

ABSTRACT

INTENSE BEAM DYNAMICS IN ARBITRARY STRUCTURES

Anthony Gee, Ph.D.
Department of Physics
Northern Illinois University, 2018
Bela Erdelyi, Director

Particle accelerators are ubiquitous in science and society and their use is still growing globally. Beam physics, the physics underlying accelerator science, is focusing in part on studies and applications where intense charged particle beams become essential. The high-intensity may cause new collective instabilities and phenomena which are difficult to be modeled by conventional means. New numerical methods must be developed to efficiently and reliably model, simulate and optimize such high currents. The University of Maryland Electron Ring (UMER) and the Fermilab Integrable Optics Test Accelerator (IOTA) are dedicated test rings to study the high intensity regimes.

A 3-D symplectic tracking code, PHAD, was recently developed, which implements the adaptive Fast Multipole Method (FMM) in the differential algebraic (DA) framework to compute accurately and efficiently the self-induced Coulomb forces, and the beam dynamics under the combined external and internal forces. However, beam-environment interactions are missing. To add the beam-wall interactions, a new theory and numerical methods are needed. Previously, the beam-wall interactions were approximated using simplistic geometries that often gave unrealistic results.

To this end, we develop the Poisson Integral Solver with Curved Surfaces (PISCS) method and implement it in the general purpose nonlinear dynamics code COSY Infinity. PISCS uses the fast multipole accelerated boundary element method in the differential algebraic framework. PISCS efficiently represents the beam-wall interaction in arbitrary structures. We implement a strategy that can include the beam-wall interaction in other space charge tracking codes too. This work presents and benchmarks PISCS with complicated geometries and includes analyses of space charge and the beam-wall interactions using the extracted transfer maps.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

MAY 2018

INTENSE BEAM DYNAMICS IN ARBITRARY STRUCTURES

BY

ANTHONY GEE
© 2018 Anthony Gee

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

DEPARTMENT OF PHYSICS

Dissertation Director:
Bela Erdelyi

ProQuest Number: 10787708

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10787708

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

ACKNOWLEDGEMENTS

Without all the support over the years, I would never have reached this point. First and foremost, I thank my dissertation advisor, Dr. Bela Erdelyi, for his counsel, his guidance, and his endless patience. Without him offering me this opportunity, none of this would be possible. I thank my family: my parents, Joseph and Lucia, and my brother, Alex for always believing in me no matter what. I thank my friends and colleagues for providing a hand when I needed it the most. I particularly want to thank the members of my research group, Edward Nissen, Sumana Abeyratne, and Andrei Patapenka for laying the foundation. I especially thank Alex Halavanau for his insightful advice and reminding me what is important during our long work nights in NICADD.

DEDICATION

I dedicate this work to my family. I finally made it.

TABLE OF CONTENTS

	Page
List of Tables	vii
List of Figures.	viii
List of Appendices.	xvii
Chapter	
1 Beam physics at the Intensity Frontier	1
2 High-intensity projects and computational challenges.	5
2.1 Fermilab’s High-intensity Program	6
2.2 UMER.	10
2.3 Computational challenges	11
2.4 Modeling space charge.	13
2.5 The beam-wall interaction	14
3 Maps and Analysis	19
3.1 Equations of motion	19
3.2 Maps and DA	21
3.3 Normal form analysis on phase space	24
4 Fast Multipole Acceleration.	29
4.1 Fast Multipole Method	29
4.2 FMM operators.	30
4.3 FMM algorithm	32
4.4 Smoothing	35

Chapter	Page
4.5 FMM and GMRES	39
4.6 Rotated multipole to local operator	41
5 Boundary Integral Theory	44
5.1 Finite element models	44
5.2 FEM versus BEM	46
5.3 Boundary integrals	47
5.4 Regularization of the BIE	49
5.5 Direct versus indirect BIE	50
5.6 Existence, uniqueness, and conditioning	51
5.7 Discretization	53
6 High order discretization	56
6.1 Interpolation.	57
6.2 Limits of flat panels.	58
6.3 Analytical derivations of parametric elements	59
6.4 Higher-order interpolation	68
6.5 General quadrature	72
6.6 Performance of the interpolation and quadrature	77
7 Poisson Integral Solver with Curved Surfaces	83
7.1 PISCS Input.	84
7.2 Computing the nodes and weights	87
7.3 Implementation.	89
7.3.1 Near boundary evaluation	89
7.3.2 Redundant collocations	93
7.3.3 Parallelization	94

Chapter	Page
7.4 Combining space charge and the beam-wall interaction	95
7.4.1 PHAD	109
8 Results and analysis	111
8.1 Performance	111
8.2 Accuracy	117
8.3 Self- and image charge forces	139
9 Summary and conclusions	158
References	160
Appendices	181

LIST OF TABLES

Table	Page
4.1 Long Beam and System Parameters	36
4.2 PIC Simulation Conditions.	37
5.1 Condition Number vs. Matrix Size n for Perfectly Conducting Sphere	53
6.1 Spherical Triangle Parameters	65
6.2 Percent Errors On Spherical Triangle	66
6.3 L_∞ -error for (6.10) using Nyström method and trapezoidal (T) or Gaussian-Legendre (GL) quadrature. n is number of terms in the quadrature. GL quadrature performs significantly better.	73
6.4 Triangular Surface Vertices For Each Interpolation Test Function.	77
7.1 Space-charge Dominated Beam Parameters	99
7.2 Imaging Triplet System Parameters	100
7.3 FODO Cell System Parameters	105
8.1 Beam Parameters	140
8.2 Sextupole Parameters	140
8.3 Average Momenta and Final Emittance after Sextupole.	147
8.4 Beam Parameters	148
8.5 FODO Parameters	148

LIST OF FIGURES

Figure	Page
2.1 Overview of average beam current vs. energy accelerator landscape [1]. The product gives the on-target power with orders of magnitude indicated by the dashed lines.	5
2.2 Overview of the DUNE setup [2]. Fermilab will provide a neutrino beam from the LBNF into a near-source detector at LBNF and the main DUNE detector at SURF.	6
2.3 Schematic of the FAST beamline, connected to IOTA [3]. The beamline provides either 150 MeV electrons or 2.5 MeV protons.	8
2.4 Schematic of the IOTA ring [3]. The total length is 40 m with 2 sections for testing the nonlinear magnets.	8
2.5 Overview of UMER layout [4]. UMER has a circumference of 11.52 m, comprised of either 36 FODO periods (shown) or 18 FODO periods [5]. The ring is injected with space-charge dominated, 10 keV electron beams. A planned extraction line on the right is shown.	10
3.1 COSY 3 rd -order 2-D representation of a FODO cell. The columns contain the Taylor coefficients for the expansions of x_f, a_f, y_f, b_f and the exponent in each DA variable. The DA variables represent the initial positions and angles, x_i, a_i, y_i, b_i . For example, ${}_a C_{(3,0,\dots,0)} = (a x^3)$ is the element in row 5, column 2.	24
3.2 Phase space ellipse defined by Twiss parameters, α, β, γ , and emittance ϵ [6]. The horizontal axis is position and vertical is momentum.	26
4.1 A snapshot of the FMM algorithm [7]. The multipole expansion is computed by MEXP, translated up a level using M2M, and moved near the target using M2L. The local expansion is translated down a level if necessary using L2L and evaluated at the target using L2P. The nearby sources are evaluated directly using P2P.	33

Figure	Page
4.2 Depiction of FMM tree data structure using $N = 20, q = 10$, where sources and targets are the same. (a) shows the C-forest with 4 C-trees and (b) shows the D-tree.	34
4.3 Comparison of radial spread from smoothed FMM, MoM, PIC, and analytical solutions. The FMM is tested using $N = 22k, 100k$. (a) shows with equal smoothing and (b) shows with separate smoothing. (c) shows the FMM runtime is $\mathcal{O}(N)$. λ had minimal impact on runtime.	38
4.4 Normed relative difference in potential and \mathbf{E} between 3DM2L and rotated M2L operators vs. FMM order using $N = 100$ particles.	42
5.1 Depiction of half the sphere Γ_ε with radius $\varepsilon = \ \mathbf{x} - \mathbf{y}\ $ around singular point $i = \mathbf{x}$ and $\mathbf{y} \in \Gamma_\varepsilon$ [8]. To find the principal value, we can integrate the sphere separately in the limit of $\varepsilon \rightarrow 0$	49
6.1 Unit triangle for (u, v) -parametrization. The three vertices are shown in (a) as red dots with local coordinates indicated. The ordering of nodes for $e = 4$ is shown in (b), bottom left to top vertex.	56
6.2 A sphere discretized with flat elements. Visually, the discretized sphere approaches the actual sphere.	58
6.3 The percent error in area from discretizing a sphere $R = 0.1$ and cylinder $R = 0.1, L = 0.1$ using flat panels. The error appears to converge to 10^{-3}	59
6.4 Comparison of parametric patches on the spherical triangle (green) given by Table 6.1 [9]. In Figure 6.4a, the quadratic patch (blue) deviates slightly in the center. In Figure 6.4b, the patch using a nonic polynomial (blue) vs. its truncated cubic (red) deviate significantly in this case.	66
6.5 Standard deviation in the difference of (6.5a) the centroid and (6.5b) the unit normal [9]. The error bound for the quadratic patch quickly shrinks until $M \sim 5000$	67
6.6 Percent error in the area vs. number of elements for the flat panel, quadratic, and truncated cubic parametrization [9]. The truncated cubic gives essentially the same error as the flat panel discretization.	67
6.7 Runtime comparison between the different methods [9]. Overall, quadratic and cubic are similar due to the integration step.	68

Figure	Page
6.8 Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a spherical triangle, $R = 1$	78
6.9 Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a polynomial surface, $z(x, y) = -x^4y^3 - x^2 + xy + y$	79
6.10 Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a trigonometric surface, $z(x, y) = \cos x^3 \sin y^2$	80
6.11 Comparison of the quadrature with mapped symmetric Chebyshev roots and Gaussian-Legendre quadrature. Integral is transformed from the unit triangle to the unit square.	81
7.1 The flowchart of PISCS. There are four major blocks based on runtime: Initialization and element setup, computation of the modified boundary conditions, solving the linear system to match the BCs, and evaluating the density. Finally, the output potentials and fields are summed and written to files with negligible runtime.	83
7.2 PISCS input file format. Each parameter is on a separate line. Optional parameters can be left blank.	85
7.3 Format for structure file. First line is the number of elements.	86
7.4 Difference in potential for a PEC sphere. While the interior shows small difference, we see significant differences near the surface (black line). These differences shrink somewhat as element order increases.	90
7.5 Difference in fields for an ideal long quadrupole. The actual surface used is a cylinder (solid black line). Some of the equipotentials (dashed black line) are shown for reference. An accurate region is evident where the norm of the difference is small and expands as element order increases.	90
7.6 Schematic of QBKIX [10]. Panel L on Γ is discretized with Gauss-Legendre nodes. $c = \mathbf{x} - \delta \mathbf{n}$ is chosen away from boundary and an equivalent density is matched on ∂B_R^c for evaluation.	92
7.7 Error of each stage in QBKIX near Γ [10]. (a) shows the regular quadrature, (b) shows after splitting the particular panel L to refine the quadrature, and (c) shows after evaluating the matched density from ∂B_R^c	93

Figure	Page
7.8 Imaging triplet rays without space charge. Focal point is at $z = 1.4$ m [11]. . .	101
7.9 Behavior of $(x a)$ and $(y b)$ vs. current in the imaging triplet, as calculated by the moment method and the fast multipole method [11].	101
7.10 Behavior of $(x x^3)$ and $(y y^3)$ vs. current in the imaging triplet, as calculated by the moment method and the fast multipole method [11].	102
7.11 Imaging triplet rays w/ current = 1.5 A [11]. The focal point has shifted to $z = 1.43$ m in X - Z and to $z = 1.44$ m in Y - Z	102
7.12 Behavior of quad gradients vs. current in the imaging triplet after fitting for imaging, as calculated by the moment method and the fast multipole method [11].	103
7.13 Behavior of $(x a)$ and $(y b)$ vs. current in the imaging triplet after fitting, as calculated by the moment method and the fast multipole method [11].	104
7.14 Behavior of the geometric aberrations $(x x^3)$ and $(y y^3)$ vs. current in the imaging triplet after fitting, as calculated by the moment method and the fast multipole method [11].	104
7.15 Imaging Triplet rays w/ current = 1.5 A after fitting the quads for imaging [11]. The focal points are at $z = 1.37$ m in X and $z = 1.35$ m in Y	105
7.16 Tune shift due to space charge from tune = 0.1362 vs. current in the FODO lattice, as calculated by the moment method and the fast multipole method [11].	106
7.17 Behavior of the quad gradients vs. current in the FODO lattice after fitting to the desired tune, as calculated by the moment method and the fast multipole method [11].	107
8.1 CAD and mesh of PEC sphere with $R = 10$ m. Mesh has 128 elements at $e = 6$	111
8.2 Runtime versus number of elements at various element orders. The runtime is particularly large for $e = 2 - 7, M = 32, e = 2, 4, 5, M = 128$, and $e = 2, M = 512, 2048$. In all these cases, runtime for BCCALC is large due to using the unrotated M2L operator with many translations corresponding to the selected q . The bottom plot shows the parallelization with respect to number of processes. The reduced efficiency at high M is due to the setup of the high-order elements, where the interpolation is not parallelized.	113

Figure	Page
8.3 Runtime breakdown for large M . BCCALC dominates the runtime for small M and is the major cause of the outliers in figure 8.2 for order 2.	114
8.4 Runtime versus number of particles for element order 2 with increasing number of MPI processes on a computing cluster (up to 4 MPI processes shown).	115
8.5 Number of iterations in GMRES versus M and e for $M = 2048$. The number of iterations stays mostly constant despite the increase in M and e , i.e. system matrix size. The prescribed tolerance is $10^{-3.5}$	116
8.6 PEC sphere with $R = 0.1$ m, $V = 0.1$ V (a) without charge and (b) with ~ 1000 charges, $\frac{q}{\epsilon_0} = -1$ based on (8.1) in the XY -plane. Position is given in meters. The spherical surface is indicated by the circle. The units are chosen so that the effects are visible with modest N	117
8.7 Potential [V] in XY -plane from PISCS for the PEC sphere with ~ 1000 electrons and $\frac{q}{\epsilon_0} = -1$. Position is given in meters. For these results, we use our NVF interpolation. The spherical surface is indicated by the circle. . . .	119
8.8 Percent error for potential at the electrons with $M = 80$ (top), $M = 1280$ (bottom). $e = 2, 3$ show improved average error compared to constant element order with the higher gain at $M = 80$	120
8.9 An example configuration for a Hall axial probe inside a quadrupole magnet.	121
8.10 CAD and mesh of cylinder with $R = 0.165$, $L = 1$. Mesh has 224 elements at $e = 10$. Multipole boundary conditions are placed on the surface.	121
8.11 Expected scalar potential and field of of magnetic (a) quad and (b) octupole without charges. Position is given in meters.	123
8.12 Potential [T·m] and field [T] in XY -plane from PISCS for the defocusing quad with $g = -100\frac{T}{m}$. Position is given in meters. For these results, we use PISCS and Vandermonde interpolation. The cylinder is indicated by the solid circle and the equipotentials by the dashed curves.	124
8.13 Potential [T·m] and field [T] in XY -plane from PISCS for the magnetic octupole with $g = -1000\frac{T}{m^3}$, using PISCS and Vandermonde interpolation. Position is given in meters. The cylinder is indicated by the solid circle and the equipotentials by the dashed curves.	125

Figure	Page
8.14 Maximum difference between theory and PISCS for (a) quadrupole and (b) octupole. There is a spike for $e = 2$, suggesting the approximation error is high, but overall higher order elements improve the accuracy.	126
8.15 Average difference between theory and PISCS for (a) quadrupole and (b) octupole.	127
8.16 CAD of (a) IOTA sextupole magnet and (b) inner surface without coils. Inner surface mesh example shown in (c) with $M = 2080, e = 6$. Current is run through coils such that the pole tips produce the magnetic field. Inner surface is used in PISCS.	129
8.17 Sextupole potential [T·m] (left) and field [T] (right) in XY -plane. Position is given in meters. The pole tips are indicated by the solid curves.	130
8.18 Potential [T·m] (left) and field [T] (right) for sextupole with no charges from PISCS with Dirichlet conditions. Position is given in meters. Mesh is based on inner surface in Figure 8.16. Boundary potential is nonzero only at the pole tips. XY -plane is shown.	131
8.19 Average difference between theory and PISCS for the sextupole magnet in XY -plane. Potential (left) and B_y (right) shown. Overall, the difference is shrinking with element order.	132
8.20 CAD and mesh of the elliptic cylinder with $R_x = 0.0254, R_y = 0.02, L = 0.1$ [m]. Mesh has 384 elements with $e = 3$. $V = 0$ V placed on surface.	133
8.21 ZY -plane of elliptic cylinder containing a beam, zero potential on the surface. Position is given in meters. We use $M = 384, e = 3$ in PISCS with Dirichlet conditions. Potential [V] (left) and field [V/m] (right) is shown. The surface is shown by the solid lines. This depicts the beam frame, where the charges are essentially static. We can see the field and potential mostly due to the beam die out rapidly away from the charges.	134
8.22 Single emission tip on substrate, CAD (left) and example mesh with $e = 4$ (right). The enclosing walls are not shown. The mesh had $M = 8158, e = 4$. Nanoemitter is 2.13 nm tall and ≤ 0.205 nm wide. The tip is spherical with $R = 0.065$ nm.	135

Figure	Page
8.23 <i>ZY</i> -plane of the nanoemitter without and with electrons. Potential [V] (left) and field [V/m] (right) is shown. The field is strongly modified due to the geometry at the emitter tip, though the direction and strength is off due to the near boundary instability in the BEM. We see the presence of electrons strongly affects the situation at the nanoscale. The emitter tip is indicated by the solid curve.	137
8.24 Absolute value of E_Y (left), E_Z (right) [V/m] of the nanoemitter without and with electrons, starting just past the emitter tip and scanned along the Z -axis [nm]. The transverse fields persist without charge, though relatively small. Without charges, E_Z rapidly approaches the parallel plate field, indicating the range of the emitter field. With charge, the transverse fields become significant, which suggests emittance growth due to space charge.	138
8.25 Comparison of tracking a 100 keV electron beam, $N = 1000$, using COSY and PISCS with an IOTA sextupole. XY (top), XA (bottom left), and YB (bottom right) shown. Positions are given in centimeters and a, b in radians. No space charge included and beam-wall interaction is negligible. The differences stem from the linear approximation and BEM error using $e = 3, M = 520$	141
8.26 Comparison of the space charge kick from method of moments and the fast multipole method. Positions are given in centimeters and a, b in radians. Comparing to COSY, the final distribution retains its shape. The beam envelope is larger due to the space charge. The FMM is also slightly stronger than the MoM.	143
8.27 Distribution after sextupole with space charge and beam-wall. Positions are given in centimeters and a, b in radians. The shape is no longer retained due to the beam-wall interaction.	144
8.28 Initial position of the beam relative to the sextupole. The beam has initially zero emittance, is uniformly distributed, and centered on the Y -axis.	145
8.29 7 TeV proton beam phase space after the sextupole. The initial beam center is at $X = 0, Y = 1.5$ cm near the top pole tip at $X = 0, Y = 2.5$ cm. Initial beam has zero emittance and radius of 0.05 cm. Final beam with only the sextupole (COSY) and with space charge (SC-FMM) are almost the same as initial.	146
8.30 Initial and final beam with $N = 20000$ after the periodic FODO cell using calculation order 8. No collective effects are included. Position is given in millimeters and a, b in radians.	149

Figure	Page
8.31 Electron beam with $N = 20000$ including collective effects at 0.3 A. MoM and FMM only include space charge and FMM+PISCS indicates space charge plus beam-wall. This is at the maximum current before losing linear stability. Position is given in millimeters and a, b in radians.	150
8.32 Beam with $N = 20000$ at $I = 10$ A with collective effects. The beam is clearly divergent and unstable in both planes. The slight difference suggests space charge is the main destabilizing cause. Position is given in millimeters and a, b in radians.	151
8.33 Final emittances [m·rad] ϵ_x (left) and ϵ_y (right) versus current. The beam-wall interaction adds a slight amount, but in all cases, the emittance grows up to an order of magnitude once in the unstable region due to collective effects.	152
8.34 Symplecticity of the periodic FODO cell. The map is symplectic to machine precision without collective effects. With collective effects, the map shows increasing symplectic error with current. The error scale suggests remnant approximation error from the moment method. The beam-wall decreases the error for small current, but it is likely an artifact.	153
8.35 Trace of transfer matrix in the XA -plane (left) and YB -plane (right). The FODO starts within the stable region. We see the stability is lost due to mainly space charge.	154
8.36 Spherical aberration versus current [A] from $(x x^3), (y y^3)$ [$1/m^2$] in the transfer map. Contribution is mostly from space charge.	154
8.37 Fractional tune shift versus current [A] in the linearly stable region. The space charge begin depressing the tune around $I = 0.005$ A, reaching a maximum shift of 0.02 in x and 0.018 in y before stability is lost.	155
8.38 Dominant 3rd (left) and 4th (right) order resonances versus current [A] in the FODO cell. 3rd order shows the space charge and SC plus beam-wall behavior is comparable. 4th order shows a weak resonance without collective effects and large jump due to collective effects.	156
8.39 Dominant 5th (left) and 6th (right) order resonances versus current [A] in the FODO cell. Similarly, there is a large jump for the 6th order resonance due to collective effects. The 5th order resonance is comparable between collective effects until $\sim 0.5A$	156

8.40 Dominant 7th (left) and 8th (right) order resonances versus current [A] in the FODO cell. The dominant 7th order resonance varies significantly with the beam-wall interaction.	156
--	-----

LIST OF APPENDICES

Appendix	Page
A Derivations	181
A.1 Regularization of boundary integral equations	182
A.2 Smoothed FMM operators	185
B Numerical methods	188
B.1 Singular value decomposition	189
B.2 Runge-Kutta integration	190
B.3 Krylov solvers	191

CHAPTER 1

BEAM PHYSICS AT THE INTENSITY FRONTIER

The field of beam physics enters an exciting time. Since the early 1900s, accelerators developed tremendously and the new technology led to marvels like the electron microscope and the Large Hadron Collider (LHC). Now, accelerators contribute to our scientific and societal needs [12], such as with proton therapy [13]. Some of the applications include scientific discovery, food safety [14], energy production [15–17], and security [18, 19]. High-intensity beams promise great potential for new scientific discoveries. Several accelerator complexes exist around the world dedicated to different regimes of energy, current, and brightness [20].

Society benefits from accelerators used in various industries and fields [21, 22]. Globally, food is safely sterilized using accelerators rather than chemicals. Food irradiation is one of the most extensively researched processing technologies [14]. Hadron beams promise new medical treatments in cancer therapy [23] and compact accelerators generate X-rays for medical use [24]. Production of several modern electronics requires accelerators [15]. The electron beam irradiation alters various material properties for industry [15, 25]. Accelerator-driven, subcritical thorium reactors promise safer generation of nuclear energy [16]. National security demands new scanning and communication technologies using accelerators [18]. Several laboratories, such as Fermilab [26], JLab [27], J-PARC [28], and ORNL [29], either plan to construct, or are constructing, modern accelerator complexes for the new wave of scientific discovery. The need for high intensity becomes prominent, driven by new demands in many scientific fields [1, 2, 26]. The wide applications of beams led to the field of beam physics, dedicated to studying the intricate beam dynamics and developing higher beam

quality in accelerators of all sizes and types [30]. The applications require a wide spread of beam parameters, such as energy, current, power, brightness, and others [6]. The goal in beam physics is to optimize the beam control such that the beam quality reaches the desired level [31].

Generally speaking, we accelerate, focus, manipulate, and bend the beam to achieve and maintain the beam parameters. The acceleration increases the beam energy and the bend keeps the beam trajectory along the desired path. In this work, we primarily concentrate on transporting the beam. Acceleration cavities can be studied independently. By definition, the beam is an ensemble of particles with nearby state vectors. Simply put, we transport a contained bunch of particles for as long as possible and maintain a narrow band of parameters. To stably transport the beam, we must analyze the destabilizing factors and design systems which minimize the causes of instability [32–34]. Beamlines are the resultant systems, comprised of many components designed to transport the beam and maintain beam parameters within some tolerance.

We can classify beam dynamics into many aspects including single-particle dynamics, multi-particle dynamics, and beam-wall interactions. Single-particle dynamics describe the motion of particles due to external electromagnetic fields [6] without self-induced forces. Various beamline components generate the external fields, e.g., dipole and quadrupole magnets. The resulting Lorentz forces entail the classical equations of motion (EoM) [35], which can be expressed in Hamiltonian form. The scientific community, including beam physicists, developed many methods to solve the Hamiltonian system and associated Maxwell equations. Computationally, some examples include finite element methods [36], finite difference methods [37], particle-in-cell (PIC) [38], and symplectic integrators [35, 39–42]. The model consists of a reference particle and many beam particles with small deviations in position and momenta. In principle, the study of the deviations gives the beam behavior including the aberrations. Akin to optics, the stable trajectories follow different paths but stay within

a beam envelope around the reference trajectory [43]. One approach solves the envelope equation, which describes the beam motion using the time evolution of the maximum transverse position [30]. The next aspect includes the self interactions, e.g., between particles within a beam or between beams, and is a primary concern for intense beams. Maxwell's equations fully describe the self-induced fields.

Other approaches involve solving the Vlasov-Maxwell equation [44], which represents the beam motion by its phase space density and includes the self-induced fields via Maxwell's equations [6], or considering a frozen charged distribution [45–47] where the overall beam shape does not vary and at most the distribution scales in size. The difficulty lies in efficiently and accurately modeling the beam's motion when the self-induced forces vary as the particles move. The prior stages assume the beam moves in free space, i.e., open boundary conditions. The beam-wall interaction considers the feedback from the surroundings on the beam or solving the equations of motion with time-dependent boundary conditions [48]. For many cases, beamline design minimizes the beam-wall interaction to negligible levels. However, the beam-wall interaction persists in all accelerators. High-intensity beams generate a strong beam-wall interaction far from ideal conditions [49, 50].

We define intensity as the beam current. Due to charged particles producing self-fields, the resultant effects limit the beam stability [30, 49]. This work focuses on two collective effects, Coulomb repulsion and beam-wall interactions. As the current density increases, Coulomb repulsion and the beam-wall interactions degrade the beam quality and limit the achievable intensity [50]. Current computational models cannot suitably model such intense beams [46, 51]. Accurate structural representation limits beam-wall modeling, particularly when the model requires widely varying length scales, i.e., between the beam and enclosure. We can model Coulomb repulsion using an overall smooth distribution, namely space charge [52]. Space charge requires a large sample size to accurately model the self-fields. We can

also model the Coulomb repulsion from the dynamics of N interacting bodies, of which the complexity conventionally scales as $\mathcal{O}(N^2)$.

To address these needs, this work describes a 3-D fast Poisson solver including arbitrary structures for simulating space charge and beam-wall interactions in the static regime [53]. We implement several techniques to reliably represent high order meshes of complicated structures. We focus on well-conditioned systems which drastically improves the convergence. We use the differential algebraic framework to extend the capability of our solver. We show how the solver is combined with the equations of motion to compute the space charge kick plus beam-wall interaction. We also extract a transfer map including these collective effects and examine some of the instabilities.

We summarize some high-intensity projects in Chapter 2 as possible areas to apply our method. We also describe the computational challenges involved in high-intensity beam simulations and highlight the advantage of our implementation in Chapter 2. We outline transfer maps in the differential algebraic (DA) framework and the maps' role in our analysis in Chapter 3 to help clarify our results and analysis. We explain the N -body acceleration using the Fast Multipole Method (FMM) in Chapter 4 to emphasize its advantage for intense beams. To explain our method, we summarize the boundary integral theory and our high-order BIE discretization in Chapters 5 and 6. We describe our implementation in Chapter 7 and two ways to include the beam-wall interaction in the beam dynamics. To measure the performance, we show results with known analytical cases and some complicated structures without general analytical solutions in Chapter 8. Finally, we demonstrate beam dynamics including collective effects in some beamline elements in Chapter 8. We also analyze the beam stability and some causes of beam loss in Chapter 8.

CHAPTER 2

HIGH-INTENSITY PROJECTS AND COMPUTATIONAL CHALLENGES

The search for new physics beyond the Standard Model has several fronts, leading to specially designed experiments around the world [20]. Since the 2014 P5 strategic report [54], the USA focuses more on the Intensity Frontier to study neutrinos and rare processes in particle and nuclear physics [55]. Figure 2.1 [1] shows the recent and planned accelerators by average beam current and energy, where the slopes indicate on-target beam power.

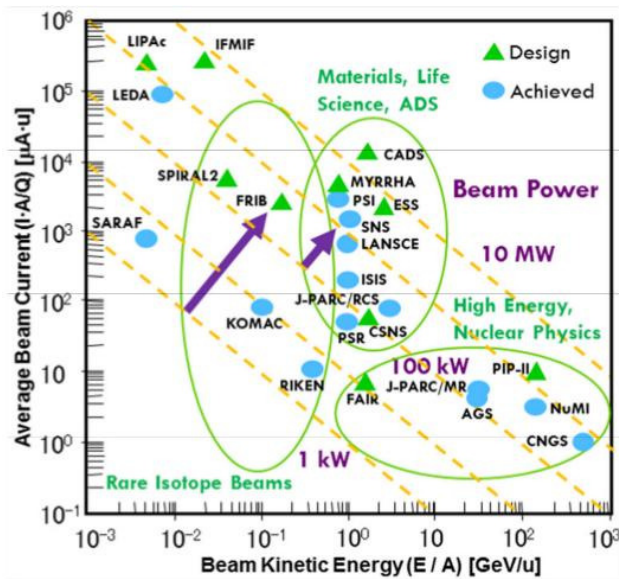


Figure 2.1: Overview of average beam current vs. energy accelerator landscape [1]. The product gives the on-target power with orders of magnitude indicated by the dashed lines.

Figure 2.1 includes accelerators for high-energy and high-intensity physics research (J-PARC, PIP-II). The Intensity Frontier experiments search for extremely rare events to explain, for instance, the matter-antimatter asymmetry in particle physics [56]. These require pushing

the current limits of accelerator technology. With the confirmation of neutrino oscillations [57, 58] and the Higgs boson [59, 60], the future studies need precise measurements within reasonable time. Neutrino studies especially need intense beams. Neutrinos interact weakly with matter, so feasible data collection requires intense neutrino beams [26]. The Deep Underground Neutrino Experiment (DUNE) is an international mega-collaboration hosted by the Sanford Underground Research Facility (SURF) in South Dakota, USA. DUNE will study neutrinos to answer some of the big questions: The origin of the matter-antimatter asymmetry and the stability of matter [61]. This chapter overviews the program at Fermilab to provide sufficient neutrinos for DUNE. We summarize two dedicated research facilities that study the dynamics and advanced transport of high-intensity beams. We also discuss the computational challenges in simulating high-intensity beams.

2.1 Fermilab’s High-intensity Program

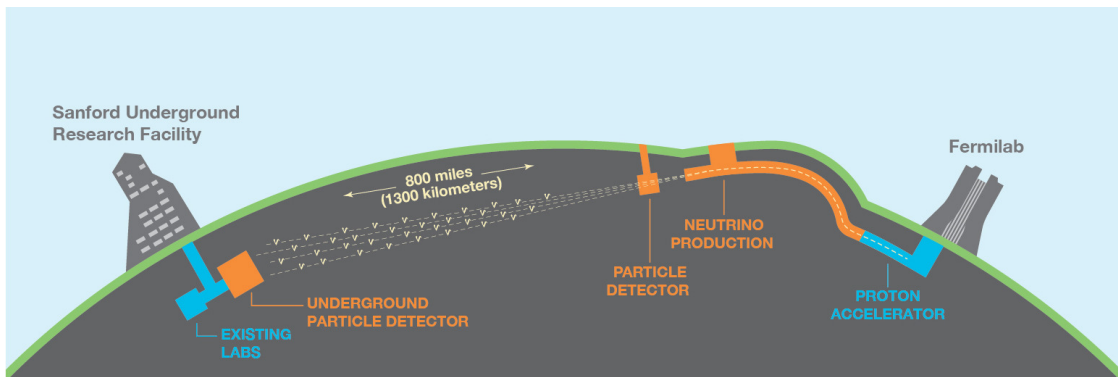


Figure 2.2: Overview of the DUNE setup [2]. Fermilab will provide a neutrino beam from the LBNF into a near-source detector at LBNF and the main DUNE detector at SURF.

Fermilab plans to provide DUNE with high-intensity neutrino beams as shown in Figure 2.2 [2]. To do so, Fermilab requires intensity upgrades to their main accelerator complex.

The upgrade project at Fermilab is called the Proton Improvement Plan (PIP) [62], with the planned PIP-II shown on Figure 2.1 near 1 MW. In the near-term, PIP requires doubling the beam power capabilities of the Fermilab Booster and Main Injector. PIP upgrades the Fermilab accelerator complex to provide a 700 kW, 120 GeV proton beam for the Neutrinos at the Main Injector (NuMI) beam and a 8 GeV Booster for low energy neutrino experiments. Currently in design phase, the immediate goal of the second phase, PIP-II [63], is to upgrade Fermilab's facilities and achieve 1.2 MW using 120 GeV protons by, e.g., introducing a continuous wave (CW) superconducting linear accelerator in the Booster. In the long run, PIP will lead to establishing a high-intensity proton facility for multi-MW experiments (PIP-III). PIP-II will also provide the needed beams to support existing experiments such as g-2 and Mu2e [26]. Sources of beam instabilities rapidly grow with intensity, leading to significant beam loss. The next generation of lattice designs will require compensating strong collective effects, which include nonlinearities. To achieve the beam power goals of PIP, Fermilab designed the Integrable Optics Test Accelerator to study intense beam transport and established the Fermilab Accelerator Science and Technology facility to generate the intense beams.

The Fermilab Accelerator Science and Technology (FAST) facility establishes a superconducting radiofrequency linear accelerator at Fermilab, in part to accomplish the goals of PIP-III [64]. FAST is designed to inject either a 150 MeV electron beam or 2.5 MeV proton beam into the Integrable Optics Test Accelerator (IOTA), an innovative ring for testing nonlinear integrable optics lattices based on present technology. Figure 2.3 shows the schematic of the FAST beamline including IOTA [3].

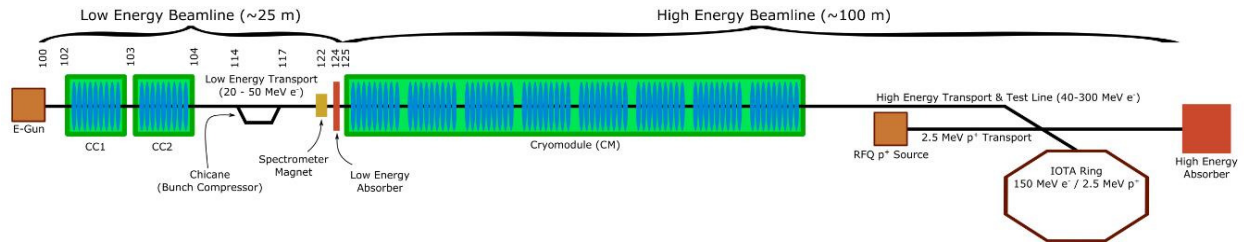


Figure 2.3: Schematic of the FAST beamline, connected to IOTA [3]. The beamline provides either 150 MeV electrons or 2.5 MeV protons.

Beam instabilities such as beam halos or space charge tune shifts are naturally nonlinear effects [50]. As the current density increases, these instabilities grow stronger and cause beam loss, which limits the achievable beam intensity [65]. The Intensity Frontier experiments require methods to suppress such effects. Integrable optics hold great promise to extend the conventional intensity limit. A schematic of IOTA is shown in Figure 2.4 [3].

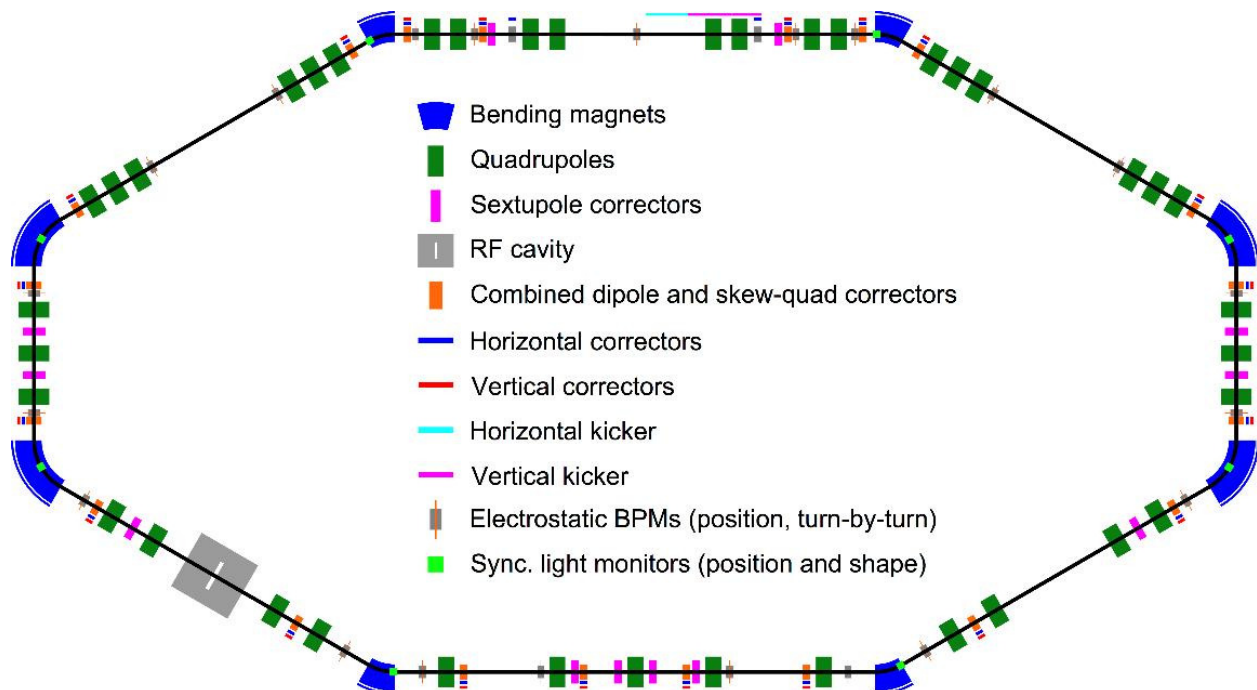


Figure 2.4: Schematic of the IOTA ring [3]. The total length is 40 m with 2 sections for testing the nonlinear magnets.

IOTA is designed for two main goals: space charge compensation [66] and proof-of-principle for nonlinear integrable optics [67]. In the future, it will be a dedicated USA facility for high intensity beam studies [3]. The storage ring will accept CW 150 MeV electrons or 2.5 MeV protons with precision control and significant lattice flexibility.

IOTA features a large pipe aperture to accommodate large amplitude beam oscillations and lattice flexibility. As shown in figure 2.4 and described in [3], the ring is formed by 6 long straight sections, of which 4 are experiment sites, and 2 short sections for dispersion suppression and chromaticity correction. The ring circumference is 40 m, compact enough for a small room. Injection occurs along the right-half of the top long straight section in figure 2.4. IOTA has 8 main bending dipoles, four 30° on the long sections and four 60° bends on the short sections. In the lower-left long section, an RF cavity will maintain the beam energy. Experiments are at the remaining four long sections.

2.2 UMER

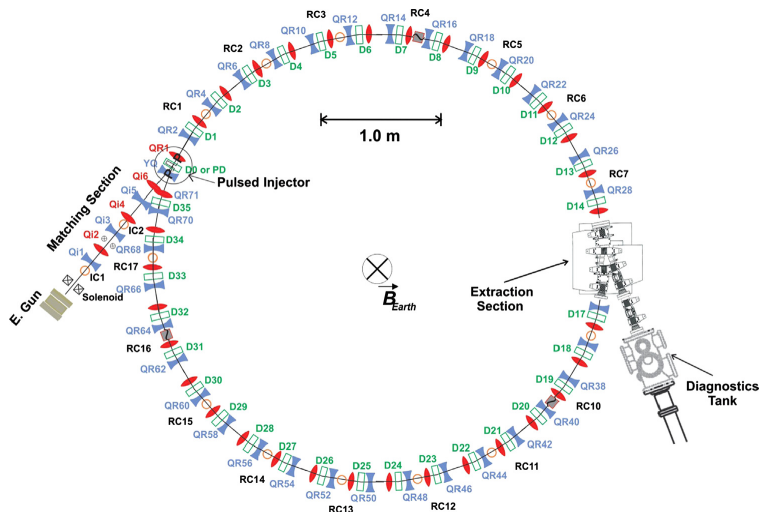


Figure 2.5: Overview of UMER layout [4]. UMER has a circumference of 11.52 m, comprised of either 36 FODO periods (shown) or 18 FODO periods [5]. The ring is injected with space-charge dominated, 10 keV electron beams. A planned extraction line on the right is shown.

The University of Maryland electron ring (UMER) is also dedicated to studying intense beams. The University of Maryland built UMER to study space-charge dominated electron beams [4, 68, 69] and, recently, nonlinear quasi-integrable optics [5, 70]. The ring layout is shown in Figure 2.5 and more details on operations may be found in Ref. [4]. Figure 2.5 shows UMER consisting of 36 FODO cells with a period of 32 cm, giving a circumference of 11.52 m. In Ref. [5], the authors also give an alternative lattice of 18 FODO periods. UMER uses 10 keV electrons at varying currents and has studied beam resonances, halos, and space charge tune shifts [4, 5]. UMER may also use a low current electron beam configuration to study space charge effects in high-energy proton and ion machines [70]. UMER has operated and characterized up to 100 mA electron beams [4] and possibly will allow for a 60 μ A electron beam for studies below the Laslett tune shift limit [5]. UMER operates the

electron beam at such low energies that the geomagnetic field accounts for 15% of the vertical bend [46]. The injection Y-section shown in Figure 2.5 has a significant additional transverse effect on the electron beam due to the odd geometry and low energy. In Ref. [46], the author approximates the kick using image charge due to a cylindrical pipe. While the image charge approach captures the overall effect quite well [46,49,71], the simplified geometry and imposed symmetry may smooth out some of the interesting features due to, e.g., corners or edges. Our implementation can consider structures with complicated geometries. We discuss some cases in Chapter 8. Computational simulation provides the most practical check for our physical model and component design. However, high-intensity beam physics challenges our current computational capabilities.

