particle trajectory tracing an ellipse where its maximum position and angle are determined by the beta function $\beta(s)$.

All the particles trajectories at a given position $s$, and thus it is a very important parameter of storage rings. When beta has a small value, the beam’s transverse size will be small, and the corresponding phase advance will be large.

In general, the charged particle beam follows a Gaussian distribution and the beam rms emittance is used to describe the phase space area in this case. The resulting phase space ellipse contains one $\sigma$ of particles. Then, the rms beam size in a ring is $\sigma(s) = \sqrt{\epsilon_{\text{rms}} \beta(s)}$. In this statistical description, the beam rms emittance $\varepsilon_{u,\text{rms}}$ can be written as

$$
\varepsilon_{u,\text{rms}} = \sqrt{\langle u^2 \rangle \langle u'^2 \rangle - \langle uu' \rangle^2},
$$

(5.5)
where the second central moments are

\begin{align*}
    \langle u^2 \rangle &= \frac{\sum u^2}{N} - \left( \frac{\sum u}{N} \right)^2 \\
    \langle u'^2 \rangle &= \frac{\sum u'^2}{N} - \left( \frac{\sum u'}{N} \right)^2 \\
    \langle uu' \rangle &= \frac{\sum uu'}{N} - \frac{\sum u \sum u'}{N^2}.
\end{align*}

The next generation nuclear physics colliders require high-intensity, high-energy hadron beams for the advances of nuclear physics, such as the electron-ion collider (EIC) that is being constructed at Brookhaven National Laboratory (BNL). This site was selected by the U.S. Department of Energy to advance nuclear science in early 2020. The most efficient way to achieve high luminosity is to cool the emittance of the ion beam. The emittance growth during the multistage acceleration is due to Coulomb collisions between the beam particles and due to some instabilities, which may result in discarding the beam. Moreover, the lack of emittance control at the collision energy leads colliders to use the beams as they are. Therefore, it is more desirable to cool high-energy hadron beams at the collision energy. Equation 5.1 indicates that this can be achieved by decreasing the transverse beam size

\[ \sigma_{x,y} = \sqrt{\varepsilon_{x,y} \beta^*_{x,y}}, \]

where \( \beta^*_{x,y} \) are the transverse beta functions at the collision (interaction) point and is expressed as \[114\]

\[ \beta (s - s_0) = \beta^* + \frac{(s - s_0)^2}{\beta^*}. \]

Different techniques were developed to decrease the phase space volume of the beam that are known as beam cooling methods. To cool heavy ion beams, there are three known methods [7]: a method based on using a co-propagating electron beam is known as electron cooling; stochastic cooling method depends on using the RF feedback system; laser cooling
which is based on using laser to cool ion beams. According to [7], it is more efficient to use electron cooling and laser cooling when cooling high-intensity low-temperature beams while cooling low-intensity high-temperature beams is better achieved by stochastic cooling. In addition, a method that incorporates both the electron and stochastic cooling and that can cool high-intensity high-temperature beams is regarded as strong electron cooling such as the coherent electron cooling discussed in [115] and its first practical scheme appeared in [14].

5.1.1 **Bunched Electron Cooling**

The electron cooling technique was proposed by Budker in 1967 to reduce the 6D phase-space volume (emittance) of ion beams in circular accelerators [116]. In this method, a cold electron beam co-propagates with a hot ion beam in a small straight section of the accelerator and the momentum is transferred from the ions to the electrons through Coulomb collisions. Repeating this brief interaction through multiple turns results in reducing the beam emittance and cooling the beam. The first electron cooling experiments were carried out in 1974 at the proton storage ring NAP-M [117].

The process of the electron cooling is analogous to the relaxation of a plasma consisting of a hot and a cold components with the exception that the electron component can be refreshed [118]. An electron beam temperature $T_e$ that is lower than the ion beam temperature $T_i$ is normally achieved when the electron velocity $v_e$ is equal to the ion velocity $v_i$. This is a result of the mass ratio between the electron and the ion in the following relation

$$T_e \approx \frac{m_e}{M} T_i,$$

where $m_e$ and $M$ are the electron and the ion masses, respectively.
If we assume that the density of the ion beam is low enough, we can ignore the mutual Coulomb interaction between the ions. For an ion moving through an electron beam, it experiences a force that is a sum of the total Coulomb interactions with all surrounding electrons. During these interactions, the momentum is transferred from the ion to the surrounding electrons with an average rate known as the friction force. An analytical expression of the friction force is usually derived in plasma physics textbooks such as [119]. Assuming that the electrons are uniformly distributed within some volume and that their velocities follow a Gaussian distribution in the electron beam reference frame, the standard analytical formula of the friction force with zero magnetic field is given by [120]

\[
F = -\frac{4\pi Z^2 e^4 n_e}{(4\pi \epsilon_0)^2 m_e} \int L(u) \frac{u}{u^3} f (v_e) \, d^3 v_e. \tag{5.7}
\]

In Eq. 5.7, \(Z\) is the ion charge number, \(e\) is the elementary charge, \(n_e\) is the electrons density, \(u = v_i - v_e\) is the electron-ion relative velocity, \(L(u)\) is Coulomb logarithm, and \(f (v_e)\) is the electrons velocity distribution function. Here, only binary collisions are presumed to contribute to the momentum transfer and to the friction force. In general, the ion beam velocity distribution has a small velocity spread, and thus the friction force is considered to be directly proportional to the ion velocity.

According to [12], the classical friction force formula is based on strong assumptions that can give inaccurate estimate of the friction force and thus the cooling time. One of these assumptions is to consider that the interaction time is infinitely long while in practice it is short and close to the plasma period in electron cooling applications. It is also assumed that the interaction domain is very large, and that the electron’s trajectory is almost a straight line. In addition, the classical formula does not consider external electromagnetic fields effects while a longitudinal magnetic field is usually present in the cooling section.
The external longitudinal magnetic field is typically applied to guide the electron beam through the cooler. If the value of this magnetic field is zero or if it is weak, the cooling is regarded as non-magnetized cooling [121]. On the other hand, the cooling is referred to as magnetized when a strong magnetic field is present. The strong longitudinal magnetic field limits the electrons transverse motion as they perform many cyclotron (Larmor) rotations within the time it takes for collisions to occur, and thus only the electrons' longitudinal velocity spread contributes to the friction force.

An analytical friction force formula for the magnetized cooling can be derived only for the case of infinitely strong magnetic field where the integral is reduced to a 1D integral. This formula was derived by [118] as follow

\[ F_{ad} = -\frac{2\pi Z^2 e^4 n_e}{(4\pi\epsilon_0)^2 m_e} \frac{\partial}{\partial \mathbf{v}} \int \frac{v^2_{\perp}}{u^2_{ad}} \mathbf{L}^{ad} f(v_e) \, dv_e, \]  

(5.8)

where the “\(ad\)” refers to the adiabatic collisions with respect to Larmor rotations and \(\mathbf{u}_{ad} = \mathbf{v}_i - \mathbf{v}_e\parallel\).

In the electron-ion Coulomb collision, the transferred momentum is described by the impact parameters in Coulomb logarithm as \(L(u) = \ln(\rho_{max}/\rho_{min})\). In practice, all possible impact parameters should be considered by the Logarithm, but the limits \(\rho_{max}\) and \(\rho_{min}\) were set as cutoffs to avoid the singularities in the logarithm. The minimum transferred momentum from the ion to the electron is determined through the maximum impact parameter \(\rho_{max}\) after which the effects of collisions are negligible. On the other hand, the minimum impact parameter \(\rho_{min}\) is obtained from the classical head-on collision where the maximum momentum is transferred. The impact parameter limits are given by

\[
\begin{align*}
\rho_{max} &= \min \left\{ d, u\tau, \lambda_D \right\} \\
\rho_{min} &= \max \left\{ \rho_L, \frac{Zr_e e^2}{u^2} \right\},
\end{align*}
\]  

(5.9)
where $d$ is the transverse electron beam size, $\tau_f$ is the time of flight in the cooling section length, $\lambda_D$ is the Debye shielding length, and $r_e$ is the classical electron radius [118]. If the cooling is magnetized, $L$ is replaced by $L^{ad}$ and $u$ is replaced by $u^{ad}$ where Larmor radius $\rho_L = m_e \Delta V_{e\perp} / (eB)$. In the case that Coulomb logarithm varies slowly with the relative velocity, it can be considered as a constant and be taken out of the integral which in turn simplifies the integral equation of the friction force.

In some typical cooling applications, simple asymptotic expressions of the friction force are usually used. Otherwise, an accurate evaluation of the friction force is accomplished numerically, especially for the magnetized cooling. If $v_i \gg v_e$, the electron velocity distribution function $f(v_e)$ can be approximated by a Delta function, and thus the friction force will be independent of $v_e$. In most practical applications, $f(v_e)$ follows an anisotropic Maxwellian and typically the longitudinal velocity spread $\Delta V_{e\parallel}$ is much smaller than the transverse velocity spread $\Delta V_{e\perp}$ in the beam frame. In this case, $f(v_e)$ takes the form

$$f(v_e) = \left[ \frac{3}{(2\pi)^2 \Delta V_{e\perp} \Delta V_{e\parallel}} \exp\left( \frac{v_{e\perp}^2}{2 \Delta V_{e\parallel}^2} + \frac{v_{e\parallel}^2}{2 \Delta V_{e\parallel}^2} \right) \right]^{-1}.$$  

Considering the non-magnetized cooling, in the limit $v_i < \Delta V_{e\perp}$, the components of the friction force can be written as [120]

$$F_{\parallel} = -\pi \sqrt{2 \pi} \frac{Z^2 e^4 n_e L (\Delta V_{e\perp}) v_{i\parallel}}{(4\pi \epsilon_0)^2 m_e \Delta V_{e\parallel}}$$

$$F_{\perp} = -\frac{4 \pi Z^2 e^4 n_e}{(4\pi \epsilon_0)^2 m_e \Delta V_{e\perp}^2} \begin{cases} L(v_{i\parallel}) - \frac{v_{i\parallel}}{\Delta V_{e\perp}} \sqrt{\frac{\pi}{2}} L(\Delta V_{e\perp}) ; & v_{i\parallel} > \Delta V_{e\parallel} \\ \frac{v_{i\parallel}}{\Delta V_{e\parallel}} \sqrt{\frac{2}{\pi}} L(\Delta V_{e\parallel}) ; & v_{i\parallel} < \Delta V_{e\parallel}. \end{cases}$$  

(5.10)
For an infinitely strong magnetic field and $v_i \gg \Delta v_{e\parallel}$, the friction force is given by [120]

$$F_{ad\perp} = -\frac{2\pi Z^2 e^4 n_e L_{ad}(v_i)}{(4\pi \varepsilon_0)^2 m_e} \frac{(v_{i\perp}^2 - 2v_{i\parallel}^2)}{v_i^3} v_{i\perp}$$

$$F_{ad\parallel} = -\frac{2\pi Z^2 e^4 n_e L_{ad}(v_i)}{(4\pi \varepsilon_0)^2 m_e} \frac{3v_{i\perp} v_{i\parallel}}{v_i^3} v_{i\parallel}$$

(5.11)

A practical empirical formula of the friction force with finite magnetic field is provided in [122]

$$F = -\frac{4\pi Z^2 e^4 n_e}{(4\pi \varepsilon_0)^2 m_e} \ln \left( \frac{\rho_{\max} + \rho_{\min} + \rho_L}{\rho_{\min} + \rho_L} \right) \frac{v_i}{\left( \sqrt{v_i^2 + v_{effe}^2} \right)^3},$$

(5.12)

where $v_{effe} = \sqrt{v_{e\parallel}^2 + \Delta v_{e\perp}^2}$ is the effective electron velocity and $\Delta v_{e\perp}$ is the velocity component that is a result of the transverse magnetic and electric fields.