2.3 Computational challenges

A beam is a group of N charged particles with directed motion along the longitudinal axis [30]. The group has approximately the same kinetic energy and momentum and occupies a relatively small region in phase space. We study the charged particle dynamics due to external, self-induced, and environmental forces. Eq. (2.1) gives the general equations of motion due to the Lorentz force.

$$\frac{d\mathbf{p}}{dt} = \mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \quad (2.1)$$

In many cases of interest, quantum effects are negligible, so we consider a classical particle picture. External forces from beamline components cause the single-particle dynamics described by Eq. (2.1). Many of the beamline components can be expressed analytically and the single-particle dynamics are well-understood [6,30,72]. Multi-particle dynamics include

the self-induced forces. Maxwell's equations in (2.2) describe the self-induced electric, \mathbf{E} , and magnetic, \mathbf{B} , fields.

$$\begin{aligned}\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0}, & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, \\ \nabla \cdot \mathbf{B} &= 0, & \nabla \times \mathbf{B} &= \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right).\end{aligned}\tag{2.2}$$

Intense beams typically contain billions ($N \sim 10^9$) of particles [6, 34]. We concentrate on a single beam in this work. Working in the beam's frame of reference gives a quasi-static picture, i.e. negligible self-induced magnetic field. Therefore, we are able to reduce Maxwell's equations to the Poisson boundary value problem (BVP) in (2.3).

$$\begin{aligned}\nabla^2 \Phi(\mathbf{x}) &= \frac{\rho}{\varepsilon_0}, & \mathbf{x} &\in \Omega. \\ \Phi(\mathbf{x}) &= g(\mathbf{x}) \quad \text{or} \quad \frac{\partial \Phi}{\partial n} = h(\mathbf{x}), & \mathbf{x} &\in \Gamma.\end{aligned}\tag{2.3}$$

Ω denotes a closed volume and Γ is the surface enclosing this volume. The Poisson BVP can be decomposed into a free-space Poisson-like partial differential equation (PDE), and a Laplace BVP describing the surface effect due to modified boundary conditions on Γ [73, 74]. The Poisson-like problem with potential $\phi(\mathbf{x})$ is well-studied [35] and has solution given by,

$$\phi(\mathbf{x}) = \frac{1}{4\pi\varepsilon_0} \int_V \frac{\rho(\mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|} dV.\tag{2.4}$$

This gives the direct self-interaction by the particles. Modeling the beam-wall interaction amounts to solving the modified Laplace equation when $\mathbf{x} \in \Gamma$. The motion due to self-induced forces is a classic example of the N -body problem, where we study the dynamics of N interacting bodies. The direct summation of the influences on each body leads to $\mathcal{O}(N^2)$ complexity. When considering $N \sim 10^6$, the operation count reaches 10^{12} . Current computational power would take extraordinarily long for high-intensity beam physics. The

scientific community developed many techniques to overcome this limitation. We primarily apply the fast multipole method. Greengard and Rokhlin developed the fast multipole method (FMM) in 1987, which achieves linear scaling, $\mathcal{O}(N)$ [75, 76]. The challenges addressed in this work involve numerically solving the 3-D Poisson BVP with $N \geq 10^6$ and complicated structures in a dynamical setting. As of this work, there is no software capable of efficiently solving this problem.

2.4 Modeling space charge

The repulsive forces between like particles tends to spread the beam, potentially causing emittance growth, scraping, and halos [6, 34]. We can describe this repulsion by intra-beam scattering and an average outward force due to a smooth charged distribution [30]. We refer to the smooth part as space charge. The moving beam generates a magnetic field, which weakens the repulsive force. The net self-force is proportional to $\frac{1}{\gamma^2}$ [30], so space charge is small in high-energy accelerators. To model space charge in 3-D, we solve the open-boundary Poisson equation for a point charge, q , at \mathbf{y} given in (2.5).

$$\begin{aligned} \nabla^2 \phi(\mathbf{x}) &= \frac{q\delta(\mathbf{y})}{\epsilon_0}, \\ \mathbf{x} \in V \quad \text{and} \quad \mathbf{y} \in V. \end{aligned} \tag{2.5}$$

The well-known solution for ϕ with N free charges at \mathbf{y}_i , $i = 1, \dots, N$ is given by (2.6) from (2.4).

$$\phi(\mathbf{x}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{q_i}{\|\mathbf{x} - \mathbf{y}_i\|}. \tag{2.6}$$

Many current solvers use particle-in-cell (PIC) methods [38]. In PIC methods, macro-particles represent samples of the beam distribution. PIC methods refine the space into

a grid and represent nearby macro-particles by an interpolated density distribution at each grid node [77]. The field is computed between each node by solving the Poisson equation using, e.g., finite difference, then interpolated back to the closest macro-particles. Accuracy of PIC methods is limited by numerical noise, interpolation error, and stability due to time step and mesh size [78]. Alternatively, Eq. (2.6) can be significantly accelerated using the fast multipole method (FMM) [75]. Instead of macro-particles, the FMM subdivides the region and represents a closely-packed group of particles using the multipole expansion of their scalar potential around an expansion center. The multipole is translated near to a target point, most importantly, without additional loss of accuracy. Within a group or with nearby particles, the interactions are directly computed as in (2.6). Many variations of the FMM were developed in recent years [79–81]. In particular, we use the differential algebraic (DA) Cartesian form described in Ref. [7]. Chapter 4 and Ref. [7] give further details on this implementation. With the FMM, Eq. (2.6) is approximated for a desired accuracy up to machine precision.

2.5 The beam-wall interaction

The beam-wall interaction originates from the electromagnetic feedback off the beamline components [6, 49, 71, 82]. The hardware constructing the beamline is generally made of metals which are affected by the electromagnetic fields due to the beam. For instance, a quadrupole focuser uses iron to form the poles and the vacuum pipe housing the beam is made of stainless steel. For most beams, the feedback is negligible. To achieve high intensity, the current must be orders of magnitude higher than before. The generated electromagnetic fields cause significant feedback, or wakefields, in this regime and may lead to instability [48, 83]. Of particular interest in this work is the force on the beam in the static regime. In Ref. [49],

the author approximates the linearized transverse motion at beam injection inside many geometries using the image charge approach. For illustration of the ideal case, we simplify the picture to a perfectly conducting, smooth cylinder surrounding a particle bunch [49,65]. The particle bunch generates an electric field, which causes the wall surface to adjust such that the field does not escape. We imagine image charges somewhere outside the cylinder which cancel the fields at the walls. The resulting image force push the particle bunch. This example shows three conditions required to cancel the image forces:

1. Cylindrical symmetry
2. Perfect conductor
3. Smooth walls

These conditions dictate beamline designs. The image charge approach approximates the major features and is extensively used [6,30,50]. Real beamlines have discontinuities, finite conductivity, and surface defects. The problem's geometry grows quite complex. To simulate the static forces due to the feedback, we need a highly accurate geometric model combined with an efficient Laplace solver for ψ in (2.7). The boundary conditions are modified by the charge distribution from (2.4).

$$\nabla^2\psi(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \quad (2.7)$$

$$\left. \begin{aligned} \psi(\mathbf{x}) &= \tilde{g}(\mathbf{x}) = g(\mathbf{x}) - \phi(\mathbf{x}) \\ \frac{\partial\psi}{\partial n}(\mathbf{x}) &= \tilde{h}(\mathbf{x}) = h(\mathbf{x}) - \frac{\partial\phi}{\partial n}(\mathbf{x}) \end{aligned} \right\}, \quad \mathbf{x} \in \Gamma. \quad (2.8)$$

Numerically, Eq. (2.7) with boundary conditions (2.8) can be solved by discretizing Ω into smaller regions where the PDE is solvable as done with the finite element method (FEM) [36]. However, this requires solving (2.7) in possibly empty or wastefully large 3-D regions. All the dynamics of interest occur in the beam region. If the beam waist is ~ 0.1 mm and the

beamline cross-section is ~ 1 cm, there is significant extra computation. Eq. (2.7) can be reformulated as an integral equation like in (2.4) using Green's identity, shown in (2.9) [73]. This forms the basis of the boundary element method (BEM) and is further explained in Chapters 5 and 6.

$$\int_{\Omega} u \nabla^2 v - v \nabla^2 u \, d\Omega = \oint_{\Gamma} u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \, d\Gamma. \quad (2.9)$$

Applying (2.9) with $u = \psi$, $v = G$ and simplifying using (2.7), $\nabla^2 G = \delta(\mathbf{x})$, we get (2.10).

$$\begin{aligned} \int_{\Omega} \psi \nabla^2 G - G \nabla^2 \psi \, d\Omega &= \oint_{\Gamma} \psi \frac{\partial G}{\partial n} - G \frac{\partial \psi}{\partial n} \, d\Gamma, \\ \Rightarrow \psi(\mathbf{x}) &= \oint_{\Gamma} \psi(\mathbf{y}) \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{y}) - G(\mathbf{x}, \mathbf{y}) \frac{\partial \psi}{\partial n}(\mathbf{y}) \, d\Gamma(\mathbf{y}), \\ \Leftrightarrow \psi(\mathbf{x}) &= \oint_{\Gamma} \tilde{g}(\mathbf{y}) \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{y}) - G(\mathbf{x}, \mathbf{y}) \tilde{h}(\mathbf{y}) \, d\Gamma(\mathbf{y}). \end{aligned} \quad (2.10)$$

In principle, we may represent $\psi(\mathbf{x})$ using (2.10) if boundary values, $\tilde{g}(\mathbf{y})$ (Dirichlet), $\tilde{h}(\mathbf{y})$ (Neumann), are known. There are additional considerations on and near Γ and is discussed in Chapter 5. We may also take the ansatz that there exists a continuous charge distribution on Γ such that it reproduces the boundary conditions in (2.7) [74]. Furthermore, since the charge is freely chosen, we could represent the potential using monopoles or dipoles. In this case, we follow the model from (2.10), replacing $g(\mathbf{y})$ or $h(\mathbf{y})$ with densities $\sigma(\mathbf{y})$ or $\eta(\mathbf{y})$. Since matching either $\psi(\mathbf{x})$ or $\frac{\partial \psi}{\partial n}(\mathbf{x})$ as $\mathbf{x} \rightarrow \Gamma$ is sufficient, we only need one layer of charges or dipoles. In this way, we form what's called the single and double layer potentials, with the respective integral operators S and D defined in (2.11) and (2.12).

$$\psi(\mathbf{x}) = S(\sigma) = \oint_{\Gamma} G(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) \, d\Gamma(\mathbf{y}), \quad (2.11)$$

$$\psi(\mathbf{x}) = D(\eta) = \oint_{\Gamma} \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{y}) \eta(\mathbf{y}) \, d\Gamma(\mathbf{y}). \quad (2.12)$$

There are several ways to discretize the integration in (2.10), (2.11), (2.12) to form a matrix equation. We chose Nyström discretization [84]. Nyström discretization approximates the integral as a sum using quadrature. As long as the quadrature weights and nodes are known, the integral is discretized such that a linear system for the density is formed. Regardless of choice, the $M \times M$ matrix is densely populated and numerical solvers require $\mathcal{O}(M^3)$ operations at worst, partially due to the matrix-vector product [73]. For instance, an iterative solver starts with an initial guess for the unknown and iteratively approaches the exact solution through information from the matrix action on each successive guess, i.e., the matrix-vector product. Krylov solvers are a type of iterative solver we use and is briefly described in Appendix B.3. The FMM can accelerate the matrix-vector product in an iterative solver, scaling like $\mathcal{O}(M)$ [73], which leads to the fast-multipole-accelerated boundary element method (FMBEM). Once the boundary conditions are matched, the FMM is again used to accelerate the evaluation of (2.10).

To integrate over Γ , we must connect the solver with a highly accurate representation of the surface. Computer-aided design (CAD) formats are widely used in practice for component design [85]. The surface is discretized into a mesh of M triangular elements, Γ_j . Flat panel elements may be used, but smaller elements are necessary to achieve an acceptable accuracy. Instead, we improve the accuracy using curved elements, requiring overall fewer elements for acceptable accuracy [86]. There are several ways to represent a curved element, but we chose a general parametric polynomial mapping to a unit triangle. We define an element order to indicate the highest polynomial order of the mapping and integrate each

element over the unit triangle. This is further explained in Chapter 6. The integral equations become as shown in (2.13), where M is the number of elements.

$$\begin{aligned}\psi(\mathbf{x}) &= \sum_{j=1}^M \int_{\Gamma_j} G(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) \, d\Gamma_j(\mathbf{y}), \\ \psi(\mathbf{x}) &= \sum_{j=1}^M \int_{\Gamma_j} \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{y}) \eta(\mathbf{y}) \, d\Gamma_j(\mathbf{y}).\end{aligned}\tag{2.13}$$

For c iterations in the matrix solver, the Laplace solver scales like $c\mathcal{O}(M)$ and the combined Poisson solver should scale like $\mathcal{O}(N) + c\mathcal{O}(M) \approx \mathcal{O}(N)$, where $N \gg M$ in the case for beam physics. As a bonus, we may separately solve Laplace problems using the BEM as if there is no beam, taking $\rho(\mathbf{x}) = 0$ in (2.3). This provides an accurate method to, for instance, check against a real magnet design. We may also isolate the beam-wall interaction by examining the modification itself. The important factors with respect to the solver are the boundary conditions and the structural representation. As mentioned earlier, either Dirichlet ($\psi(\mathbf{y})$) or Neumann ($\frac{\partial \psi}{\partial n}(\mathbf{y})$) type is sufficient. Note, there are additional considerations for matching the radiation conditions to solve the exterior problem [74]. We focus on interior problems in this work. We outlined the computational challenges for simulating high-intensity beams and mentioned some approaches. We use the strategy from the FMM [75], which accelerates our free-space Poisson solver and modified Laplace solver. To allow modeling in complicated geometries, we implement the FM-accelerated BEM to solve the Laplace equation and describe the theory and implementation in Chapters 5, 6, and 7.

CHAPTER 3

MAPS AND ANALYSIS

Generally, beamlines are beam transport devices. In practice, we first design systems with the first order map (or transfer matrix) based on the linearized equations of motion [30]. Nonlinear effects are effectively small and approximated as higher order perturbations. Since the beam is closely packed by definition, the transfer map is well described by Taylor expansions of the solution to the equations of motion. In this chapter, we give the equations of motion, explain the formalism behind transfer maps, and describe some important quantities for our analysis. In particular, we describe and apply the differential algebraic methods for the analysis of beamlines.

3.1 Equations of motion

We work with the equations of motion in optical coordinates [6, 43]. The coordinate system is defined using either time t or path length s . Consider $\mathbf{r}_0(s)$ to be the reference curve. At any point s , we define the coordinate axes by $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}})$ where s follows the beam pipe, $\hat{\mathbf{x}}$ gives the horizontal axis, and $\hat{\mathbf{y}}$ gives the vertical. The position in Cartesian coordinates is given by $\mathbf{r} = \mathbf{r}_0(s) + x\hat{\mathbf{x}} + y\hat{\mathbf{y}}$. The scaled transverse momenta with reference momentum p_0 are the angles $a = \frac{p_x}{p_0}, b = \frac{p_y}{p_0}$. We also define length of flight $l = v_0(t - t_0)$ with reference velocity v_0 and time t_0 and relative energy difference $\delta = \frac{E - E_0}{E_0}$ with reference energy E_0 . In Cartesian coordinates, any particle with position \mathbf{r} and momentum \mathbf{p} is

defined by $\mathbf{z}_r = (r_1, p_1, r_2, p_2, r_3, p_3)$. In optical coordinates, we have $\mathbf{z} = (x, a, y, b, l, \delta)$, which implicitly depends on s .

The equations of motion describe the relation between the initial state \mathbf{z}_i and final state \mathbf{z}_f . In the optical coordinates, the equations of motion due to the Lorentz force are given in (3.1), derived from (2.1). E is the energy, and $E_{x,y}, B_{x,y,z}$ are the electric and magnetic field components.

$$\begin{aligned}
\eta &\equiv \gamma - 1 = \frac{E - qV(x, y, s)}{mc^2}, \\
\chi_{e0} &\equiv \frac{p_0 v_0}{q}, \\
\chi_{m0} &\equiv \frac{p_0}{q}, \\
\frac{p_0}{p_s} &= \left(\frac{\eta(2 + \eta)}{\eta_0(2 + \eta_0)} \frac{m^2}{m_0^2} - a^2 - b^2 \right)^{-\frac{1}{2}}, \\
x' &= a(1 + hx) \frac{p_0}{p_s}, \\
a' &= \left[\frac{1 + \eta}{1 + \eta_0} \frac{p_0}{p_s} \frac{E_x}{\chi_{e0}} + b \frac{B_z}{\chi_{m0}} \frac{p_0}{p_s} - \frac{B_y}{\chi_{m0}} \right] (1 + hx) + h \frac{p_s}{p_0}, \\
y' &= b(1 + hx) \frac{p_0}{p_s}, \\
b' &= \left[\frac{1 + \eta}{1 + \eta_0} \frac{p_0}{p_s} \frac{E_y}{\chi_{e0}} - a \frac{B_z}{\chi_{m0}} \frac{p_0}{p_s} + \frac{B_x}{\chi_{m0}} \right] (1 + hx), \\
l' &= \left[(1 + hx) \frac{1 + \eta}{1 + \eta_0} \frac{p_0}{p_s} - 1 \right] \frac{k}{v_0}, \\
\delta' &= 0.
\end{aligned} \tag{3.1}$$

Defined in (3.1), χ_e, χ_m are the electric and magnetic rigidities, η is the kinetic relativistic factor, $h = \frac{1}{R}$ is the inverse radius of curvature, and $\frac{p_0}{p_s}$ is reference over longitudinal momentum. The solutions to the system in (3.1) define a transfer map, which are used for design, tracking, and analysis. We briefly describe how to solve the equations of motion using differential algebraic methods to obtain Taylor expansions of the transfer maps.

3.2 Maps and DA

The propagation of an ensemble of states may be succinctly described as a transfer map [43] from initial conditions \mathbf{z}_i and parameters ζ .

$$\mathbf{z}_f = \mathcal{M}(\mathbf{z}_i, \zeta).$$

The map \mathcal{M} captures the flow of $\mathbf{z}(s)$. For example, $x_2 = x_1 + ma_1 = x_1 + (x|a)a_1$, defines a linear flow in x and $(x|a)$ would be an element in \mathcal{M} . The map can be composed and inverted [43]. Using composition, we concatenate maps. Clearly,

$$\mathcal{M}_{t+\Delta t} = \mathcal{M}_t \circ \mathcal{M}_{\Delta t}.$$

Concatenation implies any map can be composed of several maps. We assume the maps are weakly nonlinear and $\mathcal{M} = \mathbf{M} + \mathcal{M}_n$, where \mathbf{M} is the linear matrix and \mathcal{M}_n is the nonlinear remainder. The maps are required to preserve the origin [43], i.e. the origin is always \mathbf{r}_0 . Since we work with nearby state vectors, we may represent maps using Taylor expansions around the origin. These are efficiently computed by differential algebraic methods.

Differential algebra (DA) allows computations of numerical derivatives to machine precision [43, 87, 88]. The algebra may be defined for arbitrary order n and number of variables v , denoted ${}_nD_v$. For example, using ${}_1D_1$, set members are defined by $(q_0, q_1) \equiv q_0 + \delta q_1$, where $q_0, q_1 \in \mathbb{R}$ and infinitesimal δ . This is also called a DA vector. The operations forming the

algebra are given below with scalars t, n . We also define the derivative operator ∂ to form the differential algebra $({}_1D_1, \partial)$.

$$\begin{aligned}(q_0, q_1) + (r_0, r_1) &= (q_0 + r_0, q_1 + r_1), \\ t(q_0, q_1) &= (tq_0, tq_1), \\ (q_0, q_1) \cdot (r_0, r_1) &= (q_0r_0, q_0r_1 + q_1r_0), \\ (q_0, q_1)^n &= (q_0^n, nq_0^{n-1}q_1), \\ \partial(q_0, q_1) &\equiv (0, q_1).\end{aligned}$$

Similarly, we define an antiderivative operator, $\partial^{-1}(q_0, q_1) \equiv (c, q_0, \frac{q_1}{2})$, with arbitrary constant c [43, 46]. Evaluating a function with a DA vector $(x, 1)$ automatically gives its first order Taylor expansion.

$$f(x + \delta) = f(x) + \delta f'(x).$$

These operations may be extended to any order and number of variables [43]. Clearly, we compute the Taylor expansion for any finite function using DA variables since δ is infinitesimally small.

DA methods are particularly efficient with differential equations. Consider an ODE reformulated as follows.

$$\begin{aligned}\frac{\partial \mathbf{z}}{\partial s} &= \mathbf{f}(\mathbf{z}, s), \\ \Rightarrow \mathbf{z} &= \mathcal{M}(\mathbf{z}) = \mathbf{z}_0 + \int_{s_0}^s \mathbf{f}(\mathbf{z}, s') ds' .\end{aligned}$$

DA methods use ∂_{v+1}^{-1} to transform this into an arithmetic problem in ${}_n D_{v+1}$. In fact, this iteration leads to a unique fixed point in $n + 1$ iterations only involving arithmetic operations [43], reducing computational complexity significantly.

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{0}, \\ \mathbf{z}_{k+1} &= \mathcal{M}(\mathbf{z}_k), \quad k = 1, \dots, n + 1. \end{aligned}$$

Using fixed point iteration and DA variables, we integrate the optical equations of motion. Techniques applying the DA methods have been implemented in COSY Infinity [89], which forms the backbone of our implementation. COSY Infinity includes a beam physics package, where the common transport elements are solved to any prescribed order. The resulting transfer map is represented by a Taylor expansion where the DA variables are the initial optical coordinates [46]. The map is described using the indices $\mathbf{i} = (i_1, i_2, i_3, i_4, i_5, i_6)$ for the exponent in each DA variable, the final optical coordinates $v = x, a, y, b, l$, or δ , and the Taylor coefficients ${}_v C_{\mathbf{i}}$.

$$\mathcal{M} = \begin{pmatrix} \sum_{\mathbf{i}} {}_x C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \\ \sum_{\mathbf{i}} {}_a C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \\ \sum_{\mathbf{i}} {}_y C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \\ \sum_{\mathbf{i}} {}_b C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \\ \sum_{\mathbf{i}} {}_l C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \\ \sum_{\mathbf{i}} {}_\delta C_{\mathbf{i}} x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6} \end{pmatrix}.$$

Alternatively, only the Taylor coefficients ${}_v C_{\mathbf{i}}$ are written using parenthetical notation ${}_v C_{\mathbf{i}} \equiv ({}_v | x^{i_1} a^{i_2} y^{i_3} b^{i_4} l^{i_5} \delta^{i_6})$, leaving the corresponding DA variables implicit. Then, the Taylor coefficients are called the map elements. We use parenthetical notation for the rest of this work. Figure 3.1 shows a 2-D example of the COSY representation using a FODO cell. This example cell is comprised of drift, focusing quad, drift, defocusing quad, and drift, which

provides overall transverse focusing. The FODO cell is one of the most common transport elements in any beamline [30].

-0.101258231	-5.15074168	0.00000000	0.00000000	100000
0.192156166	-0.101258231	0.00000000	0.00000000	010000
0.00000000	0.00000000	-0.101258231	-0.875273266	001000
0.00000000	0.00000000	1.13078602	-0.101258231	000100
-26.6713942	-64.1800089	0.00000000	0.00000000	300000
-9.39411143	-32.4595182	0.00000000	0.00000000	210000
-1.56192942	-8.45923612	0.00000000	0.00000000	120000
-0.137291092	-1.11200924	0.00000000	0.00000000	030000
0.00000000	0.00000000	-9.87586672	12.0186913	201000
0.00000000	0.00000000	-5.38708955	3.79289177	111000
0.00000000	0.00000000	-1.05840245	0.522939669	021000
0.00000000	0.00000000	11.4495858	-25.3711611	200100
0.00000000	0.00000000	3.58320532	-5.99633233	110100
0.00000000	0.00000000	0.497586116	-0.480327188	020100
-2.15303594	-12.1339113	0.00000000	0.00000000	102000
-0.589091484	-4.78617458	0.00000000	0.00000000	012000
5.21776198	16.7318101	0.00000000	0.00000000	101100
0.817379824	4.60851919	0.00000000	0.00000000	011100
-5.61498170	-13.7404217	0.00000000	0.00000000	100200
-0.666500396	-2.21316826	0.00000000	0.00000000	010200
0.00000000	0.00000000	-5.47802968	2.57361371	003000
0.00000000	0.00000000	7.41224959	-5.71596514	002100
0.00000000	0.00000000	-5.52613175	8.65220322	001200
0.00000000	0.00000000	2.76581643	-6.45770896	000300

Figure 3.1: COSY 3^{rd} -order 2-D representation of a FODO cell. The columns contain the Taylor coefficients for the expansions of x_f, a_f, y_f, b_f and the exponent in each DA variable. The DA variables represent the initial positions and angles, x_i, a_i, y_i, b_i . For example, ${}_a C_{(3,0,\dots,0)} = (a|x^3)$ is the element in row 5, column 2.

The transfer maps provide a concise representation of the motion. The maps can be used for tracking a set of initial conditions. More importantly, the map contains a wide breadth of information on the beamline.

3.3 Normal form analysis on phase space

Transfer maps are also used with linear accelerators and many of the following concepts apply. We concentrate on periodic systems in this work. Our cases of interest are Hamiltonian. Liouville's theorem states the phase space density is conserved in Hamiltonian

systems [6, 30]. This property allows us to study the stable beam motion using phase space. For illustration, we focus on (x, a) phase space using the transfer matrix. We assume linear stability, which restricts the determinant $D = 1$ and the trace $|T| < 2\sqrt{D}$ for the 2- \times -2 transfer matrix [43]. The eigenvalues are given in (3.2) [43].

$$\begin{aligned}\lambda_{1,2} &= de^{\pm i\mu}, \\ d &= \sqrt{D}, \\ \mu &= \text{Sign}((x|a)) \arccos\left(\frac{T}{2\sqrt{D}}\right).\end{aligned}\tag{3.2}$$

The transfer matrix is expressible by the Twiss parameters. We define the Twiss parameters in (3.3) [43] from the transfer map. The Twiss parameters define an ellipse in phase space by (3.4). The phase space ellipse is shown in Figure 3.2 [6].

$$\begin{aligned}\alpha &= \frac{(x|x) - (a|a)}{2d \sin \mu}, \\ \beta &= \frac{(x|a)}{d \sin \mu}, \\ \gamma &= -\frac{(a|x)}{d \sin \mu},\end{aligned}\tag{3.3}$$

$$\beta\gamma - \alpha^2 = 1.$$

$$\gamma x^2 + 2\alpha xa + \beta a^2 = \epsilon.\tag{3.4}$$

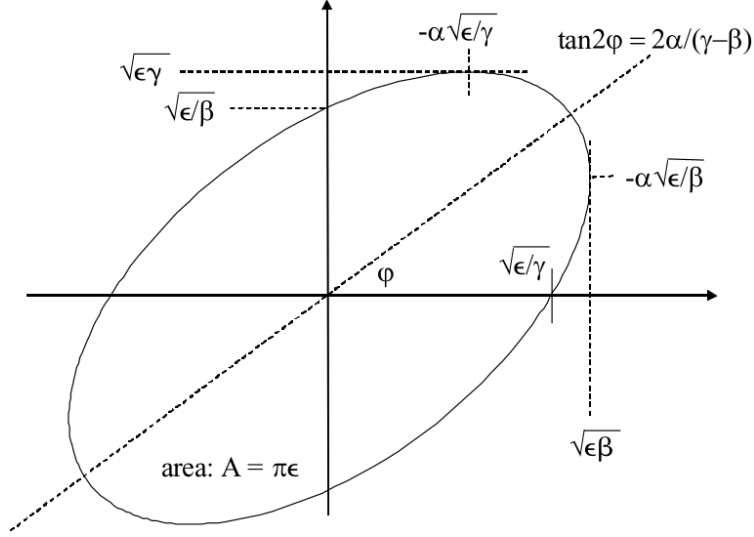


Figure 3.2: Phase space ellipse defined by Twiss parameters, α, β, γ , and emittance ϵ [6]. The horizontal axis is position and vertical is momentum.

In Figure 3.2, ϵ is called the emittance and $\pi\epsilon$ is the area of the ellipse. The emittance must be preserved in linear systems, which means the Hamiltonian motion amounts to transformations of the Twiss parameters [6, 30]. In Ref. [43], the author discusses the nonlinear normal form and the DA algorithm to approximate it, which are implemented in COSY Infinity [89].

As an example of the normal form, consider when $d = 1$ and $|\lambda_{1,2}| = 1$. The transfer matrix using the Twiss parameters becomes as shown [43].

$$\mathbf{M} = \begin{pmatrix} \cos \mu + \alpha \sin \mu & \beta \sin \mu \\ -\gamma \sin \mu & \cos \mu - \alpha \sin \mu \end{pmatrix}.$$

The transfer matrix \mathbf{M} does not change the phase space ellipse, aptly named the invariant ellipse. If \mathbf{M} is a one-turn matrix for a ring, then the ellipse must be the same after every turn [43]. The condition $|\lambda_{1,2}| = 1$ implies the eigenvectors are equal. If we form a basis

using the real and imaginary parts of the eigenvectors, we define coordinates such that the stable motion is a circle [43]. We call this the linear normal form \mathbf{N} .

$$\mathbf{N} = \begin{pmatrix} \cos \mu & \sin \mu \\ -\sin \mu & \cos \mu \end{pmatrix}.$$

In general, the normal form \mathcal{N} comes from the transformation of \mathcal{M} , where \mathcal{A} is the normalizing transform.

$$\mathcal{A} \circ \mathcal{M} \circ \mathcal{A}^{-1} = \mathcal{N}.$$

For the previous linear case, the transformation matrix \mathbf{A} is given below with $i \equiv \sqrt{-1}$ [43].

$$\mathbf{A} = \begin{pmatrix} \frac{1-i\alpha}{2i\beta} & -\frac{1}{2} \\ \frac{1+i\alpha}{2i\beta} & \frac{1}{2} \end{pmatrix}.$$

The Twiss parameters have physical meaning. β indicates the beam width and γ indicates the max divergence. α measures correlation between position and momentum. The beta function, $\beta(s)$, from (3.3) is particularly interesting for beams. When considering focusing elements, the envelope equation shows the beta function oscillates with amplitude $\sqrt{\beta}$ [30]. The frequency of oscillations is called the tune [43]. We may compute the tune by [6, 30],

$$\mu = \int_{s_0}^s \frac{1}{\beta(s')} ds'.$$

So, in practice, the beta function of a beam is tracked around a ring and integrated to compute tunes. In simulation, tracking a beam of large N is time consuming. Normal form analysis of the transfer maps almost immediately gives the tunes. As mentioned, stable motion appears as a circle in normal form coordinates. The tune is simply the angle of rotation around this circle.

The normal form shows the stable motion in the normal basis moves along a circle. The tune μ is the angular advance and is constant. If the tune hits integer multiples, the Twiss parameters rapidly grow. These are called resonances [6]. Aberrations and other nonlinear effects cause tune shifts, which may lead to resonances. The self-induced tune shifts are particularly important for high-intensity beams [6,30]. High-intensity beams induce persistent aberrations which cause tune shift and may destabilize the beam. Modeling space charge and simulating the beam is mainly to compute and compensate the space charge tune shifts [90–92]. In Ref. [49], the author derives and discusses image charge or Laslett tune shifts for vacuum chambers with different cross-sections. In Ref. [46], the author analyzes the space charge tune shift in UMER using transfer maps and normal form analysis and also discusses the image charge kick in the UMER injector. Building on this work, we include the beam-wall interaction in our analysis and combine the space charge and beam-wall kick. We have given the optical equations of motion and discussed transfer maps and normal form analysis. We briefly mention differential algebraic methods, which provide powerful tools for our analysis.

CHAPTER 4

FAST MULTIPOLE ACCELERATION

4.1 Fast Multipole Method

The fast multipole method (FMM) was originally developed in 1987 by Greengard and Rokhlin [75]. Since then, the FMM has been developed and applied for a variety of problems [93–96]. Due to its flexibility, performance, and rigorous error bounds, the FMM was ranked among the top ten algorithms of the 20th century [97]. The fast multipole method solves the problem of N interacting bodies, e.g., celestial objects. The interaction $f(\mathbf{x}, \mathbf{y})$ is known between any pair, so the simplest solution $\phi(\mathbf{x})$ for an object at \mathbf{x} is the pairwise sum due all other objects at \mathbf{y}_j where $\mathbf{x} \neq \mathbf{y}$.

$$\phi(\mathbf{x}) = \sum_{j=1}^N f(\mathbf{x}, \mathbf{y}_j).$$

For a static distribution of particles with charge q , $f(\mathbf{x}, \mathbf{y}) = G(\mathbf{x}, \mathbf{y})q = \frac{q}{4\pi\epsilon_0\|\mathbf{x}-\mathbf{y}\|}$. Computing the sum at every particle location leads to $\mathcal{O}(N^2)$ operations. The FMM instead considers interactions between clusters. The main advantage stems from the translation operators, which is described in the DA context [98]. Refs. [7, 99] describe the Cartesian DA adaptive 3-D FMM. We give an overview of the implementation in this chapter.

4.2 FMM operators

Consider the potential in Cartesian coordinates with arbitrary units such that $\frac{q_i}{4\pi\epsilon_0} = 1$. Let \mathbf{x} be the target and \mathbf{x}_i be sources. With $r \equiv \|\mathbf{x}\|$, $r_i \equiv \|\mathbf{x}_i\|$, we separate target from the sources below.

$$\begin{aligned}\phi(\mathbf{x}) &= \sum_{i=1}^N \frac{1}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}}, \\ &= \frac{1}{r} \sum_{i=1}^N \left[1 + \frac{r_i^2}{r^2} - \frac{2x_ix}{r^2} - \frac{2y_iy}{r^2} - \frac{2z_iz}{r^2} \right]^{-\frac{1}{2}}.\end{aligned}$$

The multipole expansion in DA variables, d_r, d_x, d_y, d_z is,

$$\begin{aligned}d_r &= \frac{1}{r}, & d_x &= \frac{x}{r^2}, \\ d_y &= \frac{y}{r^2}, & d_z &= \frac{z}{r^2}, \\ \Rightarrow \phi(d_x, d_y, d_z) &= d_r \sum_{i=1}^N \left[1 + r_i^2 d_r^2 - 2x_i d_x - 2y_i d_y - 2z_i d_z \right]^{-\frac{1}{2}}, \\ &= d_r \cdot \phi_M.\end{aligned}$$

ϕ_M is thus given as its Taylor expansion in the DA framework and is valid for points outside a certain radius. This is also named the far multipole expansion in Refs. [7, 98]. d_r is the square root of infinitesimals, so while it cannot be expanded, it is carried through the following operations and evaluated at the end.

Let ϕ be at the origin $(0, 0, 0)$. We can translate the expansion to (x'_0, y'_0, z'_0) by shifting the DA variables, which defines a map, \mathcal{M}_1 .

$$\begin{aligned}
d'_x &= \frac{x - x'_0}{r'^2} = \frac{x'}{r'^2} \Rightarrow d'_y = \frac{y'}{r'^2}, \quad d'_z = \frac{z'}{r'^2}, \\
r' &= \sqrt{x'^2 + y'^2 + z'^2}, \\
R &\equiv (1 + r_0'^2 d_r'^2 + 2x'_0 d'_x + 2y'_0 d'_y + 2z'_0 d'_z)^{-1}, \\
d_r &= d'_r \cdot \sqrt{R}, \\
\mathcal{M}_1 &\equiv \begin{cases} d_x = (d'_x + x'_0 d_r'^2) \cdot R \\ d_y = (d'_y + y'_0 d_r'^2) \cdot R, \\ d_z = (d'_z + z'_0 d_r'^2) \cdot R \end{cases} \\
&\Rightarrow \phi'_M = \phi_M \circ \mathcal{M}_1, \\
&\Rightarrow \phi' = \phi'_M \cdot d'_r \cdot \sqrt{R}.
\end{aligned}$$

\mathcal{M}_1 is called the M2M translation and translates a multipole expansion to another center in the valid region. Again, given ϕ_M at the origin, we translate the multipole expansion into a local expansion at (x'_0, y'_0, z'_0) close to the target, so the DA variables are simply $d'_x = x - x'_0$, etc. The map \mathcal{M}_2 is given by,

$$\begin{aligned}
\mathcal{M}_2 &\equiv \begin{cases} d_x = (x'_0 + d'_x) \cdot R \\ d_y = (y'_0 + d'_y) \cdot R, \\ d_z = (z'_0 + d'_z) \cdot R \end{cases} \\
R &\equiv ((x'_0 + d'_x)^2 + (y'_0 + d'_y)^2 + (z'_0 + d'_z)^2)^{-1}, \\
&\Rightarrow \phi_L = \phi_M \circ \mathcal{M}_2, \\
&\Rightarrow \phi = \phi_L \cdot \sqrt{R}.
\end{aligned}$$

The local expansion is valid within a certain radius containing the target in contrast to the multipole expansion. This defines the M2L translation. The M2L translation is well-known as the most compute-intensive operator in the FMM algorithm [100–102]. This implementation optimizes the M2L operator using rotation [7]. Finally, we translate a local expansion to another local center, which amounts to a shift as above. This is also known as the L2L translation. For completeness, the map \mathcal{M}_3 is given by,

$$\mathcal{M}_3 \equiv \begin{cases} d_x = x'_0 + d'_x \\ d_y = y'_0 + d'_y \\ d_z = z'_0 + d'_z \end{cases}.$$

The local expansion is evaluated using the target coordinates in the L2P operator. In the neighborhood of a target, the near neighbors cannot be expanded, so the interaction is directly computed as usual using the P2P operator or Coulomb potential. The neighborhood is the target box and all adjacent boxes (9 in 2-D, 27 in 3-D). As shown in Refs. [7, 98], the relative error in the potential behaves like $\mathcal{O}(10^{-p+1})$ while runtime behaves linearly with number of particles N . The error comes from the multipole expansion and the translations contribute no additional error.

4.3 FMM algorithm

The FMM steps may be summarized into setup, upward pass, downward pass, near neighbors, and evaluation. Figure 4.1 shows the steps in 2-D [7].

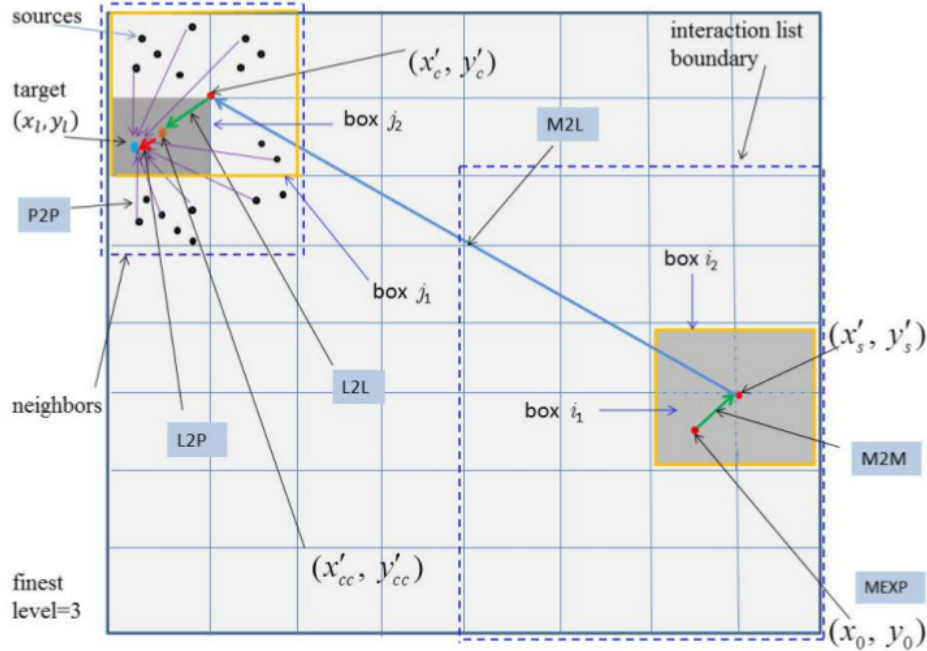


Figure 4.1: A snapshot of the FMM algorithm [7]. The multipole expansion is computed by MEXP, translated up a level using M2M, and moved near the target using M2L. The local expansion is translated down a level if necessary using L2L and evaluated at the target using L2P. The nearby sources are evaluated directly using P2P.

Starting with setup, the FMM subdivides the space into boxes (2-D squares or 3-D cubes) where the box center is the expansion center. The COSY space charge package, COMFY, applies a uniform subdivision and ends after forming a prescribed grid of boxes. The adaptive FMM used in our BEM implementation continues after the level 2 subdivision. We only give a brief summary of the octree data structure. Ref. [7] gives further details. The space is uniformly subdivided twice (level 2). Then, individual boxes are subdivided and organized into a tree structure until every box neighborhood contains at most the maximum number of neighboring particles as specified by the clustering parameter q . The set of source box interactions is organized into a C-forest containing C-trees. The set of target boxes is called the D-tree. An example depiction is given in figure 4.2, where sources and targets are the same and $N = 20, q = 10$.

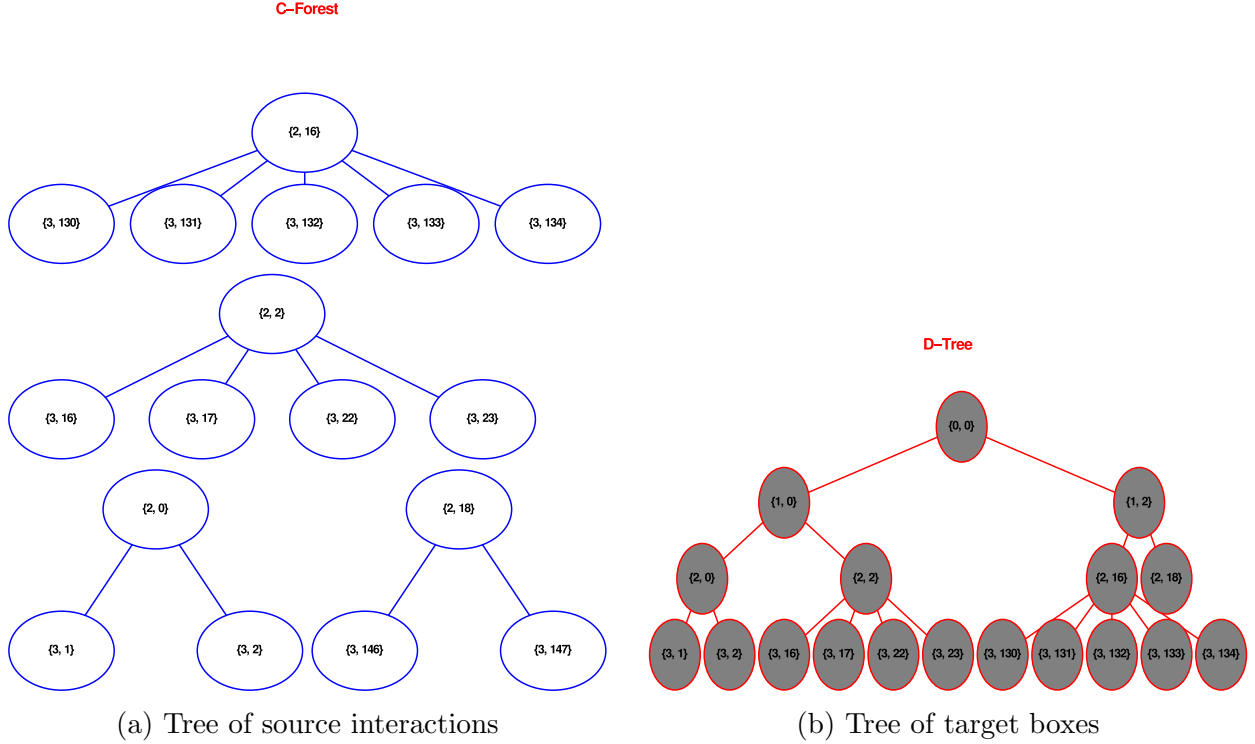


Figure 4.2: Depiction of FMM tree data structure using $N = 20$, $q = 10$, where sources and targets are the same. (a) shows the C-forest with 4 C-trees and (b) shows the D-tree.

The C-forest is used in the FMM upward pass and the D-tree is used in the downward pass. Once organized, the FMM performs the upward pass. The multipole expansion is computed for every source at the center of its box. The expansion is translated using the M2M operator up the C-tree to the highest node. In the downward pass, the M2L translates the expansion to the target neighborhood and L2L to the target node. Near neighbors are summed directly using P2P. In the evaluation step, L2P evaluates the expansion at the target position. L2P and P2P then give the potential and fields on the target. For the nonadaptive FMM, the M2M and L2L operators are not used. Otherwise, the steps are similar.