Once the friction force is calculated, the cooling rate $\tau^{-1}$ can be estimated as follow

$$\tau^{-1} = \left| \frac{\mathbf{F}}{M v_i} \right| = \left| \frac{1}{v_i} \frac{d\mathbf{v}_i}{dt} \right|,$$

(5.13)

and the cooling time $\tau$ is the inverse of the cooling rate. The cooling rate can be also defined from the exponential change of the emittance during cooling as

$$\frac{1}{\tau} = \frac{1}{\varepsilon} \frac{d\varepsilon}{dt}.$$ 

Moving to the lab frame, Lorentz transformation gives a factor of $\gamma$ in the electron density $n_{elab} = \gamma n_e$ and another factor of $\gamma$ for the cooling time from the time dilation $\tau_{lab} = \gamma \tau$. Then, the cooling time is divided by $\eta$, the ratio of the length of the cooling section to the circumference of the ring. This gives

$$\tau_{lab} = \frac{\gamma^2}{\eta} \tau.$$ 

(5.14)
This indicates that cooling at high energies becomes difficult as the cooling times become very large. At low energy, typical DC electron cooling is widely applied (magnetized and non-magnetized) where the discrepancy between asymptotic formulas and numerical simulations is considered acceptable (about a factor of 2) [123, 13]. However, the asymptotic expressions significantly overestimate the friction force at high energy [123].

For non-magnetized cooling, the imposed minimum impact parameter \( \rho_{\text{min}} \) in the standard friction force formula 5.7 suggests that all collisions due to impact parameters lower than \( \rho_{\text{min}} \) can be neglected [13]. These collisions are known as close encounters or strong collisions because they greatly change the particles’ velocities. Although strong collisions are rare, it has been shown by [13] that they significantly change the friction force. In their example, numerical simulations indicated that the friction force is 8.3% lower than predicted by the standard formula. They also show that the magnitude of the friction force is considerably overestimated when finite time effects are ignored. In addition, [12] has demonstrated that any analytical formula does not give an accurate description of the full dynamics due to ignoring close encounters.

Most of the available numerical simulation tools that estimate the cooling time rely on evaluating one of the traditional formulas of the friction force. Some of these tools combine that with parameters that can be tuned depending on the particular application and may provide a good estimate of the cooling time especially at low energy (e.g. BETACOOL [10] and JSPEC [11]). However, the approximations and limitations of the friction force in describing the dynamics of the electron cooling signals the importance of high precision, direct collisional numerical simulations that are based on a minimum set of assumptions. Our collisional numerical method PHAD is designed to tackle such a difficult problem provided that it also can include any external electromagnetic fields. The importance of our tool becomes apparent as the frontier of particle physics requires high-brightness, high energy ion beams such as the new electron-ion collider that is being constructed at BNL.
Cooling high energy ion beams requires electron beams accelerated to high energies. In conventional electron cooling, the DC electron beam is accelerated by an electrostatic high voltage which can accelerate the electrons to a kinetic energy $< 10 \text{ MeV}$. Higher energies are difficult to achieve due to the technical limitations of the high power and high voltage. Thus, acceleration with the radio-frequency (RF) fields in an energy-recovering linac system is a promising approach to accelerate electrons to tens MeV and cool hadrons at high energies [124]. However, using the RF results in a bunched electron beam and such a cooling method differ from the conventional electron cooling. To explore the bunched electron cooling technique, a collaboration team from Jefferson Lab (JLab) in USA and the Institute of Modern Physics (IMP) in China conducted experiments to explore low energy cooling with bunched electron beams [8]. Indeed, their experiment in 2016 was the first to show bunched electron cooling [125, 124].

Since PHAD method is different from the methods used by other cooling simulation tools, we needed to benchmark it with experimental data and not with other available cooling codes. The IMP experiments are suitable for benchmarking PHAD because they are performed at low energy and thus have a short cooling time. Through our collaboration with JLab and BNL, we were able to get the required information about the IMP experiments and their results to benchmark PHAD. In particular, we performed simulations related to the experiments conducted in 2017 that was reported in [126]. Later experiments were performed in 2018 and 2019 to answer more questions with improved data quality [8].

5.1.1.1 IMP Bunched Electron Cooling Experiments

The IMP experiments were conducted at the CSRm storage ring that has a racetrack shape with a circumference of 161 m [127]. The layout of the ring includes four identical arc
sections, where the cooler and the RF cavity are located in dispersion-free sections [127]. To cool the ion beam, a bunched electron beam is provided through the RF cavity. While the energy in these experiments was much lower than the high energy of the electron-ion collider to which the bunched cooling is going to be applied, the purpose of these experiments was to demonstrate and to explore the bunched electron cooling process. The cooling was achieved for both coasting and bunched ion beams where the decrease of the momentum spread, and the bunch length of the ion beam was observed. Using bunched ion beams allows to examine the dependence of the cooling time on the electron beam parameters such as pulse length and peak current.

In the 2017 experiments [126], the ion beam consisted of $\text{C}^{6+}$ ions with a kinetic energy of $7\,\text{MeV/nucleon}$ and longitudinal cooling of both coasting and bunched ion beams was achieved in less than 2 s. For the cooling of bunched ion beam, an RF voltage of $1.2\,\text{kV}$ was applied. Also, a longitudinal magnetic field of $0.1\,\text{T}$ was present in the cooling section. The number of ions was $1.3 \times 10^8$ with an RMS bunch length of $\approx 135\,\text{ns}$. The electron beam had a peak current of $65\,\text{mA}$ in a pulse length of $1\,\mu\text{m}$ and beam radius of $\approx 1.25\,\text{cm}$. Due to lack of measurements, cooling of the transverse emittance was not studied until later experiments where the measurements were improved.

Early results of the 2018 experiments suggested that cooling is faster for higher peak currents of the electron beam when the pulse length is fixed [128]. Later analysis of the experimental results from 2018 and 2019 showed that the cooling rate increases for longer pulse lengths of the electron beam when the peak current is fixed [8].
5.1.1.2 PHAD Simulations of Bunched Electron Cooling

We performed electron cooling simulations in order to benchmark PHAD with the IMP experiments that were conducted in 2017. Specifically, we considered bunched electron cooling of bunched ion beams. PHAD was used to model the interactions between the electrons and the ions in the presence of the external longitudinal magnetic field in the cooling section. Since PHAD is an add-on to COSY INFINITY, the nonlinear effect of the rest of the CSRm ring on the ion beam was modeled by COSY INFINITY transfer maps. However, the dependent variable of the transfer maps is the arc-length \( s \) while it is the time \( t \) for PHAD. Thus, we have applied the appropriate transformation of the dependent variable between \( t \) and \( s \) as the ion beam propagates through the cooling section and the rest of the ring.

Due to technical limitations, we cannot simulate the full beams with their large number of particles. Hence, we considered part of the beams with a lower number of particles. In our initial plans, we tried to simulate an ion beam of \( \leq 1000 \) ions, and electron beams of a few \( 10^5 \) electrons. During these simulations, we have observed two issues. First, the bunch length of the ion beam increases significantly after the beam propagates through the IMP ring elements. To deal with that, we have increased the RF voltage such that the bunch length remains relatively constant when the ions pass through the ring. The second issue was the long time it takes the simulations to complete one turn which was in order of days. This long computational time was expected because PHAD is a collisional method that considers all individual particles, and because modeling the electron-ion interactions requires an additional care. However, we have decided to decrease the number of electrons by using macro-electrons to speed up the simulations.

For the benchmarking, we used the ratio of the number of electrons in the electron pulse to the number of ions in the ion bunch from the IMP experiment to set up the number of
particles in our simulations. We performed two sets of simulations with this ratio which is \( \approx 3121 \). With respect to the ion beam bunch length, the first simulation Sim.1 includes a shorter electron pulse length as shown in Fig. 5.3a. A longer electron pulse length (and thus a lower current) of the electron beam was considered in Sim.2, the second simulation as shown in Fig. 5.3b. In the next set of simulations, we disregarded the ratio in order to study the effect of the electron beam current and the pulse length on the cooling time. The next two simulations considered a fixed electron beam current and compared the cooling times of a short electron bunch length in Sim.3 to that of a long electron bunch length in Sim.4. The last simulation Sim.5 considered the same electron bunch length as in Sim.4, but with a lower current and we compared their cooling times. The initial spatial configuration of the simulations Sim.3, Sim.4, and Sim.5 are illustrated in Fig. 5.4

![Figure 5.3](image)

(a) Short pulse.  
(b) Long pulse.

Figure 5.3: The initial spatial distributions of the bunched electron cooling simulations: (a) Sim.1 of short pulse and (b) Sim.2 of long pulse. The blue points represent the protons and the orange points are the electrons.

The common parameters between all the simulations are listed in Table 5.1 along with PHAD parameters, while the each simulation specific parameters are presented in Table 5.2. In all simulations in this section, the spatial distribution of the ion beam was a Gaussian while that of the electron beam was uniform.
After each turn, the simulations gives the 6D coordinates of the ions which we use to calculate the eigen emittances (see Appendix B) and plot their change with time. The main reason for using the eigen emittances and not the projected emittances is that they do not decrease with time unless there is a dissipative force such as the cooling force. Thus, their decrease in our simulations is evident of cooling. Since the simulated time is relatively small comparing to the cooling time, we fit the resulted longitudinal eigen emittance with a linear fit to evaluate $d\varepsilon/dt$ and estimate the cooling time. We will first present the simulations
Table 5.1: The parameters used in the bunched electron cooling simulations to benchmark PHAD with the IMP Experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ions RMS $\varepsilon_x/\varepsilon_y$ (mm mrad)</td>
<td>0.15/0.1</td>
</tr>
<tr>
<td>Ions $\Delta p/p$</td>
<td>$2.5 \times 10^{-4}$</td>
</tr>
<tr>
<td>Electrons radius (m)</td>
<td>$7.5 \times 10^{-3}$</td>
</tr>
<tr>
<td>Electrons $T$ [long./trans.] (eV)</td>
<td>$10^{-5}/0.01$</td>
</tr>
<tr>
<td>Length of the cooler (m)</td>
<td>3.4</td>
</tr>
<tr>
<td>Beta function in the cooler (m)</td>
<td>10</td>
</tr>
<tr>
<td>Magnetic field $B_z$ (T)</td>
<td>0.1</td>
</tr>
<tr>
<td>PHAD timestep (m)</td>
<td>$\sim 0.28$</td>
</tr>
<tr>
<td>Number of PHAD timesteps</td>
<td>100</td>
</tr>
<tr>
<td>$q$</td>
<td>60</td>
</tr>
<tr>
<td>FMM order</td>
<td>6</td>
</tr>
<tr>
<td>Accuracy</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>Simò maximum order</td>
<td>10</td>
</tr>
<tr>
<td>Type of time bins</td>
<td>equal-widths</td>
</tr>
<tr>
<td>Number of time bins</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5.2: The parameters of each simulation of the bunched electron cooling.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sim.1</th>
<th>Sim.2</th>
<th>Sim.3</th>
<th>Sim.4</th>
<th>Sim.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ions</td>
<td>1000</td>
<td>1000</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Ions RMS $\sigma_z$ (m)</td>
<td>0.22</td>
<td>0.22</td>
<td>$5.5 \times 10^{-3}$</td>
<td>$5.5 \times 10^{-3}$</td>
<td>$5.5 \times 10^{-3}$</td>
</tr>
<tr>
<td>Number of electrons</td>
<td>15604</td>
<td>15604</td>
<td>15604</td>
<td>40124</td>
<td>15604</td>
</tr>
<tr>
<td>Electrons charge (e)</td>
<td>-200</td>
<td>-200</td>
<td>-900</td>
<td>-3500</td>
<td>-900</td>
</tr>
<tr>
<td>Electrons length (m)</td>
<td>0.22</td>
<td>2.2</td>
<td>$5.5 \times 10^{-3}$</td>
<td>$5.5 \times 10^{-2}$</td>
<td>$5.5 \times 10^{-2}$</td>
</tr>
<tr>
<td>Electron current (mA)</td>
<td>$83.3 \times 10^{-3}$</td>
<td>$8.33 \times 10^{-3}$</td>
<td>15</td>
<td>15</td>
<td>1.5</td>
</tr>
<tr>
<td>RF voltage (kV)</td>
<td>80</td>
<td>80</td>
<td>8800</td>
<td>8800</td>
<td>8800</td>
</tr>
</tbody>
</table>

results along with the longitudinal cooling, and then comment on the transverse emittances later in the section.