4.4 Smoothing

A common numerical issue with N-body codes occurs near singularities, i.e. collisions. The force spikes at this instant and the equations of motion give artificially large jumps. Assuming a moderate step, such collisions might appear as additional beam loss. One possible solution uses Plummer smoothing [103, 104]. Plummer smoothing was first used for gravitational potentials. A general smoothed potential and force [103] is given by,

$$\begin{aligned}\hat{\Phi}(\mathbf{x}, \lambda) &= \frac{k}{\lambda} \phi\left(\frac{\|\mathbf{x}\|}{\lambda}\right), \\ \hat{\mathbf{F}}(\mathbf{x}, \lambda) &= -\frac{k}{\lambda^2} \phi'\left(\frac{\|\mathbf{x}\|}{\lambda}\right) \frac{\mathbf{x}}{\|\mathbf{x}\|}.\end{aligned}$$

where ϕ is the kernel function, e.g., Green's function, λ is the smoothing parameter, and k is a physical constant. We implemented the smoothed Coulomb form for the 2-D FMM [105]. For target \mathbf{x} , source \mathbf{x}_0 in arbitrary units, the smoothed potential becomes,

$$\phi(\mathbf{x}) = \frac{1}{2} \log((x - x_0)^2 + (y - y_0)^2 + \lambda^2).$$

This leads to an additional DA variable, $d_\lambda = \frac{\lambda}{r^2}$ and $r^2 = \|\mathbf{x}\|^2 + \lambda^2$. In the original FMM, smoothing is impossible because the target and source could not be separated in the expansion with λ present. In the DA Cartesian form, the strategy used is quite general and extends to the 3-D FMM easily. Besides redefining r^2 , the maps gain an additional component mapping λ . Denoting R as $R_{1,\lambda}$ for the M2M operator, \mathcal{M}_1 gains,

$$d_\lambda = d'_\lambda \cdot R_{1,\lambda}.$$

In the M2L operator, $d'_\lambda = \lambda$, so we evaluate before translation. Denoting R as $R_{2,\lambda}$ for the M2L operator, \mathcal{M}_2 gains,

$$d_\lambda = \frac{\lambda}{r^2} = \lambda \cdot R_{2,\lambda}.$$

The L2L operator does not change since d_λ has been evaluated. The smoothed FMM operators are given in Appendix A.2.

To test smoothing, we used the radial spread of a long, parallel electron beam in a drift [105]. This is an analytical case where space charge is the dominant factor. Assuming collisionless dynamics, the theoretical radial spread R with drift length z and initial radius r_0 is given by [30],

$$R \approx 1 + 5.87 \times 10^{-5} \left(\frac{z}{r_0} \right)^2.$$

The beam and system parameters are reproduced in Table 4.1 [105]. These parameters give a beam with high space charge and clearly distinguishes the beam dynamics with and without collisions.

Table 4.1: Long Beam and System Parameters

Species	Electron
Energy	100 keV
Current	1 A
Spatial distribution	Uniform circular
Initial radius	1 cm
Initial emittance	0 m-rad
Drift length	0.5 m
Radial spread (Theory)	1.521

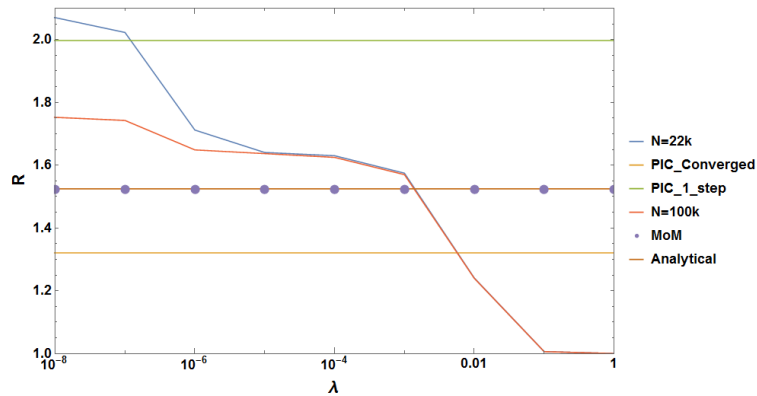
We compared the 2-D COSY space charge package, COMFY, versus the 3-D PIC code, IMPACT-T. In COMFY, we compute the transverse space charge using the moment method (MoM) or the smoothed FMM. The moment method computes the beam dynamics due to

an average repulsive force using the statistical moments of a smooth distribution [46]. The moment method smooths out collisions by definition. We show this comparing the FMM with $N = 22\text{k}, 100\text{k}$. The IMPACT-T parameters are reproduced in table 4.2 [105].

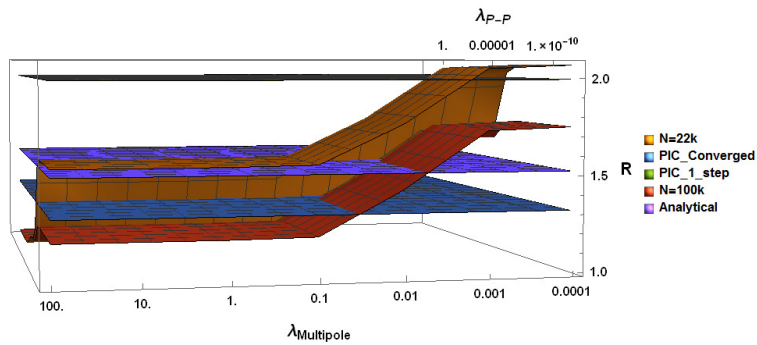
Table 4.2: PIC Simulation Conditions

Code	IMPACT-T
No. of particles	300000
Bunch length	1.03 m
No. of slices (transverse)	64
No. of slices (longitudinal)	32
i^{th} Z-slices picked	16-17
Total travel time in drift	3.1 ns
Convergent time step	0.01 ns

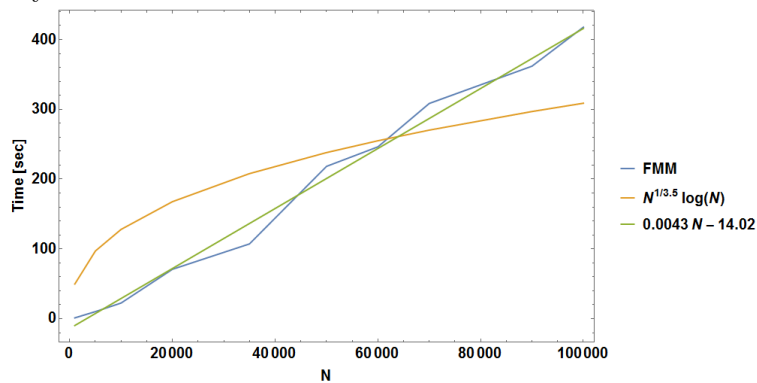
We picked the parameters in Table 4.2 for IMPACT-T to represent a long beam. We study the transverse spread using the middle Z-slices, where the longitudinal effect is minimized from symmetry. We chose the number of particles such that a transverse slice is comparable to the number of particles, N , used in COMFY. The results are shown in Figure 4.3.



(a) FMM with $\lambda_{P-P} = \lambda_{\text{Multipole}}$, MoM, and PIC vs. analytical



(b) FMM with separate smoothing, MoM, and PIC vs. analytical



(c) FMM runtimes

Figure 4.3: Comparison of radial spread from smoothed FMM, MoM, PIC, and analytical solutions. The FMM is tested using $N = 22\text{k}, 100\text{k}$. (a) shows with equal smoothing and (b) shows with separate smoothing. (c) shows the FMM runtime is $\mathcal{O}(N)$. λ had minimal impact on runtime.

The smoothing formulation allows separate parameters λ_{P-P} for P2P and $\lambda_{\text{Multipole}}$ for the expansion. Figure 4.3a shows the FMM with $\lambda = \lambda_{P-P} = \lambda_{\text{Multipole}}$. The collisional effect is reduced with higher N and an optimal smoothing parameter. In Figure 4.3a, the FMM equals the analytical spread when $\lambda = 1.4 \times 10^{-3}$. The FMM cases behave similarly when $10^{-6} \leq \lambda \leq 10^{-3}$. The MoM already assumes a collisionless beam, so it matches the analytical spread. The PIC code after reaching convergence in time steps shows it underestimates the radial spread in this space charge regime. We expect the realistic beam with collisions would have slightly higher radial spread than predicted. When independently varying the smoothing, Figure 4.3b shows the collision effect is strongest from the P2P operations. We get the FMM cases to converge varying λ_{P-P} but not with $\lambda_{\text{Multipole}}$. The runtime of the FMM is barely affected by λ . Figure 4.3c shows the runtime scales linearly with N when uniform cluster density is optimized.

4.5 FMM and GMRES

The boundary element method (BEM) forms a linear system to solve the Laplace equation from (2.13). The linear system is of form $\mathbf{M}\mathbf{x} = \mathbf{b}$, where \mathbf{b} is known. As mentioned previously, the iterative solver uses information from the matrix-vector product, $\mathbf{M}\mathbf{x}$, to iteratively approach a solution. For our implementation, the system matrix \mathbf{M} of size m is iteratively solved using the restarted generalized minimal residual method (GMRES) [106, 107]. GMRES is a type of Krylov solver, which forms a subspace where the solution exists and iteratively shrinks that space to approximate the solution. Appendix B.3 discusses some important properties of Krylov solvers in the context of GMRES. GMRES forms a basis of orthonormal vectors, \mathbf{Q} using, e.g., the Arnoldi iteration. Starting with an initial residual $\mathbf{r}_i = \mathbf{b} - \mathbf{M}\mathbf{x}_i = \|\mathbf{q}_1\|\mathbf{q}_1$, the Arnoldi iteration projects the components of each basis vector

\mathbf{q}_k of \mathbf{Q} such that $k = 1, \dots, m$ are orthonormal. A new basis vector is formed by $\mathbf{M}\mathbf{q}_{k-1}$. The restarted GMRES has a nested loop. The outer loop computes the i^{th} residual \mathbf{r}_i while the inner loop forms \mathbf{Q} from \mathbf{r}_i . After the inner loop, the outer loop forms \mathbf{x}_{i+1} and checks for convergence. GMRES has an intrinsic error computation for the solution \mathbf{x}_{i+1} at the end of the inner loop [106, 107]. In principle, the outer loop has one \mathbf{M} product and the inner loop has another. GMRES has two tolerance parameters. The absolute tolerance is set for the outer loop and the relative tolerance is set for the inner loop. GMRES exits once the absolute tolerance is satisfied or after the maximum number of iterations. If not satisfied, GMRES reruns the inner loop with the latest solution \mathbf{x}_{i+1} .

The FMM may also be viewed as a fast matrix-vector product approximation [108, 109]. Many iterative solvers, specifically GMRES, require computing products of the system matrix and some vector rather than the matrix itself. The FMM accelerates GMRES, and in turn the BEM, by quickly approximating, e.g., $\mathbf{M}\mathbf{q}_{k-1}$. In addition, we save significant memory since \mathbf{M} is never stored. The matrix-vector approximation may cause some issues with the intrinsic error computation [110–113]. Iterative solvers assume the matrix-vector product is exact. By introducing the FMM to GMRES, we add some error and must be analyzed. We write this as,

$$\tilde{\mathbf{M}}\mathbf{x} = (\mathbf{M} + \mathbf{E})\mathbf{x}.$$

where $\|\mathbf{E}\|$ generally changes. The iterative solver builds the Krylov space based on $\tilde{\mathbf{M}}$. The residual estimation in GMRES is limited by \mathbf{E} and deviates from the true residual at some point. In Ref. [110], the authors show the deviation is bounded by the product of $\|\mathbf{E}\|$ and the norm of the Krylov basis vectors. This suggests convergence as long as this product stays bounded. In Ref. [113], the authors show GMRES converges using the original FMM with order p as long as $p \sim -\log_2(\varepsilon)$, where ε is the prescribed accuracy. That is, the accuracy of the inner loop's product may be relaxed as the residual estimation shrinks. The relaxation

strategy allows some speedup since the inner loop requires smaller p as the Krylov space is formed. Letting $p_{\min} = 2$, we implemented the strategy from Ref. [113] to select the inner loop order with ε being the ratio of absolute tolerance with the estimated residual. The outer loop stays at p_{\max} . This can lead to $\sim 30\%$ speedup in the BEM [113]. We see similar behavior.

Often, GMRES is paired with a preconditioner. The preconditioner transforms an ill-conditioned problem to a well-conditioned one, i.e. reduces the condition number. Commonly for the BEM, GMRES is paired with methods such as incomplete LU factorization and block Jacobi diagonalization [73, 114]. Most methods require the explicit matrix \mathbf{M} to form a preconditioner. We wish to avoid storing \mathbf{M} and focus on matrix-free preconditioners. We found some matrix-free methods exist [115–117]. However, these are not suitable for our implementation. We saw the condition number for our test cases grew quite slowly and GMRES converged rapidly to an acceptable tolerance. Instead, we allowed the option to use a lower order GMRES to find an initial guess. Otherwise, our initial guess is simply $\mathbf{x}_0 = \mathbf{0}$. In our tests, we found the preconditioning does not have a significant speedup and leave it off by default.

4.6 Rotated multipole to local operator

Besides the direct pairwise summation, the M2L operator is the most compute-intensive step in the FMM algorithm [102]. This becomes most apparent when $p > 5$ [7]. The M2L operator translates the expansions at the top level nodes in the C-forest to the top nodes in the D-tree. We can imagine this analogous to the pairwise summation, which requires $\mathcal{O}(N^2)$ operations to pair each particle with every other. Although significantly less, the M2L operator pairs each far cluster to every other target cluster. Then, we need to optimize

the M2L operator to reduce runtime. The method used in Ref. [7] rotates the map such that the 3-D translation reduces to 1-D. The coordinate system is rotated an angle θ about an arbitrary axis $\hat{\omega}$ such that the translation is aligned with $\hat{\mathbf{z}}$. The axis-angle rotation, where θ is the angle between translation $\mathbf{r} = (x, y, z)$ and $\hat{\mathbf{z}}$, is given by [7],

$$R = \begin{pmatrix} t\omega_x^2 C & t\omega_x\omega_y - S\omega_z & t\omega_x\omega_z + S\omega_y \\ t\omega_x\omega_y + S\omega_z & t\omega_y^2 C & t\omega_y\omega_z - S\omega_x \\ t\omega_x\omega_z - S\omega_y & t\omega_y\omega_z + S\omega_x & t\omega_z^2 C \end{pmatrix},$$

$$C \equiv \cos \theta, \quad S \equiv \sin \theta, \quad t \equiv 1 - \cos \theta,$$

$$\omega_x = -\frac{y}{\|\mathbf{r}\|}, \quad \omega_y = -\frac{x}{\|\mathbf{r}\|}, \quad \omega_z = 0.$$

Given multipole expansion ϕ_M , the local expansion is $\phi_L = \phi_M \circ \mathcal{M}'_2 = \phi_M \circ R^{-1} \circ \mathcal{M}_2 \circ R$, where $R^{-1} = R^\top$. Some numerical instability is present in the rotated operator. Figure 4.4 shows the relative difference as p increases.

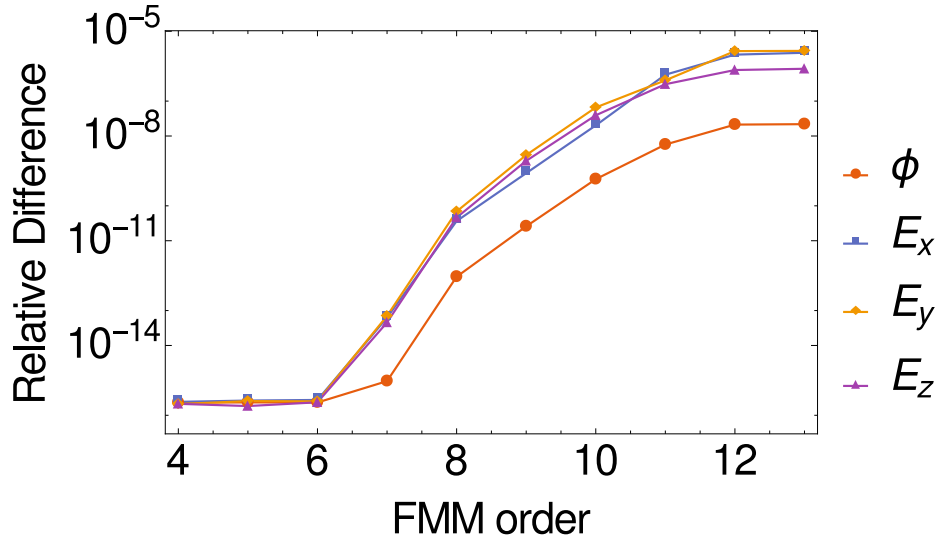


Figure 4.4: Normed relative difference in potential and \mathbf{E} between 3DM2L and rotated M2L operators vs. FMM order using $N = 100$ particles.

We used 100 particles to compare the non-rotated and rotated M2L operators (3DM2L, 1DM2L). Ref. [7] shows significant speedup with the 1DM2L, but the accuracy drops when the rotation is not exact due to the limits of machine precision. The slight error could mean the translation is not aligned with \mathbf{r} in a few cases, leading to the errors in figure 4.4. For now, we implemented both rotated and unrotated M2L operators. By default, we use the unrotated M2L for the modified BC and density evaluation. We use the rotated M2L for the matrix solver. This can be adjusted using the global flag `RFLAG`.

We have established the fast multipole method in the differential algebraic framework and described the implementation based on Ref. [7]. In this form, the FMM is used to accelerate pair interactions between point-like particles. We have included a smoothed form of the FMM operators and shown an optimal range exists. The FMM also accelerates matrix-vector products and saves significant memory in iterative solvers since the matrix is not stored. A major weakness of the boundary element method is the dense system matrix. Iterative solvers require many matrix-vector products, which we accelerate using the FMM.

CHAPTER 5

BOUNDARY INTEGRAL THEORY

As mentioned in Chapter 2, the Poisson equation (2.3) may be formulated as an integral equation. The full equation with Green's function, $G(\mathbf{x}, \mathbf{y})$, can be derived using (2.9) and the solution is given in (5.1). The goal is to describe Φ in the domain Ω bounded by Γ , where Ω_- is the interior domain and Ω_+ is the exterior domain. The unit normal $\mathbf{n}(\mathbf{x}) \equiv \mathbf{n}_x$, $\mathbf{x} \in \Omega$ is chosen to be pointing out towards Ω_+ .

$$\Phi(\mathbf{x}) = \phi(\mathbf{x}) + \psi(\mathbf{x}) = \int_{\Omega} G(\mathbf{x}, \mathbf{y}) \frac{\rho(\mathbf{y})}{\epsilon_0} d\Omega(\mathbf{y}) + \oint_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) g(\mathbf{y}) - G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}). \quad (5.1)$$

This chapter explains some numerical methods used in solving the Poisson boundary value problem. We derive and compare the finite element method (FEM) and the boundary element method (BEM) to justify our choice in implementing the BEM for applications in beam physics. Our implementation uses the indirect form for the BEM. For Dirichlet conditions, we choose to use the double layer representation and for Neumann conditions, we choose the single layer representation. The later sections explain our choice for representing the potential ψ in Ω_- .

5.1 Finite element models

Equation (2.3) may be solved by discretizing Ω into Ω_i , $i = 1, \dots, M$ such that $\Omega_i \subset \Omega$ and $\bigcup_{i=1}^M \Omega_i = \Omega$. Commonly, the differential equations, including Poisson, are solved using the finite element method (FEM). The FEM breaks down Ω into 3-D element shapes,

such as tetrahedrons, represented by Lagrange shape functions, $N_i(\mathbf{x})$. The Lagrange shape functions are linear combinations of Lagrange basis functions, $l_j(x)$, defined by the property, $l_i(x_j) = \delta_{ij}$. In 1-D, with given point set x_m , $m = 1, \dots, k+1$, the j th polynomial $l_j(x)$ is given by,

$$l_j(x) = \prod_{m \neq j}^{k+1} \frac{x - x_m}{x_j - x_m}.$$

The FEM approximates a solution to (5.1) within each Ω_i . This section is based on work from Refs. [36, 118].

Given the problem posed by (2.3), we wish to find an unknown function, $\Phi(\mathbf{x})$, such that it satisfies $\nabla^2 \Phi(\mathbf{x}) = f(\mathbf{x})$ with the given boundary conditions, $\Phi(\mathbf{x}) = g(\mathbf{x})$ or $\frac{\partial \Phi}{\partial n}(\mathbf{x}) = h(\mathbf{x})$ when $\mathbf{x} \in \Gamma$. The solution may be approximated by $\Phi \approx \tilde{\Phi} = \sum_{j=1}^n N_j a_j = \mathbf{N}\mathbf{a}$, where a_j are unknown. To find a_j , the problem is recast into the weak formulation. Observe $\int_{\Omega} v(\nabla^2 \Phi - f) d\Omega = 0$ still satisfies the original PDE and $\int_{\Gamma} \bar{v}b d\Gamma = 0$ where $b = \Phi - g$ or $\frac{\partial \Phi}{\partial n} - h$ satisfies the boundary conditions. So, the sum satisfies the BVP. If we perform an integration by parts on the first term and use the divergence theorem, we get the weak form in (5.2).

$$\begin{aligned} \int_{\Omega} v \nabla^2 \Phi d\Omega &= \int_{\Omega} \nabla \cdot (v \nabla \Phi) - \nabla \Phi \cdot \nabla v d\Omega, \\ &= - \int_{\Omega} \nabla \Phi \cdot \nabla v d\Omega + \int_{\Gamma} v \frac{\partial \Phi}{\partial n} d\Gamma, \\ \Rightarrow - \int_{\Omega} \nabla \Phi \cdot \nabla v + v f d\Omega + \int_{\Gamma} v h + \bar{v} b d\Gamma &= 0. \end{aligned} \quad (5.2)$$

Inserting $\tilde{\Phi}$ into (5.2) and choosing $v = \sum_{i=1}^n w_i \delta a_i$, $\bar{v} = \sum_{i=1}^n \bar{w}_i \delta a_i$, we get (5.3) with $i = 1, \dots, n$.

$$\delta a_i \left[- \int_{\Omega} \nabla(\mathbf{N}\mathbf{a}) \cdot \nabla w_i + w_i f d\Omega + \int_{\Gamma} w_i h + \bar{w}_i b d\Gamma \right] = 0. \quad (5.3)$$

Just about any set of independent functions works for w_i [36]. The Galerkin method is mainly used in practice, which corresponds to $w_i = N_i$, the shape functions. Since δa_i is arbitrary, the terms in brackets leads to the symmetric matrix equation for a_j in (5.4) and gives the Galerkin FEM for the Poisson equation.

$$\sum_{j=1}^n \left[\int_{\Omega} -\nabla N_j \cdot \nabla N_i \, d\Omega \right] a_j - \int_{\Omega} N_i f \, d\Omega + \int_{\Gamma} N_i h + N_i b \, d\Gamma = 0, \quad (5.4)$$

$$\Leftrightarrow \mathbf{L}\mathbf{a} + \mathbf{f} = 0.$$

It should be noted the elements for this form are chosen by the shape function and all the functions are C^1 . As the element size, e , shrinks, the number of a_j increases. Since the shape functions, N_j , are polynomials, there is also an associated polynomial order, p . Clearly, the convergence of the approximation is then dependent on e and p . In fact, the order of error ends up being $\mathcal{O}(e^{p+1})$ [36].

5.2 FEM versus BEM

The FEM shows great performance and many fields have applied it [36]. As shown, only the appropriate shape functions are needed as well as the conditions $f \rightarrow \rho$, $h \rightarrow \frac{\partial \Phi}{\partial n}$, and b . However, when considering only the effect from Γ , the FEM loses efficiency due to the element discretization. There are several advantages for the BEM versus the FEM [119]. The FEM requires discretization of the whole domain. For 3-D, this is expensive in computational and storage cost [119]. The BEM only requires discretization of Γ , which is represented as a surface. This reduces the 3-D mesh and the problem dimension to 2-D. The advantage shows when considering the solution convergence by reducing mesh size e . If mesh size e is reduced, the FEM generally gains more elements than the BEM. The FEM computes a

solution for Ω in (5.4). This may be advantageous if the region of interest is all of Ω [36]. However, when considering only small subsets, i.e. Ω_- , the BEM has the advantage [120]. The BEM computes a solution for Γ first and evaluates it in Ω separately.

The BEM approximates the boundary conditions only. The FEM approximates the PDE through the shape functions, which affects all aspects of the problem [74]. The BEM error is isolated to the boundary. This suggests better convergence for the BEM by reducing discretization error [119]. The Galerkin FEM leads to a sparse, symmetric matrix. This is more advantageous as many fast solvers exist for this case. The BEM system matrix, while smaller, is dense and nonsymmetric [74]. Direct or iterative solvers take longer in this case. Iterative solvers are accelerated using the fast multipole method to mitigate this disadvantage [73]. The integrals in the FEM are simpler to evaluate than the BEM. As shown, the FEM integrals amount to integrating the Lagrangian shape functions or its derivatives [36]. This reduces it to a sum almost immediately due to the shape function properties. The BEM requires integration of the Green's function, which includes integrating singularities when $\mathbf{x} = \mathbf{y}$. For our specific case, the BEM is expected to outperform the FEM after introducing curvilinear elements and the FMM acceleration [73, 121]. We note the form in (5.4) combined with the shape functions is sensitive to noise or error, i.e., an ill-conditioned linear system. Our chosen boundary integral form does not have this sensitivity and is discussed later.

5.3 Boundary integrals

The FEM shares many concepts with the BEM as they are related. The main difference appears in the derivation of the BEM. From earlier with (2.10), we formed the boundary integral equation (BIE) using Green's identity. As shown in Ref. [119], the BIE derivation

is similar to the FEM. However, it requires knowing the fundamental solution, i.e. Green's function G , that satisfies (5.5) for linear operator \mathcal{L} .

$$\mathcal{L}(G) = \delta. \quad (5.5)$$

When $\mathcal{L} = \nabla^2$, the 3-D Green's function is given by (5.6).

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi\|\mathbf{x} - \mathbf{y}\|}. \quad (5.6)$$

It is due to (5.5) that we may write a boundary integral solution [74]. Using Green's identity (2.9), $\Phi(\mathbf{x})$ is directly represented in (5.1) as the sum of the particular and homogeneous solution to the Poisson BVP. That is, the Poisson equation is decomposed to the open-boundary Poisson equation and closed-boundary Laplace equation. Φ is the composition of ϕ and ψ , or the sum for the linear operator case. This allows separate treatment in our implementation. We focus on ψ for the rest of this chapter. We call (5.1) the direct boundary integral equation.

From the direct BIE, we need both boundary conditions, g, h , on Γ [73]. If either is unknown, we would need to solve (5.1). There are some advantages in that the form easily includes mixed conditions, i.e. g on Γ_1 and h on Γ_2 with $\Gamma = \Gamma_1 \cup \Gamma_2$, or linear combinations $ag+bh$ with coefficients a, b . The implementation would require rearranging the matrix before solving. There is also inherent physical meaning in the direct BIE since the unknowns are the boundary conditions themselves [73]. Eq. (5.5) allows an indirect BIE with the ansatz that a charge distribution on Γ exists which reproduces the boundary conditions [74]. Note that the ansatz is interpretation dependent and not attached to physical objects [122]. The single layer and double layer potential from (2.11), (2.12) represent the potential due to this distribution if they satisfy additional constraints. These potentials must satisfy the boundary

conditions for a unique interior solution. In addition, ψ must match the correct behavior as $\mathbf{x} \rightarrow \infty$ for a unique exterior solution. To further compare the direct and indirect BIE, we show their behavior as $\mathbf{x} \rightarrow \Gamma$.

5.4 Regularization of the BIE

The behavior of the BIE on Γ is similar for the direct and indirect forms [123]. Our derivation is given in Appendix A.1 for the indirect BIE in the Cauchy principal value sense [124]. We provide an intuitive description here. We define $\Omega \subset \mathbb{R}^3$ with at least piecewise continuous boundary, Γ . The single and double layer operators are given in (2.11) and (2.12). $G(\mathbf{x}, \mathbf{y})$ is singular when $\mathbf{x} = \mathbf{y}$, but the single and double layer integrals are finite [74]. Let $\mathbf{x}, \mathbf{y} \in \Gamma$. We can define a spherical surface Γ_ε centered at fixed \mathbf{x} with some radius ε with $\mathbf{y} \in \Gamma \cap \Gamma_\varepsilon$ as shown in Figure 5.1 [8].

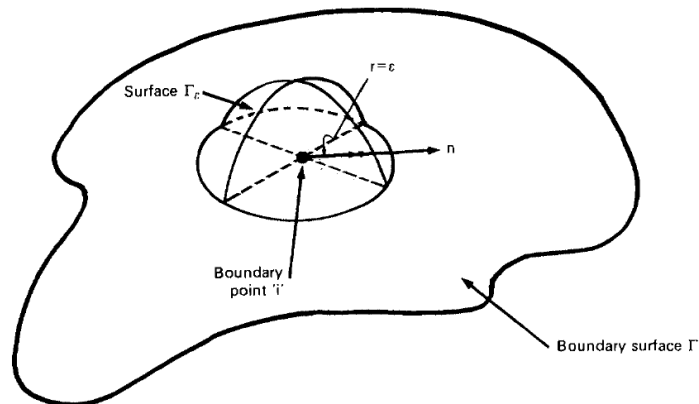


Figure 5.1: Depiction of half the sphere Γ_ε with radius $\varepsilon = \|\mathbf{x} - \mathbf{y}\|$ around singular point $i = \mathbf{x}$ and $\mathbf{y} \in \Gamma_\varepsilon$ [8]. To find the principal value, we can integrate the sphere separately in the limit of $\varepsilon \rightarrow 0$.

When $\|\mathbf{x} - \mathbf{y}\| > \varepsilon$, $\mathbf{x} \neq \mathbf{y}$ and we preserve continuity. When $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon$, we consider the integral over the sphere as $\varepsilon \rightarrow 0$. Remember that the indirect BIE assumes a density on the

boundary that reproduces the BC. We could imagine instead of the singular point charge $\sigma(\mathbf{y})$ or $\eta(\mathbf{y})$, we spread out the singularity onto the surface, so $\mathbf{y} \in \Gamma_\varepsilon$ and $\varepsilon = \|\mathbf{x} - \mathbf{y}\|$. The singularity concentrates at \mathbf{x} as $\varepsilon \rightarrow 0$. Considering the single layer potential, $G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi\varepsilon}$ is constant over the sphere. So, we sum up the singularity, which gives the surface area. Overall, this goes to 0 as $\varepsilon \rightarrow 0$. Similarly, we consider the double layer potential. Now we must consider $\mathbf{n}(\mathbf{y}) \cdot \nabla G$, where $\mathbf{n}(\mathbf{y}) = \frac{\mathbf{x}-\mathbf{y}}{\varepsilon}$ is not constant. Summing the flux suggests we only capture half of the singularity [119]. This depends on the surface area, which may differ if Γ is not smooth. We showed the singular integrals are bounded and its principal value is simply $\frac{1}{2}$ when smooth. If it is locally nonsmooth, we end up with $1 - \frac{\phi}{4\pi}(1 - \cos\theta)$, which is really $\frac{\text{inner solid angle}}{4\pi}$ [119].

5.5 Direct versus indirect BIE

We have established the two main BIE formulations. We consider our requirements in beam physics. Generally, we wish to know the effect due to the presence of charge. We know the boundary conditions change via (2.7). The various beamline components output modified fields as a result. To assess the magnitude, we care most about the interior in the beam region. We need the field near or on the surface in rare cases. Let's assume we are given homogeneous BC, e.g., Dirichlet. The direct form requires knowing Neumann type as well. We must manipulate the matrix equation after discretization to isolate the known and unknowns, which can be unnecessarily complicated. In addition, the discretization leads to essentially two integrations [125]. Considering numerical integration by quadrature, we have twice as many terms in the sum with half of them unknown. After solving, we then evaluate with two integrations. The direct form solves for the boundary conditions accurately, but

we need the evaluation to be efficient. The indirect method solves for the fictitious density without the algebra. We also evaluate with half the terms compared to the direct form.

When we evaluate somewhere near the surface or outside the structure, the direct form is more stable than the indirect form [8, 74, 126]. When on the surface, we want the boundary condition. Either it is given or the direct form solves for it. When outside the structure, the direct form already satisfies the radiation condition [125]. In other words, the direct form may be used for interior or exterior problems without any changes. On the surface, we may use the given BC or the solved BC, possibly interpolated along the surface. The direct form allows flexible cases, but the beam must only be in the interior. The particles cannot go outside the structure, and if near or on the surface, we call that beam loss. The indirect form efficiently represents the interior region we need. If near the surface, the indirect form requires additional techniques [127–129].

5.6 Existence, uniqueness, and conditioning

Previously, we assumed a solution exists to the Poisson equation. Fredholm theory states the solution exists for a linear and compact operator. This section briefly describes the Laplacian operator in the context of Fredholm theory, following Ref. [130]. For complete, normed spaces, Λ and Ξ , a compact operator is a linear operator that maps a subset of Λ to a closed subset of Ξ . In other words, the operator is bounded. The Laplacian is compact [74].

We defined the layer potential operators earlier. We may now represent ψ when $\mathbf{x}, \mathbf{y} \in \Gamma$ for Dirichlet (5.7) and Neumann conditions (5.8).

$$\psi(\mathbf{x}) = -c(\mathbf{x})\eta(\mathbf{x}) + D(\eta), \quad \text{Dirichlet BC,} \quad (5.7)$$

$$\frac{\partial \psi}{\partial n_x}(\mathbf{x}) = c(\mathbf{x})\sigma(\mathbf{x}) + \frac{\partial S}{\partial n_x}(\sigma), \quad \text{Neumann BC,} \quad (5.8)$$

$$c(\mathbf{x}) = \begin{cases} 1 & , \text{ if } \mathbf{x} \in \Omega_-, \\ \frac{1}{2} & , \text{ if } \mathbf{x} \in \Gamma, \Gamma \text{ smooth,} \\ \frac{\text{internal solid angle}}{4\pi} & , \text{ if } \mathbf{x} \in \Gamma, \Gamma \text{ not smooth.} \end{cases}$$

We implicitly assume σ, η are continuous on Γ . We know that G is singular, but as shown earlier, the integral is uniformly bounded and equicontinuous. Thus, the integral operators are uniformly bounded, equicontinuous, and compact. The Fredholm alternative theorem shows a unique solution exists as stated in Theorem 5.1 [130].

Theorem 5.1 (Fredholm alternative). *Let \mathcal{X} be a Banach space, and let $\mathcal{K} : \mathcal{X} \rightarrow \mathcal{X}$ be a compact operator. Then the equation $(\lambda - \mathcal{K})x = y, \lambda \neq 0$ has a unique solution $x \in \mathcal{X}$ if and only if the homogeneous equation $(\lambda - \mathcal{K})z = 0$ has only the trivial solution $z = 0$. In such a case, the operator $\lambda - \mathcal{K} : \mathcal{X} \xrightarrow[\text{onto}]{1-1} \mathcal{X}$ has a bounded inverse $(\lambda - \mathcal{K})^{-1}$.*

If we have a compact operator, there exists a unique solution to $(\lambda - \mathcal{K})x = y$. This form is also known as a Fredholm integral equation of the second kind.

Since we are concerned with the numerical solution, we also must consider the conditioning of the linear system. As mentioned in Chapter 4, the boundary element method gives a linear system of form $\mathbf{M}\mathbf{x} = \mathbf{b}$. The condition number is defined by,

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|.$$

The condition number indicates the effect on the result due to a small change in the input. In terms of error, an ill-conditioned problem or large κ is sensitive to errors due to approximation. A well-conditioned or small κ approaching 1 indicates low error sensitivity. We have shown $(\lambda - \mathcal{K})$ is bounded for (5.7), (5.8) and Theorem 5.1 shows the inverse must be bounded. From the definition of κ , this suggests the condition number is also finite. Using the conducting sphere as an example linear system from the BEM, we show the resulting condition numbers in Table 5.1 as the sphere is more finely discretized using flat panel elements. For this example, the matrix size is n , where n is the number of elements.

Table 5.1: Condition Number vs. Matrix Size n for Perfectly Conducting Sphere

n	Dirichlet DL	Neumann SL	Dirichlet SL
80	2.8911	2.8671	1579.5
320	2.9116	2.9333	2624.7
1280	2.9512	2.9681	20561.
5120	2.9748	2.9851	1.604×10^6

We can see the second kind integral equations, Dirichlet double layer (DL) and Neumann single layer (SL), asymptotically tend to $\mathcal{O}(1)$ as $n \rightarrow \infty$. However, the conditioning of the Dirichlet SL which is a first kind integral equation, rapidly degrades with n . For the BEM, it is crucial for the condition number to be bounded. For our purpose, the indirect Dirichlet double layer and Neumann single layer operators demonstrate good behavior. In addition, we expect the wellposedness to minimize the number of iterations for our iterative matrix solver [131].

5.7 Discretization

The integral operator is compact and the BIE of the second kind is well-conditioned. We discretize the operator by splitting the surface into a mesh of triangular surface elements,

$\Gamma = \bigcup_{j=1}^n \Gamma_j$ and apply the operator to Γ_j . This gives the discrete BIE in (5.9) with $c(\mathbf{x})$ as defined earlier.

$$\begin{aligned}\psi(\mathbf{x}) &= -c(\mathbf{x})\eta(\mathbf{x}) + \sum_{j=1}^n \int_{\Gamma_j} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y})\eta(\mathbf{y}) \, d\Gamma(\mathbf{y}), \\ \frac{\partial \psi}{\partial n_x}(\mathbf{x}) &= c(\mathbf{x})\sigma(\mathbf{x}) + \sum_{j=1}^n \int_{\Gamma_j} \frac{\partial G}{\partial n_x}(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y}) \, d\Gamma(\mathbf{y}).\end{aligned}\tag{5.9}$$

To numerically solve (5.9), we must make some assumptions for σ, η on Γ_j . For instance, we can assume they are constant on Γ_j which gives the flat panel collocation method where we only need to integrate $\frac{\partial G}{\partial n}$ for the system matrix [73]. This assumption has strict limitations and tends to give low accuracy solutions [74], which is discussed in Chapter 6. The Galerkin discretization is mentioned in the context of the FEM. In general, the Galerkin discretization is well-behaved and highly accurate [36, 125, 132]. However, it requires the trial function, v in (5.2). For the BIE, the Galerkin method requires an additional surface integration of a trial function [74, 133]. This implies four integrations on Γ_j in (5.9), which is approximated using numerical integration. We chose the Nyström method where the surface integral is directly discretized using quadrature [130].

Quadrature approximates the integration by a sum of function evaluations multiplied by weights, $\int_{\Gamma} f(\mathbf{x}) \, d\Gamma \approx \sum_{i=1}^m w_i f(\mathbf{x}_i)$ [134]. The implementation requires summation of point-like evaluations on Γ_j and the accuracy of the integration is determined by the quadrature rule [135], i.e. choice of nodes \mathbf{x}_i and weights w_i , and m . We assume the quadrature converges as $m \rightarrow \infty$. The Nyström method approximates (5.9) as (5.10).

$$\begin{aligned}\psi(\mathbf{x}) &\approx -c(\mathbf{x})\eta(\mathbf{x}) + \sum_{j=1}^n \sum_{k=1}^m w_{jk} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}_{jk})\eta(\mathbf{y}_{jk}), \\ \frac{\partial \psi}{\partial n_x}(\mathbf{x}) &\approx c(\mathbf{x})\sigma(\mathbf{x}) + \sum_{j=1}^n \sum_{k=1}^m w_{jk} \frac{\partial G}{\partial n_x}(\mathbf{x}, \mathbf{y}_{jk})\sigma(\mathbf{y}_{jk}),\end{aligned}\tag{5.10}$$

$$\mathbf{y}_{jk} \in \Gamma_j.$$

The original Nyström method interpolates, e.g., η , using the form (5.11).

$$\eta(\mathbf{x}) = \frac{1}{c(\mathbf{x})} \left[-\psi(\mathbf{x}) + \sum_{j=1}^n \sum_{k=1}^m w_{jk} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}_{jk}) \eta(\mathbf{y}_{jk}) \right]. \quad (5.11)$$

The main reason for choosing the Nyström method is the point-like pairwise summations of the Green's function [130]. The Nyström provides a middle ground between the compute-intensive, high-accuracy Galerkin method and the point-like, but low-accuracy flat panel collocation method [74].

We have shown the boundary integral form has many benefits for our intended application. The indirect form is also more efficient than the direct form when only considering Ω_- , sufficiently away from Γ . Fredholm integral theory shows the second kind BIEs have many desirable properties for the numerical solution [130]. In particular, the wellposedness suggests low error sensitivity and fast convergence for iterative solvers [136]. Finally, we chose the Nyström discretization because of its compatibility with the FMM as well as being less compute-intensive than the Galerkin method while providing comparable accuracy [74]. The BEM error is mostly due to surface discretization and numerical integration [73]. We next show the flat panel discretization of Γ has significant limitations and discuss high-order interpolation schemes. Also, many BEM implementations use Gaussian quadrature to approximate the integrals in the system matrix [137]. However, this requires a flexible representation of the surface elements, which is difficult to implement for surface elements embedded in 3-D [138, 139]. In many cases, the elements are determined using fixed order polynomials, for instance using B -splines or Lagrangian shape functions [140, 141]. Our implementation allows for a variable element order representation using Vandermonde-like parametrization.

CHAPTER 6

HIGH ORDER DISCRETIZATION

This chapter discusses the high-order discretization scheme used in our fast BEM implementation. Each element Γ_i is parametrized to the unit triangle with local coordinates (u, v) , shown in Figure 6.1.

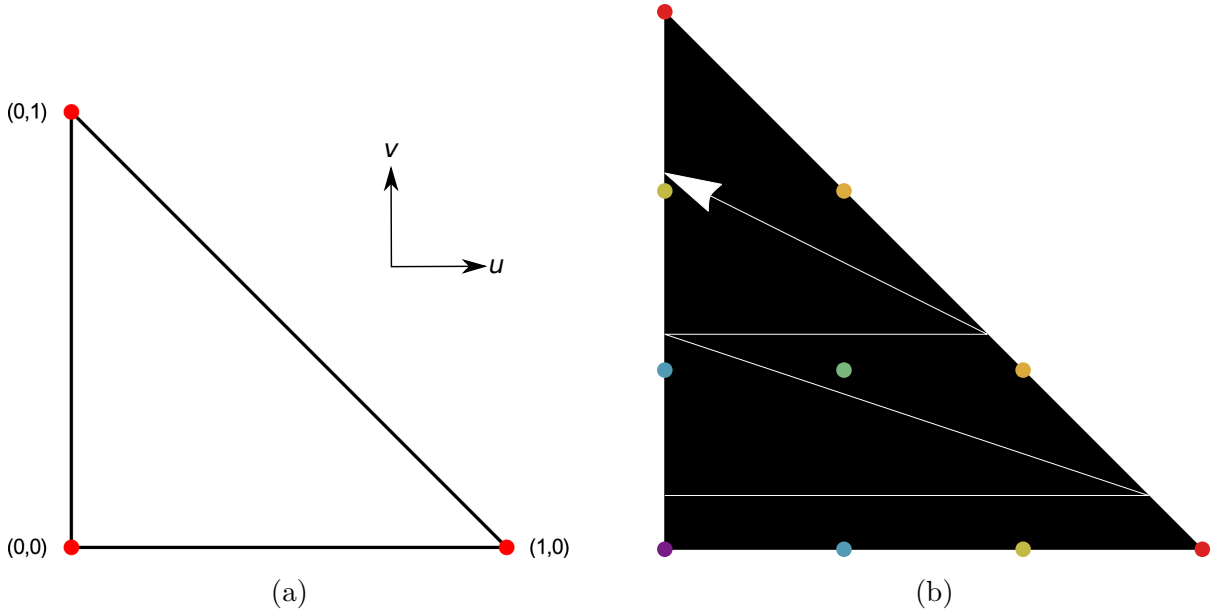


Figure 6.1: Unit triangle for (u, v) -parametrization. The three vertices are shown in (a) as red dots with local coordinates indicated. The ordering of nodes for $e = 4$ is shown in (b), bottom left to top vertex.