The simulation results of Sim.1 and Sim.2 are shown in Fig. 5.5 and Fig. 5.6, respectively. In both figures, the longitudinal emittance decreases with time which indicates cooling while the transverse emittance shows an oscillatory behavior that is mainly due to the longitudinal magnetic field. The estimated cooling times from the simulations and from the analytical formulas are shown in Table 5.3. Both cooling times of Sim.1 and Sim.2 are accurately
consistent with the experiments. The cooling time of Sim.2 with the long pulse is slightly higher because the longer pulse gave a lower current which cooled in a slower rate. As for the estimation from the analytical formulas, some of the resulted cooling times were inaccurate, and a few were relatively close which may be due to the low energy to which the analytical estimates are better.

Figure 5.5: Eigen emittances change with time that resulted from the electron cooling simulation Sim.1: (a) longitudinal with a linear fit (red) and (b) transverse.

Figure 5.6: Eigen emittances change with time that resulted from the electron cooling simulation Sim.2: (a) longitudinal with a linear fit (red) and (b) transverse.

To study the effect of varying the electron beam pulse length at a fixed current, the simulation results of Sim.3 and Sim.4 are shown in Fig. 5.7 and Fig. 5.8, respectively. The estimated cooling times from the simulations and from the analytical formulas are shown in
Table 5.3: Longitudinal cooling time of Sim.1 and Sim.2 from the analytical formulas, the experiment, and the simulations.

<table>
<thead>
<tr>
<th></th>
<th>Sim.1</th>
<th>Sim.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation 5.10</td>
<td>9.6 s</td>
<td>94 s</td>
</tr>
<tr>
<td>Equation 5.11</td>
<td>0.62 s</td>
<td>6.1 s</td>
</tr>
<tr>
<td>Equation 5.12</td>
<td>0.34 s</td>
<td>3 s</td>
</tr>
<tr>
<td>Simulation</td>
<td>3 s</td>
<td>6.4 s</td>
</tr>
<tr>
<td>Experiment</td>
<td>≤ 2 s</td>
<td>≤ 2 s</td>
</tr>
</tbody>
</table>

Table 5.4. The table shows that the estimated cooling time from the simulations is slightly longer for the shorter pulse in Sim.3 which agrees with the experiments. Since most of the parameters of the simulations are similar except for the number of electrons and the electron pulse lengths, the main parameter that changes in the analytical formulas is the density of the electron beam $n_e$. It occurred that $n_e$ is the same for both Sim.3 and Sim.4, and thus their analytical estimations of the cooling times were the same. Yet, most of the analytical estimations are considerably different.

In the case of varying the electron beam current at a fixed pulse length, we compare the simulation results of Sim.4 in Fig. 5.8 and Sim.5 in Fig. 5.9. The estimated cooling times from the simulations in Table 5.4 indicate that the cooling time is longer for the lower current from Sim.5, which is also in agreement with the experiment.

![Figure 5.7: Eigen emittances change with time that resulted from the electron cooling simulation Sim.3: (a) longitudinal with a linear fit (red) and (b) transverse.](image-url)
Figure 5.8: Eigen emittances change with time that resulted from the electron cooling simulation Sim.4: (a) longitudinal with a linear fit (red) and (b) transverse.

(a) Longitudinal eigen emittance.  
(b) Transverse eigen emittances.

Figure 5.9: Eigen emittances change with time that resulted from the electron cooling simulation Sim.5: (a) longitudinal with a linear fit (red) and (b) transverse.

(a) Longitudinal eigen emittance.  
(b) Transverse eigen emittances.

Table 5.4: Longitudinal cooling time of Sim.3, Sim.4, and Sim.5 from the simulations and the analytical formulas.

<table>
<thead>
<tr>
<th></th>
<th>Sim.3</th>
<th>Sim.4</th>
<th>Sim.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation 5.10</td>
<td>0.21 s</td>
<td>0.21 s</td>
<td>1.8 s</td>
</tr>
<tr>
<td>Equation 5.11</td>
<td>8.6 ms</td>
<td>8.6 ms</td>
<td>0.051 s</td>
</tr>
<tr>
<td>Equation 5.12</td>
<td>3.9 ms</td>
<td>3.9 ms</td>
<td>0.025 s</td>
</tr>
<tr>
<td>Simulation</td>
<td>0.068 s</td>
<td>0.056 s</td>
<td>0.14 s</td>
</tr>
</tbody>
</table>

To further show that the reduction of the longitudinal eigen emittance in our simulations is due to electron cooling, we propagated the ion beams through the lattice of the ring using transfer maps without the cooling section. In all simulations, the longitudinal eigen
emittance did not decrease with time when the ion beam was propagated through the ring without the cooler. We provide here one example from Sim.1 in Fig. 5.10 which shows that the reduction of the longitudinal eigen emittance is indeed due to electron cooling.

![Graph](image)

Figure 5.10: Comparison between the resulted longitudinal eigen emittance of the ion beam propagated through the ring elements including the cooling section and when the cooler is not included. Clearly, the reduction of the eigen emittance when including the cooler is due to the electron cooling.

Our results in this section are the first microscopic simulations of electron cooling with a very accurate prediction of cooling times. Unlike other electron cooling codes, our code does not depend on tuning or fitting some code parameters and only considers the dynamics of the system.

**Transverse Emittances in our Simulations.** Our simulations were designed to predict the longitudinal electron cooling. We have not observed any cooling of the transverse emittances as shown in the figures presented earlier in this section. While the eigen emittances change with time should be invariant under linear transformations, their behavior under nonlinear transformations is not clear. Here, we have considered nonlinearities through both the ring and the cooler and observed a nonlinear behavior of the eigen emittances.
Specifically, the transverse eigen emittances exhibited sinusoidal oscillations as shown in Fig. 5.5b, Fig. 5.6b, Fig. 5.7b, Fig. 5.8b, and Fig. 5.9b.

The 4D emittance $\epsilon_{4d} = \sqrt{\epsilon_1 \epsilon_2}$ is invariant under linear forces and in the presence of a solenoidal magnetic field. However, the nonlinearities considered here have affected the behavior of the 4D emittance depending on the strength of the nonlinear electron-ion interaction forces. For Sim.1 and Sim.2, the nonlinear electron-ion interaction forces where relatively weak due to the low currents of the electron beams. Thus, the 4D emittances were relatively invariants as shown in Fig. 5.11. When the nonlinear electron-ion interaction forces were strong for the case of the high currents of the electron beams in Sim.3 and Sim.4, the 4D emittances were not invariants and increased for both simulations. This is illustrated in Fig. 5.12a and Fig. 5.12b where the 4D emittance of Sim.4 showed a higher increase due to the higher charge of the electron beam compared to that of Sim.3. Since the electron beam of Sim.5 has a lower current than that of Sim.3 and Sim.4, the 4D emittance was relatively invariant as shown in Fig. 5.12c.

We have compared the resulted transverse eigen emittances of our cooling simulations with the ones resulted from only the ring transfer maps without including the cooler in which we found that $\epsilon_1$ and $\epsilon_2$ did not change with time. As an example, this behavior is shown
Figure 5.12: The 4D emittance of the electron cooling simulations: (a) Sim.3, (b) Sim.4, and (c) Sim.5. The 4D emittance was not invariant when the nonlinear electron-ion interaction forces were strong in Sim.3 and Sim.4, but was relatively invariant for Sim.5 when the nonlinear forces were relatively weak.

For the simulation Sim.1 with and without the cooling section in Fig. 5.13. The solenoid’s longitudinal magnetic field is known to cause oscillations of the transverse projected emittances and to introduce coupling in the $x$-$y$ plane. In the linear case, these oscillations and the effect of coupling do not appear in the eigen emittances, and hence the oscillations of $\epsilon_1$ and $\epsilon_2$ are mainly due to the nonlinear contribution of the longitudinal magnetic field to the dynamics in the cooling section.

Between all the five electron cooling simulations, only the transverse eigen emittances from Sim.4 presented in Fig. 5.8b showed an increase of $\epsilon_1$ while the oscillations of both $\epsilon_1$ and $\epsilon_2$ decreased with time. The main characteristic of Sim.4 that is different from the rest
of the simulations is the very high charge due to the high current in a short pulse. Since it is difficult to interpret the eigen emittance when nonlinearities are included, we look at the projected transverse emittances illustrated in Fig. 5.14 in which the emittances oscillate and grow with time. We have determined earlier that the oscillations are due to the solenoid field, so now we need to consider the growth of the transverse $\varepsilon_{x,y}$.

We have seen from Hill’s equation 5.2 and its solution in Eq. 5.3 that the transverse motion of a single particle through a periodic lattice is oscillatory around a closed orbit. Those equations describe a particle of an ideal momentum $p_0$ where the momentum spread $\delta = (p - p_0)/p_0 = \Delta p/p_0$ is zero. A particle with $\delta$ oscillates about a closed orbit different from that of the reference particle. For a small $\delta$, the Hill’s can be written as

$$u''(s) + K(s)u(s) = \frac{\delta}{\rho(s)},$$

(5.15)
Figure 5.14: The transverse projected emittances from the electron cooling simulation Sim.4 showing oscillations due to the solenoid field and growth to the strong electron-ion interactions.

where $\rho(s)$ is the radius due to bending magnets (dipoles fields) at $s$. The solution to Eq. 5.15 can be divided into a homogeneous solution $u_\beta(s)$ due to $\delta = 0$ and an inhomogeneous solution $u_\delta(s)$ due to $\delta \neq 0$ as

$$u(s) = u_\beta(s) + u_\delta(s).$$

(5.16)

The closed orbit for the off-momentum particle is displaced by $u_\delta(s)$ from the closed orbit of the reference particle, and $u_\beta(s)$ is the betatron oscillations around this new closed orbit. A general expression of $u_\delta(s)$ is given by

$$u_\delta(s) = D(s)\delta,$$

(5.17)

where $D(s)$ is called the dispersion function which is a characteristic of the machine like the beta function.
For a distribution of particles, the beam size becomes

\[ \langle u^2 \rangle = \langle u_{\beta}^2 \rangle + 2D \langle u_{\beta} \rangle \langle \delta \rangle + D^2 \langle \delta^2 \rangle. \] (5.18)

This equation indicates that the beam size of a particle distribution with a momentum spread \( \delta \) tends to increase due to dispersion unless there is cooling. As a result, the transverse projected emittances \( \varepsilon_{x,y} \) will increase as well. This could happen in our simulations since they were specifically designed to cool longitudinally. In Sim.4, the high charge results in strong nonlinear electron-ion interactions that significantly contribute to the growth of \( \varepsilon_{x,y} \).

Since the initial horizontal and vertical emittances are slightly different, the smaller initial emittance \( \varepsilon_y \) experiences stronger nonlinear interactions and thus increases more than the larger initial emittance \( \varepsilon_x \). For the transverse eigen emittances \( \epsilon_{1,2} \) here, the decrease of the amplitude of their oscillations with time is possibly due to the very fast cooling of the longitudinal emittance (a result of the high charge) which compresses \( \delta \) rapidly and that changes the effect of the dispersion through the rest of the ring.

### 5.1.2 Modulator Section of the Coherent Electron Cooling

The luminosity in Eq. 5.1 was derived based on some assumptions of an ideal head-on collisions of the beam bunches [113]. In real machines, some effects such as the crossing angles, the collision offset, and the hourglass effect need to be included. Because the transverse beam size \( \sigma \) varies with \( \beta(s) \) which also varies along the collision region, this variation generates a geometrical effect called the hourglass effect described by the coefficient \( h(x) \) given as [14]

\[ h(x) = \frac{\sqrt{\pi}}{x} e^{1/x^2} \text{erfc}(1/x). \]
Then, the luminosity can be written as

\[ \mathcal{L} = \frac{N_1N_2f_{\text{rev}}n_b}{4\pi\beta^*\varepsilon} h \left( \frac{\sigma_s}{\beta^*} \right). \]  

(5.19)

As a result, the transverse beam size depends on \( \beta^* \) and on the longitudinal position \( s \) along the collision region described by the bunch length \( \sigma_s \). The hourglass effect imposes the limitation \( \beta^* \geq \sigma_s \) when \( h > 0.75 \) [14]. This limitation enables lowering \( \beta^* \) by reducing the bunch length \( \sigma_s \) which increases the luminosity.

In modern colliders, the ion beams are accelerated to very high energies such as the Large Hadron Collider (LHC) which accelerates protons to 7 TeV. The high intensity requirement makes it challenging to cool the beams with standard cooling techniques like the traditional electron cooling and the stochastic cooling. Thus, a novel new cooling method called the coherent electron cooling (CeC) was proposed to cool high-intensity high-energy beams more efficiently [14, 129]. The feasibility of CeC technique was introduced in 1980 [115, 129], but a practical scheme with a detailed theory was first proposed in 2007 [14, 129].

The CeC system consists of three sections: a modulator, an amplifier, and a kicker. In the modulator, cooling electrons co-propagates with the ions at the same velocity and the electron density is modulated by the ions. Then, the density modulations are amplified in the amplifier section. In the kicker, the amplified electron density modulation is fed back to the ions which receive energy kicks toward their central energy. As a result, the ion beam emittance and energy spread are reduced, and the beam is cooled. Different CeC systems were proposed and theoretically explained, and they vary mainly in the amplification method [130]. Currently, cooling using variants of the CeC system is regarded as the strong electron cooling.

At Brookhaven National Laboratory (BNL), an electron-ion collider (EIC) is being constructed by modifying the existing Relativistic Heavy Ion Collider (RHIC). This site was
selected by the U.S. Department of Energy to advance nuclear science in early 2020. In order to test the CeC concept before it can be applied to achieve high luminosity in EIC, Proof-of-Principle (PoP) experiments were proposed in [131] and commissioned at RHIC in the past few years.

The first section of any CeC system, the modulator, is based on the Coulomb interactions between the ions and the electrons. The ion beam and the electron beam co-propagate in the straight line of the modulator with the same velocity. Each ion attracts the surrounding electrons until there are a total charge equal to that of the ion but of opposite sign, a process known as Debye shielding (or screening), which modulates the electron beam density. Because these density modulations are going to be used in the rest of the CeC system, it is essential to understand the modulation process and get an accurate estimate of the cooling time. For an ion moving in an infinite, uniform electron distribution, an analytical solution of the modulation process is calculated in [132]. For a non-uniform electron distribution, different numerical approaches were carried out such as in [15, 16, 17, 18]. These numerical methods were benchmarked with the analytical solution of the uniform electron beam without external fields and have not been verified by experimental results yet.

The underlying physical mechanism of many plasma phenomena such as Debye shielding is directly related to collisions, and hence an $N$-body approach is crucial [19]. Therefore, we performed simulations of the modulation process using our collisional method PHAD which can deal with any particle distributions and easily include realistic external fields along the beamline.

We considered the modulator section of the PoP CeC experiments in RHIC at BNL. In these experiments, the modulator section is a 3 m long with $\beta = 4.2$ m, and it includes four quadrupoles described in [18]. For the ion beam, it consists of fully stripped Gold ions Au$^{+79}$ that are accelerated to an energy of $\gamma = 42.9$. The velocity of the electron beam is matched to that of the ion beam.
Our goal is to quantify the best initial configuration of the ion beam that can give strong density modulations of the electron beam and examine some cooling limitations. As noted by [18], particle simulations result in a shot noise much larger than the density modulations which makes it difficult to observe the modulation signal. To extract the signal, we apply two procedures. The first one follows the method used by [18] where two simulations are performed with the same electron distribution, but one of them includes the ions and the other is without the ions. Throughout the modulator section, the resulted electron distributions are subtracted to extract the modulation signal. The second procedure is to apply signal averaging where we perform many simulations with new random generations of the electrons’ positions and momenta of the same initial conditions and take the average of the results of these simulations. We choose to use the first approach in all our simulations of various configurations because it gives a much clearer signal that is easier to quantify, and we perform only one simulation by the second approach.

Since the size of the electron beam in the PoP CeC experiments at RHIC described in [18] are large compared to the lengths related to the ion’s shielding (Debye lengths) and because the far away electrons do not contribute to the shielding, we consider only a small part of the beam. Because the longitudinal Debye length is much smaller than the transverse one, we consider a longitudinal slice of the electron beam with a relatively smaller transverse size than the one in the experiments. The modulator simulations were performed by PHAD where we used a Gaussian distributed electron beam with the parameters described in Table 5.5 along with PHAD parameters. These parameters in Table 5.5 were applied in all the following simulations unless it is noted otherwise.

We propagated a beam of electrons through the modulator to observe the effect of the quadrupoles on the transverse beam size of the electron beam. As shown in Fig. 5.15, the horizontal beam size starts decreasing after the first quadrupole at 0.6 m, reaches a minimum after the second quadrupole at 1.2 m, and increases after that. The vertical beam
size behaves in the opposite way and reaches a maximum after the second quadrupole at 1.2 m. Therefore, we expect the transverse density modulations due to the ions to be affected by these changes of the transverse beam size which is due to the quadrupoles. Because of the Gaussian distribution of the electrons, this variation of the beam size would influence the strength of the modulation signal for off-axis ions transversely where the number of electrons around them will vary as well.

5.1.2.1 Simulations with One Ion

We consider a single ion at the center of the electron beam and we extract the modulation signal along the modulator section. Then, we will study the effect of varying the location of the ion and its momentum with respect to the electron beam reference frame. Figure. 5.16 shows the density modulations of the electron beam due to a centered ion resulted from our PHAD simulation. In the longitudinal direction, the signal increases with the propagation
Figure 5.15: Density plots from PHAD simulations demonstrating the change of the electron beam transverse size as the beam propagates through the lattice of the modulator section of the CeC.

distance while the evolution of the transverse modulations reflects the effect of the modulator quadrupoles on the transverse beam size. The horizontal modulation signal reaches maximum when the horizontal beam size is minimum at 1.2 m. For the vertical modulations, the vertical beam size is minimum at about 1.8 m where the signal is maximum.

Because only the longitudinal modulations are related to the concept of the CeC, we will mostly present only the results of the density modulations in the longitudinal direction in the remaining simulations. Before we continue with different ion configurations, we show that our choice of simulating a longitudinal slice gives results similar to the one by a larger beam of the same density. Thus, we increase both the longitudinal beam size and the number of
(a) Longitudinal modulations. 
(b) Horizontal modulations. 
(c) Vertical modulations.

Figure 5.16: PHAD simulations of density modulations of the electron beam due to a single centered ion through the modulator section of the PoP CeC at RHIC. The longitudinal signal increases with the propagation distance while the transverse density varies according to the modulator lattice.

electrons by a factor of 2 and compare its results with the previous results. In this case,
the larger beam has $2 \times 10^5$ electrons and its longitudinal RMS size is $2\sigma_z$. The results are shown in Fig. 5.17 where the resulted longitudinal density modulations are comparable.

![Figure 5.17: Comparison of the resulted longitudinal density modulations from two simulations of the same density, but differ in size and the number of electrons. One simulation included $10^5$ electrons within $\sigma_z$, and the other included $2 \times 10^5$ electrons within $2\sigma_z$ of the same density.](image)

Now, we consider an off-axis ion with respect to the center of the electron beam. Let the position of the ion be at $(x_{ion}, 0, 0)$, and we vary $x_{ion}$ as $\pm \sigma_x$ and $2\sigma_x$. The results are shown in Fig. 5.18 where the decline of the signal compared to the centered ion is clear. The longitudinal signal is about the same for $x_{ion} = \pm \sigma_x$ and it decreases as the distance from the $x$-axis center increases as for the $x_{ion} = 2\sigma_x$ case. The reason for this decrease is the Gaussian distribution of the electrons where the number of electrons around the ion decreases the further the ion is displaced from the $x$-axis center, and it almost disappear for more than $\sigma_x$ distances.

Since the effect of the modulator lattice on the transverse size of the electron beam differ between the horizontal and the vertical, we also consider an ion off the center of the $y$-axis
(a) $x_{\text{ion}} = \sigma_x$.

(b) $x_{\text{ion}} = -\sigma_x$.

(c) $x_{\text{ion}} = 2\sigma_x$.

Figure 5.18: PHAD simulations of the longitudinal density modulations of electron beams through the modulator section of the PoP CeC at RHIC with a single ion located at $(x_{\text{ion}}, 0, 0)$ for different $x_{\text{ion}}$. The signal is reduced by the displacement from the center of $x$-axis compared to the centered ion.

by $\sigma_y$ and $2\sigma_y$. The resulted longitudinal density modulations along the modulator section are shown in Fig. 5.19. Compared to the signal of the centered ion, the signal decreases
for the ion at $\sigma_y$ and decreases more at $2\sigma_y$ reflecting the lower number of electrons further from the electron beam core. We also notice that the longitudinal signal is less affected by displacing the ion off the $y$-axis compared to displacement of the ion off the $x$-axis by the same amount. The reason for that is that the vertical size of the electron beam is larger than the horizontal size and the ion sees more electrons throughout most of the modulator.

Figure 5.19: PHAD simulations of the longitudinal density modulations of the electron beams through the modulator section of the PoP CeC at RHIC with a single ion located at $(0, y_{ion}, 0)$ for: (a) $y_{ion} = \sigma_y$ and $y_{ion} = 2\sigma_y$. The signal decreases as the ion displacement from the center increases.

Although a longitudinal slice of the electron beam should be represented by a uniform distribution longitudinally, we chose a Gaussian distribution to show the effect of the ion placed off the core of the electron beam. Accordingly, we consider an ion to be off the center of the $z$-axis by about $0.5\sigma_z$ and then by $\sigma_z$. We compare the resulted modulation signals with the one from the centered ion as shown in Fig. 5.20 where it can be seen that the signal decreases as we move away from the center of the electron beam. The decreased signal reflects the decrease of the number of electrons away from the core of the electron beam. If
we compare the signal due the ion at $\sigma_z$ with the one due to the ion at $-\sigma_z$, both off the $z$-axis, we see in Fig. 5.21 that the strength of both signals is the same which is due to the symmetry of the electron beam longitudinally.