Higher element order requires more nodes. The assumed node ordering is shown in figure 6.1b with $e = 4$. The global coordinates for each node are given as Cartesian triplets, $\mathbf{x} = (x_1, x_2, x_3)$. The outward unit normals for each node is given as $\mathbf{n}(\mathbf{x}) = (n_1, n_2, n_3)$. We form a polynomial mapping between the global and local coordinates. Each element is described by three polynomials, $\mathbf{P}(u, v) = (P_1, P_2, P_3)$ where e , the polynomial degree,

is the element order. The usual polynomial parametrization uses the Lagrangian shape functions as with the FEM. However, since we use integrals containing $\frac{\partial G}{\partial n}$, we need normals with accurate direction and length along the surface. Normals from the shape functions are somewhat difficult to extract and are not well-reproduced. We examined several techniques which address interpolation of the normals. In this chapter, we start by discussing the need for high-order elements. We also discuss analytical quadratic and cubic elements. We derive our variable order polynomial parametrization. We discuss a suitable quadrature. We show the accuracy of the interpolation and quadrature and finally discuss some of the known limitations.

6.1 Interpolation

Interpolation is the process of fitting given data y to some function, $y = f(x)$. In 3-D, we fit a surface to the data set, (\mathbf{x}_i, y_i) . The interpolated surface must pass through the given data, i.e. $f(\mathbf{x}_i) = y_i$. In contrast, the approximated surface satisfies $|f - y| \leq \epsilon$ for small ϵ . For our purpose, the approximate surface is not expected to reproduce the normals adequately as the approximation may introduce artificial smoothness. We focus on the interpolation requirements. Interpolating a small object such as a surface or solid is commonly needed in graphical applications. Many CAD programs use parametric descriptions for modeling. Models are described using NURBS (non-uniform rational basis splines) among other entities [142]. Splines mainly parametrize an object using control points combined with Bernstein polynomials [143]. Splines give good approximations but often do not satisfy the interpolation conditions and introduce artificial smoothness [144]. Splines perform well for graphics but do not guarantee accurate normals. In particular, given the nodes with outward unit normals forming a triangular element, we wish to reproduce a continuous (C^0) surface

satisfying the interpolation conditions with continuous tangent planes. This is known as geometric continuity G^1 .

6.2 Limits of flat panels

The flat panel element is considered to be $e = 0$. Clearly, flat panels cannot reproduce the curvature nor form G^1 elements. While simple to implement, the discretization introduces significant error. Figure 6.2 compares the discretized spheres using 80 elements versus 1280 elements. Figure 6.3 shows the percent error in the area versus the number of flat panel elements for a sphere with radius 0.1 and a cylinder with radius 0.1 and length 0.1 in arbitrary units.

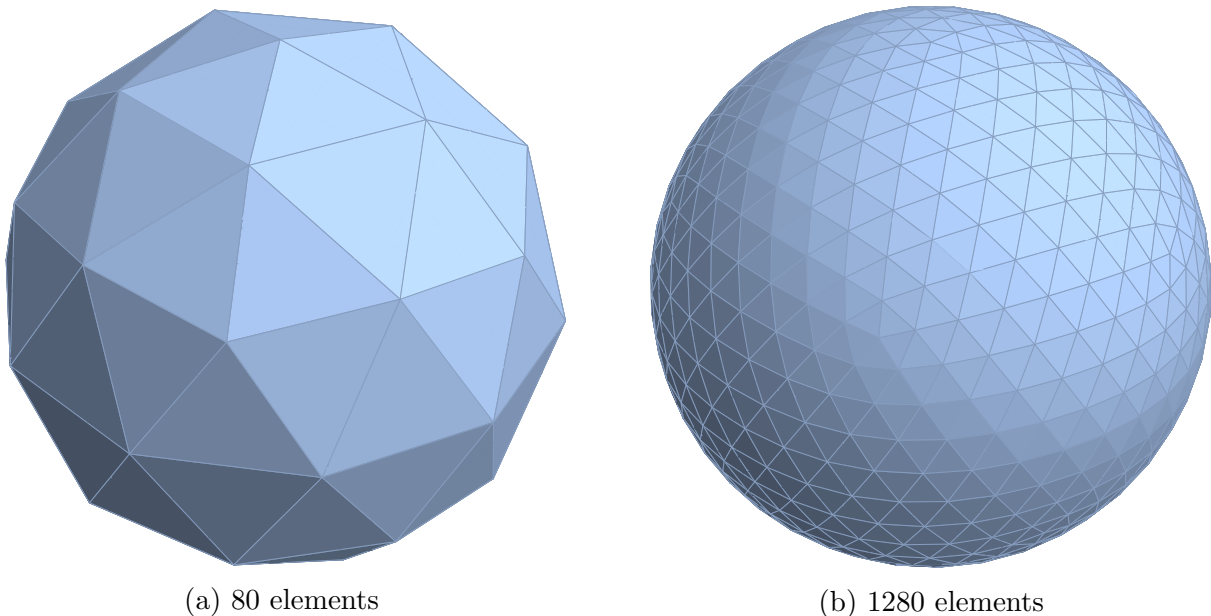


Figure 6.2: A sphere discretized with flat elements. Visually, the discretized sphere approaches the actual sphere.

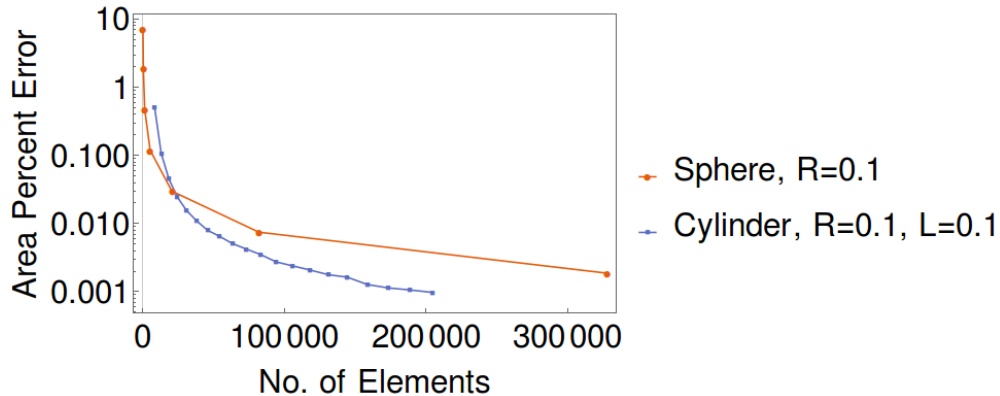


Figure 6.3: The percent error in area from discretizing a sphere $R = 0.1$ and cylinder $R = 0.1, L = 0.1$ using flat panels. The error appears to converge to 10^{-3} .

Appearance-wise, the discretized sphere converges to the actual sphere. From figure 6.3, the minimum expected error using flat panels is $\mathcal{O}(10^{-3})$ with $n \sim 10^5$. Chapter 8 discuss the flat panel error in the BEM.

6.3 Analytical derivations of parametric elements

We skip $e = 1$ since linear elements give similar results to constant elements. Instead, we consider quadratic elements, $e = 2$. In Ref. [145], the author shows an analytical solution for quadratic elements. With $\mathbf{x} = (x_1, x_2, x_3) \in \gamma$, where γ denotes a smooth triangular patch, we look for polynomials of the form (6.1),

$$\mathbf{x}(u, v) = \mathbf{a}_{00} + \mathbf{a}_{10}u + \mathbf{a}_{01}v + \mathbf{a}_{20}u^2 + \mathbf{a}_{11}uv + \mathbf{a}_{02}v^2. \quad (6.1)$$

We adjust the derivation in Refs. [145, 146] slightly to fit the unit triangle in Figure 6.1. Let the vertices \mathbf{x} and associated outward unit normals \mathbf{n} be denoted by,

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{x}(0, 0), & \mathbf{x}_2 &= \mathbf{x}(1, 0), & \mathbf{x}_3 &= \mathbf{x}(0, 1), \\ \mathbf{n}_1 &= \mathbf{n}(0, 0), & \mathbf{n}_2 &= \mathbf{n}(1, 0), & \mathbf{n}_3 &= \mathbf{n}(0, 1).\end{aligned}$$

To start, consider a single curve. The equation for a quadratic curve between \mathbf{x}_0 and \mathbf{x}_1 , with $\mathbf{d} = \mathbf{x}_1 - \mathbf{x}_0$ and \mathbf{c} is unknown, is given by,

$$\mathbf{x}(u) = \mathbf{x}_0 + (\mathbf{d} - \mathbf{c})u + \mathbf{c}u^2.$$

We may interpret \mathbf{c} as a curvature parameter. We require the normals to be perpendicular to the tangents at the endpoints $\mathbf{x}_0, \mathbf{x}_1$. In other words,

$$\begin{aligned}\mathbf{n}_0 \cdot \left. \frac{d\mathbf{x}}{du} \right|_{u=0} &= 0, & \mathbf{n}_1 \cdot \left. \frac{d\mathbf{x}}{du} \right|_{u=1} &= 0, \\ \frac{d\mathbf{x}}{du} &= \mathbf{d} + (2u - 1)\mathbf{c}, \\ \Rightarrow & \left\{ \begin{array}{l} \mathbf{n}_0 \cdot (\mathbf{d} - \mathbf{c}) = 0 \\ \mathbf{n}_1 \cdot (\mathbf{d} + \mathbf{c}) = 0 \end{array} \right\}, \\ \Leftrightarrow & \begin{pmatrix} \mathbf{n}_0^\top \\ \mathbf{n}_1^\top \end{pmatrix} \mathbf{c} = \begin{pmatrix} \mathbf{n}_0^\top \mathbf{d} \\ -\mathbf{n}_1^\top \mathbf{d} \end{pmatrix}.\end{aligned}$$

We solve this using the generalized inverse. Since the system is small, we build the inverse directly.

$$\begin{aligned}
\mathbf{A}^\dagger &= \lim_{\alpha \rightarrow 0} \{\mathbf{A}^* \mathbf{A} + \alpha \mathbf{I}\}^{-1} \mathbf{A}^*, \\
\Rightarrow \begin{pmatrix} \mathbf{n}_0^\top \\ \mathbf{n}_1^\top \end{pmatrix}^\dagger &= \begin{pmatrix} \mathbf{n}_0^\dagger & \mathbf{n}_1^\dagger \end{pmatrix}^\top, \\
&= \lim_{\alpha \rightarrow 0} \begin{pmatrix} \mathbf{n}_0 & \mathbf{n}_1 \end{pmatrix} \frac{1}{(1 + \alpha)^2 - b^2} \begin{pmatrix} 1 + \alpha & -b \\ -b & 1 + \alpha \end{pmatrix}, \\
&= \begin{cases} \frac{1}{1 - b^2} \begin{pmatrix} \mathbf{n}_0 & \mathbf{n}_1 \end{pmatrix} \begin{pmatrix} 1 & -b \\ -b & 1 \end{pmatrix} & , \text{ if } b \neq \pm 1, \\ \frac{1}{2} \begin{pmatrix} \mathbf{n}_0 & \pm \mathbf{n}_0 \end{pmatrix} & , \text{ if } b = \pm 1, \end{cases} \\
b &\equiv \mathbf{n}_0 \cdot \mathbf{n}_1 = \mathbf{n}_0^\top \mathbf{n}_1.
\end{aligned}$$

Thus, we can define a function for \mathbf{c} as follows. For simplification, first let,

$$\begin{aligned}
\boldsymbol{\nu} &\equiv \frac{\mathbf{n}_0 + \mathbf{n}_1}{2}, \\
\Delta \boldsymbol{\nu} &\equiv \frac{\mathbf{n}_0 - \mathbf{n}_1}{2} \Rightarrow b = \mathbf{n}_0 \cdot (\mathbf{n}_0 - 2\Delta \boldsymbol{\nu}), \\
\Delta b &\equiv \mathbf{n}_0 \cdot \Delta \boldsymbol{\nu} \Rightarrow b = 1 - 2\Delta b, \\
d &\equiv \mathbf{d} \cdot \boldsymbol{\nu}, \quad \Delta d \equiv \mathbf{d} \cdot \Delta \boldsymbol{\nu}.
\end{aligned}$$

After some algebra, we get (6.2).

$$\mathbf{c}(\mathbf{d}, \mathbf{n}_0, \mathbf{n}_1) = \begin{cases} \frac{\Delta d}{1 - \Delta b} \boldsymbol{\nu} + \frac{d}{\Delta b} \Delta \boldsymbol{\nu} & , \text{ if } b \neq \pm 1, \\ \mathbf{0} & , \text{ if } b = \pm 1. \end{cases} \quad (6.2)$$

Now we consider the triangular patch. Let $\mathbf{d}_{ij} \equiv \mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{c}_{ij} \equiv \mathbf{c}(\mathbf{d}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ as defined in (6.2). The equations for the three edges of the triangle are

$$\begin{aligned}
\mathbf{x}(u, 0) &= \mathbf{x}_1 + (\mathbf{d}_{12} - \mathbf{c}_{12})u + \mathbf{c}_{12}u^2, \\
\mathbf{x}(0, v) &= \mathbf{x}_1 + (\mathbf{d}_{13} - \mathbf{c}_{13})v + \mathbf{c}_{13}v^2, \\
\mathbf{x}(u, 1 - u) &= \mathbf{x}_3 + (\mathbf{d}_{32} - \mathbf{c}_{32})u + \mathbf{c}_{32}u^2, \\
\mathbf{d}_{32} + \mathbf{d}_{13} &= \mathbf{d}_{12}.
\end{aligned} \tag{6.3}$$

Evaluating the general form along the edges, the coefficients in (6.4) must match with (6.3).

$$\begin{aligned}
\mathbf{x}(u, 0) &= \mathbf{a}_{00} + \mathbf{a}_{10}u + \mathbf{a}_{20}u^2, \\
\mathbf{x}(0, v) &= \mathbf{a}_{00} + \mathbf{a}_{01}v + \mathbf{a}_{02}v^2, \\
\mathbf{x}(u, 1 - u) &= \mathbf{a}_{00} + \mathbf{a}_{01} + \mathbf{a}_{02}, \\
&\quad + (-\mathbf{a}_{01} - 2\mathbf{a}_{02} + \mathbf{a}_{10} + \mathbf{a}_{11})u, \\
&\quad + (\mathbf{a}_{02} - \mathbf{a}_{11} + \mathbf{a}_{20})u^2.
\end{aligned} \tag{6.4}$$

After equating the coefficients and solving, we end up with,

$$\begin{aligned}
\mathbf{a}_{00} &= \mathbf{x}_1, & \mathbf{a}_{20} &= \mathbf{c}_{12}, \\
\mathbf{a}_{10} &= \mathbf{d}_{12} - \mathbf{c}_{12}, & \mathbf{a}_{11} &= \mathbf{c}_{12} + \mathbf{c}_{13} - \mathbf{c}_{23}, \\
\mathbf{a}_{01} &= \mathbf{d}_{13} - \mathbf{c}_{13}, & \mathbf{a}_{02} &= \mathbf{c}_{23}.
\end{aligned}$$

After substitution to (6.1), we get the form (6.5).

$$\begin{aligned}
\mathbf{x}(u, v) &= \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1 - \mathbf{c}_{12})u + (\mathbf{x}_3 - \mathbf{x}_1 - \mathbf{c}_{13})v, \\
&\quad + \mathbf{c}_{12}u^2 + (\mathbf{c}_{12} + \mathbf{c}_{13} - \mathbf{c}_{23})uv + \mathbf{c}_{23}v^2.
\end{aligned} \tag{6.5}$$

We have defined a form suitable for quadratic elements. We assume the variation along an element is small and may be described using $e = 2$. Note by including the normals, the patch may be described by three nodes and normals.

Continuing to $e = 3$, we found another analytical solution from Refs. [147,148]. However, the performance is lacking in our tests. In this case, we form a tensor product of two cubic curves on the triangular element mapped to a unit square. Following the procedure from Ref. [148], we start with the curve between \mathbf{x}_1 and \mathbf{x}_2 and $0 \leq u \leq 1$. We also require the normals to be perpendicular to the tangents at the endpoints.

$$\begin{aligned}\mathbf{x}(u) &= \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \mathbf{a}_3 u^3, \\ \mathbf{x}(0) &= \mathbf{a}_0, \quad \mathbf{x}(1) = \mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3, \\ \mathbf{a}_1 \cdot \mathbf{n}_1 &= 0, \\ (\mathbf{a}_1 + 2\mathbf{a}_2 + 3\mathbf{a}_3) \cdot \mathbf{n}_2 &= 0.\end{aligned}$$

In Ref. [148], the authors make an approximation where the normal is parallel to the second derivative, $\frac{d^2 \mathbf{x}}{du^2} \propto \mathbf{n}$. This simplifies the system considerably and gives the following.

$$\begin{aligned}\mathbf{a}_2 &= \alpha \mathbf{n}_1, \\ \mathbf{a}_3 &= \frac{1}{3}(\beta \mathbf{n}_2 - \alpha \mathbf{n}_1).\end{aligned}$$

With this approximation and some algebra, we can solve for the coefficients α, β and fully describe (u) .

$$\begin{aligned}\alpha &= \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot [\mathbf{n}_1(\mathbf{n}_2 \cdot \mathbf{n}_2) + \frac{1}{2}\mathbf{n}_2(\mathbf{n}_1 \cdot \mathbf{n}_2)]}{\frac{2}{3}(\mathbf{n}_1 \cdot \mathbf{n}_1)(\mathbf{n}_2 \cdot \mathbf{n}_2) - \frac{1}{6}(\mathbf{n}_1 \cdot \mathbf{n}_2)^2}, \\ \beta &= \frac{-(\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{n}_2 - \frac{1}{3}\alpha(\mathbf{n}_1 \cdot \mathbf{n}_2)}{\frac{2}{3}(\mathbf{n}_2 \cdot \mathbf{n}_2)}.\end{aligned}$$

This also allows us to describe the unit normal using curvature $\mathbf{K}(u)$ [148].

$$\mathbf{n}(u) = \text{sgn}(u) \frac{\mathbf{K}(u)}{\|\mathbf{K}(u)\|},$$

$$\mathbf{K}(u) = \frac{1}{\left\|\frac{d\mathbf{x}}{du}\right\|^2} \left[\frac{d^2\mathbf{x}}{du^2} - \frac{\frac{d\mathbf{x}}{du} \cdot \frac{d^2\mathbf{x}}{du^2}}{\left\|\frac{d\mathbf{x}}{du}\right\|^2} \frac{d\mathbf{x}}{du} \right].$$

The sgn function always chooses the sign such that the orientation is maintained. Returning to the triangular element, we may parametrize in u any two joined edges using the above equations. Once the parametric curves is known, we may form another parametric curve in v spanning across the element, giving the tensor product. To integrate, the final step is to map coordinates to the unit square, using the following.

$$\begin{cases} u = r + s, & v = \frac{s}{r + s} & , \text{ if } r + s \neq 0, \\ u = 0, & v = 0 & , \text{ if } r + s = 0. \end{cases}$$

This procedure is inefficient. Due to the tensor product, you would ultimately get a nonic polynomial ($e = 9$) or a nonic patch. For practical implementation, we tested the 3rd order Taylor expansion of the nonic patch or truncated cubic below.

We compare both approaches in Ref. [9]. We test a spherical triangle as a single element and collectively using the flat elements forming a discretized sphere. We compare the interpolated centroids and corresponding unit normal versus the actual entities. We evaluate the centroid and unit normal at $(1/3, 1/3)$ for the quadratic patch and equivalently $(2/3, 1/2)$ for the nonic and truncated cubic patch. We also check the computed area using the area element $\iint \left\| \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v} \right\| du dv$. The spherical triangle is formed by picking 3 points in the first octant of a sphere centered at $(0, 0, 0)$ with radius R and forming the corresponding spherical arcs. This prevents an angle $> \frac{\pi}{2}$ between the normals. The spherical triangle area is

given by $R^2(A + B + C - \pi)$ where A, B, C are the interior angles. The spherical triangle parameters are given in Table 6.1.

Table 6.1: Spherical Triangle Parameters

Vertices	(8.93, 4.49, 0.27)
	(3.08, 6.60, 6.85)
	(8.63, 0.54, 5.02)
Center	(7.75, 4.37, 4.56)
Radius	10
Area	28.970

Figure 6.4 compares the parametrizations on the spherical triangle, shown in green. As shown in figure 6.4b, the nonic patch (blue) has a slight deviation, but the truncated cubic (red) does not interpolate the element. Table 6.2 gives the percent errors on the spherical triangle. We show the L_2 -norm in the case of the centroid and unit normal. Table 6.2 shows high percent error on all indicators for the truncated cubic. Since the truncated cubic does not interpolate the surface, the error in unit normal is inconclusive. Table 6.2 also shows the quadratic patch performs slightly better than the nonic patch.

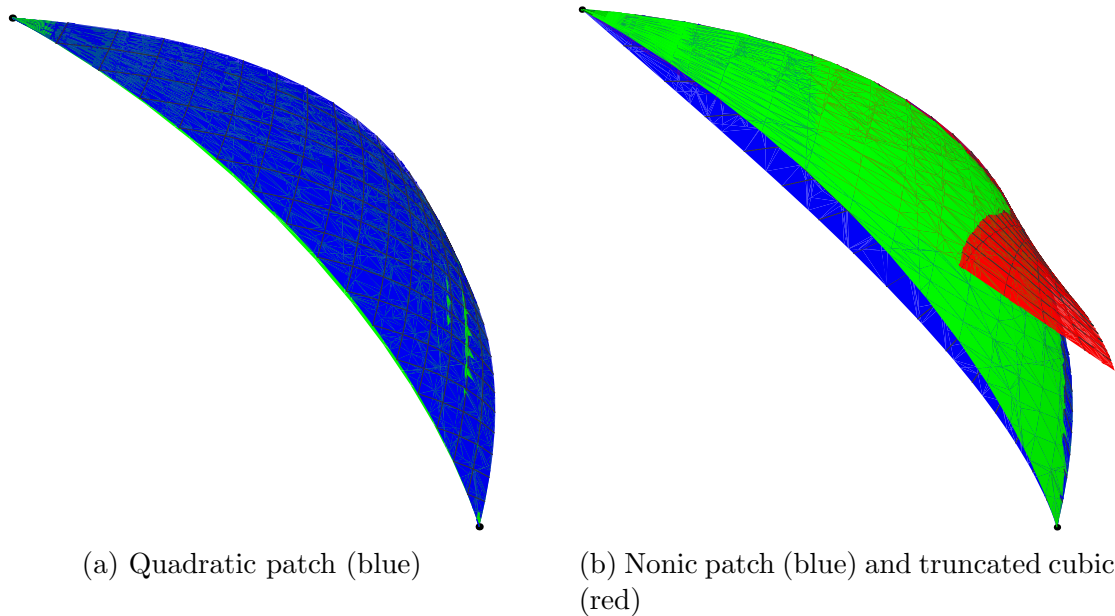


Figure 6.4: Comparison of parametric patches on the spherical triangle (green) given by Table 6.1 [9]. In Figure 6.4a, the quadratic patch (blue) deviates slightly in the center. In Figure 6.4b, the patch using a nonic polynomial (blue) vs. its truncated cubic (red) deviate significantly in this case.

Table 6.2: Percent Errors On Spherical Triangle

	% error in centroid	% error in normals	% error in area
Quadratic	0.79	0.44	-0.33
Nonic	1.63	6.49	1.53
Truncated cubic	4.19	30.5	13.6

We consider practical elements by applying the parametrizations on a discretized sphere. Due to computational limits using the nonic patch, we could only apply the quadratic and truncated cubic form. Figure 6.5 shows the standard deviation between the x-component of the computed and actual centroids and unit normals versus the number of elements. We found the behavior is similar for x, y, z . The error bound is decreasing rapidly for the quadratic patch. However, the truncated cubic patch maintains a relatively high error. Figure 6.5b shows the unit normals from the truncated cubic form does not meet our needs.

Figure 6.6 shows the area percent error versus number of elements. The truncated cubic patch behaves about the same as the flat panel elements while the quadratic patch gives better results. When we compare runtimes versus number of elements in figure 6.7, the times for the quadratic and truncated cubic rapidly increase due to the integration time.

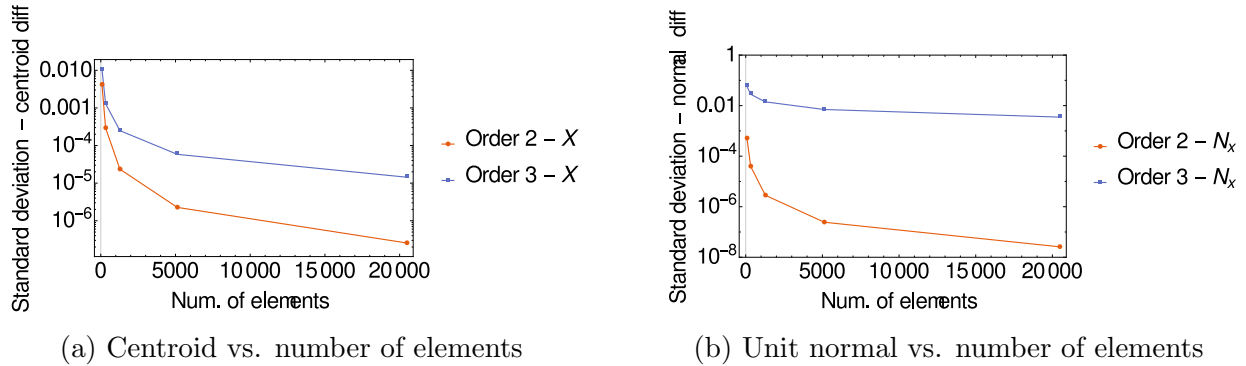


Figure 6.5: Standard deviation in the difference of (6.5a) the centroid and (6.5b) the unit normal [9]. The error bound for the quadratic patch quickly shrinks until $M \sim 5000$.

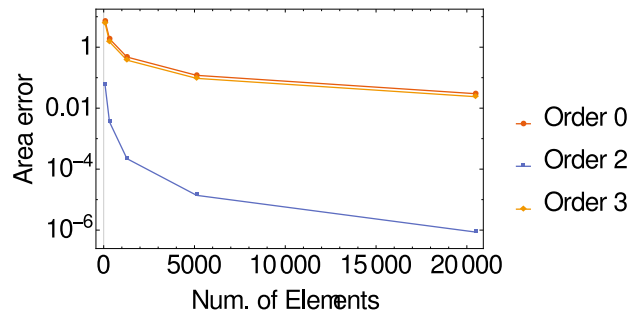


Figure 6.6: Percent error in the area vs. number of elements for the flat panel, quadratic, and truncated cubic parametrization [9]. The truncated cubic gives essentially the same error as the flat panel discretization.

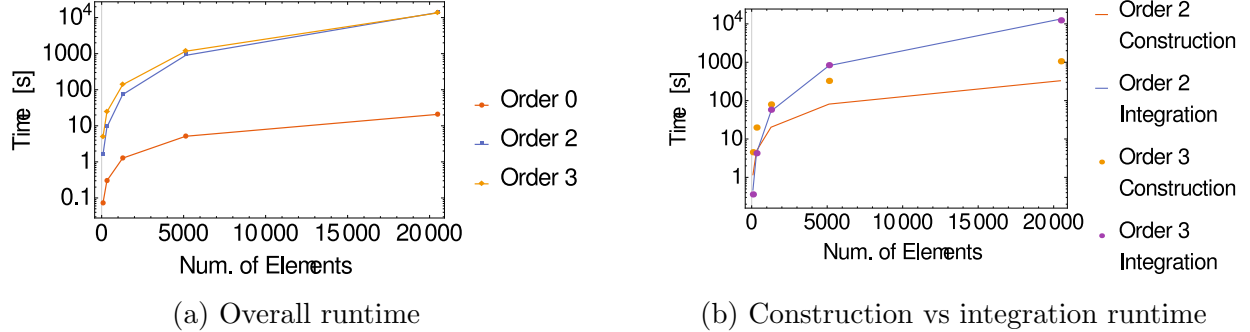


Figure 6.7: Runtime comparison between the different methods [9]. Overall, quadratic and cubic are similar due to the integration step.

From our results, the analytic cubic form does not work for our needs. The computed points and unit normals tend to have a high error and perform about the same as flat panel elements. However, the analytic quadratic form from Ref. [145] works extremely well. We can expect the runtime to improve when we use an efficient integration scheme.

6.4 Higher-order interpolation

From our results, we found an analytical solution for an interpolating polynomial with $e = 2$. However, this is the limit for analytical solutions. Instead, from Ref. [149], we may build a Vandermonde-like rectangular matrix \mathbf{M} including the unit normals. In general, \mathbf{M} is ill-conditioned, but we numerically solve it using singular value decomposition (SVD), described in Appendix B.1. In Refs. [150,151], the authors discuss surfaces with Pythagorean normal fields or PN surfaces, defined below.

Definition 1. Let \mathcal{X} be a real algebraic surface in \mathbb{R}^3 and let us denote by $\mathbf{n}_{\mathbf{p}} \in S^2$ a unit normal vector on sphere S^2 at regular point $\mathbf{p} \in \mathcal{X}$. Then the δ -offset $O_{\delta}(\mathcal{X})$ of \mathcal{X} is defined as the closure of the set $\{\mathbf{p} \pm \delta \mathbf{n}_{\mathbf{p}} \mid \mathbf{p} \in \mathcal{X}\}$.

If \mathcal{X} is rational with $\mathbf{x} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ its parametrization, the parametrized form of the offset is $\mathbf{x}(u, v) \pm \delta \mathbf{n}_{\mathbf{x}}(u, v)$. The surface is called a rational PN surface if $\|\mathbf{x}_u \times \mathbf{x}_v\|^2 = \sigma^2$ with rational function $\sigma(u, v)$ [150]. We may also say if a rational normal field exists, we may interpolate a rational PN surface of parametric form. As defined, we may find a general parametric surface. The parametrization is more fully explored in Ref. [149] with polynomial PN surfaces instead. We start with a polynomial parametrized surface $\mathbf{x}(u, v)$. In Ref. [149], the authors define $\Gamma(\mathbf{x}_1, \dots, \mathbf{x}_k)$ as the Gram determinant or area element for \mathbb{R}^2 . For polynomial vector fields and regardless of parametrization, the area element has $\Gamma(\mathbf{x}_u, \mathbf{x}_v) = f^2 \Gamma(\mathbf{n})$ for non-zero scalar $f(u, v)$. If we start with a k^{th} -degree polynomial normal field $\mathbf{n}(u, v)$ such that $\|\mathbf{n}(u, v)\|^2$ is a perfect square, we must have a polynomial area element. Also, given \mathbf{n} , we can find polynomial vector fields $\mathbf{x}_u, \mathbf{x}_v$ such that (6.6) are satisfied.

$$\begin{aligned} \mathbf{x}_u \cdot \mathbf{n} &= 0, \\ \mathbf{x}_v \cdot \mathbf{n} &= 0, \\ \frac{\partial \mathbf{x}_u}{\partial v} - \frac{\partial \mathbf{x}_v}{\partial u} &= 0. \end{aligned} \tag{6.6}$$

In Ref. [149], the authors continue by showing how they can interpolate the surface $\mathbf{x}(u, v)$ by first interpolating $\mathbf{n}(u, v)$. In their examples, they chose stereographic projection. However, as stated, the parametrization is freely chosen.

For our case, we are given k Cartesian nodes \mathbf{b} and outward unit normals \mathbf{n} on a surface embedded in \mathbb{R}^3 sufficient to form a polynomial $\mathbf{P}(u, v)$ of degree e . We assume the nodes are matched with corresponding (u, v) on the unit triangle such that distortion is minimized.

We wish to satisfy the following conditions for $l = 1, \dots, k$. We need to consider every component, $d = 1, 2, 3$, separately due to the change in dimensionality of the conditions.

$$\mathbf{a} \equiv (a_1, a_2, a_3),$$

$$P_d(u, v) = \sum_{i=0}^e \sum_{j=0}^i a_{d,i-j,j} u^{i-j} v^j, \quad (6.7a)$$

$$n_d(u_l, v_l) = n_{d,l}, \quad (6.7b)$$

$$P_d(u_l, v_l) = b_{d,l}, \quad (6.7c)$$

$$\mathbf{P}_u \cdot \mathbf{n}(u, v) = 0, \quad (6.7d)$$

$$\mathbf{P}_v \cdot \mathbf{n}(u, v) = 0. \quad (6.7e)$$

Eq. (6.7a) gives the general polynomial form for each component. We evaluate this to form the interpolation matrix that satisfies the conditions (6.7b)–(6.7e). The Vandermonde-like properties come from condition (6.7c). The procedure in Ref. [149] starts with finding an interpolated normal field $\mathbf{n}(u, v)$. This would lead to two matrices to solve, first giving a polynomial $\mathbf{n}(u, v)$ using (6.7b), then $\mathbf{P}(u, v)$ using the rest. Note the matrix is generally square for $\mathbf{n}(u, v)$ and rectangular for $\mathbf{P}(u, v)$. Eqs. (6.7d) and (6.7e) imply it is possible to evaluate extra nodes (u, v) such that the second matrix is square. In practice, we found this caused significant distortion, though this may depend on choice of extra nodes. In the end, we decided to form only the rectangular system \mathbf{M} without $\mathbf{n}(u, v)$ and solve for $\mathbf{P}(u, v)$ from (6.7c)–(6.7e). \mathbf{M} is formed by ordering columns and filled-in zero elements such that

$a_{d,i-j,j}$ is a single unknown vector of length $m = 3\binom{e+2}{2} = \frac{3(e+1)(e+2)}{2}$. The matrix equation is shown in (6.8).

$$\begin{aligned}
\mathbf{u}_l^i \mathbf{v}_l^j &\equiv \mathbf{I}_{3 \times 3} \left. u^i v^j \right|_{u=u_l, v=v_l}, \\
\mathbf{u}^i \mathbf{v}^j \mathbf{n}_{l,u} &\equiv \begin{bmatrix} n_{1,l} & n_{2,l} & n_{3,l} \end{bmatrix} \left. \frac{\partial u^i v^j}{\partial u} \right|_{u=u_l, v=v_l}, \\
\mathbf{M} \mathbf{a} &= \mathbf{b}, \\
\begin{bmatrix} \mathbf{u}_1^0 \mathbf{v}_1^0 & \mathbf{u}_1^1 \mathbf{v}_1^0 & \mathbf{u}_1^0 \mathbf{v}_1^1 & \cdots & \mathbf{u}_1^1 \mathbf{v}_1^{e-1} & \mathbf{u}_1^0 \mathbf{v}_1^p \\ \vdots & & & & & \\ \mathbf{u}_k^0 \mathbf{v}_k^0 & \cdots & & & & \\ \mathbf{u}^0 \mathbf{v}^0 \mathbf{n}_{1,u} & \cdots & & & & \\ \vdots & & & & & \\ \mathbf{u}^0 \mathbf{v}^0 \mathbf{n}_{k,u} & \cdots & & & & \\ \mathbf{u}^0 \mathbf{v}^0 \mathbf{n}_{1,v} & \cdots & & & & \\ \vdots & & & & & \\ \mathbf{u}^0 \mathbf{v}^0 \mathbf{n}_{k,v} & \cdots & & & & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ \vdots \\ b_{3,k} \\ \vdots \\ \mathbf{0}_{2k \times 1} \end{bmatrix}. \tag{6.8}
\end{aligned}$$

We call this normal vector field (NVF) interpolation. We consider the minimum number of nodes to form \mathbf{P} . From Ref. [149], we prescribe an o^{th} -degree polynomial for $\mathbf{n}(u, v)$, where $k = \binom{o+2}{2}$. For order e , \mathbf{P} has $3\binom{e+2}{2}$ coefficients. From (6.8), \mathbf{M} is $5\binom{o+2}{2} \times 3\binom{e+1}{2}$. \mathbf{M} is solvable if $5\binom{o+2}{2} \leq 3\binom{e+1}{2}$. Solving the minimum case with positive root, we get (6.9).

$$o = \left\lfloor \frac{1}{10} \left(-15 + \sqrt{5} \sqrt{5 + 12p + 12p^2} \right) \right\rfloor. \tag{6.9}$$

We have the minimum number of nodes for e^{th} -order elements. Eq. (6.9) does not work for $e = 3$ because the corresponding $\mathbf{n}(u, v)$ is inaccurate. As a final note, we address the orientation for the unit normal \mathbf{n} . We choose the outward normals, but the corresponding

sign is unclear. To choose the correct sign, we must compare a computed normal direction against another vector. We assume the given normals always point outwards and an element does not have widely varying normals. We choose the sign from the dot product between a computed normal and the mean of the given normals.

6.5 General quadrature

We have derived a method to interpolate high-order elements for $e \geq 4$. We must consider a suitable quadrature to apply Nyström discretization. Quadrature takes the form, $\int f(x) dx \approx \sum_i w_i f(x_i)$. Commonly, Gaussian quadrature is used due to its efficiency and accuracy. However, Gaussian quadrature is originally derived for 1-D integrals of the form, $\int_{-1}^1 f(x) dx$. Any interval $[a, b]$ may be changed to $[-1, 1]$ through a change of variables as shown below.

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) dx.$$

Another common alternative is trapezoidal quadrature, also derived for 1-D. To apply to 2-D, Fubini's theorem is used to form double sums to approximate the surface integral, thus retaining the properties from the 1-D form. To illustrate the performance, we reproduce an example from Ref. [130] in Table 6.3. Table 6.3 gives the Nyström error for a 1-D integral equation of the second kind (6.10) with $\lambda = 50$.

$$\lambda x(t) - \int_0^1 e^{st} x(s) ds = y(t). \quad (6.10)$$

Table 6.3 shows slow convergence for the trapezoidal quadrature but rapid convergence for the Gaussian-Legendre quadrature. We can expect this behavior with higher dimension

integrals using Fubini's theorem. However, there are some practical complications due to using triangular elements.

Table 6.3: L_∞ -error for (6.10) using Nyström method and trapezoidal (T) or Gaussian-Legendre (GL) quadrature. n is number of terms in the quadrature. GL quadrature performs significantly better.

n_T	$\ x - x_n\ _\infty$	n_{GL}	$\ x - x_n\ _\infty$
2	5.44×10^{-3}	1	9.81×10^{-3}
4	1.37×10^{-3}	2	2.18×10^{-4}
8	3.44×10^{-4}	3	1.86×10^{-6}
16	8.61×10^{-5}	4	8.47×10^{-9}
		5	2.39×10^{-11}

The Nyström method for surface integrals and Fubini's theorem leads to $\sum_i^a \sum_j^b$ over an element, where a, b are number of terms in the quadrature. The Nyström method requires interpolating the density from (5.11) and evaluating where needed if the quadrature nodes change in the sum. To avoid the additional integration, we fix the quadrature nodes. If there are c elements, a Nyström BEM matrix row would have $c \cdot a \cdot b$ elements. To maintain uniqueness, we would need $d = ab$ evaluation nodes per element. We may equate the evaluation nodes with the interpolation nodes. However, if the interpolation and quadrature node set coincide, we get additional singularities in the BEM matrix. The quadrature rule must be selected to avoid that. Let's consider using quadrature over a triangle. Ref. [152] mentions the surface is oversampled using a tensor product quadrature formula, which results from Fubini's theorem and $a = b$. The problem is constructing the set of d points that maintains the convergence from Gaussian quadrature on a triangle and having $d = \frac{(o+1)(o+2)}{2}$, where o comes from (6.9). In Refs. [152]– [153], the authors discuss using the Fekete points, which are the Gauss-Lobatto points for $[-1, 1]$ in 1-D. The Fekete points are shown to improve the conditioning and accuracy of quadrature, much like with Gaussian quadrature [152]. In addition, the Fekete points may be found in any domain and polynomial degree. The Fekete

points are defined such that they maximize the determinant of a Vandermonde matrix [152] for a given polynomial. Several algorithms, e.g., Ref. [154], are proposed to find the Fekete point set in general as optimization problems. Unfortunately, these are not computationally feasible for our purposes.

With our interpolation method, we need to know what nodes to select a priori to avoid intersections between interpolation and quadrature node set. Gaussian quadrature weights were originally derived based on the point set which minimizes the polynomial interpolation error using 1-D Lagrange basis functions [134]. Using orthogonal polynomials and their roots, such as Legendre polynomials, gives the rapid convergence in the quadrature. We may consider the same strategy using Lagrange polynomials. The general derivation for the weights starts with the Lagrange basis functions. In Ref. [155], the authors derive a quadrature rule for the integral over $[-1, 1] \times [-1, 1]$ and map to Bézier triangles. The nodes are chosen using a Newton-Cotes rule, which is described in Ref. [156]. To find a general method on a triangle, we use a similar derivation for the quadrature weights with the 2-D Lagrange basis functions on the triangle and bypass the mapping. To prevent additional singularities, we must find interpolation and quadrature node sets without common points. Some possible sets for quadrature include the roots of Lobatto and Chebyshev polynomials. We follow Ref. [157], where the authors symmetrically map the Lobatto polynomial roots onto the triangle for interpolation. We instead use the p th-order Chebyshev nodes on $[0, 1]$, c_i , with the same symmetric mapping to (u, v) . The formula for the nodes and the mapping is given in (6.11). The Chebyshev nodes are the roots of the 1-D Chebyshev polynomials of the first kind [158]. The node set provides stable quadrature weights for 1-D integration [134,158].

That is, the weights are positive and sum to 1. This is not strictly true after the mapping. However, our tests show this node set is adequate for our quadrature up to $p = 20$.

$$\begin{aligned}
c_i(p) &= \frac{1}{2} \left(1 + \cos \left(\frac{\pi(2i+1)}{2(p+1)} \right) \right), \quad i = p, \dots, 0, \\
u &= \frac{1}{3} (1 + 2c_j - c_i - c_k), \\
v &= \frac{1}{3} (1 + 2c_i - c_j - c_k), \\
i &= 1, \dots, p+1, \quad j = 1, \dots, p+2-i, \quad k = 1, \dots, p+3-i-j.
\end{aligned} \tag{6.11}$$

When $p = 3n$, $n = 1, 2, \dots$, the middle quadrature node coincided with our interpolation node. Before computing the Lagrange basis, we shift the quadrature node \mathbf{u}_c by $k(\mathbf{u}_{c+1} - \mathbf{u}_c)$, where \mathbf{u}_{c+1} is the next node and $k = 1/10$. We choose the factor k based on empirical studies. While not optimal, we have derived a general high-order interpolation and quadrature. Now that the nodes are selected, we compute the weights using the Lagrange interpolating polynomial formed from Lagrange basis functions $l_i(u, v)$.

$$\begin{aligned}
f(u, v) &\approx \sum_{i=1}^n l_i(u, v) f_i, \\
\int f(u, v) \, dudv &\approx \sum_{i=1}^n w_i f_i, \\
\Rightarrow w_i &= \int l_i(u, v) \, dudv.
\end{aligned}$$

Even though we selected the Chebyshev roots, the mapped nodes did not display optimal convergence. We did notice the weights tended to be positive until higher orders. The flexibility of the interpolation and quadrature is necessary to implement a high-order Nyström BEM.