Figure 5.20: Comparison between the longitudinal density modulations of $10^5$ electrons for an ion located at $(0, 0, z_{ion})$ when $z_{ion} = 0, 0.5\sigma_z,$ and $\sigma_z$. The reduction of the signal due to off-centered ion is a result of the decrease of the number of electrons.

Going back to an ion at $(0, 0, 0)$, we vary the ion’s velocity with respect to the electron beam velocity. In the frame of electron beam, the velocity spread of the electron beam is $\sigma_{v_z}$. We start with an ion moving with a velocity of $\sigma_{v_z}$ and $3\sigma_{v_z}$ with respect to the electron beam. Because the electrons velocity spread in the CeC experiments is much larger than the one we used in our simulations, we also simulated a moving ion with a velocity of $10\sigma_{v_z}$ with respect to the electron beam. The results are shown in Fig. 5.22 and compared to a stationary ion with respect to the electron beam. For $\sigma_{v_z}$ ion, there is almost no change in the signal while there is a small change for the $3\sigma_{v_z}$ ion where it is slightly asymmetric. The asymmetry is clear for the $10\sigma_{v_z}$ ion with a small displacement as the ion itself has moved from the center of the electron beam.
\[ z_{ion} = \sigma_z. \]

\[ z_{ion} = -\sigma_z. \]

Figure 5.21: PHAD simulations of the longitudinal density modulations of the electron beam through the modulator section of the PoP CeC at RHIC with a single ion located at \((0, 0, z_{ion})\) for \(z_{ion} = \pm \sigma_z\). The signals are displaced to the location of the ions and are of the same strength.

5.1.2.2 Simulations with More than One Ion

We performed simulations that include more than one ion at different locations longitudinally, and propagated them surrounded by the electron beam through the modulator section. First, we considered two ions in which one ion is positioned at \((0, 0, \sigma_z)\) and the other is at \((0, 0, -\sigma_z)\), and extract the modulation signal. In Fig. 5.23, there are two peaks around the positions of both ions in the longitudinal direction, and the strength of the signal in the transverse direction is about twice that due to one ion. Then, we add one more ion at the center of the electron beam \((0, 0, 0)\), and get the density modulations due to the three ions in Fig. 5.24. Similar to the two ions case, there are three peaks around the positions of the ions longitudinally, and the strength of the signal in the transverse direction is about three times that due to one ion.
Figure 5.22: PHAD simulations of the longitudinal density modulations of the electron beam due to a centered ion, moving with respect to the electron beam through the modulator section of the PoP CeC at RHIC. The signal due to the moving ion is compared to that of the stationary ion for ion’s velocities: (a) $\sigma_v$, (b) $3\sigma_v$, and (c) $10\sigma_v$.

The variation of the strength of the longitudinal signal around the three ions in Fig. 5.24a reflects the Gaussian distribution of the electrons. When the longitudinal distribution of the electrons is uniform, the three signals should be the same. We show that this is the case with a uniform longitudinal electron distribution of length $4\sigma_z$ in Fig. 5.25.

The presented simulations in this section showed clear signals around the ions where we placed the ions about $d = \sigma_z$ apart longitudinally. This leads to the following question: how the signal would be affected if the ions were much closer to each other? The expectation is that, at some point, the electrons would see the ions as one with higher charge and there will be one peak around those ions. To get an estimation of at what distance between the ions this would happen, we performed simulations varying $d$ between three ions. Figure. 5.26
(a) Longitudinal modulations.  
(b) Horizontal modulations.  
(c) Vertical modulations.

Figure 5.23: PHAD simulations of density modulations of the electron beam through the modulator section of the PoP CeC at RHIC due to two ions, one is at $\sigma_z$ and the other is at $-\sigma_z$. There are two peaks around the two ions longitudinally, and the signal is about twice that of a single ion transversely.

shows our results for $d = 0.2\mu m, 0.1\mu m$ and $0.05\mu m$ at the end of the modulator section.  
For $d = 0.2\mu m$, we can see three clear signals around the ions. The three signals start to
Figure 5.24: PHAD simulations of density modulations of the electron beam through the modulator section of the PoP CeC at RHIC due to three ions placed at $-\sigma_z$, 0, and $\sigma_z$. There are three peaks around the three ions longitudinally, and the signal is about three times that of a single ion transversely.

become less well-defined for $d = 0.1 \mu m$, and they become one large signal when $d = 0.05 \mu m$.

Thus, it is possible to say that the density modulations are better when the distance between
the ions is larger than $d = 0.1\mu m$. In fact, there is no need for more cooling when the ions are as close to each other such that they cannot be distinguished anymore.

It was suggested in [133] that it is possible to perform simulations with a single ion in different configurations and extract the signal, and then apply the superposition principle to get the full effect of the ion beam on the electron beam. We have examined this proposition for three cases: two ions, three ions, and five ions. The ions were positioned on-axis transversely, and longitudinally the two ions were positioned at $\pm \sigma_z$ for the two ions case; a third ion was at the center of the axis for the three ions case; two additional ions were positioned at $\pm 0.5\sigma_z$ for the five ions case. We show in Fig. 5.27 a comparison between the extracted signal when the simulations were performed with multiple ions (blue curve) and
Figure 5.26: PHAD simulation results at the end of the modulator section showing density modulations of the electron beam due to three ions when the distance between the ions is: (a) 0.2 µm, (b) 0.1 µm, and (c) 0.05 µm. The modulation is improved when the ions are placed more than 0.1 µm apart.

when the resulted signals were a superposition of each signal of individual ions separately (orange curve). For the two ions case in Fig. 5.27a, the signals from both ways were almost the same. However, the signals started to differ when more ions are included. This difference is small for the three ions case in Fig. 5.27b, and it is noticeably large for the five ions case in Fig. 5.27c. As we added ions into the same density and bunch length of the electrons, the distance between the ions decreases. While electrons can move freely, relatively, to shield a single ion, that is not the case in the presence of other ions’ attracting forces. Thus, our results suggest that the superposition principle should be used carefully as it can give inaccurate signals when any ions are close longitudinally.
Figure 5.27: Comparison between the modulation signals when the signal is a result of one simulation of multiple ions (orange), and when the signal is a superposition of signals due to individual ions (blue). The density modulations are due to: (a) two ions, (b) three ions, and (c) five ions.

5.1.2.3 Signal Averaging Simulations with One Ion

Using the same parameters from Table 5.5 and a single ion at the center of the electron beam, we performed 20 runs with freshly random generation of the electrons’ positions and momenta. Then, we performed signal averaging of the results where the signal tends to accumulate, and the noise is reduced. In Fig. 5.28, we show the results of the signal averaging at three different propagation distances with an inset for the region where the signal should be. We see that the electron beam longitudinal size increases as it propagates through the modulator section and the signal becomes more visible towards the end. At
0.6 m, the signal is not visible indicating a weak signal, but it becomes clear (stronger) for propagation distances 1.8 m and the end of the modulator section at 3 m.

(a) At propag. dist. = 0.6 m.  
(b) At propag. dist. = 1.8 m.  
(c) At propag. dist. = 3 m.

Figure 5.28: PHAD simulations of density modulations of the electron beam due to a centered ion through the modulator section of the PoP CeC at RHIC using signal averaging for 20 runs at different propagation distances: (a) 0.6 m, (b) 1.8 m, and (c) 3 m (end of the modulator section). The inset of each plot shows the region where the signal should be, represented by the small bump in (b) and (c)

5.1.2.4 Remarks on the Simulation Results of the Modulator

In [17] and [18], simulations were performed with a larger beam size and density of the electrons than in our simulations. However, the behavior of the resulted density modulations for the centered, off-axis transversely, and off-reference momentum ions are similar in both
our simulations and in those references. The modulation signal due to a centered ion is
the strongest and it starts to decrease as the ion is displaced off-axis transversely reflecting
the decrease of the number of electrons that can shield the ion and the focusing effects of
the modulator lattice. An off-reference momentum ion results in a slightly smaller signal
with some asymmetry compared to the one due to the ion of reference momentum. Since
PHAD simulation is based on first principles, thus our simulations support the modulation
simulation results in [17, 18].

Our simulations results suggest that the best longitudinal density modulations due to
the ions are achieved when the ions are well-aligned with the center of the electron beam
transversely, and far from the electron beam edges longitudinally. Also, it is important
to have a distance larger than 0.1µm between the ions longitudinally to get a well-defined
strong signal around each ion, and thus achieve better cooling. When two ions are so close
such that they are indistinguishable by the electron beam, that means no further cooling
can be accomplished. One also has to be careful when using the superposition principle to
combine the signals due to individual ions as that can become inaccurate, especially when
ions are relatively close.

5.2 Relaxation of Certain Beam Perturbations

Common charged particle systems involve a velocity distribution that follows a Maxwellian
distribution which can be perturbed from its equilibrium for different reasons. This pertur-
bation causes the system to oscillate in order to restore its equilibrium. Depending on the
system, these oscillations can be damped through Landau damping, or it could grow by the
inverse Landau damping (instability) [119].
In plasma physics, the interaction between an electron beam and a neutral plasma has been observed to excite longitudinal electrostatic waves known as Langmuir waves both in experiments and also in numerical simulations. One of the most important beam-plasma interactions is the weak beam-plasma instability, also known as the bump-on-tail (BoT) instability, which served as a testbed for various theories on nonlinear plasma-wave interactions (see [134] and the references therein). The bump-on-tail problem was first observed experimentally in 1960’s [135] and it has applications in various fields such as astrophysics, cosmical geophysics, and fusion plasma [136, 137, 138]. A BoT system consists of a spatially uniform and neutral plasma where a small energetic electron beam propagates through this plasma which perturbs the plasma equilibrium and excites the electrostatic Langmuir waves that grows through its interaction with beam particles. The energetic beam appears as bump on the tail of the electrons’ velocity distribution and hence the name bump-on-tail.

In accelerators, the longitudinal momentum of a charged particle beam generally follows a Gaussian distribution with a specific momentum spread. This momentum distribution can deviate from a Gaussian by some perturbation through a brief interaction between the beam and an external field such as an RF voltage or a laser. This brief interaction causes the beam to oscillate and excite longitudinal waves that are quickly damped by the system due to the momentum spread. As a result, the beam’s momentum distribution is slightly modulated, but it relaxes towards equilibrium after some time. It has been shown that modulating the momentum distribution of the beam with some frequency $\omega_1$ and follow it (after a relatively long time) with another frequency $\omega_2$ results in an echo signal with a frequency related to both $\omega_1$ and $\omega_2$ [139]. The phenomenon of beam echoes has different applications such as in measuring diffusion rates in high energy synchrotrons [140], the generation of short-wavelength radiation [141], and free electron laser [142].

Whether it is a wave interaction with a neutral plasma or with a charged particle beam, this interaction is usually described through the kinetic model which is based on Vlasov
equation (collisionless Boltzmann equation). In this equation, collisions are ignored, and the physical mechanism is provided by the collective particle interaction effects described by a phase space particle distribution function, evolving in an electromagnetic field, that can relax towards equilibrium. For each particle species, Vlasov equation is given by

$$\frac{\partial f}{\partial t} + \mathbf{v}(\mathbf{p}) \cdot \nabla f + q(\mathbf{E} + \mathbf{v}(\mathbf{p}) \times \mathbf{B}) \cdot \nabla_p f = 0,$$

where $f$ is the phase space distribution function, $\mathbf{v}(\mathbf{p}) = \mathbf{p}/\gamma m$, $\mathbf{E}$ and $\mathbf{B}$ are the electric and magnetic fields, respectively. In a time scale that is shorter than the collision time, there exists an infinite number of equilibrium solutions to Vlasov equation.