Our implementation involves integrating polynomials. The quadrature is sufficient for applying Nyström discretization. We use (6.12) to map the unit triangle to the unit square.

$$\int_0^1 \int_0^{1-u} f(u, v) \, dv \, du = \int_0^1 \int_0^1 (1-u) f(u, (1-u)v) \, dv \, du. \quad (6.12)$$

Using the change of variables and interpolation polynomial $\mathbf{P}(u, v)$, our integral becomes,

$$\begin{aligned} \int f(\mathbf{y}) \, d\Gamma &= \int_0^1 \int_0^{1-u} \mathbf{P}(u, v) \|\mathbf{n}(u, v)\| \, dv \, du, \\ &= \int_0^1 \int_0^1 \mathbf{P}(u, (1-u)v) \|\mathbf{n}(u, (1-u)v)\| (1-u) \, dv \, du. \end{aligned} \quad (6.13)$$

Using this form, the discrete BEM equation is given in (6.14) where c is the number of elements, d is the number of quadrature nodes per element, $\mathbf{x}_i : i = 1, \dots, cd$ are the interpolation nodes, $\mathbf{y}_k : k = 1, \dots, cd$ are the quadrature nodes, w_j are the quadrature weights, and all terms are collected under a single sum.

$$\begin{aligned} \psi(\mathbf{x}_i) &= C(\mathbf{x}_i) \eta(\mathbf{x}_i) + \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}_i, \mathbf{y}) \eta(\mathbf{y}) \, d\Gamma(\mathbf{y}), \\ &= C(\mathbf{x}_i) \eta(\mathbf{x}_i) + \sum_{k=1}^c \int_0^1 \int_0^1 (1-u) \frac{\partial G}{\partial n_k}(\mathbf{x}_i, \mathbf{y}_k(u, (1-u)v)) \|\mathbf{n}_k(u, (1-u)v)\| \eta(\mathbf{y}_k) \, dv \, du, \\ &= C_i \eta_i + \sum_{j=1}^{cd} w'_j \frac{\partial G_{ij}}{\partial n_j} \eta_j. \end{aligned} \quad (6.14)$$

Eq. (6.14) has all terms under a single sum, so we may precompute all nodes and weights $w'_j \equiv w_j \|\mathbf{n}_j\|$ after interpolating the elements. The factor $(1-u)$ is included in the weights w_j when computing $\int l_j(u, v) \, dv \, du$. This simplifies our implementation when setting up the matrix and helps when accelerating the solver using the FMM since we only consider nodes

rather than elements. Ultimately, we implement (6.14) to form the complete system matrix equation for $\boldsymbol{\eta}$, shown below with $m \equiv cd$ and $\frac{\partial G_{ij}}{\partial n_y} \equiv G_{ij,n_y}$.

$$\boldsymbol{\psi} = \mathbf{M}\boldsymbol{\eta},$$

$$\begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{pmatrix} = \begin{pmatrix} C_1 + w'_1 G_{11,n_y} & w'_2 G_{12,n_y} & \cdots & w'_m G_{1m,n_y} \\ w'_1 G_{21,n_y} & C_2 + w'_2 G_{22,n_y} & & \vdots \\ \vdots & & \ddots & \\ w'_1 G_{m1,n_y} & \cdots & & C_m + w'_m G_{mm,n_y} \end{pmatrix} \begin{pmatrix} \eta \\ \eta_2 \\ \vdots \\ \eta_m \end{pmatrix}.$$

6.6 Performance of the interpolation and quadrature

The NVF interpolation procedure works for $e \geq 4$. When $e = 2$, the quadratic patch has better performance. So, our implementation uses the quadratic patch when $e = 2$ and the general interpolation when $e \geq 4$. To use $e = 3$, the NVF interpolation requires $o = 2$ from (6.9). The corresponding matrix system is ill-conditioned but gives acceptable accuracy in the interpolation.

Table 6.4: Triangular Surface Vertices For Each Interpolation Test Function.

Function	Triangle vertices (x,y)
$\sqrt{1 - x^2 - y^2}$	(0.01, 0.02), (0.3, 0.02), (0.01, 0.4)
$-x^4 y^3 - x^2 + xy + y$	(-0.2, -0.1), (-0.2, 0.4), (0.5, -0.1)
$\cos(x^3) \sin(y^2)$	$(-\frac{\pi}{6}, -\frac{\pi}{6}), (-\frac{\pi}{6}, \frac{\pi}{6}), (\frac{\pi}{6}, -\frac{\pi}{6})$

To test the NVF interpolation, we choose three functions, $\sqrt{1 - x^2 - y^2}$, $-x^4 y^3 - x^2 + xy + y$, $\cos(x^3) \sin(y^2)$ over a small triangular surface. The vertices are given in Table 6.4. We test this scheme against Vandermonde interpolation with only coordinates using SVD to control

the conditioning in both cases. We compare the behavior of the interpolated quantities using the maximum difference over a triangular grid of 3403 points on three analytical surfaces. Figures 6.8, 6.9, and 6.10 show the comparison over the surfaces with Tol being the SVD tolerance.

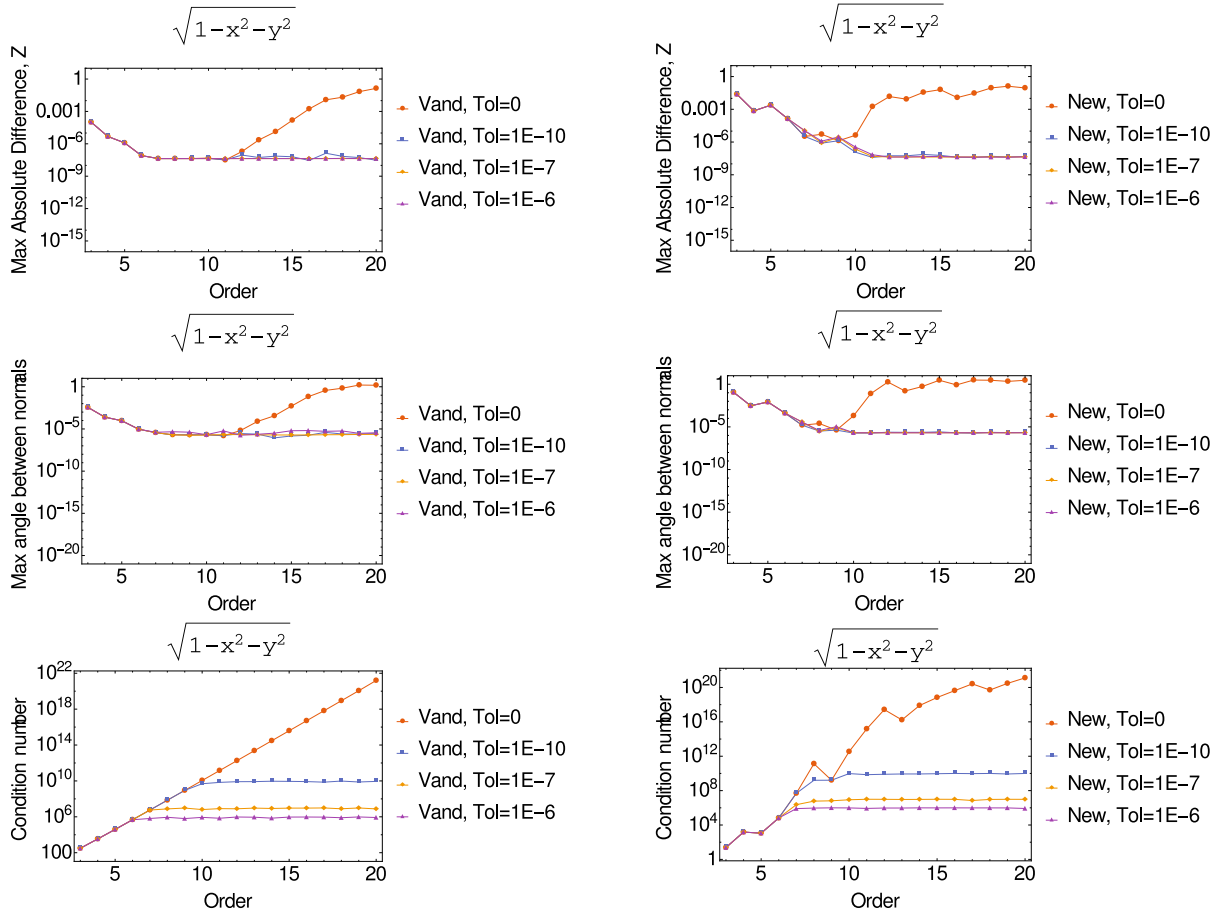


Figure 6.8: Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a spherical triangle, $R = 1$.

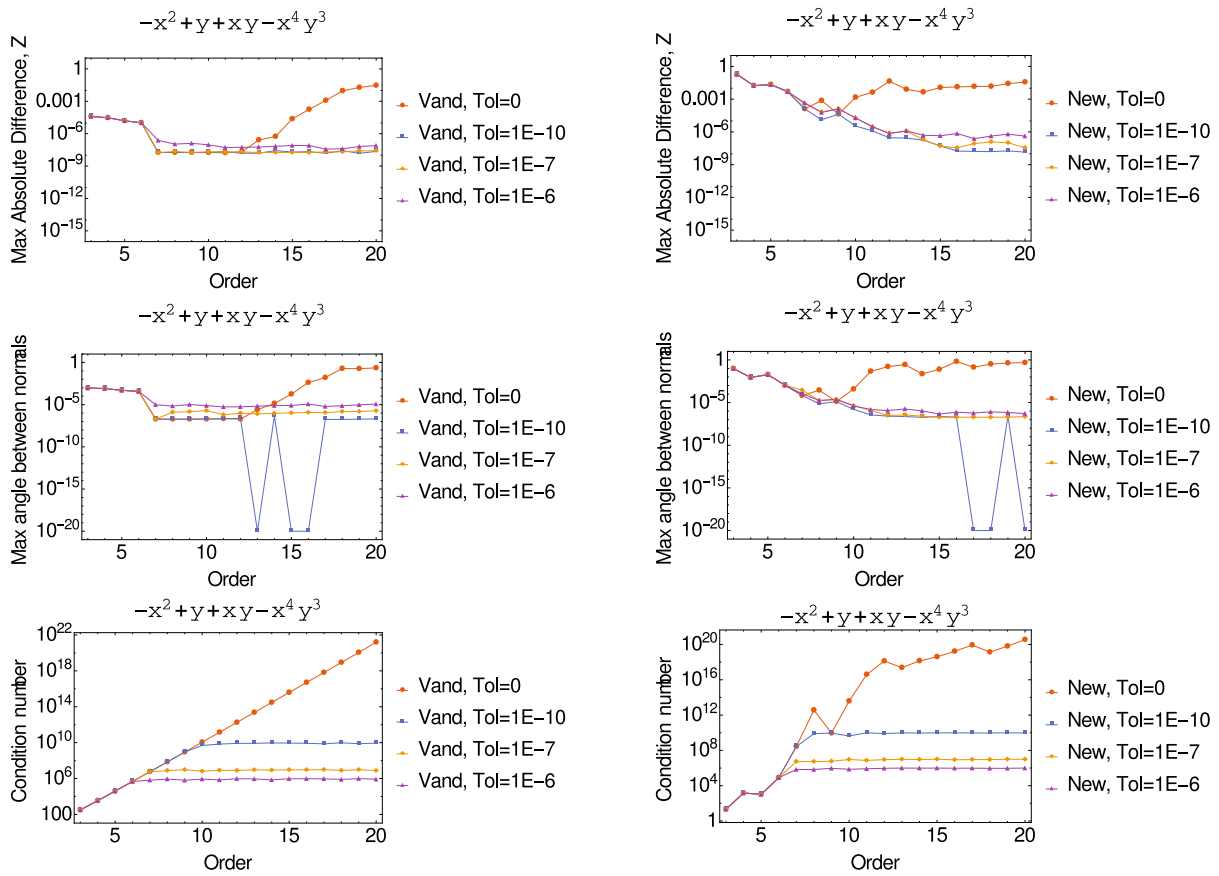


Figure 6.9: Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a polynomial surface, $z(x, y) = -x^4 y^3 - x^2 + xy + y$.

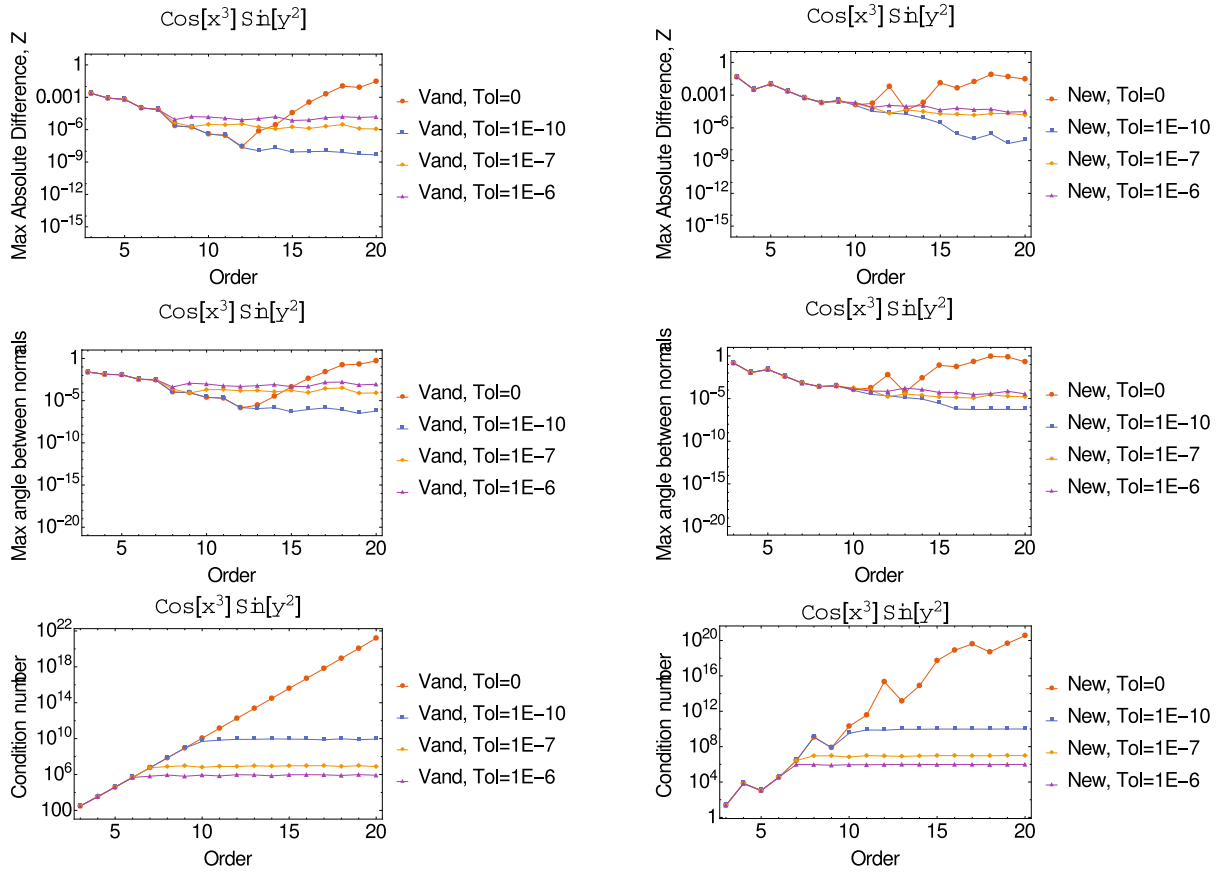


Figure 6.10: Comparison of basic Vandermonde interpolation (left) and the NVF interpolation (right) using the maximum difference on a triangular grid on a trigonometric surface, $z(x, y) = \cos x^3 \sin y^2$.

To compare the analytical and interpolated quantities, we test the difference in the coordinates and the angle between normals as well as the conditioning. We show the maximum absolute difference in z as it displays the highest error of the coordinates. Overall, Vandermonde interpolation shows better stability than the NVF interpolation. While the conditioning grows faster, the SVD truncates the smallest singular values and maintains the accuracy up to order 20.

We implement two versions of the interpolation. For Chapter 8, we use the Vandermonde interpolation unless stated otherwise. We set the SVD tolerance at 1×10^{-10} . The integration requires mainly the length of the normal at different positions. The normal direction is

sensitive to the errors in position. In the BEM, we require accurate normal direction for $\frac{\partial G}{\partial n}$. We have established a possible quadrature rule for general polynomials. We wish to compare against Gaussian-Legendre quadrature for an analytical case. We chose to integrate the functions, $f(x, y) = -x^4y^3 - x^2 + xy + y$ and $f(x, y) = \cos(x^3) \sin(y^2)$, on a unit triangle mapped to the unit square $[0, 1] \times [0, 1]$ by (6.12). The integral is then (6.13) with $\mathbf{P}(u, v) = 1$. Figure 6.11 compares the quadrature behavior. We checked the integral with and without the change of variables and confirmed they are equivalent.

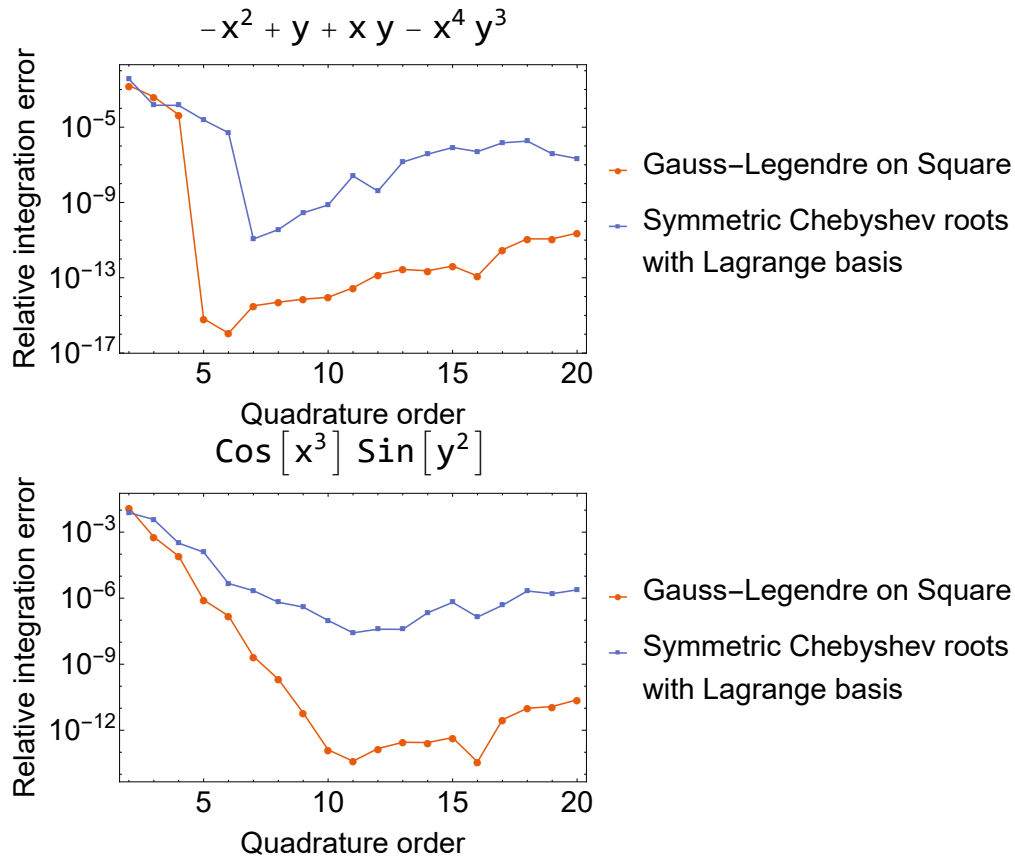


Figure 6.11: Comparison of the quadrature with mapped symmetric Chebyshev roots and Gaussian-Legendre quadrature. Integral is transformed from the unit triangle to the unit square.

From our tests, the high-order interpolation and quadrature on the triangle lose some accuracy when the order is greater than 9. Our cases of interest do not need higher element order

than 10, so this is acceptable. The Fekete point set is promising, but there is no feasible algorithm for finding the set in general yet. The general techniques we have described allow our implementation to form flexible algorithms compatible with the fast multipole-accelerated Nyström discretization.

CHAPTER 7

POISSON INTEGRAL SOLVER WITH CURVED SURFACES

We develop the Poisson Integral Solver with Curved Surfaces (PISCS) for the study of high-intensity beams in arbitrary structures. PISCS applies the Nyström boundary element method accelerated by the fast multipole method to solve the Poisson equation with complicated boundaries. Figure 7.1 gives a condensed flowchart of the program.

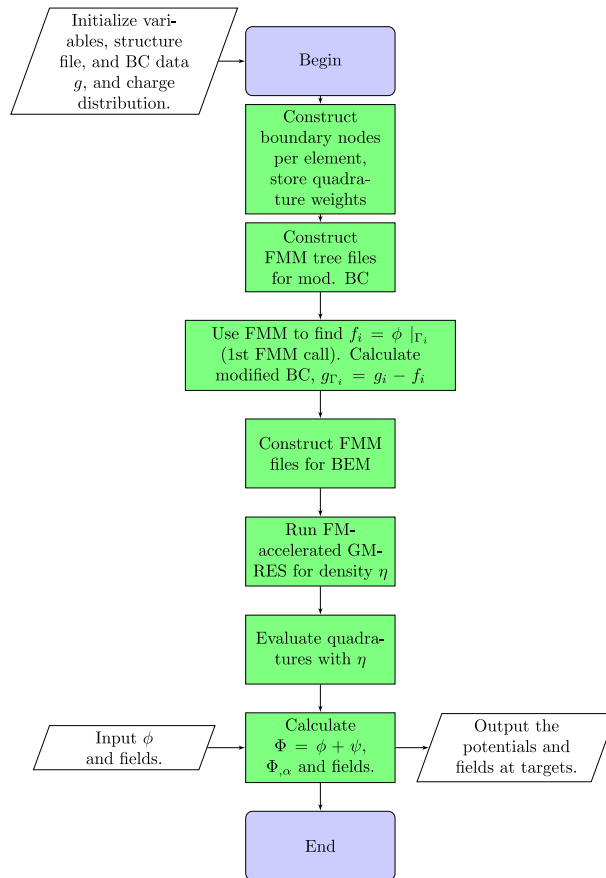


Figure 7.1: The flowchart of PISCS. There are four major blocks based on runtime: Initialization and element setup, computation of the modified boundary conditions, solving the linear system to match the BCs, and evaluating the density. Finally, the output potentials and fields are summed and written to files with negligible runtime.

PISCS is written in COSYScript for COSY Infinity v9.2 from Michigan State University (http://bt.pa.msu.edu/index_cosy.htm) [89]. The package may be found on the NICADD Beam Physics Code Repository (<http://niu.edu/beamphysicscode/>). We assume COSY has been installed and setup correctly. At this point, the COSY beam physics package is separate from PISCS. A manual and examples are on the NICADD Beam Physics Code Repository for more details on usage. In this chapter, we describe the implementation and discuss using the Poisson BVP for solving the equations of motion. Note PISCS uses the Cartesian coordinate system based on the structure, not the beam. This is relevant when discussing the merge with beam physics tracking codes. In the transverse case, we assume the structure's longitudinal axis is aligned with the beam's longitudinal axis, which gives the same transverse coordinates. This comes from cylindrically symmetrical transport designs [6]. We discuss some approaches to evaluating the BEM near the surface. We identify some bottlenecks and parts that are parallelized. We explain how to combine PISCS with the space charge package, COMFY [46], and discuss integrating PISCS into the 3-D FMM symplectic tracking code, PHAD, [7].

7.1 PISCS Input

The parameter file is an ASCII text file where each line is a parameter (default is `parameters.dat`). The format is given in Figure 7.2.

- 1: Temporary folder
- 2: Particle (or evaluation points) input file
- 3: Source species (Filename for multispecies, else keywords)
- 4: Structure file
- 5: Clustering parameter q (Default= 0.05)
- 6: FMM executable
- 7: FMM order p
- 8: Boundary element order e
- 9: Boundary condition type
- 10: Boundary conditions (file or number)
- 11: Preconditioning (Optional)
- 12: Field type (Optional)

Figure 7.2: PISCS input file format. Each parameter is on a separate line. Optional parameters can be left blank.

The temporary folder stores all temporary files and must exist. The default is `./tmp/`. The particle file contains all the positions in Cartesian coordinates with one line per particle. The source species may be a filename for multispecies where each line is the charge of the corresponding particle. Otherwise, the following keywords may be used.

1. `protons`
2. `electrons`
3. `positrons`
4. `antiprotons`
5. `targets`

Most are self-explanatory. `targets` means the particle positions are BEM evaluation points and assume no charged particles in the region.

The structure file gives the number of elements and nodes per element according to the ordering in chapter 6. We assume the correct number of nodes per element is given for the prescribed element order. The first line is the number of elements. Each line afterwards contains the position and unit normal of a node. There is the option to input ASCII `.st1`

files, but this is restricted to flat elements and not recommended. The procedure to generate the structure file requires the program `gmsh`. `gmsh` is an open source high order finite element mesh generator and further details may be found in [159,160] (<http://gmsh.info/>). We use `gmsh` to either form or manipulate structures from CAD files and generate high order surface meshes. The `gmsh` mesh order and our element order e match, but the number of nodes per element is different due to including normals in our formulation. The node ordering is also different. A COSYScript program, `gmshparser.fox`, is included in the PISCS package for the normal vector field interpolation. `gmshparser.fox` converts `gmsh`'s `.msh` format to PISCS's format. We recommend a high order in `gmsh` for your desired element order. Use the program `gmshrearrange.fox` for Vandermonde interpolation without normals. For `gmshrearrange.fox`, the `gmsh` order is the same as the PISCS order. Constant and linear order are treated as constant and output flat panel elements. An optional scaling factor scales all the dimensions (default=1). The default for the output file is `struct.dat`. The format for the structure file is shown in figure 7.3. Each element block is comprised of the necessary nodes.

$$\begin{array}{l} \# \text{ elements} \\ \left. \begin{array}{l} (x \ y \ z \ n_x \ n_y \ n_z) \\ \# \text{ nodes per element} \end{array} \right] \# \text{ elements} \end{array}$$

Figure 7.3: Format for structure file. First line is the number of elements.

The clustering parameter q controls the maximum number of targets in a neighborhood for any box and determines the level of subdivision in the adaptive FMM. If $q \geq 1$, q is the number of particles. If $q < 1$, then it is the fraction of total number of particles.

The clustering parameter is only used when computing the modified boundary conditions. Otherwise, we set q such that the maximum number of targets in a neighborhood is ≤ 1000 to speedup the BEM solver and evaluation. The FMM executable refers to an external program called `fmmcpp` written to construct the octree files for the adaptive FMM. The program is written in C++ and the latest stable source code plus makefile is packaged with PISCS. `fmmcpp` must have been compiled beforehand. The boundary condition (BC) types are Dirichlet (0 or keyword `Dirichlet`) by default and Neumann (1 or keyword `Neumann`). The boundary conditions may be a filename containing the BC values for each corresponding node in the structure file or a number which assumes constant BC. The preconditioning flag is optional (0 for none, 1 to compute an initial guess with low order FMM). The field type is electrostatic (E) or magnetostatic (B). The default is electrostatic.

7.2 Computing the nodes and weights

PISCS first reads in the parameters and sets up the elements. If $e = 0, 1$, each element consist of its centroid and the mean unit normal. If $e > 1$, PISCS performs the interpolation as discussed in Chapter 6 with DA variables $(d_u, d_v) = (u, v)$. This allows us to use the COSY DA operations. From the interpolation polynomials $\mathbf{P}(u, v)$, we compute the interpolation and quadrature nodes for every (u, v) according to the corresponding rule. Similarly, the normals at the nodes are computed as needed. The lengths is stored with the quadrature weights. PISCS computes the quadrature weights using Lagrange basis functions. However,

we could not find a suitable general-order derivation for the basis functions on the unit triangle. Instead, since the basis function property requires,

$$\ell_j(u_i, v_i) = \delta_{ij},$$

$$\boldsymbol{\ell}(u, v) \equiv \begin{pmatrix} \ell_1(u, v) \\ \vdots \\ \ell_n(u, v) \end{pmatrix},$$

we may form the Vandermonde system for n nodes,

$$\mathbf{P}_e(u, v) \equiv \begin{pmatrix} 1 & u & v & \dots & u^1 v^{e-1} & u^0 v^e \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{P}_e(u_1, v_1) \\ \vdots \\ \mathbf{P}_e(u_n, v_n) \end{pmatrix} \mathbf{a} = \mathbf{I}.$$

where \mathbf{a} are the coefficients, \mathbf{I} is the identity matrix, and invert using singular value decomposition. Then, $\boldsymbol{\ell}(u, v) = \mathbf{P}_e \cdot \mathbf{a}$ gives the vector of 2-D basis functions. We assume the element order is small for our purposes. The method may be used for high order, but it rapidly grows inefficient as $e > 5$. The basis functions are integrated on the unit triangle to compute the quadrature weights. We used a modified adaptive Runge-Kutta integrator from the COSY beam physics package for accuracy [89]. The integrator is modified to integrate the 2-D Lagrangian basis functions on the unit triangle by Fubini's theorem [134]. The transformation $(u, v) \rightarrow (u, (1-u)v)$ allows integration over the unit square. This setup gave acceptable performance for tolerance of $\varepsilon = 10^{-10}$. From (6.14), the integral requires $w_j \|\mathbf{n}_j\|$, which can be stored together as w'_j .

7.3 Implementation

We have precomputed the quadrature quantities including the area element. Some modifications are made to the multipole expansion and P2P procedures from [7] to accommodate the different kernel function, $\frac{\partial G}{\partial n}$, but the translation operators are the same. For the FMM to work with GMRES, we structure the summation similar to the Coulomb potential, $\sum_j q_j G(\mathbf{x}, \mathbf{y}_j)$. However, rather than charges, we have the trial density vector $\boldsymbol{\eta}$ from the solver. $\frac{\partial G}{\partial n} w'_j$ takes the place of the expanded kernel function. The rest is straight-forward. Using the techniques from chapter 6, we form a square Nyström matrix of size number of points \times number of elements which ensures a unique density. We next discuss some known issues and limitations of PISCS. PISCS is formulated such that the boundary conditions and the output potentials and fields are in SI units.

7.3.1 Near boundary evaluation

We addressed the singular integrals on Γ . However, the indirect BEM gives an unstable evaluation near the boundary due to near-singular integrals. Since we form a discrete density $\boldsymbol{\eta}$, the resultant quadrature Q of integration \mathcal{I} is subject to the error in each term of the sum. The error in the solution is bounded by something of the form [134],

$$|\mathcal{I} - Q| \leq C \sum_{i=1}^n |\eta(\mathbf{y}_i) - \eta_i| \left| \frac{\partial}{\partial n} G(\mathbf{x}, \mathbf{y}_i) \right|.$$

where C is an arbitrary constant. If a quadrature term has a large jump, i.e. the singularity, so does the error bound. In this case, the jump comes from $\frac{\partial G}{\partial n}$ when $\mathbf{x} = \mathbf{y}_i$ while $|\eta(\mathbf{y}_i) - \eta_i|$ remains finite and nonzero. To illustrate the instability, figures 7.4 and 7.5 shows density

and vector plots at the center slice of a sphere and an ideal quadrupole. We show the difference between the theoretical and BEM results using constant, quadratic, and cubic elements. We note regions where the difference is large, particularly near Γ . The error is somewhat mitigated using higher order elements, extending the accurate region. However, only increasing element order is somewhat inefficient for this issue, so we researched other possibilities.

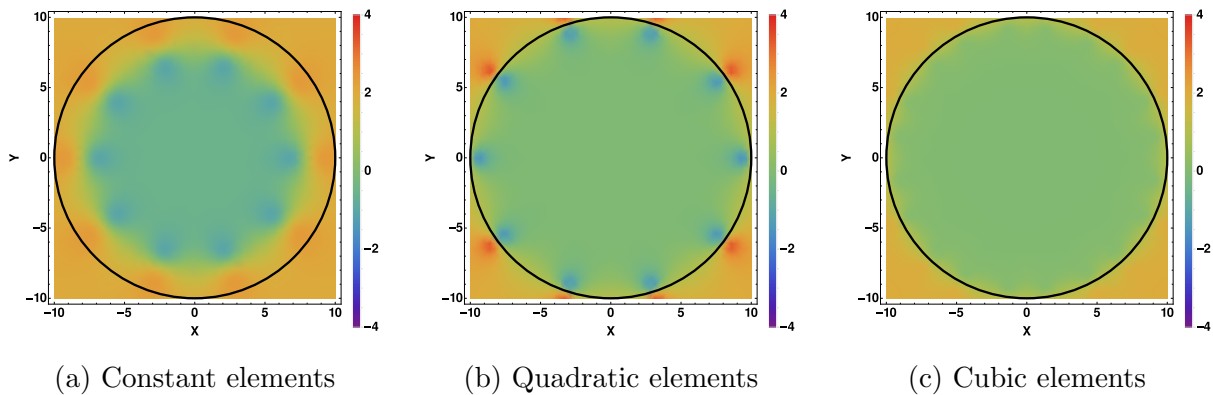


Figure 7.4: Difference in potential for a PEC sphere. While the interior shows small difference, we see significant differences near the surface (black line). These differences shrink somewhat as element order increases.

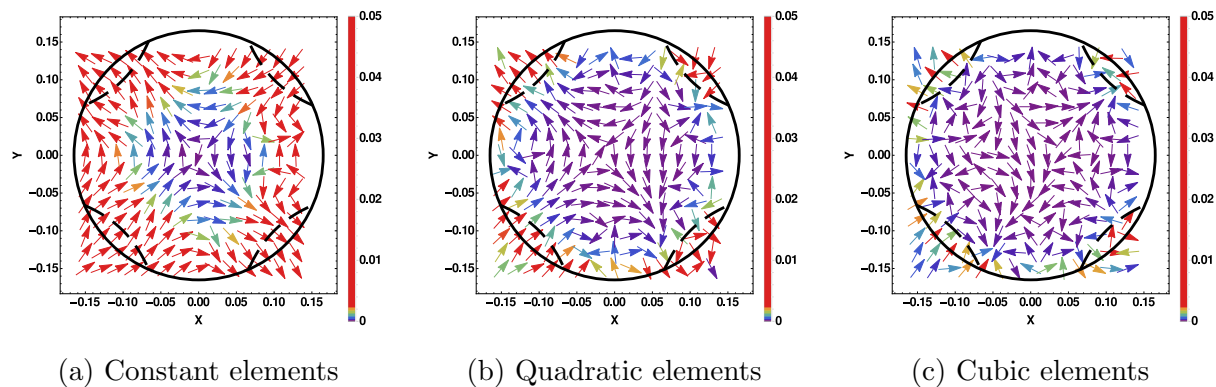


Figure 7.5: Difference in fields for an ideal long quadrupole. The actual surface used is a cylinder (solid black line). Some of the equipotentials (dashed black line) are shown for reference. An accurate region is evident where the norm of the difference is small and expands as element order increases.

While we came across other methods, we considered two possibilities to resolve the instability. These are the locally-corrected Nyström (LCN) and quadrature by expansion (QBX). LCN requires solving a linear system to find a kernel function near the singularity that reduces the quadrature error [127,161,162]. The corrected kernel is used when near a singularity and the regular Green's function otherwise. LCN has not reached an optimal level that we may implement, so we put it aside. QBX requires an expansion away from a singularity before evaluation, which fits our implementation based on a preliminary study [163,164].

The locally-corrected Nyström method replaces the kernel function K in the quadrature with a corrected kernel L at specific nodes.

$$\int f(y)K(x, y) dy \approx \sum_i w_i f(y_i)K(x, y_i) \rightarrow \sum_i w_i f(y_i)L(x, y_i).$$

While the optimal derivation of L is an open topic, the authors in Ref. [162] mention finding L such that a known set of variables giving the near field is correct for specific samples. The known set is represented by basis functions B , so the condition is,

$$\sum_i w_i B_k(y_i)L(x, y_i) = \int B_k(y)K(x, y) dy.$$

If the number of points and basis functions are equal, we form a square matrix equation for $w_i L(x, y_i)$. In Ref. [162], the authors state the accurate integration of $\int B_k(y)K(x, y) dy$ costs the most operations in this formulation. Using variants of Gaussian quadrature in Ref. [162], the authors demonstrate the effectiveness of the local correction. However, we expect the LCN to be inefficient for our implementation at this time.

Quadrature by expansion for the BEM is introduced by Ref. [163]. We know the quadrature gives accurate results sufficiently away from Γ . The accuracy is maintained by forming a local expansion sufficiently away from Γ before evaluation. In Ref. [163], the authors show

the high accuracy region depends on the tolerance ϵ and grid spacing h . The authors in Ref. [164] further expand on Ref. [163] and combines the FMM with QBX. Somewhat recent, the authors in Ref. [10] extend QBX further to several elliptic equations of form $\mathcal{L}\phi = 0$, $\phi = f$ on Γ , named quadrature by kernel-independent expansion (QBKIX). A schematic of the setup is shown in Figure 7.6 [10].

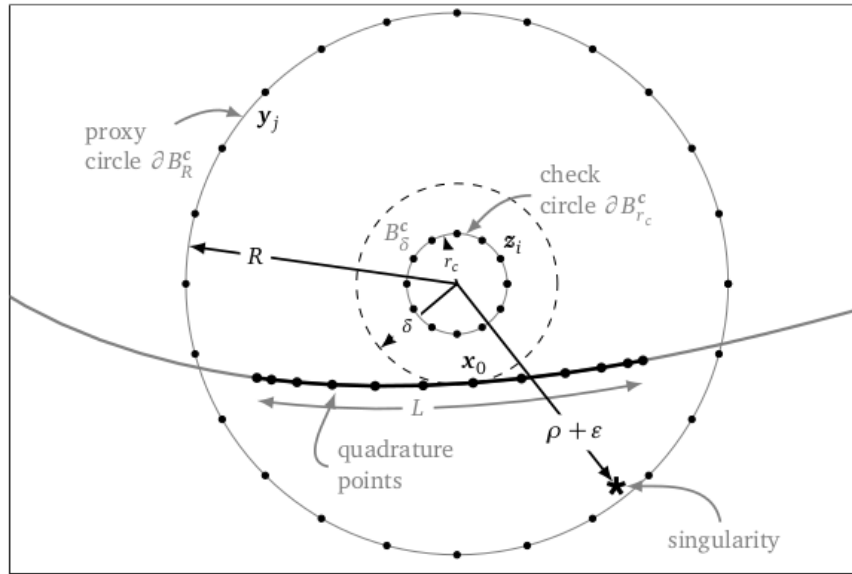


Figure 7.6: Schematic of QBKIX [10]. Panel L on Γ is discretized with Gauss-Legendre nodes. $c = \mathbf{x} - \delta \mathbf{n}$ is chosen away from boundary and an equivalent density is matched on ∂B_R^c for evaluation.

The basic steps to evaluate at \mathbf{x} lead to interpolating a density on an equivalent boundary around the expansion point c . However, this requires solving an ill-conditioned least squares system to match an equivalent boundary. In Figure 7.6, the equivalent boundary is ∂B_R^c and the evaluation region is inside B_δ^c . The error at each stage of QBKIX is shown in figure 7.7.

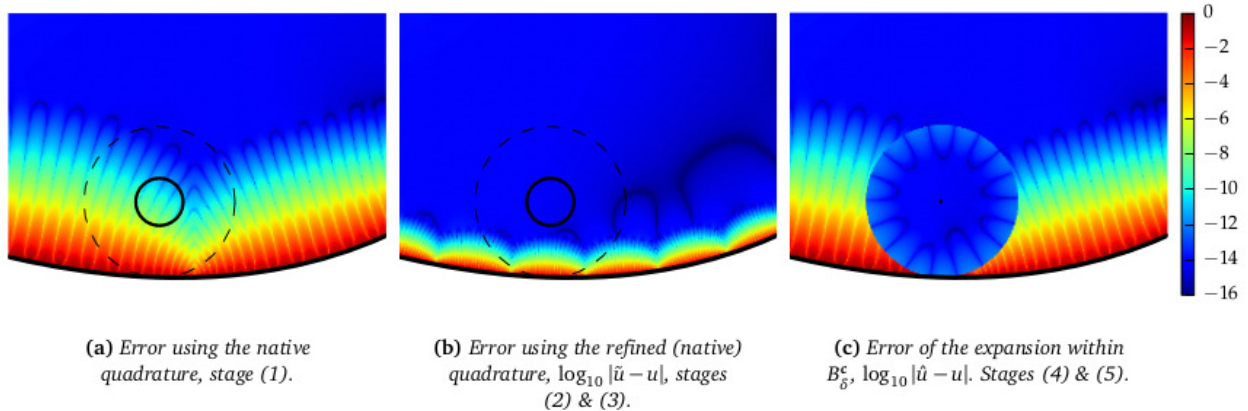


Figure 7.7: Error of each stage in QBKIX near Γ [10]. (a) shows the regular quadrature, (b) shows after splitting the particular panel L to refine the quadrature, and (c) shows after evaluating the matched density from ∂B_R^c

The QBKIX algorithm has additional optimizations for evaluation sets [10]. QBX and related variants have great potential for us, but we have not fully explored these concepts at this time. Details on convergence and error with QBX may be found in other works such as Refs. [165, 166]. In this work, we note that the expansion should be easily done with DA methods as it bears striking similarity to a translation operator in the FMM. The near boundary evaluation should not significantly affect our goal in this work.

7.3.2 Redundant collocations

When discretizing a structure, every element must share at least one edge. Following the closed Newton-Cotes rule, we specify several collocation nodes including the edges where we match the boundary condition. Any element sharing the edge similarly shares the collocation nodes. This leads to redundant rows in the system matrix and inflates the size, particularly with high order elements. We could identify the unique nodes, but then we must adapt the quadrature rule, which currently only considers the interior of an element. Clearly, we lose significant accuracy by simply reducing the quadrature order to fit the unique collocation

nodes. Without the reduction, we lose the uniqueness of the density. We wish to note here that this also is a limiting factor for the high-order discretization. However, it is not a significant issue for this work since we consider element orders < 10 .

7.3.3 Parallelization

Although PISCS is not optimally parallelized, we identified and parallelized some major bottlenecks. The modified Runge-Kutta integrator allowed some parallelization by subdividing the outer integral and using MPI. Since we transformed to the unit square, we may integrate over either variable first. From some tests, we found subdividing the outer integral displayed more stable speedup. The FMM is parallelized in Ref. [7] and retained in PISCS. In GMRES, the major bottleneck is the matrix-vector product in the outer and inner loops, which is accelerated by the FMM. The second major bottleneck comes from other dot products when projecting the Krylov basis vectors. We formulated these using COSY's intrinsic routines which turned out faster than a rough parallelization.

In addition, we added various utility functions for general use with matrices and vectors. A major bottleneck when using high-order elements comes from the polynomial interpolation matrix. The high-order elements require large, sparse matrices for interpolation and to compute the 2-D Lagrangian basis functions for the quadrature weights. The SVD algorithm used for matrix inversion is not optimized nor parallelized. The Lagrangian basis functions are computed once and require only one matrix inversion, which is not significant. However, the NVF interpolation requires the normals on each element, which changes the interpolation matrix and needs the SVD for each element. The SVD takes longer as the matrix size grows due to element order. While higher element order suggests fewer elements, the increased runtime does not optimally scale. The Vandermonde interpolation without

normals does not have this problem. Instead, the interpolation matrix is the same for all elements and is inverted by SVD once. We apply the inverse to each coordinate set. The runtime is significantly less for high-order elements. At this time, we have not parallelized the interpolation, but the Nyström quadrature is parallelized through the FMM. We next discuss applying PISCS combined with solving the equations of motion.