In the kinetic approach, the kinetic limit $N \to \infty$ does not always provide an accurate description of the physical phenomena and a microscopic $N$-body approach is important. Specifically, the dynamics in the limit $N \to \infty$ are valid on a finite time interval and it differs from the $N$-body description for a finite $N$ or for a very long-time interval $t \to \infty$ [19]. In addition, general kinetic approaches consider the One Component Plasma (OCP) model in which the plasma is regarded as infinite with spatial periodicity in 3D and may work only for a uniform plasma. This model uses a Coulomb potential that is smoothed by imposing a cut-off on the smallest impact parameter between particles encounters. However, the Coulomb potential from the interactions of impact parameters smaller than the cut-off represents the dynamics in time scales shorter than the collision time relevant for the Vlasov model, and thus the resulted dynamics are not fully correct [22]. It has been suggested that actual, finite $N$ physical behaviors may be overlooked by employing the kinetic model [20, 21, 22, 19]. While applying the collisionless methods for a collisionless plasma in which the long-range electrostatic interactions dominate over the pair-wise collisions seems justified, collisions have a crucial contribution to the dynamics even in collisionless plasmas. Thus,
the microscopic model from the $N$-body classical mechanics can give the most fundamental explanation of the underlying physics [19].

In this section, we will consider a charged particle beam that initially have a uniform spatial distribution and a Gaussian momentum distribution that is slightly perturbed, and study its relaxation to a Gaussian using our numerical simulation methods. These simulations are an initial step to study other charged particle systems phenomena such as beam echo. Since beam echoes are very sensitive to diffusion, which is a consequence of Coulomb collisions, it has been shown that echoes provide a very efficient method to measure diffusion and collision rate [143]. This method is important for the performance of a variety of accelerators where beam diffusion can lead to effects that results in particle amplitude growth [143]. The generation of beam echoes by both protons and electrons has been recently studied in the Integrable Optics Testing Accelerator (IOTA) at Fermilab [144]. Our simulations can also be used for studying the relaxation of the BoT.

We consider perturbations in the longitudinal direction where the average longitudinal momentum of the beam is $p_0$ and the unperturbed Gaussian momentum distribution function $f_0(p)$ is

$$f_0(p) = \frac{n_0}{\sigma_p \sqrt{2\pi}} \exp\left(-\frac{(p - p_0)^2}{2\sigma_p^2}\right),$$

where $\sigma_p^2$ is the variance, and $n_0$ is the density of the particles in the beam. We will assume that the momentum distribution is perturbed such that a small group of particles of density $n_b$ have gained a momentum $\delta p$, and their average momentum is slightly higher than the rest of the particles. This perturbation appears as bump on the tail of the momentum Gaussian distribution where the distribution function can be written as

$$f(p) = \frac{n}{\sigma_p \sqrt{2\pi}} \exp\left(-\frac{(p - p_0)^2}{2\sigma_p^2}\right) + \frac{n_b}{\sigma_{p_b} \sqrt{2\pi}} \exp\left(-\frac{(p - p_0 - \delta p)^2}{2\sigma_{p_b}^2}\right),$$
where $n$ is the density of the unperturbed group of particles and $\sigma_{p_b}^2$ is the variance of bump.

Our simulations consider both proton beams and electron beams while varying the beam density, energy, momentum spread, and the size of the perturbation. The particle beams will propagate in a drift with no external electromagnetic fields. With the assumption that the equilibrium solution is a Gaussian distribution, we study under which conditions the momentum distribution relaxes to the equilibrium and how the spatial distribution is affected. All the simulations in this section were performed using the Simò integrator unless noted otherwise. The order of the Simò integrator used in these simulations was 10, and the required accuracy was $10^{-10}$.

5.2.1 Electron Beam

We perform our simulations for an electron beam of 5 MeV kinetic energy that is uniformly distributed over a cylinder of a radius 0.4 mm and longitudinal length 1.6 mm. First, we examine the effect of electron beam density on the relaxation time by varying the number of particles while fixing the momentum spread $n_b$ and the density of the perturbation $n_p$. Then, we explore how the momentum spread $\Delta p/p_0 = (p - p_0)/p_0$ affects this relaxation process. Last, we vary the perturbation density $\Delta p/p_0$ as a percentage of the initial beam density $n_0$ for a fixed number of particles and momentum spread.

5.2.1.1 Effect of Beam Density

We start with a beam of $N = 10^4$ electrons, $n_b = 10\% n_0$ and $\Delta p/p_0 = 3.3 \times 10^{-6}$ where the initial phase space distribution is shown in Fig.5.29a. Our simulation shows that the final momentum distribution relaxes to a Gaussian and the resulted phase space
distribution can be seen in Fig 5.29b. The evolution of the momentum distribution with time is shown in Fig. 5.30a where it gradually relaxes to the Gaussian distribution in about 200 ns. Figure 5.30b shows that the longitudinal spatial distribution was not affected here.

(a) Initial phase space distribution. (b) Final phase space distribution.

Figure 5.29: The phase space distribution of a beam of $10^4$ electrons of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed initially with $n_b = 10\% n_0$. Simulated by the Simô integrator, the momentum distribution relaxes towards a Gaussian after about 200 ns.

(a) Momentum distribution. (b) Longitudinal density.

Figure 5.30: The Simô integrator simulation of a beam of $10^4$ electrons of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed initially with $n_b = 10\% n_0$: (a) the evolution of the momentum distribution and (b) the initial and final longitudinal spatial distribution.
Now, we vary the beam density by increasing the number of particles from $10^4$ electrons to $4 \times 10^4$ and $10^5$ electrons. The simulations results are demonstrated in Fig. 5.31 where we notice that the momentum distribution relaxed to a Gaussian and the spatial distribution was not affected for both densities. Figures 5.31a and 5.31b show that the relaxation is faster for high densities ($50$ ns for $N = 4 \times 10^4$ and $30$ ns for $N = 10^5$), which could be due to the increase of collisions between particles as the density increases.

Figure 5.31: The Simó integrator simulation of two electron beams, $4 \times 10^4$ and $10^5$ electrons, both of initial uniform spatial distributions and Gaussian momentum distributions of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that are slightly perturbed with $n_b = 10\% n_0$. The relaxation time decreases with increasing beam density, and the spatial longitudinal density was not affected here.
5.2.1.2 Effect of Momentum Spread

We increased the momentum spread to $3.3 \times 10^{-5}$ and kept the density of the perturbation at $n_b = 10\% n_0$. For a small density beam of $10^4$ electrons, the momentum distribution depicted in Fig. 5.32a did not relax to a Gaussian even after a long time compared to the smaller $\Delta p/p_0$ in Fig. 5.30a. In fact, Fig. 5.32b shows that the spatial distribution became similar to the momentum distribution, i.e., a Gaussian with a small perturbation. The resulted phase space distribution is shown in in Fig. 5.32c where the simulations were performed for about 20$\mu$s. The large $\Delta p/p_0$ spreads the particles in space faster than the small $\Delta p/p_0$ which results in less collisions in addition to the $n_b$ particles moving away from the bulk which gives the results seen here.

If we increase the beam density by using $N = 4 \times 10^4$ electrons, the simulation results in Fig. 5.33 shows that the momentum distribution relaxes towards a Gaussian and the spatial distribution becomes a Gaussian as well. The relaxation time was about 4.4$\mu$s which is much longer than the 50 ns relaxation time for the beam of the same density but with smaller $\Delta p/p_0$ in Fig. 5.31a. These results suggests that even that the large $\Delta p/p_0$ spreads the particles in space, the system can relax towards equilibrium if the beam density is large enough where more particles collisions occur. Also, the long relaxation time causes the spatial distribution to change from uniform to Gaussian.

5.2.1.3 Effect of the Perturbation Size

Considering a beam of $10^5$ electrons and small $\Delta p/p_0 = 3.3 \times 10^{-6}$, we vary $n_b$ from the previously $10\% n_0$ to $20\% n_0$ and $30\% n_0$. The simulations results are illustrated in Fig. 5.34 where it can be seen that increasing $n_b$ slightly increases the relaxation time with about
Figure 5.32: The Simò integrator simulation of a beam of $10^4$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distribution did not relax to a Gaussian and the longitudinal spatial distribution became similar to the momentum distribution.

43.4 ns for $n_b = 20\% n_0$ and 50 ns for $n_b = 30\% n_0$. In addition, Fig. 5.34a and Fig. 5.34b show that the center of the relaxed Gaussian distribution was shifted slightly towards the perturbation and this shift increases as $n_b$ increases.
5.2.2 Non-Relativistic Proton Beam

In this section, we present our simulations of a non-relativistic proton beam of 2.5 MeV kinetic energy. The spatial distribution of the beam is uniform over a cylinder of a radius 0.4 mm and longitudinal length of 1.6 mm. We follow Section 5.2.1 and we examine the effect of beam density, the momentum spread $\Delta p/p_0$, and the density of the perturbation $n_b$. 

Figure 5.33: The Simò integrator simulation of a beam of $4 \times 10^4$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distribution relaxed to a Gaussian and the longitudinal spatial density changed from uniform to a Gaussian.
(a) Momentum distribution with $n_b = 20\%n_0$.
(b) Momentum distribution with $n_b = 30\%n_0$.

(c) Longitudinal density with $n_b = 20\%n_0$.
(d) Longitudinal density with $n_b = 30\%n_0$.

Figure 5.34: The Simò integrator simulations of a beam of $10^5$ electrons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with different $n_b$. Increasing $n_b$ moderately increases the relaxation time and slightly shifts the center of the Gaussian distribution towards the perturbation. The Longitudinal density was not affected here.

5.2.2.1 Effect of Beam Density

We set $n_b = 10\%n_0$ and $\Delta p/p_0 = 3.3 \times 10^{-6}$, and we compare the relaxation of the beam between $N = 4 \times 10^4$ and $N = 10^5$ protons. The simulations results are demonstrated in Fig. 5.35 where the momentum distribution relaxed to a Gaussian and the spatial distribution was not affected for both densities. Similar to the electron beams, Fig. 5.35a and Fig. 5.35b show that the relaxation is faster for high densities (about 47 ns for $N = 4 \times 10^4$ and 17 ns
for $N = 10^5$) which could be due to the increase of collisions between particles as the density increases.

### 5.2.2.2 Effect of Momentum Spread

We increased the momentum spread to $3.3 \times 10^{-5}$ and kept the density of the perturbation at $n_b = 10\% n_0$. For a beam of $N = 4 \times 10^4$ protons, the momentum distribution depicted in Fig. 5.36a did not fully relax to a Gaussian even after a long time compared to the smaller $\Delta p/p_0$ in Fig. 5.35a. In addition, Fig. 5.36b shows that the spatial distribution became similar to the momentum distribution, i.e., a Gaussian with a very small perturbation. The resulted phase space distribution is shown in in Fig. 5.36c where the simulations were performed for about 1.2\,µs. The large $\Delta p/p_0$ spreads the particles in space faster than the small $\Delta p/p_0$ which results in less collisions in addition to the $n_b$ particles moving away from the bulk which gives the results seen here.

### 5.2.2.3 Effect of the Perturbation Size

Considering a beam of $10^5$ protons and $\Delta p/p_0 = 3.3 \times 10^{-6}$, we vary $n_b$ from 10\%$n_0$ to 20\%$n_0$ and 30\%$n_0$. The simulations results are illustrated in Fig. 5.37 which shows that increasing $n_b$ increases the relaxation time with about 33\,ns for $n_b = 20\%n_0$ and 37\,ns for $n_b = 30\%n_0$. In addition, Fig. 5.37a and Fig. 5.37b show that the center of the relaxed Gaussian distribution was shifted slightly towards the perturbation and this shift increases as $n_b$ increases.
Figure 5.35: The Simò integrator simulations of two beams consisting of $4 \times 10^4$ and $10^5$ protons, which were initially of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distributions relaxed to Gaussian and the longitudinal spatial densities did not change, and the relaxation time is faster for the higher density beam.
Figure 5.36: The Simò integrator simulation of a beam of $4 \times 10^4$ protons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-5}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distribution did not relax to a Gaussian and the longitudinal spatial distribution became similar to the momentum distribution.