7.4 Combining space charge and the beam-wall interaction

We have combined Nyström discretization and high-order interpolation to reduce the discretization and quadrature error. PISCS computes the potentials and fields from the static beam-wall interaction for arbitrary structures whereas the image charge approach requires a simplified geometry [49]. We have given the equations of motion in (3.1). We can combine an integrator with our potentials and fields to solve the EoM. First, we explain the relativistic connection between BCs in the lab and beam frame. We briefly outline Strang splitting, the space charge transfer map extraction, and describe the integration of PISCS with COMFY. We also briefly discuss integrating PISCS with the 3-D FMM tracking code, PHAD.

We primarily deal with relativistic situations. We must consider the Lorentz boosted quantities between the lab frame (unprimed) and beam frame (primed). We boost in z along the beam axis, i.e. $\beta \approx \beta_z$. We assume the static boundary conditions are given in the lab frame and the longitudinal field component is zero. We also assume the output is

given in the lab frame. The boosted electromagnetic fields may be found in Ref. [35]. The electrostatic BC are,

$$\begin{aligned}\phi' &= \gamma\phi, \\ E'_x &= \gamma E_x, & E'_y &= \gamma E_y, \\ B'_x &= \frac{\gamma\beta}{c} E_y, & B'_y &= -\frac{\gamma\beta}{c} E_x.\end{aligned}$$

Using scalar potential, the magnetostatic BCs are,

$$\begin{aligned}\phi'_M &= \gamma\phi_M, \\ E'_x &= -\frac{\gamma\beta}{c} B_y, & E'_y &= \frac{\gamma\beta}{c} B_x, \\ B'_x &= \gamma B_x, & B'_y &= \gamma B_y.\end{aligned}$$

We find something similar using vector potential where $A_\perp = 0, A_z \neq 0$. We mainly require a factor of γ in both cases between frames. We boost the relevant boundary conditions to the beam frame before computing the change due to the charges. We simply combine this with the unit conversion. Similarly, we boost back at the end when converting to SI units.

Using DA methods and fixed point iteration, we may integrate the equations of motion [43]. If the motion is described by two PDEs, the solution of the sum is approximated by a composition of the individual solutions as shown [7, 46].

$$\begin{aligned}\frac{d\mathbf{z}_1}{ds} &= \mathbf{g}_1(\mathbf{z}_1; s) \rightarrow \mathbf{z}_1(s) = \mathbf{f}_1(s), \\ \frac{d\mathbf{z}_2}{ds} &= \mathbf{g}_2(\mathbf{z}_2; s) \rightarrow \mathbf{z}_2(s) = \mathbf{f}_2(s), \\ \Rightarrow (\mathbf{z}_1 \circ \mathbf{z}_2)(s) &= \mathbf{f}_1\left(\frac{s}{2}\right) \circ \mathbf{f}_2(s) \circ \mathbf{f}_1\left(\frac{s}{2}\right) + \mathcal{O}(s^3).\end{aligned}$$

This is known as Strang splitting [167, 168] and corresponds to composition of the maps, i.e. $\mathbf{z}_f(s) = \mathcal{M}(\mathbf{z}_i(s)) = (\mathcal{M}_1(\frac{s}{2}) \circ \mathcal{M}_2(s) \circ \mathcal{M}_1(\frac{s}{2}))(\mathbf{z}_i(s))$. Strang splitting has quadratic convergence [169]. The COSY beam physics package [89] solves (3.1) using a DA-based integrator [43] for several common transport elements and produces Taylor expansions of the transfer map. For general cases, COSY integrates using Runge-Kutta methods, which are outlined in Appendix B.2. In principle, to solve the equations of motion including the beam-wall interaction, we run an integrator for each particle after computing and summing the self potential and fields with the boundary contribution. We discuss a practical alternative.

In Ref. [46], the author combines the transverse space charge kick with several COSY built-in beamline elements using Strang splitting. In Ref. [46], the author computes the space charge using two approaches, the method of statistical moments (MoM) and the fast multipole method. Once the potentials and fields are computed, the transverse space charge kick is given by integrating the transverse part of (3.1). In Refs. [46, 170], the authors describe how the MoM and FMM are used to compute the 2-D potentials and fields for a beam in the COSY space-charge package COMFY. The FMM uses the nonadaptive 2-D version of the algorithm from Chapter 4 and Ref. [7]. The nonadaptive FMM subdivides the region evenly. In areas of dense clusters, the algorithm requires more direct summations, which ultimately scales like $\mathcal{O}(N \log N)$ [75, 171]. Even so, the nonadaptive FMM is sufficient for the transverse case. The FMM kick is computed using the linear approximation described later. The moment method summarized next represents the space charge kick as a transfer map. For full details, we refer to Ref. [46].

We assume the beam's charge density is expressible as an expansion of smooth continuous functions with unknown coefficient [46]. The method of statistical moments assumes a statistical distribution exists which is represented by the given charges [172]. In our case, the particles in the beam are samples of the charged density distribution, similar to the Vlasov approach [30]. Any statistical distributions may also be represented by its moments

up to some order [46, 173]. If two distributions have the same moments, they are considered statistically identical [173]. The moments are given below where $f(x, y)$ is the distribution function [46]. For point charges $f(x, y) = \delta(x - x_i)\delta(y - y_i)$, so the moments become a sum.

$$M_{nm} = \int_{x_l}^{x_u} \int_{y_l}^{y_u} x^n y^m f(x, y) dx dy,$$

$$\Leftrightarrow M_{nm} = \sum_i x_i^n y_i^m.$$

Similarly, we can Taylor expand $f(x, y)$ with unknown coefficients and match the moments. This ultimately gives a matrix equation for the Taylor coefficients [46].

$$\sum_i x_i^n y_i^m = \sum_{ij} T_{ij} \int_{-1}^1 \int_{-1}^1 x^{n+i} y^{m+j} dx dy.$$

Note the bounds are chosen to minimize numerical errors [46]. The ill-conditioned matrix is solved by singular value decomposition [46]. There are other steps involving scaling and Duffy transform [46], but at this point, we could compute the usual Coulomb potential as its Taylor expansion around (x_0, y_0) where D_{nm} are the scaled Taylor coefficients.

$$\phi(x_0, y_0) = \sum_{nm} D_{nm} \int_{-a}^a \int_{-a}^a x^n y^m \ln\left(\sqrt{(x - x_0)^2 + (y - y_0)^2}\right) dx dy.$$

In COSY, ϕ is Taylor expanded as a DA object around the reference trajectory \mathbf{r}_0 . Similarly, the fields are also DA objects. Integrating the motion due to the space charge potentials and fields for the reference particle gives a zero-length Taylor expansion of the kick or a space charge map [46]. Composing the space charge map and element map with Strang splitting thus adds the space charge kick for the beamline elements. In addition, we apply normal form analysis to extract different effects due to space charge [11, 46].

We can extract the self-consistent space charge transfer map from simulation, opening new possibilities in the field of beam physics [11, 170]. We perform normal form analysis to study some effects in space-charge dominated beams [11]. We can combine the map extraction method with just about any tracking code. We use the space charge map from the MoM to track the transverse beam motion [11]. We also combine the space charge map with the FMM [11]. The MoM and FMM necessarily produce slightly different trajectories due to approximations [171], so we check that the map extraction procedure itself smooths out the differences. Each element includes a single space charge kick at its center assuming open boundary conditions. To emphasize the space charge effect, we limit the beamline element maps to first order and computes the space charge kick up to eighth order [11]. When we previously compared the MoM and FMM [171], the results at the same space charge order are comparable. The space-charge dominated beam parameters are given in Table 7.1 for all runs [11].

Table 7.1: Space-charge Dominated Beam Parameters

Species	Proton
No. of particles	5000
Energy [MeV]	5
Shape	Ellipse
Initial spatial distribution	Uniform
Initial maximum radius [m]	0.001
Initial angle distribution	Uniform
Initial maximum angle [rad]	0.03
Initial emittance (X,Y) [m]	(7.63 , 7.50)

We chose the number of particles, $N = 5000$, for speed and acceptable accuracy. We use two test cases, an imaging triplet of quads and a periodic FODO cell to study some aspects of multi-particle beam dynamics. The triplet we set up consists of an outer drift, quad (Q1),

inner drift, quad (Q2), inner drift, quad (Q3), and outer drift. The system parameters are given in Table 7.2 [11] and its first order map is shown in (7.1) [11].

Table 7.2: Imaging Triplet System Parameters

Aperture radius [m]	0.08
Outer drift lengths [m]	0.25
Inner drift lengths [m]	0.15
Q1 length [m]	0.2
Q2 length [m]	0.2
Q3 length [m]	0.2
Initial Q1 [T/m]	0.65525
Initial Q2 [T/m]	-0.66566
Initial Q3 [T/m]	0.65525

$$\mathbf{M}_{\text{Triplet}} = \begin{pmatrix} -1 & 3.20 \times 10^{-14} & 0 & 0 \\ 5.69 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1.91 \times 10^{-14} \\ 0 & 0 & -4.39 & -1 \end{pmatrix}. \quad (7.1)$$

To illustrate the motion, we show a ray trace of the system without space charge in Figure 7.8 [11]. We draw 3 independent rays in the X - Z and Y - Z projections using the transfer map. We include an extra end drift of 6.25 cm to show the focal point at $z = 1.4$ m. The extra drift is not included in our maps [11].

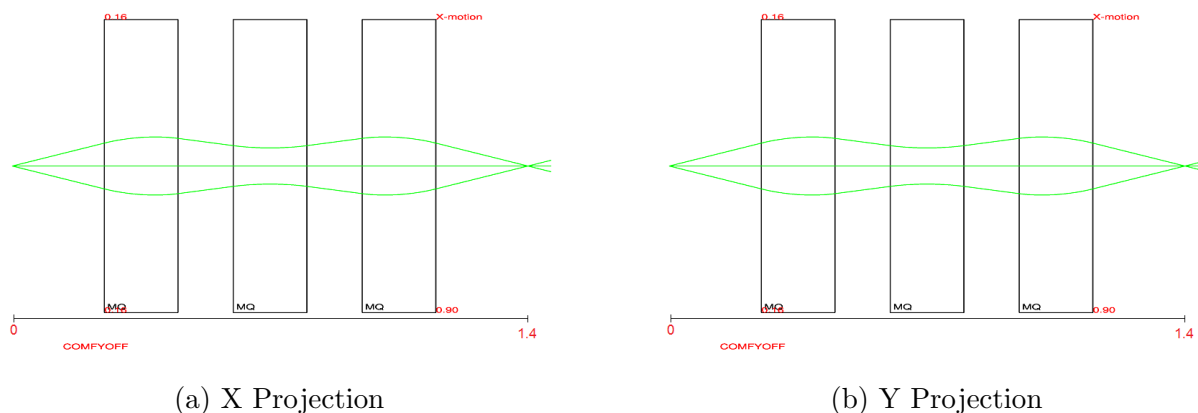


Figure 7.8: Imaging triplet rays without space charge. Focal point is at $z = 1.4$ m [11].

We fix the settings and increase the representative current of the beam with fixed particle number to isolate the space charge effects. The triplet shows interesting behavior with current. The map elements $(x|a)$ and $(y|b)$ are almost zero with the given settings but grow significantly with current from Figure 7.9 [11].

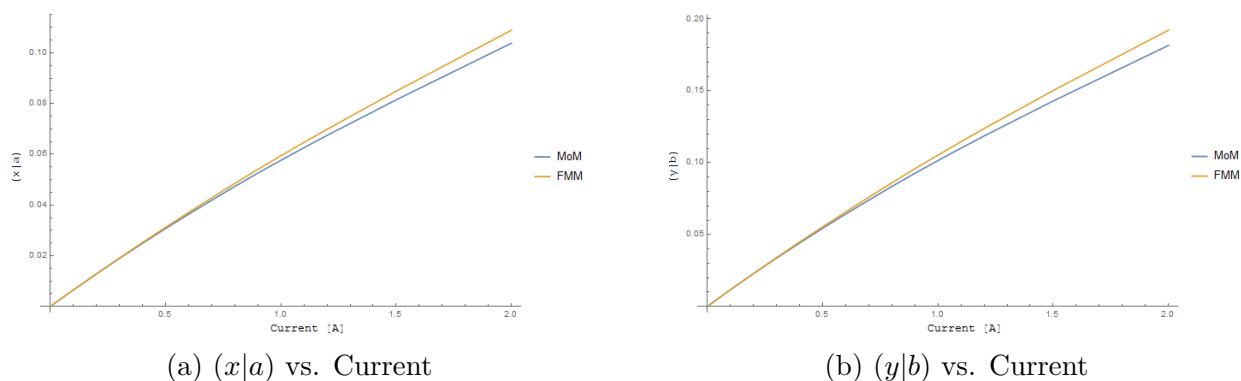


Figure 7.9: Behavior of $(x|a)$ and $(y|b)$ vs. current in the imaging triplet, as calculated by the moment method and the fast multipole method [11].

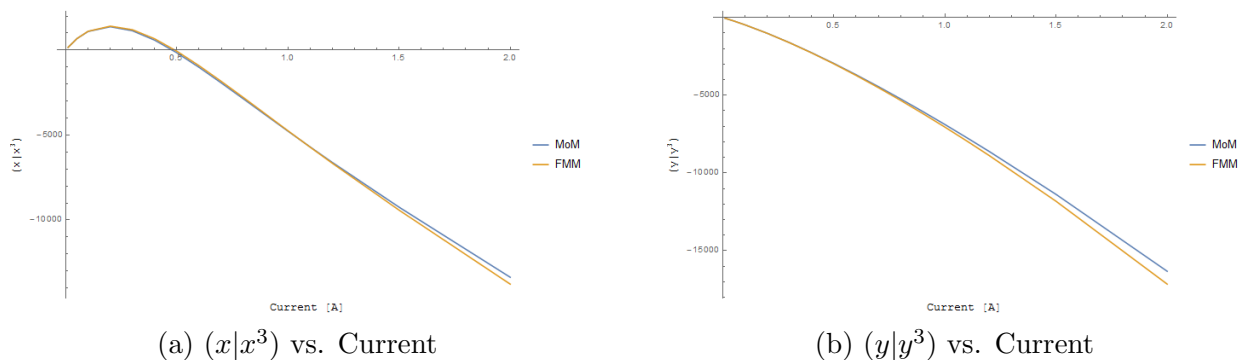


Figure 7.10: Behavior of $(x|x^3)$ and $(y|y^3)$ vs. current in the imaging triplet, as calculated by the moment method and the fast multipole method [11].

We also show the behavior of the third order geometric aberrations $(x|x^3)$ and $(y|y^3)$ as a function of current in Figure 7.10 [11]. Since the system map is limited to first order, the third order aberrations are due to space charge [11]. The FMM predicts a slightly stronger self-field, which is seen in previous testing [171]. This is likely due inclusion of collisional forces which are negligible in the MoM [11]. Space charge appears to inflate $(x|x^3)$ until around 0.25 A before reversing direction and becoming negative.

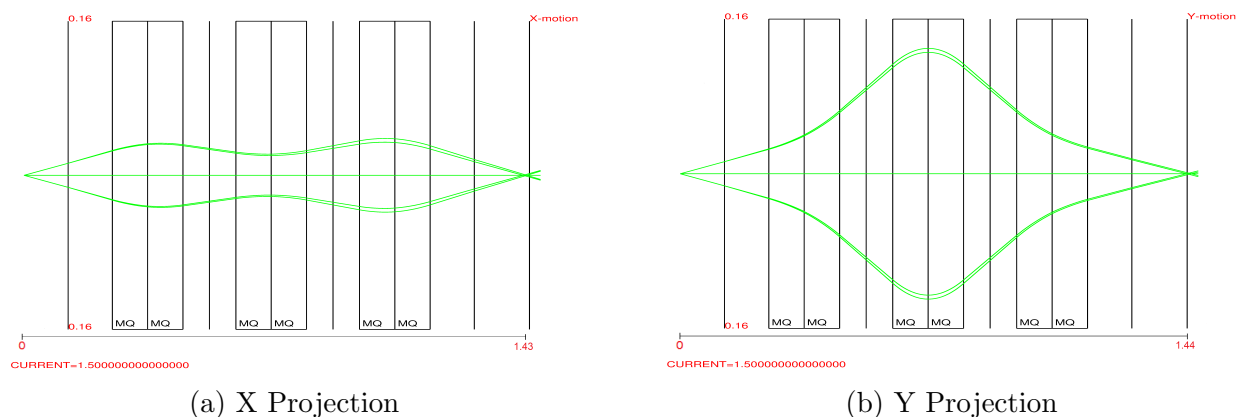


Figure 7.11: Imaging triplet rays w/ current = 1.5 A [11]. The focal point has shifted to $z = 1.43$ m in X - Z and to $z = 1.44$ m in Y - Z .

The focal point shifts at 1.5 A in Figure 7.11 [11]. Although subtle, the focal point is now around $z = 1.43$ m in X - Z , Figure 7.11a, and $z = 1.44$ m in Y - Z , Figure 7.11b, indicating

slight astigmatism. The shift of the focal point is also observed due to third order terms in the system elements if included, which are spherical aberrations [11]. The maximum width of the rays are increased slightly and the rays split at the point of maximum width. The split appears in both projections to different extents. So, a ray in Y - Z shifts in X and vice versa, which explains the extra trajectory. This split seems to come from nonlinear coupling introduced by fourth order space charge terms [11]. To reduce or eliminate the effect of space charge, we fit the quad gradients for each current to match our desired parameters. We give priority to preserving the imaging property. We fit the quad gradients with the condition of only minimizing $(x|a)$ and $(y|b)$. The quad gradients steadily grew with current as shown in Figure 7.12 [11].

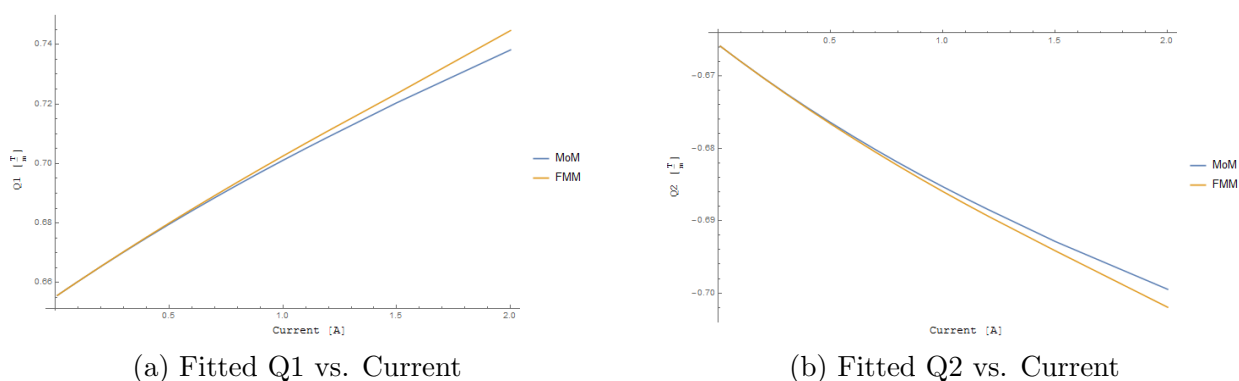


Figure 7.12: Behavior of quad gradients vs. current in the imaging triplet after fitting for imaging, as calculated by the moment method and the fast multipole method [11].

Figure 7.13 plots the absolute value of $(x|a)$ and $(y|b)$ [11]. $(x|a)$ and $(y|b)$ oscillate between positive and negative values when using the FMM. The deviation from 0 increases by about two or three orders of magnitude in $(x|a)$ and four or five orders of magnitude in $(y|b)$. The self-field appears stronger in Y than X , leading to the larger increase in $(y|b)$. The peak at 0.6 A in $(x|a)$ appears due to difficulty in minimizing $(x|a)$. We can also see fitting to the MoM is generally smoother than for the FMM. Most likely, this is due to the inclusion of collisional forces.

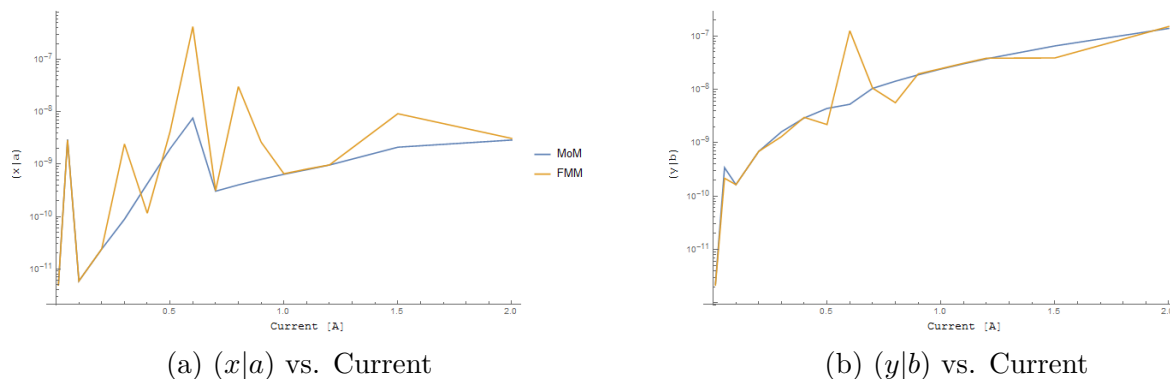


Figure 7.13: Behavior of $(x|a)$ and $(y|b)$ vs. current in the imaging triplet after fitting, as calculated by the moment method and the fast multipole method [11].

The behavior of $(x|x^3)$ and $(y|y^3)$ after fitting is shown in Figure 7.14 [11]. The third order aberration $(x|x^3)$ shows different behavior than before, as the two results start to diverge around 0.3 A. We found this is due to the different quad fits between the MoM and FMM [11]. When using the settings found with the MoM, the FMM predicted similar $(x|x^3)$ and $(y|y^3)$ but $(x|a)$ and $(y|b)$ increased by seven orders of magnitude. For $(y|y^3)$, both methods predict the aberration behaves similar to Figure 7.10b.

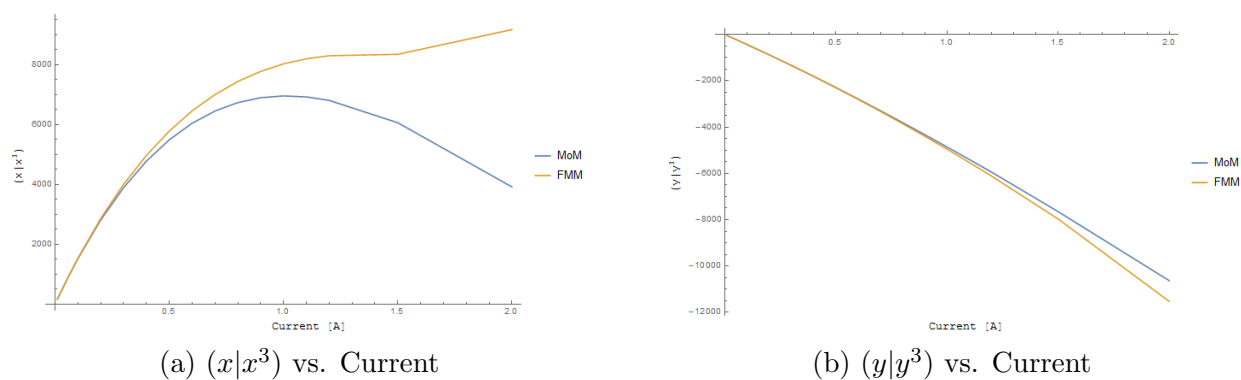


Figure 7.14: Behavior of the geometric aberrations $(x|x^3)$ and $(y|y^3)$ vs. current in the imaging triplet after fitting, as calculated by the moment method and the fast multipole method [11].

The ray diagram for 1.5 A after fitting is shown in Figure 7.15 [11]. Figure 7.15 shows the fitted quad gradients are overcompensating. The focal points are now around $z = 1.37$ m in

X - Z and $z = 1.35$ m in Y - Z . The maximum width is slightly less than in Figure 7.11, but the split in the ray is still present.

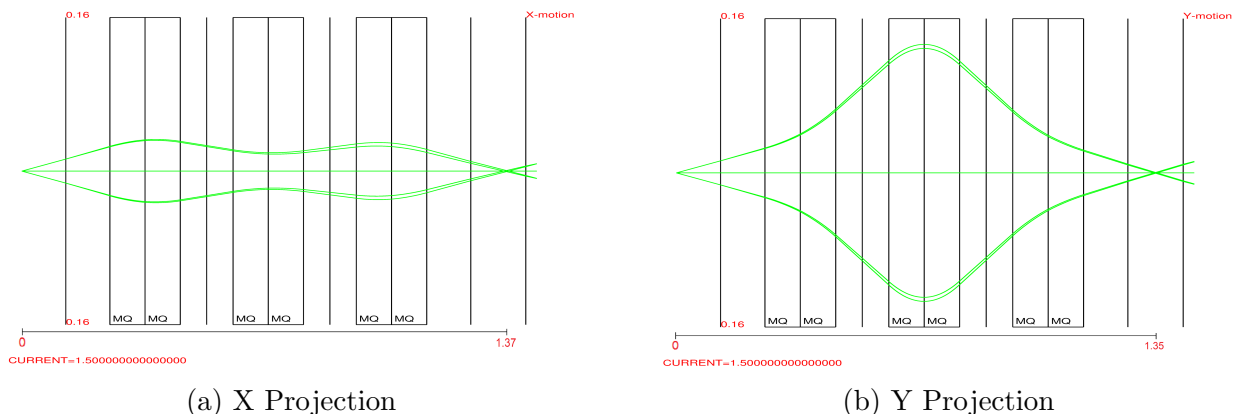


Figure 7.15: Imaging Triplet rays w/ current = 1.5 A after fitting the quads for imaging [11]. The focal points are at $z = 1.37$ m in X and $z = 1.35$ m in Y .

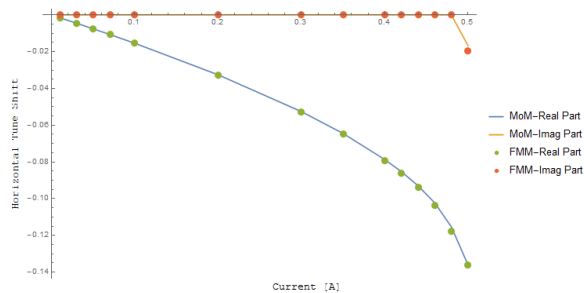
The FODO cell consists of one half quad (Q1), inner drift, full quad (Q2), inner drift, and one half quad (Q1). The system parameters are shown in Table 7.3 [11] and the first order map is given in (7.2). We choose to match the system to a horizontal and vertical tune away from resonance and study its behavior [11].

Table 7.3: FODO Cell System Parameters

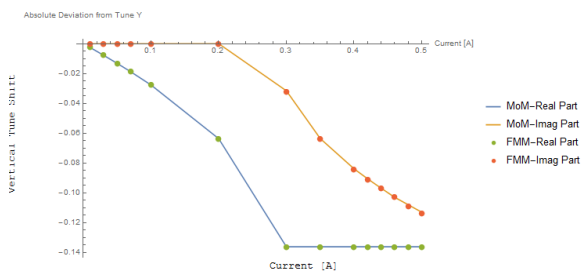
Aperture radius [m]	0.08
Inner drift lengths [m]	0.15
Q1 length [m]	0.1
Q2 length [m]	0.2
Initial Q1 [T/m]	-0.39045
Initial Q2 [T/m]	0.39045
Horizontal tune	0.1362
Vertical tune	0.1362

$$\mathbf{M}_{\text{FODO}} = \begin{pmatrix} 0.656 & 0.425 & 0 & 0 \\ -1.34 & 0.656 & 0 & 0 \\ 0 & 0 & 0.656 & .951 \\ 0 & 0 & -.599 & 0.656 \end{pmatrix}. \quad (7.2)$$

Settings for the quads in the FODO cell are easily found such that the calculated tunes almost perfectly matched our chosen parameters. Its behavior changed significantly with higher intensity. From Figure 7.16, the deviation in the horizontal and vertical tune rapidly increases with current, displaying greater rate in the vertical [11]. For μ_x , the system stays periodic until 0.5 A, where the tune becomes imaginary. μ_y 's periodicity is lost around 0.3 A. Both the MoM and FMM predict the same tune. The differences displayed in the triplet map elements suggest small deviation between the two methods, which would be negligible in calculating the tune.



(a) Space charge induced horizontal tune shift vs. Current



(b) Space charge induced vertical tune shift vs. Current

Figure 7.16: Tune shift due to space charge from tune = 0.1362 vs. current in the FODO lattice, as calculated by the moment method and the fast multipole method [11].

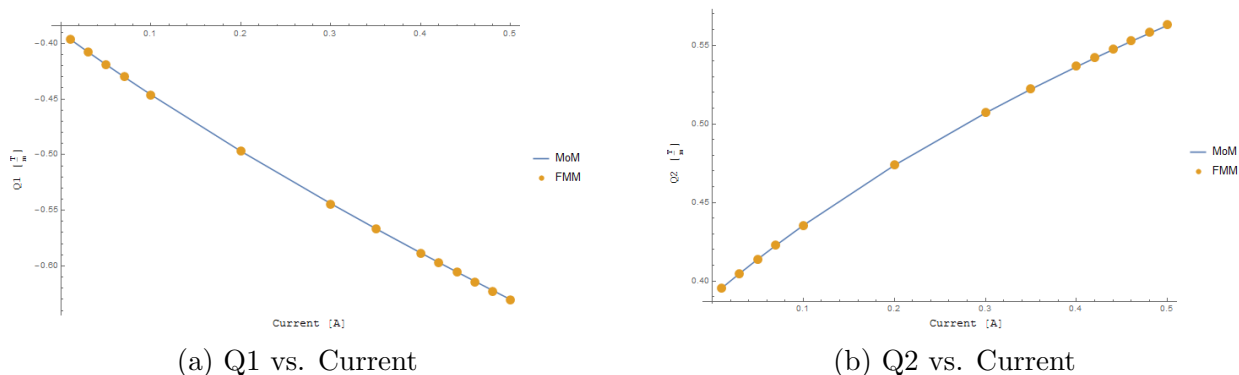


Figure 7.17: Behavior of the quad gradients vs. current in the FODO lattice after fitting to the desired tune, as calculated by the moment method and the fast multipole method [11].

We fit the quad gradients of the FODO Cell by matching the calculated tune to our desired tune in figure 7.17 [11]. Matching the tune grew difficult as current increased. Vertical stability is lost around 0.3 A. To preserve stability, we allowed for higher tune fitting tolerance [11]. The MoM and FMM give essentially identical results.

We showed some of the potential from the space charge map extraction method. We compared the MoM and FMM for the transverse space charge kick using COMFY [46]. The two methods give similar results with small differences at high currents. These indicate the collisions in the FMM versus the mean-field nature of the MoM [11]. For the spherical aberration, we found large discrepancy due to the different settings. The triplet case showed imaging properties are lost rapidly due to space charge, particularly as the spherical aberrations grew. The FODO cell loses its stability rapidly as the space charge increases, particularly in the vertical plane. When the cell is re-fitted, the bare tunes cannot be recovered exactly. The space charge map analysis shows great potential for designing new systems to control high-intensity beams [11]. We next consider including the beam-wall interaction.

Earlier, we assume a short kick and the linear approximation to solve (3.1) combined with the FMM. If needed, we increase the number of kicks per element to reduce the length. Ref. [46] shows the convergence as number of kicks increases. Strang splitting and the linear

approximation are used to include additional effects for the University of Maryland Electron Ring (UMER) [46]. The geomagnetic fields and image charges contribute additional forces as straight nonzero-length kicks. In Ref. [46], the author computes the image charge kick assuming a cylindrical pipe in the 2-D approximation. Using PISCS, we can compute the analogous potentials and fields for general geometries and solve the EoM. At high-energy or for short kicks, the evolution for a, b over path length ℓ is approximately,

$$a_f = a_i + \ell a',$$

$$b_f = b_i + \ell b'.$$

In general, a', b' from (3.1) are coupled through $\frac{p_0}{p_s}$. The linear approximation is only valid when $\frac{m^2 \eta (2 + \eta)}{m_0^2 \eta_0 (2 + \eta_0)} \gg a^2 + b^2$ or when the kick length is sufficiently small [46]. The kicks due to the FMM, geomagnetic field, and image charge are modeled using this approximation. In contrast, the transfer maps represent the zero-length kick until evaluated. We could compute the beam-wall kick in the linear approximation and combine with the space charge kick. However, this breaks quadratic convergence from Strang splitting [169]. Recalling a', b' from (3.1), the change in angle is due to the potential and fields in the factors, $\eta, \mathbf{E}, \mathbf{B}$, where $\mathbf{B} \propto \mathbf{E}$ due to relativity. If we compute and sum the potentials and fields prior to integrating, we perturb the EoM solution but preserve the convergence, where the perturbation is from the beam-wall interaction. For the FMM, we simply sum the potentials and fields, then apply the linear propagation. We cannot sum potentials and fields with the space charge transfer map due to how the MoM Taylor expands the charge density. At the moment, our indirect BIE form cannot be Taylor expanded to define a self-consistent transfer map. The map split is given by,

$$\mathcal{M}_{\text{elem+SC}}(s) = \mathcal{M}_{\text{elem}}\left(\frac{s}{2}\right) \circ \mathcal{M}_{\text{SC}}(s) \circ \mathcal{M}_{\text{elem}}\left(\frac{s}{2}\right).$$

We add the beam-wall interaction to the potentials and fields in $\mathcal{M}_{\text{SC}}(s)$ using the FMM module. Although not directly composed in the map, the distribution is perturbed for later kicks, which indirectly changes the space charge and the resulting kick. In this way, we expect to preserve the quadratic convergence from Strang splitting.

7.4.1 PHAD

The program Particles High-order Adaptive Dynamics (PHAD) from Ref. [7] is developed for particle tracking with space charge. PHAD employs several novel concepts. We briefly summarize relevant parts here and refer to Ref. [7] for further details. PHAD works in Cartesian coordinates centered around the beam, so it solves the equations of motion Y from (2.1) due to the Lorentz force of self-induced and external fields [7]. PHAD is used to study the time evolution of a high-intensity, charged particle beam. Consider a test particle in the vicinity of the beam distribution. The forces from nearby particles are sensitive to small changes in distance and vary rapidly in time. The forces at long range slowly vary in time and are insensitive to small changes in distance. The long range forces can be collectively modeled using the mean-field. The equation of motion of the test particle is split into the near region with small time step and far region with large time step. We can employ Strang splitting to combine the solutions, which leads to composing alternating solutions for the near region $\phi_{t,k}^{[1]}$ and the far region $\phi_{t,k}^{[2]}$. The final solution after n time steps with stepsize h is given below [7].

$$Y_n = \phi_{h/2,2n}^{[2]} \circ \phi_{h,n}^{[1]} \circ \phi_{h/2,2n-1}^{[2]} \circ \phi_{h/2,2n-2}^{[2]} \circ \phi_{h,n-1}^{[1]} \circ \phi_{h/2,2n-3}^{[2]} \circ \dots \\ \phi_{h/2,4}^{[2]} \circ \phi_{h,2}^{[1]} \circ \phi_{h/2,3}^{[2]} \circ \phi_{h/2,2}^{[2]} \circ \phi_{h,1}^{[1]} \circ \phi_{h/2,1}^{[2]}(Y_0).$$

PHAD combines three integrators for computing the kicks at different steps to form an efficient symplectic tracking code. The three integrators employ different forms of the Picard iteration [99,174,175]. PHAD uses two long range integrators combined with the FMM, one for odd steps before propagating the particles and another for even steps after propagating the particles. PHAD uses the third Picard integrator to compute $\phi^{[2]}$ at the start of a step. The basic order is Picard integrator, even step integrator, and odd step integrator. The FMM steps, L2P or P2P, are not used in PHAD. These are replaced by the Picard iteration, which propagates the particles in the near region. In Ref. [7], the author demonstrates PHAD's efficacy by simulating electron cooling for Jefferson Lab's Electron Ion Collider (JLEIC) in the Energy Recovery Linac (ERL) section.

The current version of PHAD solves the $6N$ coupled relativistic equations of motion using two fixed step sizes for long and near regions [7]. Similar to space charge, we consider the beam-wall contribution as a set of additional forces. PHAD uses Strang splitting to insert the kick from near and far regions. While not done at this time, we plan to merge PHAD with PISCS to include the beam-wall interaction. In the equations of motion, this amounts to summation of the fields due to the induced charge before integrating the equations of motion. The main challenge is understanding which step(s) must include the beam-wall interaction without losing the symplecticity. In this work, we demonstrate the beam-wall interaction in the transverse equations of motion using PISCS merged with COMFY.

CHAPTER 8

RESULTS AND ANALYSIS

8.1 Performance

We first study the performance and limits of our implementation, PISCS, in terms of runtime and memory usage. The perfect spherical conductor is the simplest structure with an analytical solution including charges to perform this study. We use a 3D Gaussian distribution of 1000 protons inside a perfect conducting sphere (radius $R = 10$ m) with constant potential $V = 1$ V as our example. Figure 8.1 shows the CAD and example mesh with 128 elements, $e = 6$.

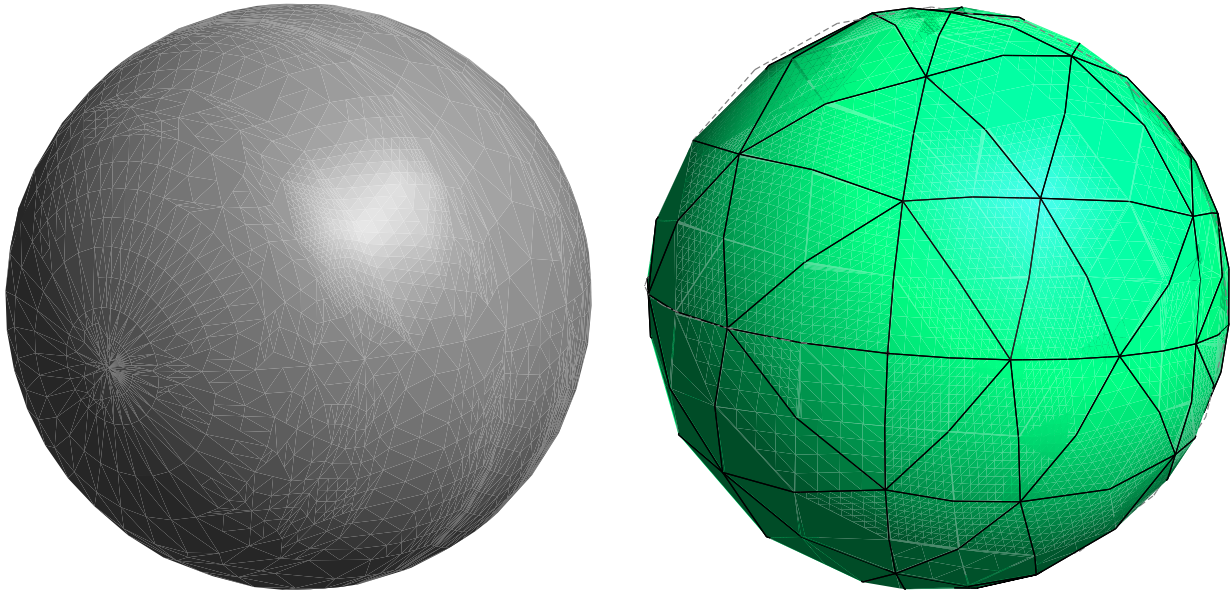


Figure 8.1: CAD and mesh of PEC sphere with $R = 10$ m. Mesh has 128 elements at $e = 6$.

While we only show results with Dirichlet boundary conditions in this section, PISCS with Neumann BC has approximately the same performance. For these tests with particles, we automatically set the FMM clustering parameter, q , depending on the number of particles, N . If $N \leq 20000$, then $q = 0.05N$. Otherwise, parameter $q = \frac{1000}{N}$, i.e. each box has at most 1000 particles. This is to limit the near interactions that slow down the FMM. This has some issues with small N as shown in Figure 8.3.

The performance depends on two major modules, the setup for high-order elements and the FMM. We classify PISCS with five major steps in the algorithm.

1. Setup (BCREAD)
2. Free-space Poisson solver (BCCALC)
3. BIE solver (BCMATRIX)
4. BEM evaluation (BCEVAL)
5. Self plus BEM (OUTPUTRES)

The setup in BCREAD computes all the points and normals using the interpolation scheme. The element order has the most effect here as it determines the size of the Vandermonde matrix to be inverted. Besides BCREAD and OUTPUTRES, each step requires at least one FMM call on different sets of points. BCCALC computes the self potentials and fields as well as the modified BC for the Laplace equation. The self potentials are saved temporarily until OUTPUTRES. BCMATRIX uses the FMM to accelerate GMRES, so it requires the most FMM calls. BCEVAL requires one FMM call to evaluate the density $\boldsymbol{\eta}$ at the desired points. OUTPUTRES sums the potentials and fields and uses the least amount of time. For our expected applications, the dominant steps generally are (longest to shortest): BCCALC, BCMATRIX, BCEVAL, BCREAD, OUTPUTRES. The runtime in each step strongly depends on either number of elements or number of particles. We breakdown the runtime to explain the major bottlenecks.

We are concerned with the behavior as element order and element number increases. Using our example sphere, Figure 8.2 shows the runtime versus number of elements at various element orders with FMM order 6. We use a single process for runtime unless stated otherwise. For large M , we only show element orders 0 and 2 due to an issue in Gmsh occurring for large M , $e \geq 3$ as of Gmsh v3.0.6.

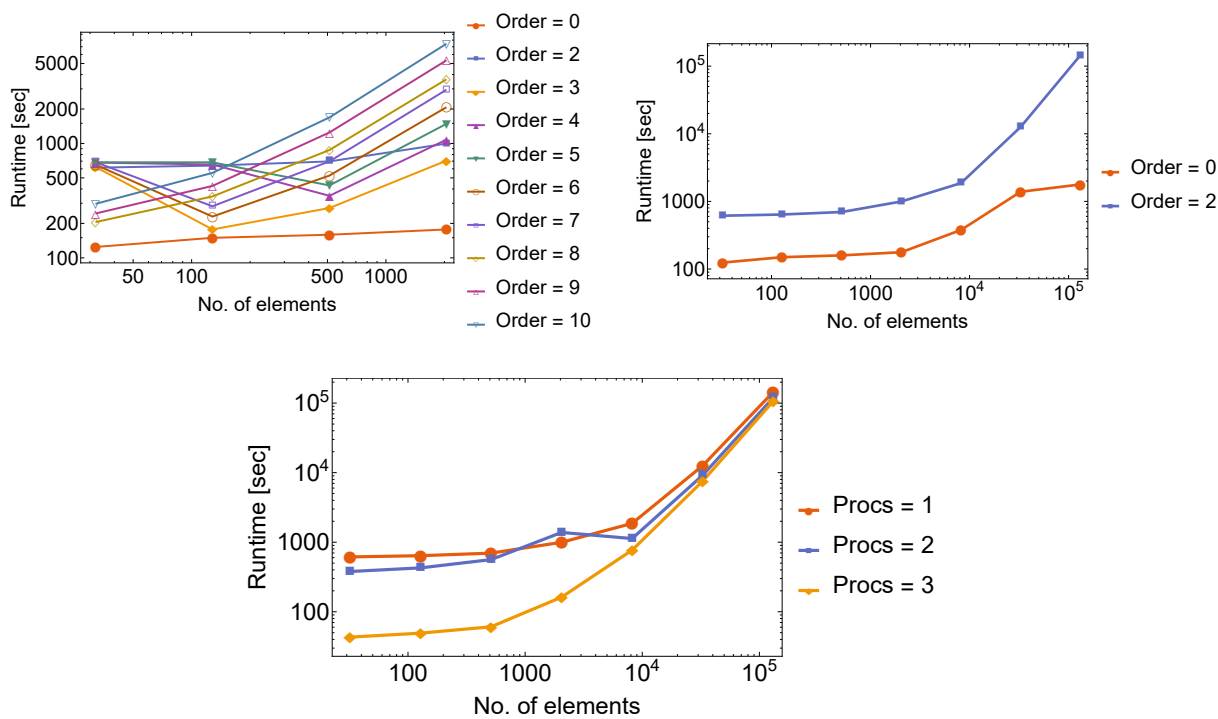


Figure 8.2: Runtime versus number of elements at various element orders. The runtime is particularly large for $e = 2 - 7$, $M = 32$, $e = 2, 4, 5$, $M = 128$, and $e = 2$, $M = 512, 2048$. In all these cases, runtime for **BCCALC** is large due to using the unrotated M2L operator with many translations corresponding to the selected q . The bottom plot shows the parallelization with respect to number of processes. The reduced efficiency at high M is due to the setup of the high-order elements, where the interpolation is not parallelized.

We see several spots where runtime is oddly large. In these cases, the time for **BCCALC** is large due to the FMM downward pass. This comes from using the unrotated M2L when the selected clustering parameter leads to many translations. There is a balance between near and far interactions which is not met for these particular parameters. However, the general runtime behavior besides these spots match what we expect. While the FMM is

expected to show linear runtime with N, M , **BCMATRIX** requires several FMM calls, so our overall behavior is $\mathcal{O}(N) + c\mathcal{O}(M)$ where c is the number of iterations in the matrix solver. The runtime breakdown is more insightful in this case, shown in Figure 8.3.

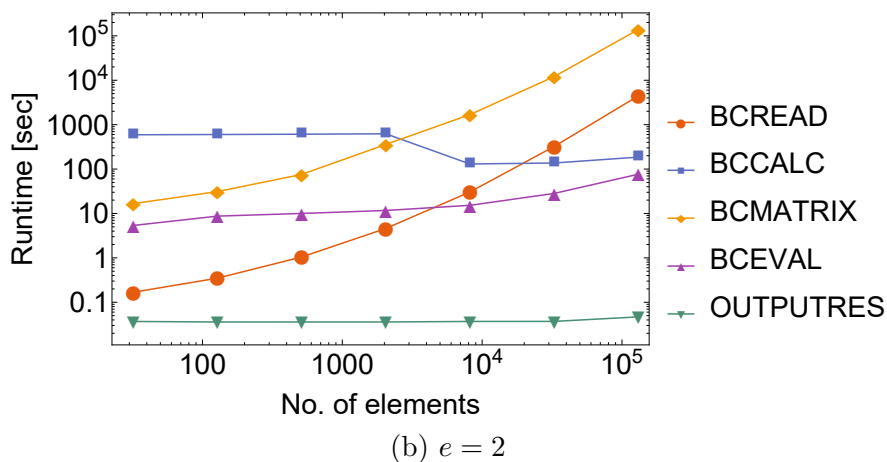
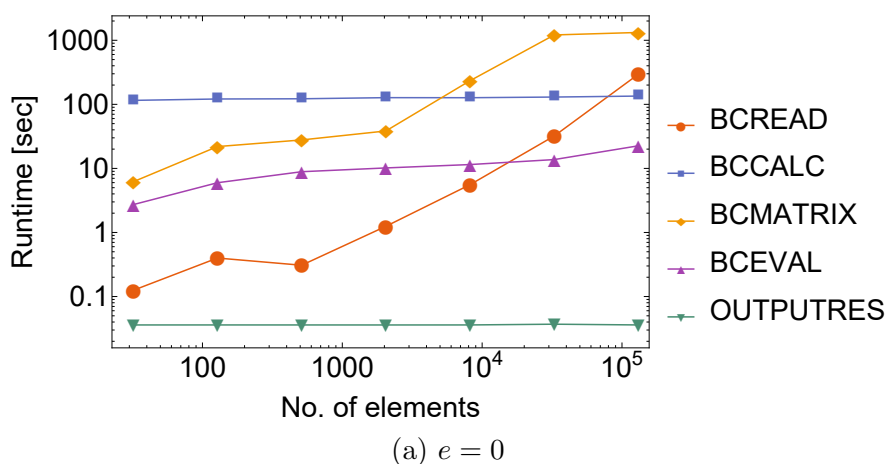


Figure 8.3: Runtime breakdown for large M . **BCCALC** dominates the runtime for small M and is the major cause of the outliers in figure 8.2 for order 2.

Figure 8.3 shows that **BCREAD** and **BCMATRIX** are affected the most by M as expected. **BCCALC** is mostly constant since $N = 1000$ for all tests. There's a slight dip for $e = 2$ due to a drop in the runtime in the FMM downward pass. Runtime around 100 seconds or less is where **BCCALC** should be. **GMRES** in **BCMATRIX** for $e = 0$ only iterates twice which gives the flat region at $M > 50000$. The exact cause is unclear. This behavior is not seen for $e = 2$,

where GMRES does iterate as expected. We measure runtime versus number of particles with increasing MPI processes to test the parallelization with respect to N , shown in Figure 8.4. We test the parallelization using a high-performance computing cluster.

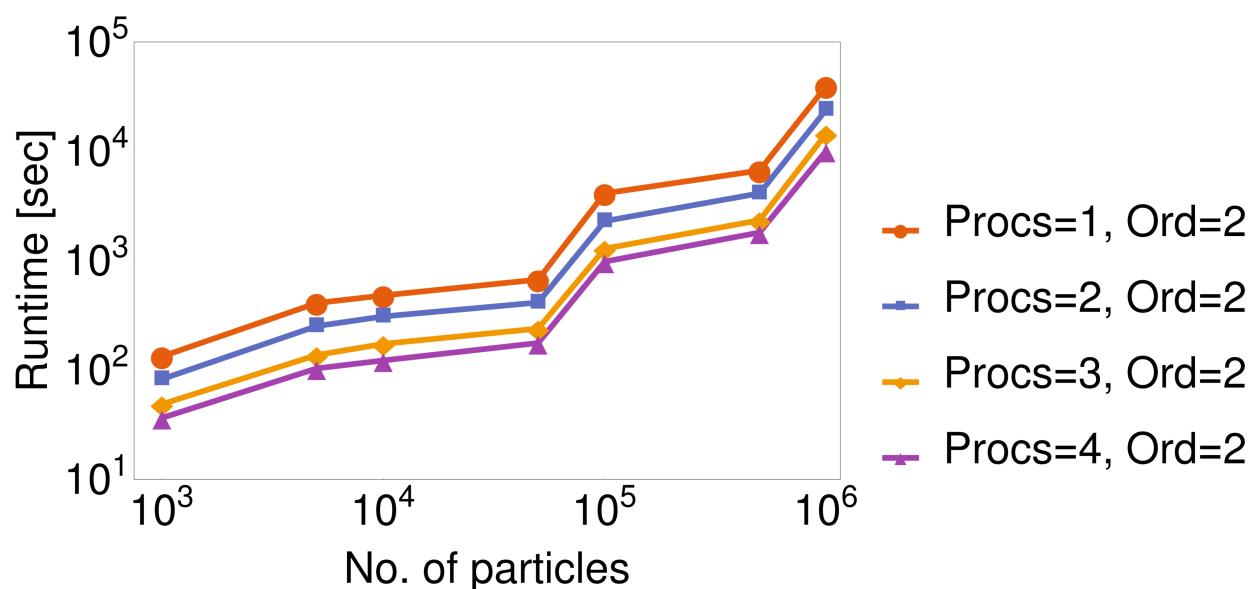


Figure 8.4: Runtime versus number of particles for element order 2 with increasing number of MPI processes on a computing cluster (up to 4 MPI processes shown).

Figure 8.4 shows the parallelization with number of particles, N , and number of MPI processes. Clearly, the parallelization is not optimal, but the runtime is significantly reduced with 4 processes. Though not shown, we successfully test up to 32 MPI processes on the same computing cluster.

We select our boundary integral formulation to be 2nd Fredholm type because of the upper bounded condition number. The number of iterations in GMRES display this clearly as the wellposedness functions as a type of analytic preconditioning. Figure 8.5 gives the number of iterations in BCMATRIX versus number of elements and element order with 2048 elements.

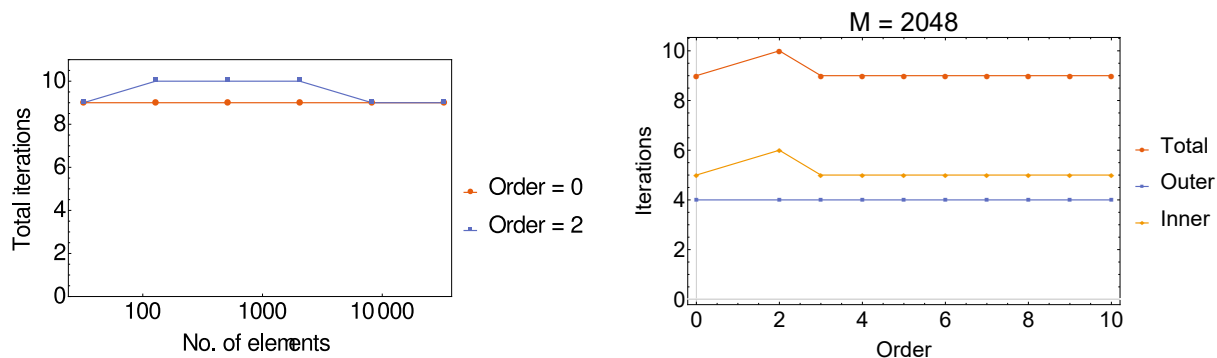


Figure 8.5: Number of iterations in GMRES versus M and e for $M = 2048$. The number of iterations stays mostly constant despite the increase in M and e , i.e. system matrix size. The prescribed tolerance is $10^{-3.5}$.

Figure 8.5 shows the bounded conditioning maintains a relatively constant number of iterations in the matrix solver to reach the desired residual of $10^{-3.5}$, which is the approximate limit for FMM order 6. Although the number of iterations vary between different structures, FMM orders, and conditions, Figure 8.5 clearly illustrates the advantage of the well-conditioned system matrix arising from Fredholm integrals of the 2nd type.

Finally, we test the limits of memory usage in PISCS with MPI processes. COSY allocates the memory required at start, so all variables and procedures are stored in RAM. While modifiable, COSY is preconfigured with ~ 2 GB [89]. The memory usage in PISCS depends mostly on N, M, e . PISCS ran successfully using the standard COSY configuration with $N \leq 10^6$ and modest M, e . With modest N , we increase the memory in COSY up to 3.5 GB for $M > 2048, e = 2$. PISCS ran out of memory when the matrix size reaches $> 3 \times 10^6$ at $M \sim 5 \times 10^5, e = 2$, though it is possible to continue increasing the memory. We have tested and presented the overall performance of PISCS. We test accuracy using an example sphere, a cylinder, and other structures.

8.2 Accuracy

The perfect conducting sphere with charges is a well-studied problem using the image charge method [35]. Without charges, the interior has constant potential and zero field determined by the boundary conditions (BC). The analytical potential $\phi(\mathbf{x})$ at observation point \mathbf{x} for a single charge q at \mathbf{y} inside the sphere is given by (8.1) for Dirichlet BCs [35].

$$\phi(\mathbf{x}) = \frac{q}{4\pi\epsilon_0} \left(\frac{1}{\|\mathbf{x} - \mathbf{y}\|} - \frac{\frac{R}{\|\mathbf{y}\|}}{\left\| \mathbf{x} - \frac{R^2}{\|\mathbf{y}\|^2} \mathbf{y} \right\|} \right) + V. \quad (8.1)$$

ϵ_0 is the permittivity, R is the sphere radius, and V is the constant potential on the sphere. For multiple charges, we simply sum (8.1) at all \mathbf{y}_i , $i = 1 \cdots N$. Based on (8.1), the potential in the central XY -plane of the sphere looks like Figure 8.6 without and with charges.

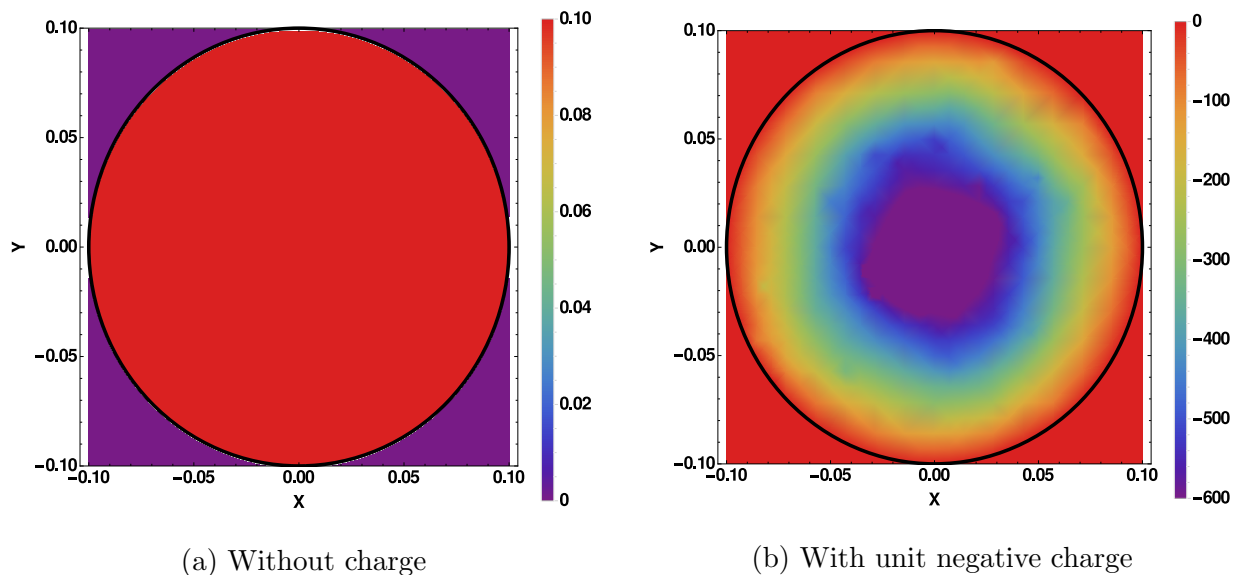


Figure 8.6: PEC sphere with $R = 0.1$ m, $V = 0.1$ V (a) without charge and (b) with ~ 1000 charges, $\frac{q}{\epsilon_0} = -1$ based on (8.1) in the XY -plane. Position is given in meters. The spherical surface is indicated by the circle. The units are chosen so that the effects are visible with modest N .

Figure 8.6 shows a sphere without and with ~ 1000 charges, where $\frac{q}{\epsilon_0} = -1$. The units are chosen so that the effects show with $N \sim 1000$. The field is completely from the charges and otherwise zero, which is not a good test for the fields from the BEM. We use this picture to check the accuracy of the potential from the BEM with charge adjusted appropriately. This test uses the normal vector field interpolation. We choose a sphere with $R = 0.1$ and $V = 0.1$, which looks similar to Figure 8.1. Figure 8.7 shows the potential in the central XY -plane from PISCS. We see the improvement from the higher element order with small M . The near boundary evaluation causes the artifacts near the surface as explained in Chapter 7.

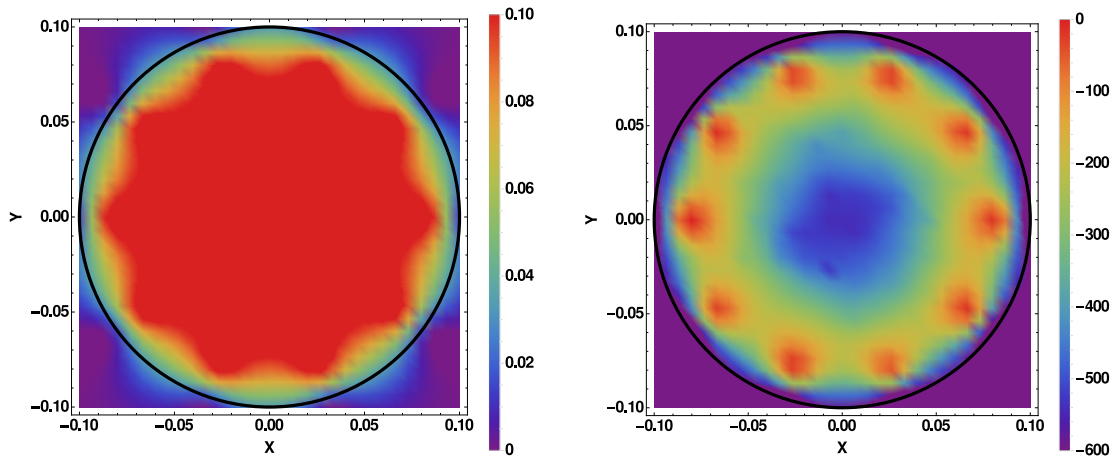
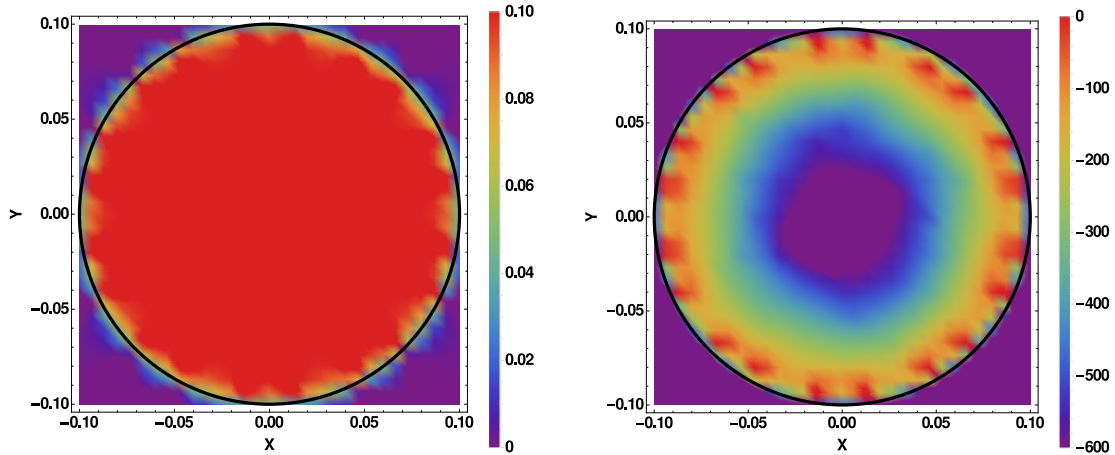
(a) $M = 80, e = 0$, w/o charge (left) and w/ charge (right)(b) $M = 80, e = 3$, w/o charge (left) and w/ charge (right)

Figure 8.7: Potential [V] in XY -plane from PISCS for the PEC sphere with ~ 1000 electrons and $\frac{q}{\epsilon_0} = -1$. Position is given in meters. For these results, we use our NVF interpolation. The spherical surface is indicated by the circle.

To quantify the accuracy of PISCS for practical use, we evaluate the potential at the location of the electrons as compared to (8.1). Figure 8.8 shows the percent error for $M = 80, 1280$. The element order improves the average error more with small M , but the higher element order ultimately gives better overall accuracy. PISCS and the NVF interpolation give good results using the example sphere. We show the practical case with a charged distribution

gives acceptable error and the high order elements improve the behavior with a small number of elements.

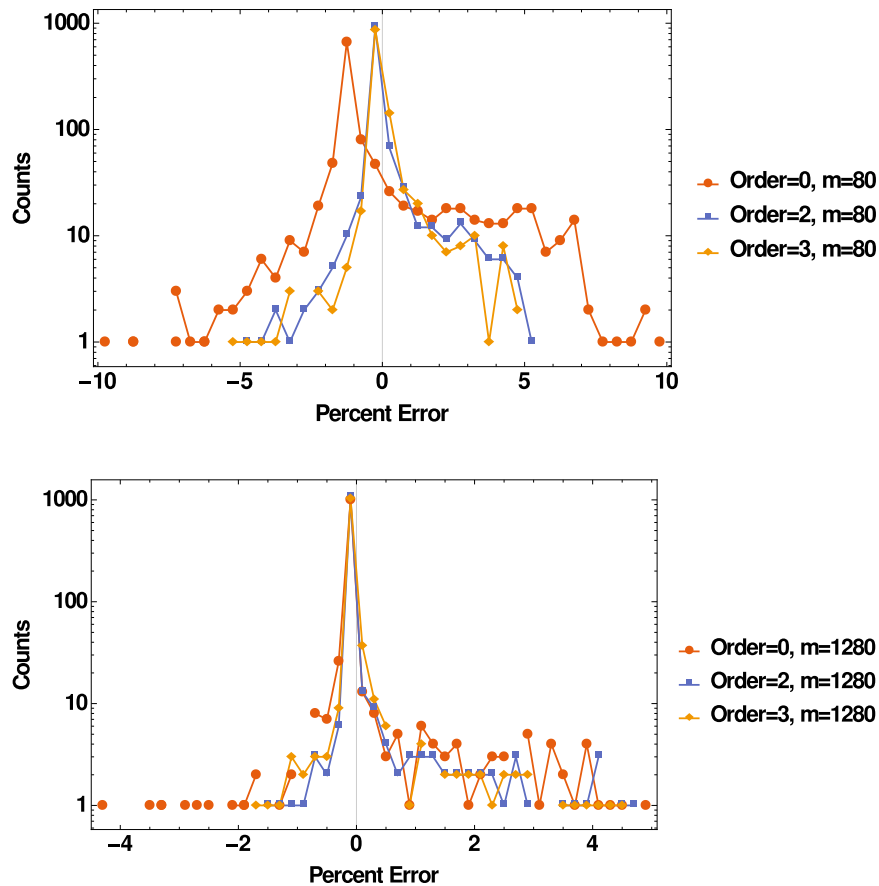


Figure 8.8: Percent error for potential at the electrons with $M = 80$ (top), $M = 1280$ (bottom). $e = 2, 3$ show improved average error compared to constant element order with the higher gain at $M = 80$.

To test the field accuracy and Neumann boundary conditions, we use an open-ended cylinder where the normal is known to minimize error in the computed boundary conditions. This structure looks similar to a Hall axial probe for measuring a quadrupole magnet. The setup might look like Figure 8.9.

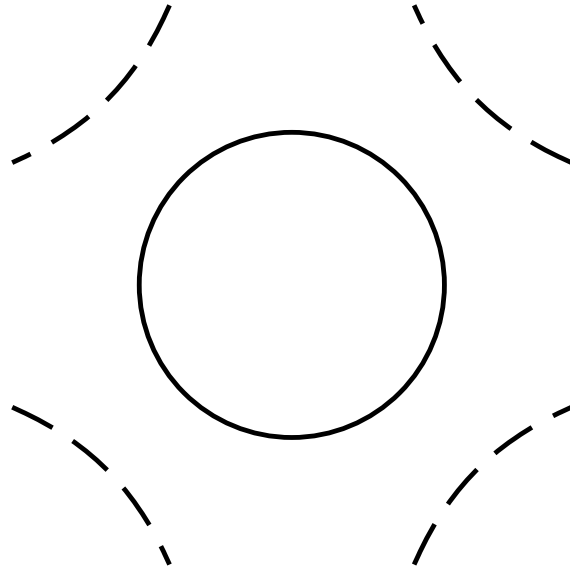


Figure 8.9: An example configuration for a Hall axial probe inside a quadrupole magnet.

For the following studies, we work in SI units to provide a realistic picture. In this case, we use PISCS with Vandermonde interpolation to show the two versions behave similarly. The CAD and an example mesh with $M = 224, e = 10$ for the cylinder is shown in Figure 8.10.

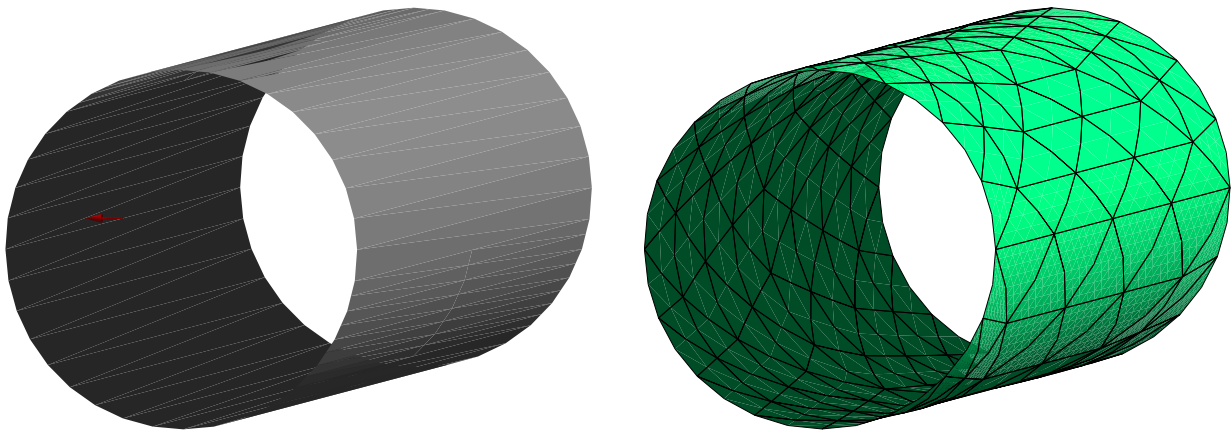


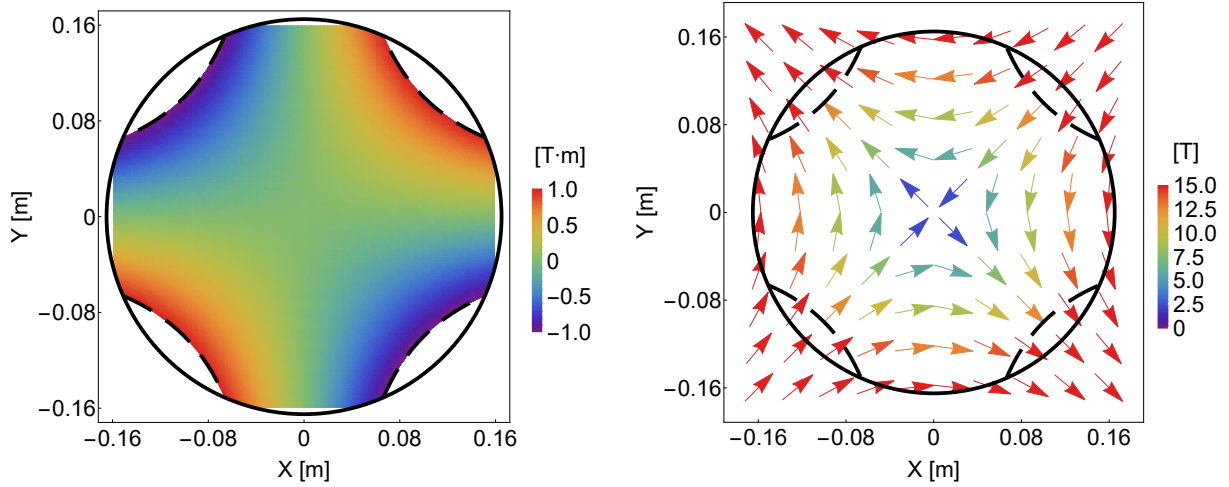
Figure 8.10: CAD and mesh of cylinder with $R = 0.165, L = 1$. Mesh has 224 elements at $e = 10$. Multipole boundary conditions are placed on the surface.

We use the analytical scalar potentials and fields for a magnetic defocusing quadrupole and octupole as our boundary conditions, given in (8.2) and (8.3). We set the gradient g to be -100 T/m for the quadrupole and -1000 T/m³ for the octupole.

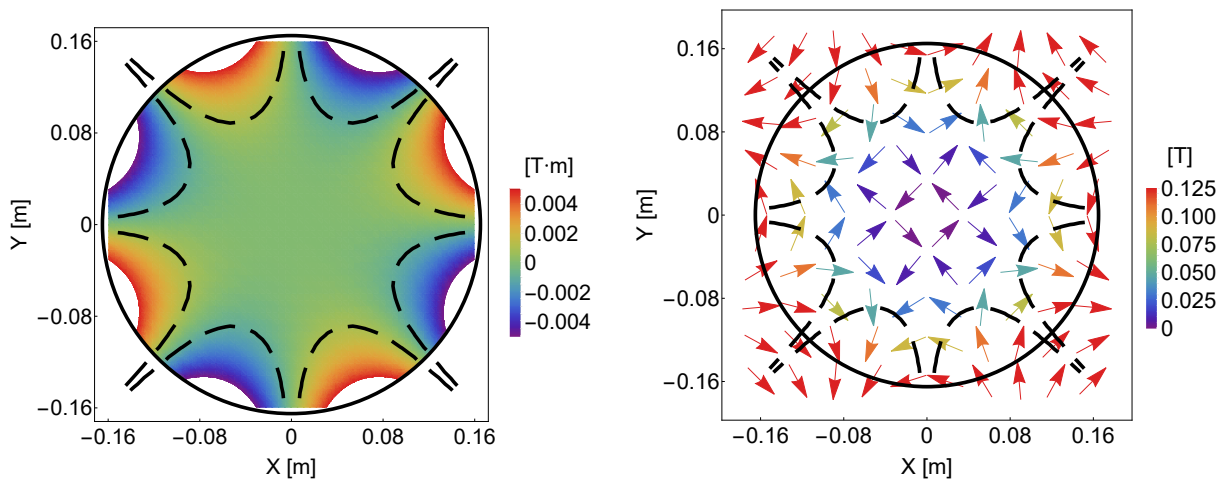
$$\begin{aligned}\phi_M(X, Y) &= -gXY, \\ B_X(X, Y) &= gY, \\ B_Y(X, Y) &= gX.\end{aligned}\tag{8.2}$$

$$\begin{aligned}\phi_M(X, Y) &= -gXY, \\ B_X(X, Y) &= \frac{1}{24}g(3X^2Y - Y^3), \\ B_Y(X, Y) &= \frac{1}{24}g(X^3 - 3XY^2).\end{aligned}\tag{8.3}$$

The cylinder has radius $R = 0.165$ m and length $L = 1$ m. The expected central XY -plane without charges is given in Figure 8.11. Figure 8.12 shows the quadrupole potentials and fields computed by PISCS without charge. Similarly, Figure 8.13 shows the octupole without charges. In these tests, we use FMM order 10.



(a) Quadrupole scalar potential (left) and field (right) in central XY-plane.



(b) Octupole scalar potential (left) and field (right) in central XY-plane.

Figure 8.11: Expected scalar potential and field of of magnetic (a) quad and (b) octupole without charges. Position is given in meters.

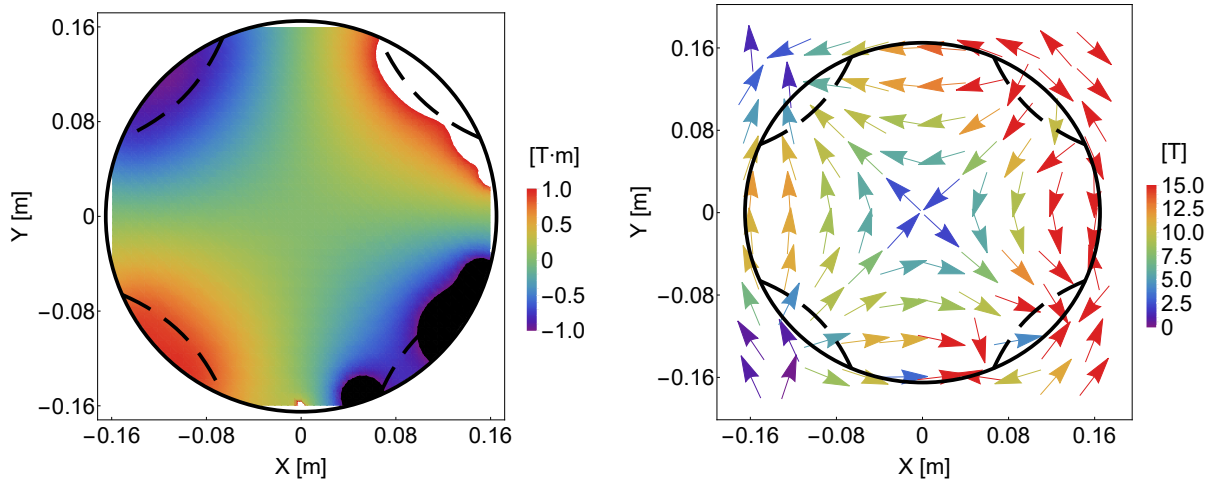
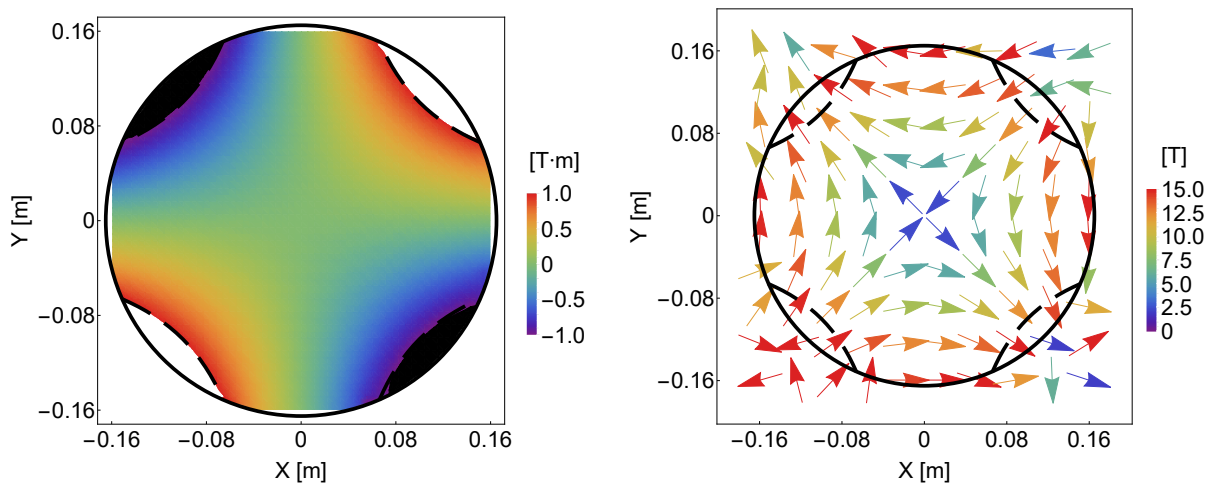
(a) $M = 224, e = 0$ (b) $M = 224, e = 10$

Figure 8.12: Potential [T·m] and field [T] in XY -plane from PISCS for the defocusing quad with $g = -100 \frac{\text{T}}{\text{m}}$. Position is given in meters. For these results, we use PISCS and Vandermonde interpolation. The cylinder is indicated by the solid circle and the equipotentials by the dashed curves.

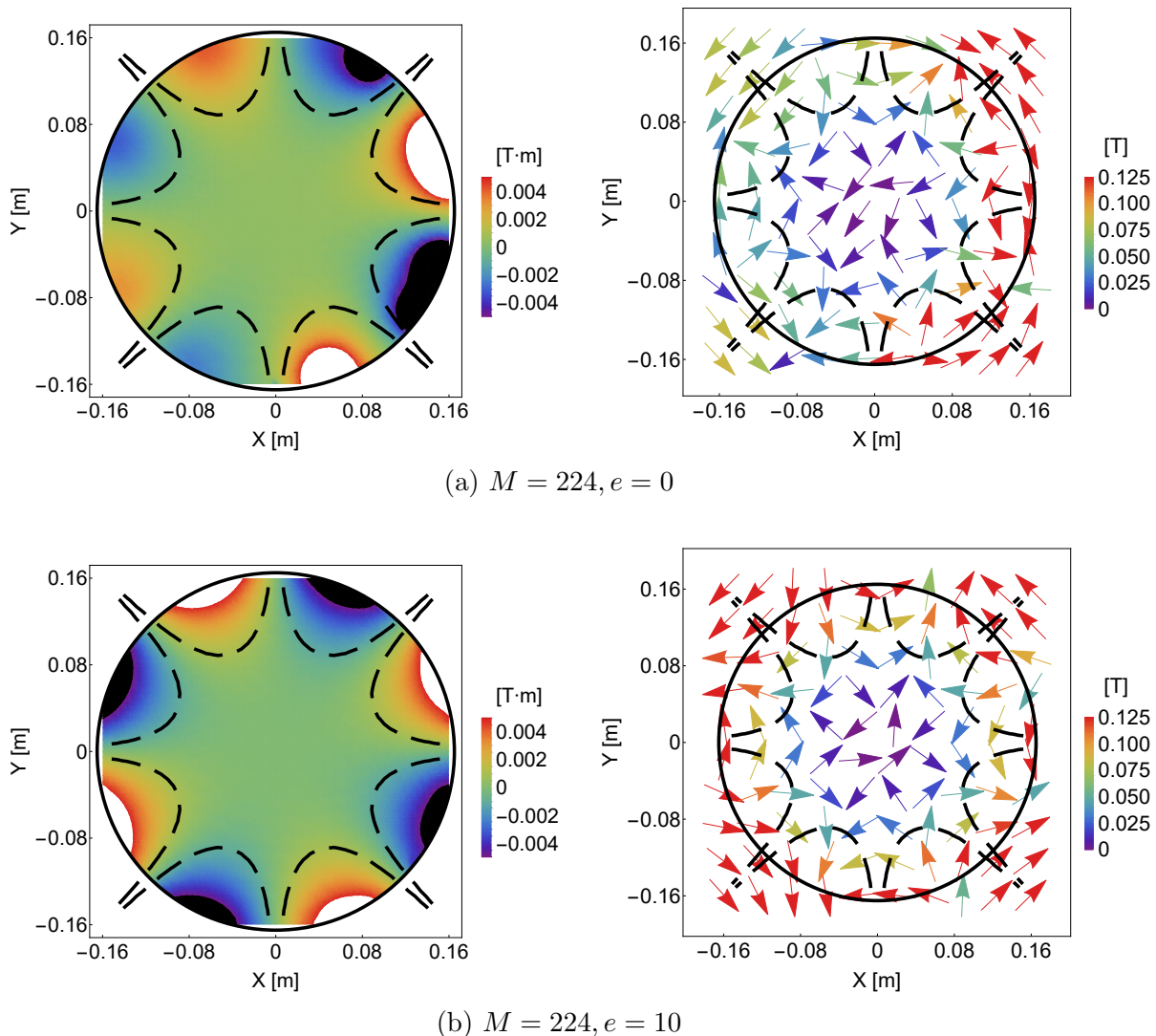
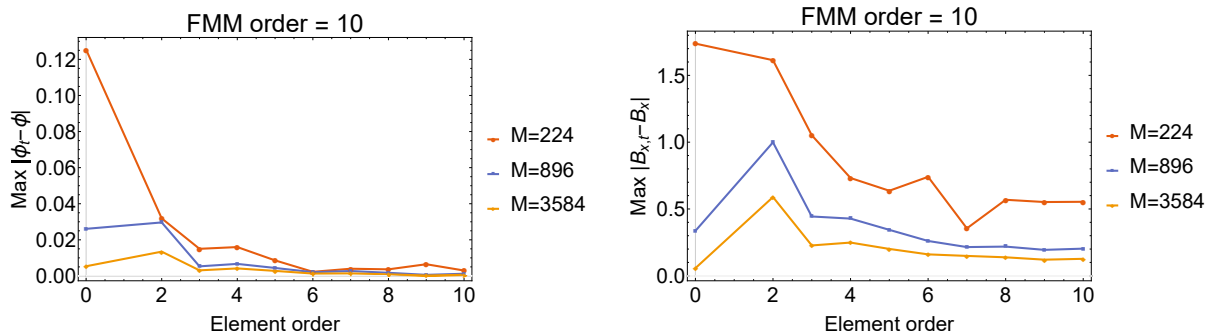


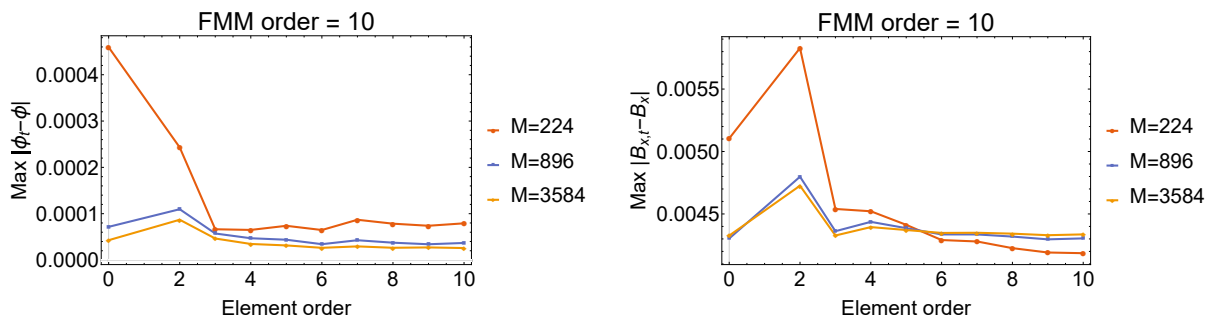
Figure 8.13: Potential [T·m] and field [T] in XY -plane from PISCS for the magnetic octupole with $g = -1000 \frac{\text{T}}{\text{m}^3}$, using PISCS and Vandermonde interpolation. Position is given in meters. The cylinder is indicated by the solid circle and the equipotentials by the dashed curves.

When element order $e = 10$, the potentials and fields look almost identical to the theory except near the surface. The computed octupole displays similar behavior in Figure 8.13. Near the center, the multipoles have near zero potentials and fields and are most sensitive to the approximations from the FMM, interpolation, and quadrature in Figures 8.12 and 8.13

We also scan the potentials and fields along y with $X = 0, Z = 0.5$ for the quadrupole and similarly for the octupole with $X = 0.033$. From (8.2), Figure 8.14 shows the maximum difference between theory and PISCS for the quadrupole and octupole versus element order and M within the stable region for the cylinder.