### 5.2.3 Relativistic Proton Beam

We consider a relativistic proton beam of the same speed as the electron beam considered in Section 5.2.1 to which the proton beam’s kinetic energy is 10 GeV. Since the electrons in our simulations were relativistic, we expect the relativistic protons to behave in a similar manner as the electrons. Thus, we consider one example to compare it with the correspondent electron case. The proton beam consisted of $4 \times 10^4$ protons with $\Delta p/p_0 = 3.3 \times 10^{-6}$ and
(a) Momentum distribution with $n_b = 20\% n_0$.

(b) Momentum distribution with $n_b = 30\% n_0$.

(c) Longitudinal density with $n_b = 20\% n_0$.

(d) Longitudinal density with $n_b = 30\% n_0$.

Figure 5.37: The Simò integrator simulations of a beam of $10^5$ protons initially with a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with different $n_b$. Increasing $n_b$ moderately increases the relaxation time and slightly shifts the center of the Gaussian distribution towards the perturbation. The Longitudinal density was not affected here.

$n_b = 10\% n_0$. The spatial distribution of the beam is uniform over a cylinder of a radius 0.4 mm and longitudinal length 1.6 mm. After about 800 ns, the momentum distribution is shown in Fig. 5.38 where no change was observed. Due to the relativistic effects, the interactions between the protons in the beam are weak and it might take very long for their perturbed momentum distribution to relax.

In order to increase the strength of the protons interactions, we have increased the density of the beam by decreasing the spatial size to a cylinder of a radius 35 µm and longitudinal
Figure 5.38: The Simò integrator simulations of a relativistic proton beam consisting of $4 \times 10^4$ protons, which were initially of a uniform spatial distribution and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10\% n_0$. The momentum distributions did not relax to a Gaussian during 800 ns.

length 0.14 mm. The results are shown in Fig. 5.39 where the momentum distribution started to relax toward a Gaussian within a few microseconds, and the longitudinal spatial density changed from uniform to a Gaussian. Compared to the relativistic electron beam, the relaxation time is much larger here and electrons spatial density did not change to a Gaussian. This could be still due to relativistic effects as the momentum of the relativistic protons is much higher than the momentum of the relativistic electrons. The Gaussian spatial density is usually a result of the long relaxation time.

5.2.3.1 Remarks on the Simulation Results

We have observed that, in general, the relaxation is faster whenever collisions are stronger which is the case for large $N$, small momentum spread $\Delta p/p_0$, and small density of the perturbation $n_b$. Also, the longitudinal density did not change for the fast relaxation and
Figure 5.39: The Simò integrator simulation of a beam of $4 \times 10^4$ relativistic protons initially with and a Gaussian momentum distribution of $\Delta p/p_0 = 3.3 \times 10^{-6}$ that is slightly perturbed with $n_b = 10\%n_0$. The initial beam density was increased by decreasing the cylinder’s dimensions in which the protons are uniformly distributed. The momentum distribution started to relax toward a Gaussian and the longitudinal spatial density changed from uniform to a Gaussian.

become of a similar shape to the momentum distribution in the case of the slow relaxation. For large $n_b$, the center of the Gaussian momentum distribution was slightly shifted towards the perturbation. Thus, if it is required to keep the spatial density uniform, the beam parameters should be set to get shorter relaxation time.

For the beam echo phenomena, these relaxation studies can provide information about which time should the second perturbation be applied. That is because the time to echo is linearly proportional to the time difference $\Delta t$ between the two applied perturbations [140].
It has been also shown in some studies of the transverse beam echo that maximum echo amplitude can depend on $\Delta t$, and that large momentum spread can generally result in a smaller maximum echo amplitude [144]. According to our simulations, large momentum spread results in a longer relaxation time and that would require larger $\Delta t$ which might be a cause of the small maximum echo amplitude.

The dependence of the relaxation times on the different parameters presented in our simulations indicates the importance of including finite $N$ effects. Our microscopic simulations can provide such effects and show the importance of collisions unlike the collisionless model that considers the kinetic limit $N \to \infty$ and ignores collisions.

While these simulations were performed by the Simù integrator, some of them took a very long CPU time. Thus, PHAD is more efficient for a full study of the beam echo phenomena. PHAD simulations will also provide finite $N$ effects and long-time behavior which may not be provided by the collisionless approach.
CHAPTER 6
SUMMARY AND CONCLUSIONS

Numerical methods are essential in beam physics, and currently the general approach in the beam physics community is based on the collisionless (kinetic) model where their algorithms and codes are widely applied. These methods are sufficient in the collisionless regime where the dynamics does not depend on Coulomb collisions. However, the collisionless methods are inadequate to describe beam dynamics phenomena where the collisional effects are important such as in the high-intensity beams which are of current interest for future accelerator applications. Yet, there are no collisional methods and algorithms designed to model charged particle beams.

Collisional methods based on first principles give very complex algorithms and it is challenging to efficiently implement them. The most accurate collisional methods are the direct methods which have a computational complexity that scales quadratically with the number of particles. Since the number of particles involved in beam physics is very large, direct methods are considered inefficient. However, important insights of beam dynamics can be provided from the direct numerical methods that considers the $N$-body nature of charged particle beams. A literature review of the available numerical integrators of the gravitational $N$-body problem show that these integrators struggle to balance between the accuracy and the efficiency requirements of the simulated problem. Most of the time, accuracy is achieved at the expense of efficiency.

The main parameters controlled to achieve accuracy and efficiency of the direct integration methods are the time stepsize, the order of integration, and controlling the truncation errors. The appropriate choice of these parameters is challenging due to the presence of close
encounters. A major progress in this area is due to the development of adaptive integrators and a few variable order integrators, but the optimal choice of the stepsize, especially the initial one, was still a challenge. Considering all these challenges, we have developed the Simò integrator and designed it to be adaptive and variable order in a unique way. One main component of the Simò integrator is a readily adaptable Picard-iteration based integrator that was previously developed by our research group. Not only that we have employed the adaptive stepping and the variable order at each time step, but also for each individual particle at each time step. In addition, the dense output generated by our integrator allows to evaluate at any point of time within the convergence interval unlike integrators that allow evaluations only at discrete points in time.

To complete the efficiency of the Simò integrator in achieving a prescribed accuracy up to machine precision, we employed a Theorem of Simò proposed for an optimal selection of the time stepsize and order such that the computational cost is minimized. An added efficiency feature of the Simò integrator is the use of time bins which prevents unnecessary repeated expensive computations of forces. Moreover, we performed full parallelization of the algorithm of the Simò integrator to be able to deal with relatively large $N$. The resulted algorithm is adaptive, variable order with dense output integrator, and with optimal selection of the particle-by-particle time stepsizes and orders. The Simò integrator is unique and is the first efficient large-scale collisional numerical method that is symplectic to machine precision. The abilities and performance of the Simò integrator were demonstrated here for different beam dynamics examples.

With all the novel properties of the Simò integrator, its computational cost is still of $O(N^2)$. In addition, it is more appropriate to model short time dynamics as it loses the symplectic feature due to the discreteness when modeling long-time dynamics. Thus, our research group have developed a more advanced collisional method that reduces the computational complexity to $O(N)$ and maintain symplecticity of the long-time dynamics. This
method addresses three main challenges of the collisional methods: the long-range pair-wise forces, accurate collision management and time stepping, and maintaining the symplectic feature of the long-time dynamics. This collisional method is termed PHAD (Particles’ High-order Adaptive Dynamics) and uses a novel adaptive multi-level FMM (fast multiple method) to separate the pair-wise interactions into near and far which reduces the complexity of force computation to an $O(N)$. PHAD also employs Strang splitting to correctly combine the solutions of the far and near regions and preserve the symplecticity of the algorithm. In the early implementation of PHAD, the Picard-iteration based time integrator of a fixed order and time stepsize was used for the time stepping. However, the properties of the Simó integrator makes it the suitable choice to model collisions in PHAD algorithm. Thus, we have incorporated the Simó integrator in PHAD giving an accurate and efficient algorithm that is adaptive both in space and time. With all these remarkable numerical advances in PHAD, it is the first efficient collisional method in beam physics with a computational complexity of $O(N)$ and that is symplectic to machine precision. In addition, PHAD is fully parallelized, and thus we can perform realistic studies and describe complex beam dynamics phenomena of large-scale particle systems using our collisional algorithms with high accuracy in the most efficient way.

We have presented simulations performed by the Simó integrator and PHAD of complicated beam dynamics that are important for current practical applications. The first application was to the electron cooling of ion beams, both the traditional and the coherent electron cooling. Both methods are important to reduce the 6D phase space of the beam and increase the luminosity required by modern colliders. We started with benchmarking PHAD with experiments of bunched electron cooling at low energy that were performed by the JLab-IMP collaboration team at the IMP facility in China. These experiments were the first to demonstrate bunched electron cooling as an alternative method to DC coolers. The results of our simulations accurately reproduced cooling times, and showed that cooling
is faster for higher currents and longer pulses. Our simulations are the first microscopic electron cooling simulations that gave an accurate prediction of the cooling times that are consistent with the experimental results. Unlike other available electron cooling codes, our code is based on first principles with minimum assumptions, and does not require tuning or fitting of any parameters in the code. Moreover, we can accurately include all the nonlinear dynamics of the whole accelerator using transfer maps in addition to the cooling section. In principle, the framework of our approach to the electron cooling can be applied to all electron cooling forms and stages.

For the coherent electron cooling, we provided PHAD simulations of the density modulations in the modulator section which is a result of Debye shielding of the ion by the surrounding electrons. Our results support the possibility of obtaining modulations signals and show that better signals, and thus better cooling, is achieved when the ions are on-axis transversely. Longitudinally, the ions have to be within the core of the electron beam and without large variations of their velocities with respect to the electron beam. Modeling multiple ions suggests that these ions have to be well-separated (longitudinally) in order to obtain a clear, well-defined signal. If the ions are very close such that they are indistinguishable by the electron cloud, they cannot be cooled any further. Moreover, we showed that the superposition principle has to be applied carefully to combine the signals due to individual ions as it might result in incorrect signals when ions are relatively close. Although we have shown that it is possible to extract the modulation signal using statistical averaging, the signal is very small with respect to the electron distribution and it is much straightforward to quantify it using the other approach of subtracting two simulations, one with the ions and one without them.

In the last application, we used the Simò integrator to study the relaxation of the perturbed longitudinal momentum distribution of beams that is of importance to applications like beam echo in beam physics and the bump-on-tail problem in plasma physics. We showed
how the relaxation time can be affected by different parameters such as the number of particles, the momentum spread, and the size of the perturbation. For the beam echo applications, our studies are important to select the time at which a second perturbation can be applied, and this time difference between the two applied perturbation can affect the resulted maximum echo amplitude. Also, our simulations provide a microscopic picture of the dynamics with clear finite $N$ effects and importance of collisions which cannot be provided by the collisionless approach that considers the kinetic limit $N \to \infty$ and neglect collisions.

The development of our collisional methods, applying numerical and performance tests, and performing the simulations of the applications took a very long computational time. Conducted on the high-performance hybrid cluster Gaea, many jobs were launched, many nodes were utilized and CPU hours were consumed. Some statistics of my utilization of Gaea are included in Appendix C.

In conclusion, both of our methods are the first collisional methods in beam physics that allow relatively efficient realistic studies of large-scale particle beams with accuracy up to machine precision; the Simò integrator is the first collisional method and PHAD is the first efficient collisional method.
REFERENCES


[125] Yuhong Zhang, Andrew M Hutton, Kevin Jordan, Thomas J. Powers, Robert A. Rimmer, Michael F. Spata, Haipeng Wang, Shaoheng Wang, He Zhang, J. Li, L.J. Mao,


APPENDIX A

THE DIFFERENTIAL ALGEBRA TECHNIQUES
The Simò integrator and PHAD codes are implemented in COSY INFINITY and rely on its Differential Algebra (DA) techniques. We summarize here the basics of DA and follow their description on Chapter 2 of [93].

Numerical methods for differentiations were considered unpractical as computing Taylor expansions of functions involves differentiation and evaluation of functions at specific values which is complicated and results in inaccuracies. The differential algebraic structure consists of three main components. The first is an operator that can extract Taylor coefficients of a function up to a specific order, and thus translates the functions to an equivalence class containing all the functions with identical Taylor expansion to the same order. The second provides well-defined arithmetic operations to the set of equivalence classes of functions which results in the truncated power series algebra (TPSA). Lastly, the analytic operations of differentiation and integration are included.

The simplest nontrivial differential algebra $D_1$ involves a set of ordered pair $(a_0, a_1)$ where $a_0$ and $a_1$ are real numbers in $\mathbb{R}$. The three set of arithmetic operations that form an algebra are addition, scalar multiplication, and vector multiplication defined as

$$(a_0, a_1) + (r_0, r_1) = (a_0 + r_0, a_1 + r_1)$$

$$t \cdot (a_0, a_1) = (t \cdot a_0, t \cdot a_1)$$

$$(a_0, a_1) \cdot (r_0, r_1) = (a_0 \cdot r_0, a_0 \cdot r_1 + a_1 \cdot r_0).$$

The vector multiplication has a unity element $(1, 0)$, and it is commutative, associative, and distributive with respect to addition. Another unique property of $D_1$ is that it is totally ordered and this order is compatible with its operations. Also, $(a_0, a_1)$ has a multiplicative inverse defined as $(a_0, a_1)^{-1} = (a_0^{-1}, -a_1/a_0^2)$ if and only if $a_0 \neq 0$. Defining a positive
infinitesimal (or a differential) $d$ as $d \overset{\text{def}}{=} (0, 1)$ which does not have a multiplicative inverse or an $n$th root in $1D_1$ for any $n > 1$ and we have

$$(a_0, a_1) = (a_0, 0) + (0, a_1) = a_0 + d \cdot a_1,$$

where the first component is regarded as the real part and the second as the differential part. To turn the algebra $1D_1$ into a differential algebra, a derivation is introduced as a map $\partial: 1D_1 \mapsto 1D_1$ by $\partial(a_0, a_1) = (0, a_1)$, and thus $(1D_1, \partial)$ is a differential algebra.

The automated computation of derivatives by $1D_1$ is the essential feature for beam physics applications. Consider that the values and derivatives of two functions $f$ and $g$ at the origin are given, we can define the operation $[\cdot]$ from the space of differentiable functions to $1D_1$ as $[f] = (f(0), f'(0))$ and $[g] = (g(0), g'(0))$. The previous arithmetic operations applies here in addition to the fact that $[f(x)] = f([x])$ for a real $x$ and $[x] = (x, 1) = x + d$. Consequently, we can evaluate the value and the derivative of $f(x)$ at any $(x, 1)$ just by using arithmetic operations. As an example, consider the following function and its derivative

$$f(x) = x^2 + \frac{1}{1 + x}, \quad f'(x) = 2x - \frac{1}{(1 + x)^2}.$$  

At $x = 2$, direct evaluations give $f(2) = 4.3$ and $f'(2) = 3.89$. However, we can evaluate both the function and its derivative using the DA representation of $x = 2 = (2, 1)$ and we get

$$f((2, 1)) = (2, 1)^2 + \frac{1}{1 + (2, 1)} = (4, 4) + \frac{1}{(3, 1)} = (4.3, 3.89).$$

This treatment can be extended for any intrinsic function $g_i$ as $g_i([f]) = [g_i(f)]$ or $g_i((a_0, a_1)) = (g_i(a_0), a_1g'_i(a_0))$. Therefore, any function can be expressed by finite basic operations and intrinsic functions in $1D_1$. 
Finally, to be able to compute the derivatives of functions in $v$ variables up to an order $n$, we need to define the differential algebra $(nD_v, \partial_1, \ldots, \partial_v)$. In this space, the functions $f$ and $g$ are said to be $f \equiv_n g$ if and only if $f(0) = g(0)$ and all their partial derivatives are equal at 0 up to the order $n$. Hence, the function $f$ has the equivalence class $[f]$ that is a set of the elements which are related to $f$. In this formulation, DA vectors are the resulting equivalence classes. The collection of all classes is $nD_v$ and the arithmetic operations are defined as follow

$$ [f] + [g] = [f + g] $$

$$ t \cdot [f] = [t \cdot f] $$

$$ [f] \cdot [g] = [f \cdot g], $$

and the derivation is the map

$$ \partial_k[f] = \left[ p_k \cdot \frac{\partial f}{\partial x_k} \right], $$

for each $k \in \{1, \ldots, v\}$ where $p_k(x_1, \ldots, x_v) = x_k$.

In the implementation of the DA techniques in COSY INFINITY software, evaluating a function in DA creates a DA vector that contains the truncated Taylor polynomial around a particular expansion point. These DA vectors are treated according to the various supported operations of DA such as arrays of DA vectors, arithmetic operations, derivation and antiderivation, evaluation, inversion, and composition.
APPENDIX B

THE CONCEPT OF EIGEN EMITTANCE
We used eigen emittances in Chapter 5 to describe the electron cooling, and thus we give an overview of their concept here following the presentation of [145].

The 6D phase space of a beam is described by the projected emittances, transverse and longitudinal. These projected emittances represent the beam dimensions in the lab frame where different planes can be coupled. Starting with the canonical coordinates $q_m$ and momenta $p_m$, the 6D phase space for each particle is given by $z = (q_1, p_1, q_2, p_2, q_3, p_3)$. For a collection of particles, the beam matrix in the lab frame is given by

$$
\Sigma = \langle (z - \langle z \rangle) \cdot (z - \langle z \rangle)^\top \rangle,
$$

and the symmetric block-matrix takes the form

$$
\Sigma = \begin{pmatrix}
\Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\
\Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\
\Sigma_{31} & \Sigma_{32} & \Sigma_{33}
\end{pmatrix}.
$$

The elements $\Sigma_{mk}$ are $2 \times 2$, and when all of elements of $\Sigma$ are zero except for $\Sigma_{mk}$ with $m = k$, the beam matrix $\Sigma$ is regarded as uncoupled.

For a coordinate plane $(q_m, p_m)$, the projected emittance $\varepsilon_m$ can be calculated from the square root of the determinant of $\Sigma_{mm}$ and is written as

$$
\varepsilon_m = \sqrt{\text{det} (\Sigma_{mm})} = \sqrt{\langle q_m^2 \rangle \langle p_m^2 \rangle - \langle q_m p_m \rangle^2}.
$$

Reference [145] shows that the projected emittances are invariant under linear uncoupled symplectic transformations. On the other hand, the beam matrix can be expressed in a frame where all the degrees of freedom are uncoupled, and thus the beam dimensions are given by the eigen emittances which are invariants under linear coupled or uncoupled symplectic
transformations. The eigen emittances are equal to the projected emittances if all the degrees of freedom are uncoupled in the lab frame.

One way to obtain the eigen emittances is by using a symplectic matrix $M$ to diagonalize the beam matrix as $M\Sigma M^\top = D$. The resulted diagonal matrix $D$ is of the form

$$D = \text{diag}(\Lambda, \Lambda), \quad \Lambda = \text{diag}(\epsilon_1, \ldots, \epsilon_n) > 0,$$

where $n$ is the number of degrees of freedom, and the diagonal elements $\epsilon_m$ are the eigen emittances. In the other way, the eigen emittances are calculated as the eigenvalues of $\Sigma J_{2n}$ where $J_{2n}$ is the $2n \times 2n$ symplectic unit matrix given as

$$J_{2n} = \text{diag}(\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix})_{n}.$$

In case of nonlinear transformations, the behavior of eigen emittances is not clear but it is generally supposed to be secular. Nonetheless, eigen emittances do not decrease unless there is a dissipative force such as cooling. This was the main advantage of using them in our application of the electron cooling since their decrease with time is a definite indication of cooling.
APPENDIX C

GAEA CLUSTER INFORMATION AND STATISTICS OF UTILIZATION
The development and the parallelization of the Simò integrator and PHAD codes were performed on Gaea, which is the NIU’s 60-node CPU/GPU hybrid cluster shown in Fig. C.1. The cluster is running Red Hat Enterprise Linux 7 operating system. Connected via Full 1 : 1 non-blocking Infiniband and Ethernet switch connectors, each compute node is an HP SL380s G7 with the following configuration

- 2 × 6-core Intel X5650 processors with HyperThreading at 2.66 GHz
- 2 TB each node: 72 GB RAM 4 × 500 GB 2.5” SATA disk drives
- 6 GB RAM 2× NVIDIA M2070 FERMI GPUs

My utilization of Gaea is summarized in the following Table C.1. As it can be seen from Table C.1, I have submitted about 4500 jobs with about 94 × 10^3 wall hours in total. The total CPU hours is about 2 × 10^6 hours which is the highest any Gaea user have utilized so far.

Figure C.1: Gaea cluster assembly at NIU’s computing facility.
Table C.1: Summary of the number of jobs launched and the computational time used for our work, which were performed on the computing cluster Gaea.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Hours per Job: The average CPU hours (number of CPU cores × wall time hours) per job</td>
<td>457.89</td>
</tr>
<tr>
<td>Total CPU Hours: The total CPU hours (number of CPU cores × wall time hours) used by all jobs</td>
<td>2,064,189.8</td>
</tr>
<tr>
<td>Max Job Size (Max Core Count): The maximum total number of processor cores used by a (parallel) job per job</td>
<td>360</td>
</tr>
<tr>
<td>Job Size per Job (Core Count): The average number of processor cores used by a (parallel) job per job</td>
<td>17.2</td>
</tr>
<tr>
<td>Job Size Weighted By CPU Hours (Core Count): The average job size weighted by CPU hours</td>
<td>43.9</td>
</tr>
<tr>
<td>Node Hours per Job: The average node hours (number of nodes × wall time hours) per job</td>
<td>65.56</td>
</tr>
<tr>
<td>Total Node Hours: The total node hours (number of nodes × wall time hours) used by all jobs</td>
<td>295,527.1</td>
</tr>
<tr>
<td>Gaea Utilization (%): The ratio of the total CPU hours consumed by jobs over a given time period divided by the maximum CPU hours that the system could deliver</td>
<td>6.09%</td>
</tr>
<tr>
<td>Number of Jobs Submitted: The total number of jobs that submitted/queued within the selected duration</td>
<td>4,422</td>
</tr>
<tr>
<td>Number of Jobs Ended: The total number of jobs that ended within the selected duration</td>
<td>4,508</td>
</tr>
<tr>
<td>Wall Hours per Job: The average time, in hours, a job takes to execute</td>
<td>20.82</td>
</tr>
<tr>
<td>Total Wall Hours: The total time, in hours, all jobs took to execute</td>
<td>93,869.4</td>
</tr>
</tbody>
</table>