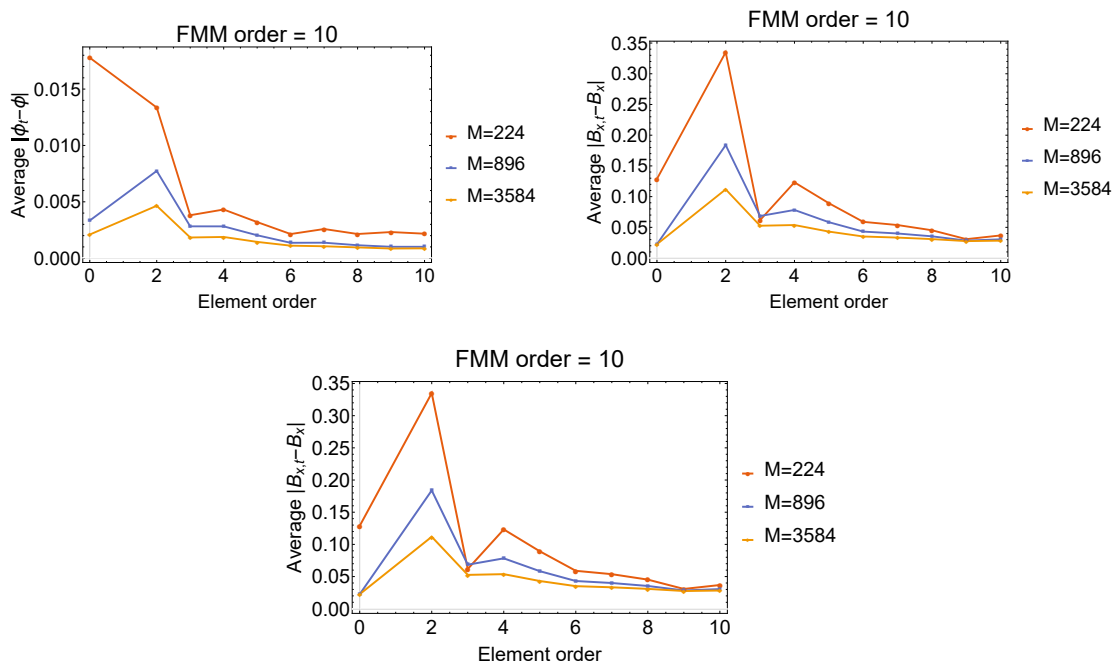
(a) Quadrupole, max difference for potential (left) and B_x (right) along y -axis, $Z = 0.5$



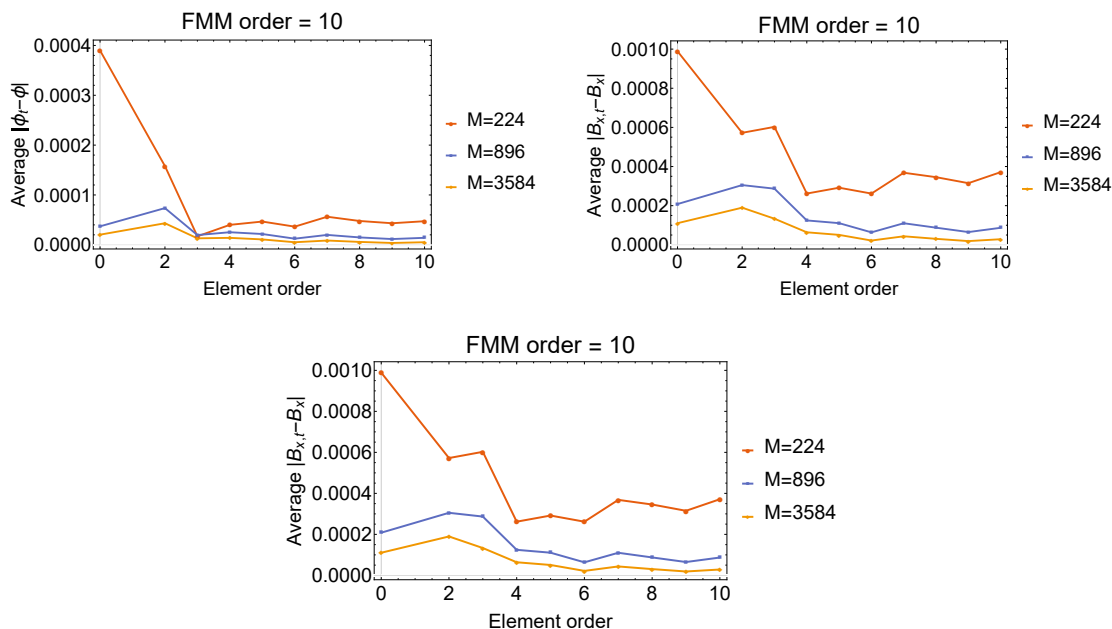
(b) Octupole, max difference for potential (left) and B_x (right) along y , $X = 0.033, Z = 0.5$

Figure 8.14: Maximum difference between theory and PISCS for (a) quadrupole and (b) octupole. There is a spike for $e = 2$, suggesting the approximation error is high, but overall higher order elements improve the accuracy.

The maximum difference tends to decrease with element order except when $e = 2$. For Vandermonde interpolation, element order $e = 2$ suggests the derivatives are linear, so the normals from $\partial_u \mathbf{x} \times \partial_v \mathbf{x}$ have high error and results in the large spike. We also know the normals and the quadrature have low accuracy for quadratic order from Figs. 6.8-6.11. Even so, the high-order elements tend to decrease the error up to order 10. We see this a bit more clearly by looking at the average difference in the central XY -plane, shown in Figure 8.15.



(a) Quadrupole, average difference for potential (top left), B_x (top right), and B_y (bottom) in XY -plane, $Z = 0.5$

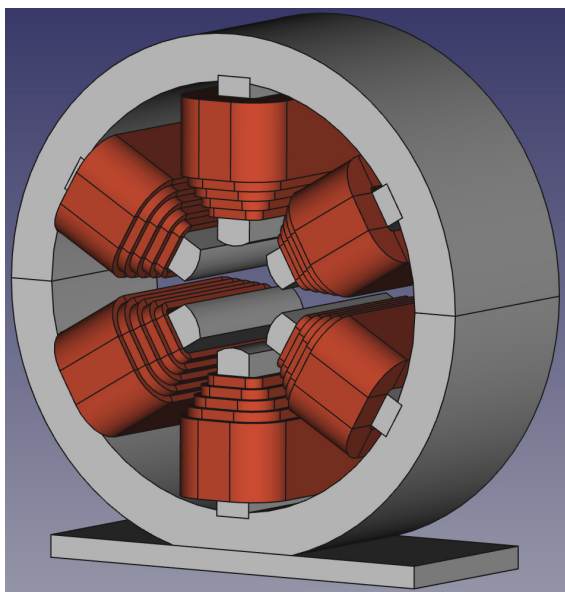


(b) Octupole, average difference for potential (top left), B_x (top right), and B_y (bottom) in XY -plane, $Z = 0.5$

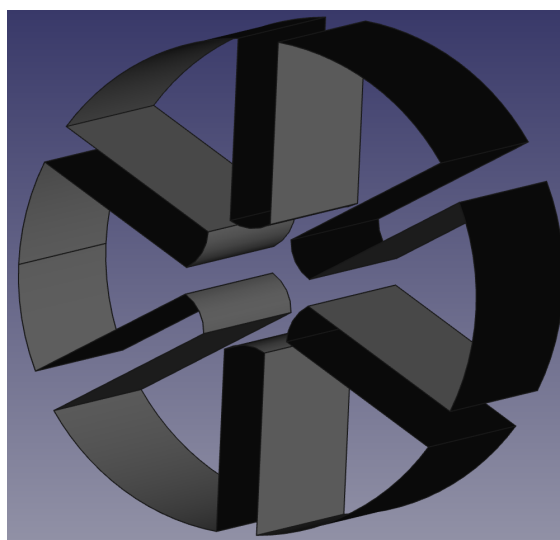
Figure 8.15: Average difference between theory and PISCS for (a) quadrupole and (b) octupole.

The average difference, shown in Figure 8.15, is steadily decreasing with element order. The behavior is a bit erratic for $M = 224$. There are some large spikes in error at low element order. We see the expected overall behavior, where higher element order improves the accuracy. When M is larger, the behavior is more stable as well.

The previous tests show PISCS with different interpolation methods works with Dirichlet and Neumann BC and simple structures. However, the BEM algorithm allows arbitrary structures, which are commonly stored as CAD files. We create various structures in CAD for study with PISCS. We next show and discuss the results. First, we examine a 10 cm length sextupole magnet with aperture radius of 2.5 cm based on a design for IOTA. Figure 8.16 shows the CAD for the IOTA sextupole, courtesy of Fermilab. We know the theoretical potentials and fields for the sextupole, so we discuss the error with no charge present.



(a) Sextupole magnet



(b) Inner surface w/o coils

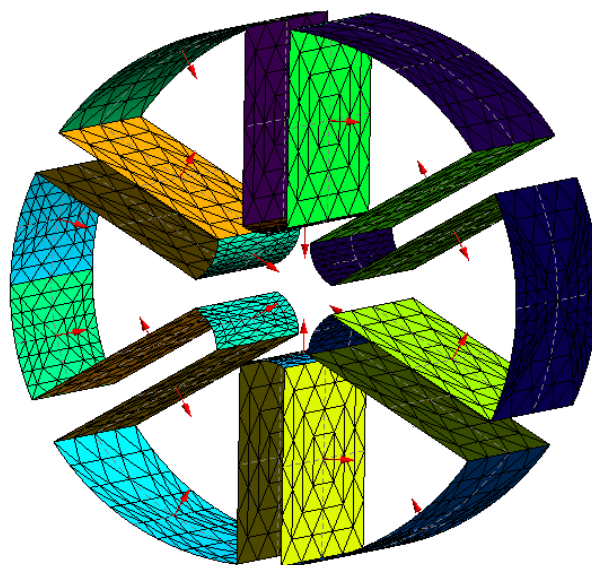
(c) Inner surface mesh, $M = 2080, e = 6$

Figure 8.16: CAD of (a) IOTA sextupole magnet and (b) inner surface without coils. Inner surface mesh example shown in (c) with $M = 2080, e = 6$. Current is run through coils such that the pole tips produce the magnetic field. Inner surface is used in PISCS.

The actual sextupole features coils wrapped around the poles such that the strongest magnetic field is produced at the pole tips. The pole tips are shaped to match the equipotentials

based on multipole theory with some correction for the coils. To form an equivalent situation with a thin shell, we trace the inner surface of the sextupole without the coils and set a nonzero potential at the tips. The sextupole potential and field is given in (8.4), where we set $g = -1000 \frac{\text{T}}{\text{m}^2}$.

$$\begin{aligned}\phi(X, Y) &= -\frac{1}{6}g(3X^2Y - Y^3), \\ B_X(X, Y) &= gXY, \\ B_Y(X, Y) &= \frac{1}{6}g(3X^2 - 3Y^2).\end{aligned}\tag{8.4}$$

From (8.4), we should get Figure 8.17 in the XY -plane.

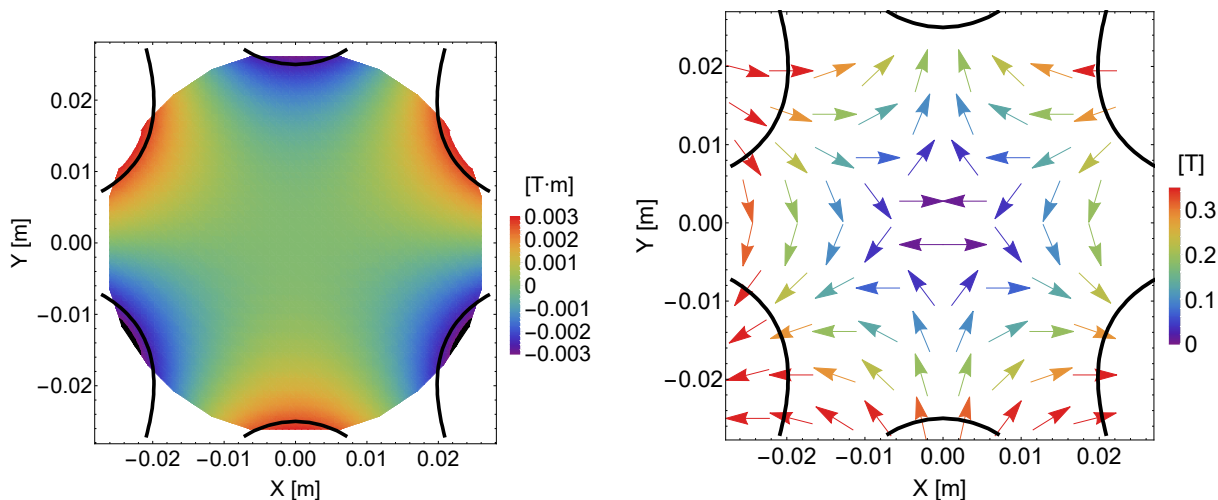


Figure 8.17: Sextupole potential [T·m] (left) and field [T] (right) in XY -plane. Position is given in meters. The pole tips are indicated by the solid curves.

We compute the Dirichlet boundary conditions using (8.4) at the pole tips and set zero everywhere else. We compute the potential and fields in a similar way to the cylinder using Vandermonde interpolation with FMM order 10. Figure 8.18 shows the potentials and fields in the XY -plane.

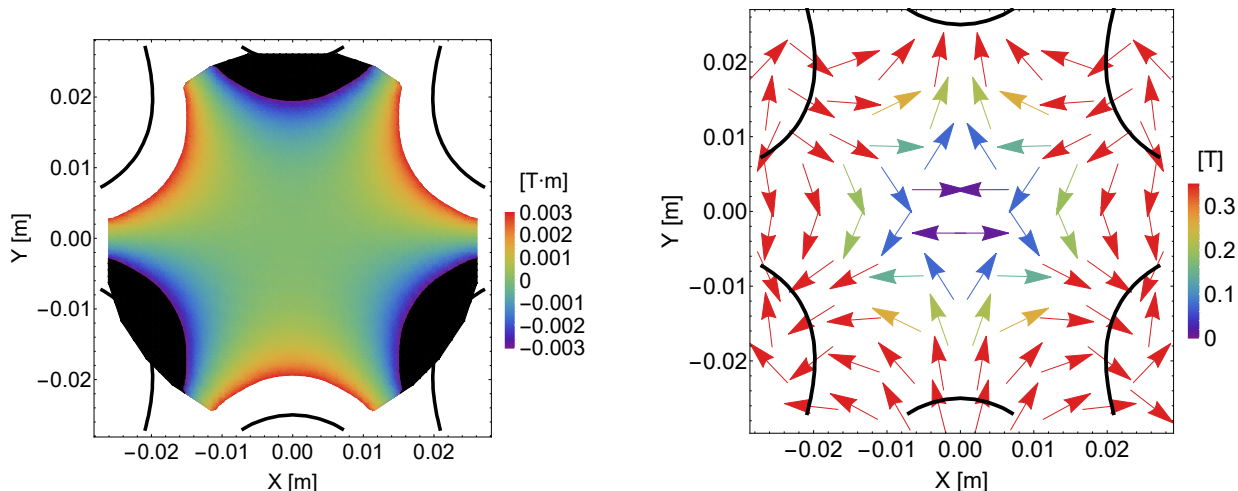
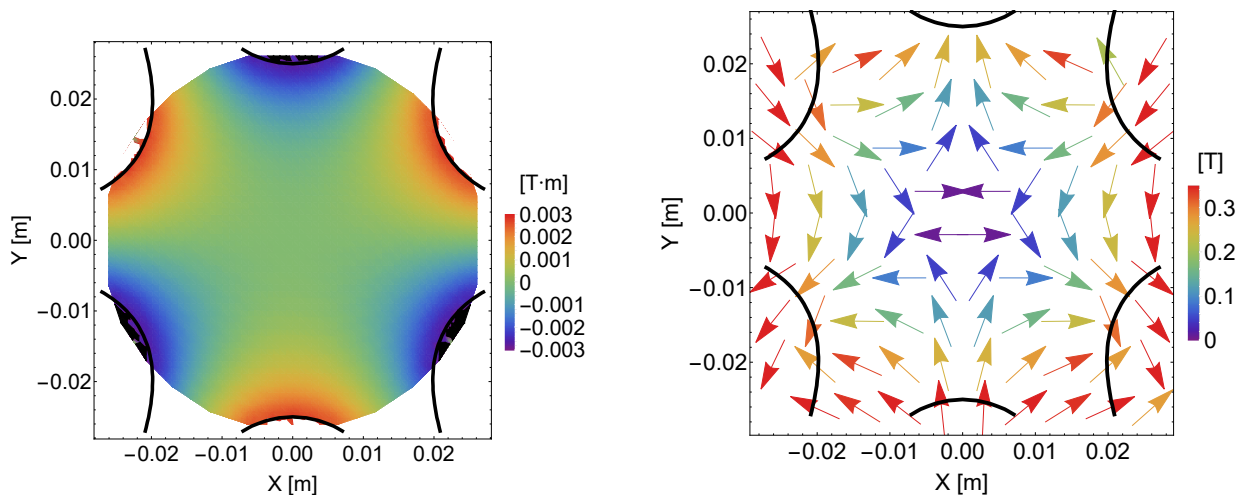
(a) $M = 2080, e = 0$ (b) $M = 2080, e = 6$

Figure 8.18: Potential $[T \cdot m]$ (left) and field $[T]$ (right) for sextupole with no charges from PISCS with Dirichlet conditions. Position is given in meters. Mesh is based on inner surface in Figure 8.16. Boundary potential is nonzero only at the pole tips. XY -plane is shown.

The potentials and fields are approaching the theory, so PISCS can handle more realistic structures. We show the average difference in the central XY -plane in Figure 8.19. Overall, the accuracy is improving, though the results seem sensitive at certain orders. While the error is large at $e = 2$, we can stabilize it by increasing M .

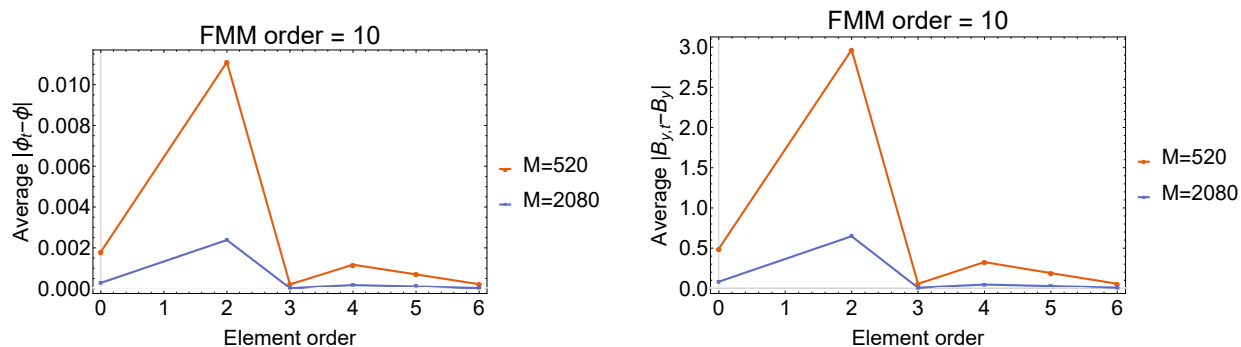


Figure 8.19: Average difference between theory and PISCS for the sextupole magnet in XY -plane. Potential (left) and B_y (right) shown. Overall, the difference is shrinking with element order.

We demonstrate PISCS behaves as expected in several test cases with some exceptions. There are some stability issues to understand for particular combinations of element orders, number of elements, and structures. Overall, the accuracy improves, drastically in some cases, with high e and small M . We next consider cases that are not as well understood. For instance, the vacuum pipe housing a beam is studied under a particular set of assumptions in the ideal case [49, 176]. In Ref. [49], the author derived the Laslett coefficients for several geometries, which describe the image charge forces in, for instance, an elliptic cylinder containing a line of charge along the central axis. When considering a high-intensity beam, the space charge forces can prevent such a charge distribution. We know cylindrical symmetry allows cancellation of the image force, but again, small defects or warping may break this symmetry. We use PISCS to form a picture of the situation for a half-filled and completely filled vacuum pipe segment with an elliptical cross-section. We assume the potential is zero on the surface, meaning the boundary conditions are completely due to the beam interaction with the metal. The pipe has dimensions of $R_x = 0.0254$, $R_y = 0.02$, $L = 0.1$ [m]. The CAD and an example mesh with $M = 384$, $e = 3$ for the elliptic cylinder is shown in Figure 8.20.

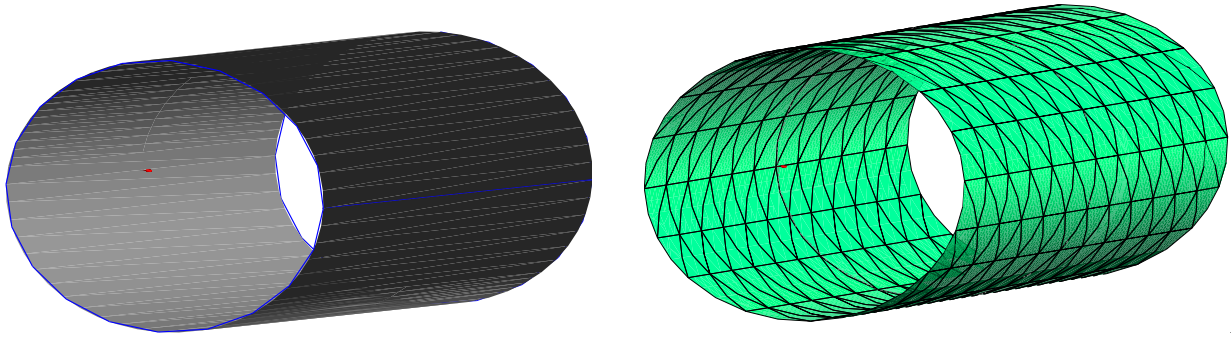


Figure 8.20: CAD and mesh of the elliptic cylinder with $R_x = 0.0254$, $R_y = 0.02$, $L = 0.1$ [m]. Mesh has 384 elements with $e = 3$. $V = 0$ V placed on surface.

We use a transverse Gaussian, longitudinally uniform electron beam with $N = 5 \times 10^4$ when half-filled and $N = 10^5$ when filled. Figure 8.21 shows the results in the central ZY -plane.

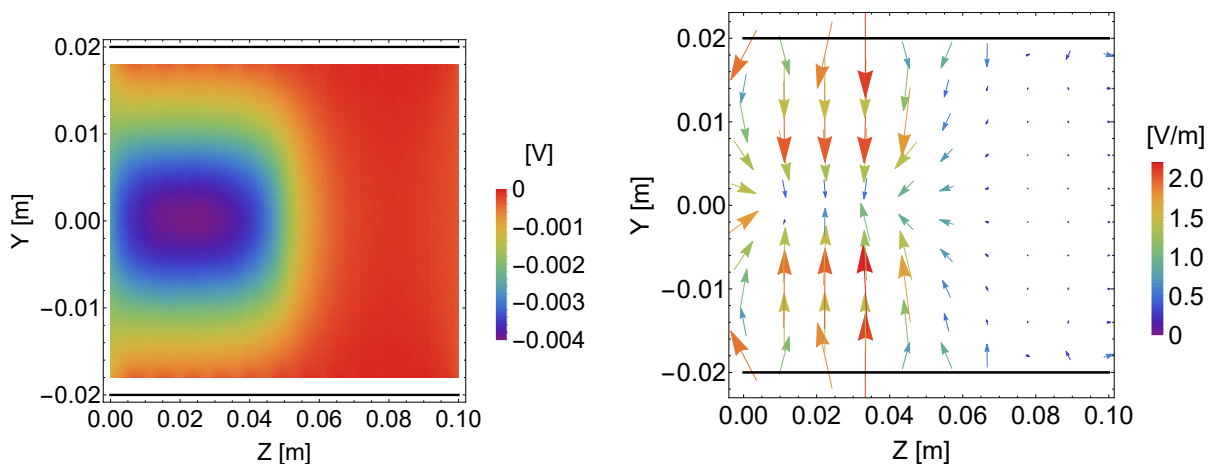
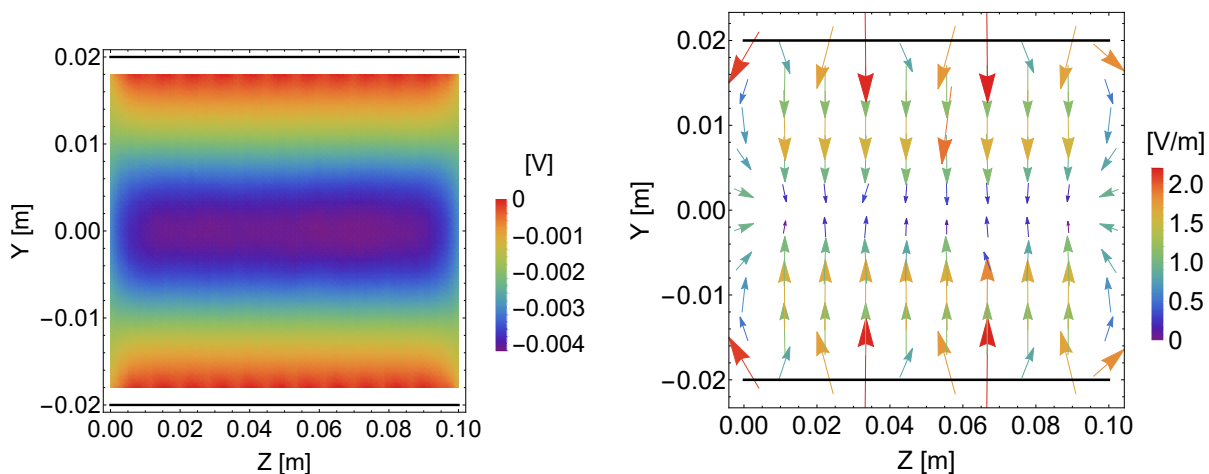
(a) Half-filled, $N = 5 \times 10^4$ (b) Filled, $N = 1 \times 10^5$

Figure 8.21: ZY -plane of elliptic cylinder containing a beam, zero potential on the surface. Position is given in meters. We use $M = 384, e = 3$ in PISCS with Dirichlet conditions. Potential [V] (left) and field [V/m] (right) is shown. The surface is shown by the solid lines. This depicts the beam frame, where the charges are essentially static. We can see the field and potential mostly due to the beam die out rapidly away from the charges.

The fields and potentials are mainly due to the beam from the static depiction in Figure 8.21. The strength rapidly drops away from the beam. If we separate the self fields and the image fields, we would see the image charge effect alone. This would require slightly modifying PISCS before attempting.

We did one final static study using PISCS. There has been some recent interest in an ultra low emittance electron source [177]. One idea uses, in essence, many sharp single-atom nanotips, which leads to an ultracold field emission cathode [178]. The biggest advantage of this setup is the extremely high field at the sharp emitter from a modest voltage. The emission probability is high and the emitted electrons are strongly accelerated almost immediately after emission. The issue with this setup appears due to the compact density of the emitted distribution. Space charge is strong which can cause large transverse emittance growth just before the acceleration. We also must consider the feedback to the emitter surface, which modifies the emission probability and possibly produces nonnegligible image forces. While PISCS is not at a state to fully study the nanoemitter array, we may consider the static picture with and without some electrons. We setup a single rounded emitter inside essentially infinitely long parallel plates. The emitter plus connected plate are shown in Figure 8.22.

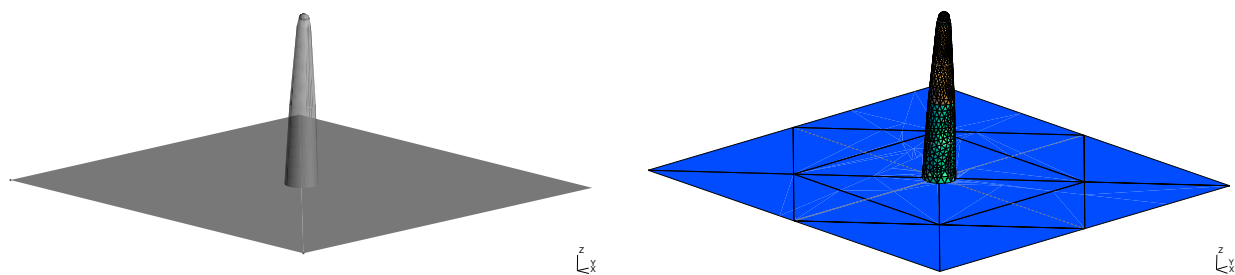


Figure 8.22: Single emission tip on substrate, CAD (left) and example mesh with $e = 4$ (right). The enclosing walls are not shown. The mesh had $M = 8158$, $e = 4$. Nanoemitter is 2.13 nm tall and ≤ 0.205 nm wide. The tip is spherical with $R = 0.065$ nm.

The nanoemitter has a height of 2.13 nm and max width of 0.205 nm. The spherical tip has a radius of 0.065 nm. We enclose the hollow tip inside a box of dimensions $L = 5$ nm, $W = 5$ nm, $H = 10$ nm. The tip is attached to the center of one plate. We set a constant potential of $V = -0.1333$ to this plate, linearly varying potential along the walls, and $V = 0$ on the other plate. We should have a constant electric field along $-Z$ with $\|\mathbf{E}\| = 1.33 \times 10^7$ except

near the emitter. We use a 3D Gaussian distribution of $N = 100,1000$ actual electrons within a narrow cone emitting from the spherical tip.

We do not expect sensible results near the surface due to the near boundary instability, but we form some qualitative insights. Figure 8.23 depicts the ZY central plane without and with the electrons.

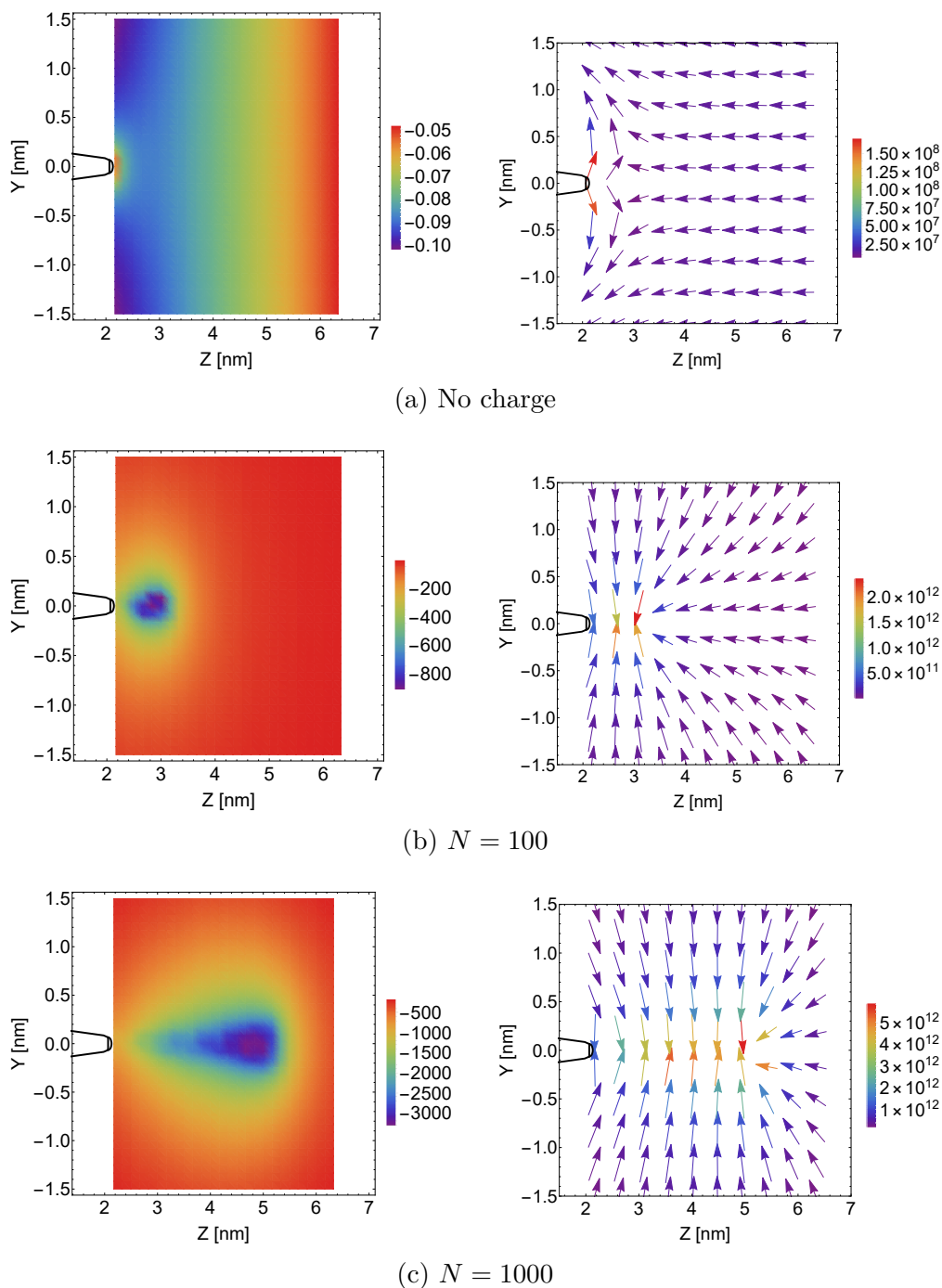


Figure 8.23: ZY -plane of the nanoemitter without and with electrons. Potential [V] (left) and field [V/m] (right) is shown. The field is strongly modified due to the geometry at the emitter tip, though the direction and strength is off due to the near boundary instability in the BEM. We see the presence of electrons strongly affects the situation at the nanoscale. The emitter tip is indicated by the solid curve.

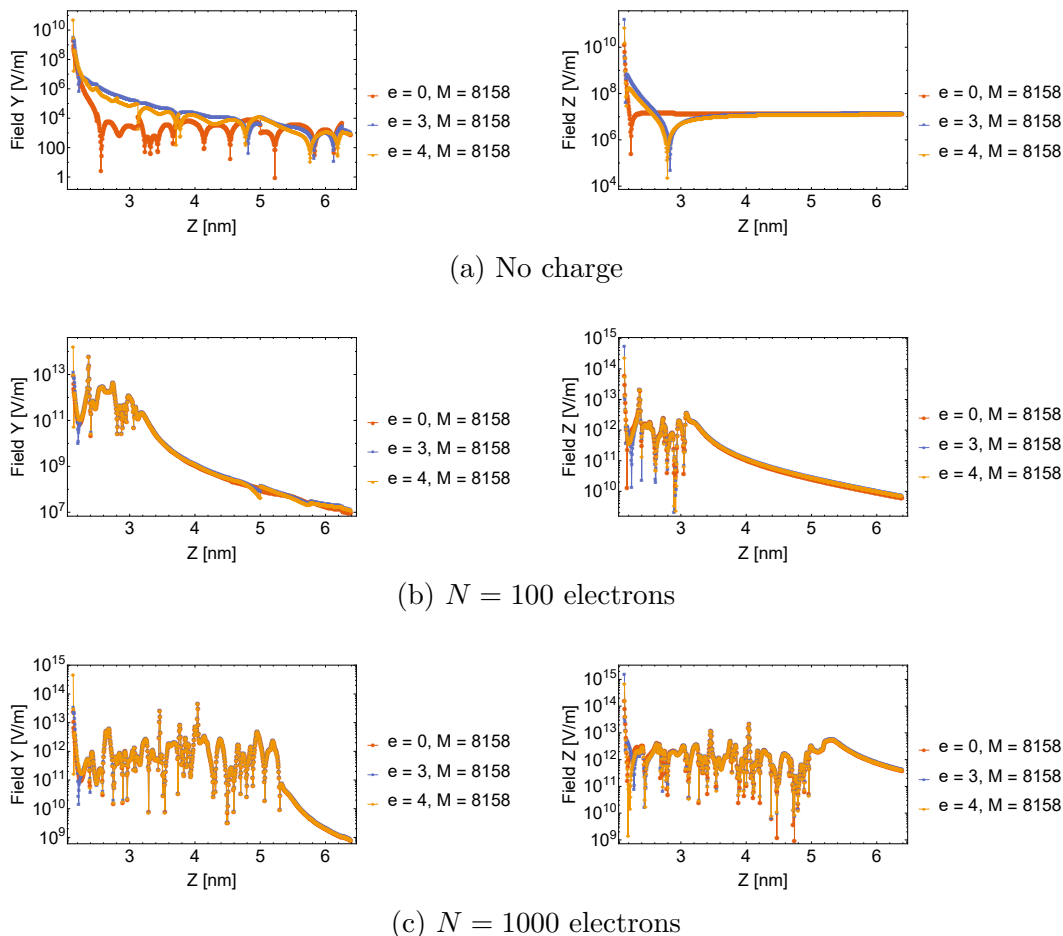


Figure 8.24: Absolute value of E_Y (left), E_Z (right) [V/m] of the nanoemitter without and with electrons, starting just past the emitter tip and scanned along the Z -axis [nm]. The transverse fields persist without charge, though relatively small. Without charges, E_Z rapidly approaches the parallel plate field, indicating the range of the emitter field. With charge, the transverse fields become significant, which suggests emittance growth due to space charge.

The potential and field near the surface of the emitter is unstable as expected in Fig. 8.23a. We see the effective range from the emitter is small, ~ 1 nm. Afterwards we only see effects due to the parallel plates. From Fig. 8.23b, 8.23c, the charges' self-field is significantly stronger than the parallel plates. Figure 8.24 shows the absolute value of E_Y , E_Z scanned along the Z -axis, starting just after the emitter tip. E_X looks similar to E_Y , so only E_Y is shown. Without charge, the transverse fields should be strongest near the emitter due to the geometry and E_Z should rapidly approach a constant equal to the parallel plate field, which is

observed in Figure 8.24a. When charges are present, the field strength increases significantly, shown in Figures 8.24b, 8.24c. The large values indicate the approximate position of the bunch. Clearly, the transverse field component is significant due to the bunch. Figures 8.24b, 8.24c suggest as the charge accumulates, the transverse force is nonnegligible. This could degrade the transverse emittance significantly. If strong enough, the space charge force still affects bunch quality at extraction. While this static depiction may not be perfectly accurate, Figure 8.24 emphasizes the self-forces cannot be ignored. We note the effects shown are within 10 nm of the emitter. A realistic setup is on the order of millimeters. This requires a thorough investigation including the emission probability which PISCS is not designed to include. We have tested and shown PISCS for various static cases. PISCS provides good accuracy and acceptable runtime. Thanks to the wellposedness, the solver requires relatively few iterations for complicated structures. We also show this is not significantly affected by matrix size. We now consider practical use for beam physics studies in terms of tracking and map analysis.

8.3 Self- and image charge forces

As previously mentioned in Chapter 7, we can combine PISCS with other beam physics packages to include boundary effects when studying, for instance, beam transport. We consider two structures where we have combined PISCS and COMFY to study the transverse effects. In the 2D approximation, we assume the presence of a long beam such that longitudinal effects are cancelled by symmetry and no fringe fields. We consider a thin slice of this beam to determine the transverse behavior with COMFY. For longitudinal symmetry, we chose a reference $Z = l/2$ for PISCS, where l is the kick length or element length. We first track a beam through an IOTA sextupole from Figure 8.16. The beam parameters are

given in Table 8.1 and the sextupole parameters are in Table 8.2. The distribution is selected to be fully contained within the aperture radius before and afterwards. The current only affects the space charge calculation. The beam-wall interaction depends on N . We choose negatively charged macroparticles to amplify the potentials and fields from PISCS to be of similar strength to space charge.

Table 8.1: Beam Parameters

Energy [MeV]	0.1
Species	Macroparticles
No. of particles	1000
Current [A]	10
Distribution	Uniform ($R = 0.01$ m)
Emittance [m·rad]	0

Table 8.2: Sextupole Parameters

Gradient [$\frac{T}{m^2}$]	10
Length [m]	0.1
Aperture radius [m]	0.025
BC type	Dirichlet
Calculation order	8
Element order	3
No. of elements	520

Figure 8.25 shows the initially parallel beam before and after the sextupole without space charge using COSY and PISCS. The sextupole is represented using $e = 3, M = 520$ for speed.

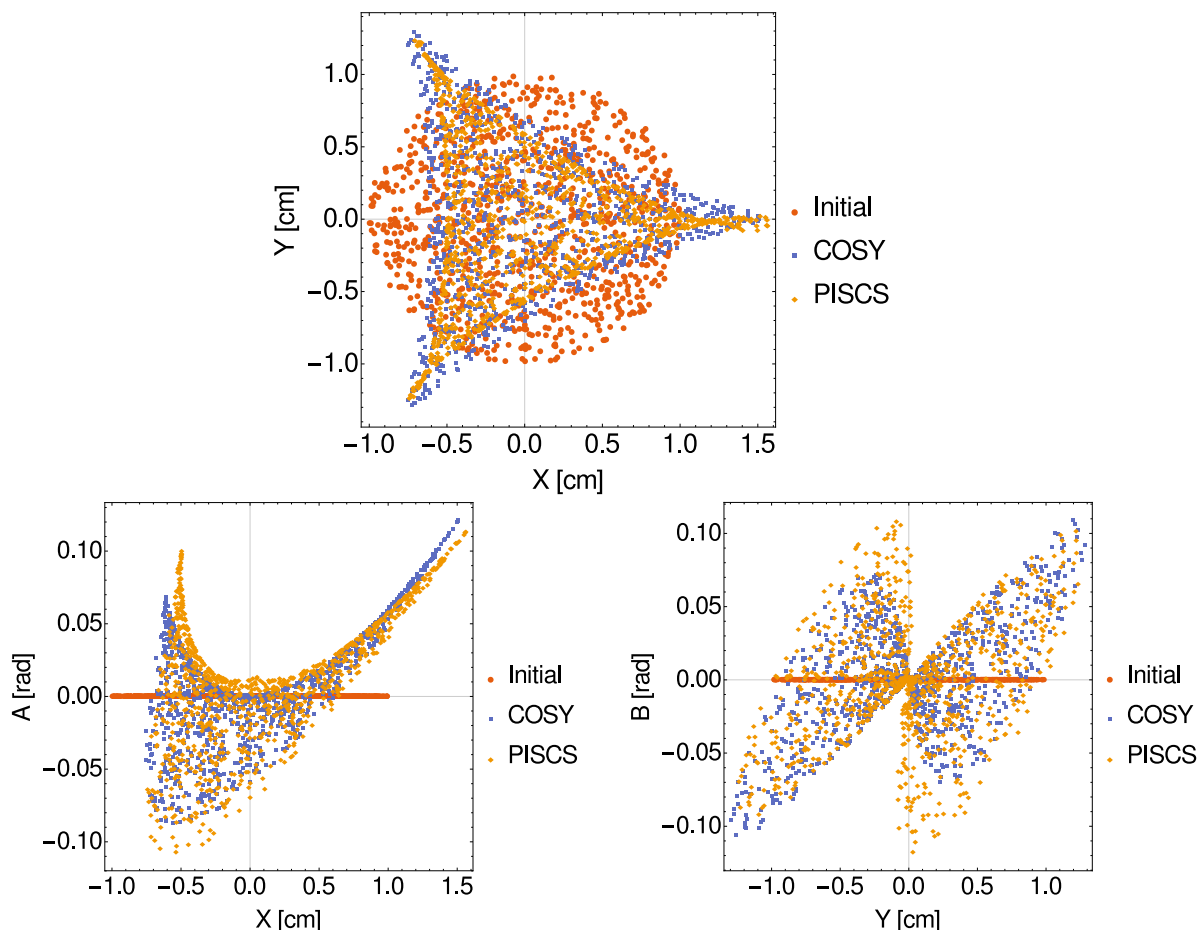


Figure 8.25: Comparison of tracking a 100 keV electron beam, $N = 1000$, using COSY and PISCS with an IOTA sextupole. XY (top), XA (bottom left), and YB (bottom right) shown. Positions are given in centimeters and a, b in radians. No space charge included and beam-wall interaction is negligible. The differences stem from the linear approximation and BEM error using $e = 3, M = 520$.

We see the differences are significant when considering practical parameters for PISCS. The time required to compute the sextupole potentials and fields is much greater than simply using COSY to compute the transfer map. Although we may improve the accuracy with higher e, M and number of kicks, we only use PISCS to isolate the beam-wall interaction. As discussed in Chapter 7, we include the potentials and fields from PISCS in the 2D-FMM module of COMFY, which shows when tracking the particles. The transfer map does not reflect the beam-wall interaction from PISCS or the space charge tracking from the FMM

for a single transport element. However, we see the effect in the map for multiple transport elements as the charge distribution is different after the first space charge kick.

The COSY beam physics package is much better at representing the kick of the common beamline elements. We describe the modification to PISCS to isolate the beam-wall interaction. The purpose of the merged PISCS-COMFY package is to study that feedback. We recall our Nyström discretization essentially gives $\mathbf{A}\boldsymbol{\eta}_1 = \mathbf{b}_1$ where \mathbf{b}_1 would be our unmodified boundary conditions. Let $f|_{\Gamma}$ represent the potential or field of the charges on the surface Γ . The modified equation due to charges is,

$$\begin{aligned} \mathbf{A}\boldsymbol{\eta}_2 &= \mathbf{b}_1 - f|_{\Gamma}, \\ \Rightarrow \mathbf{A}(\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1) &= \mathbf{A}\boldsymbol{\eta}_c = -f|_{\Gamma}. \end{aligned}$$

We use macroparticles to make this small effect visible. These small effects could build up and lead to instabilities only visible in repetitive systems such as rings. Using the same initial beam, we include space charge with the MoM and the FMM modules in COMFY. Figure 8.26 shows the final distributions. There are slight differences between the two methods, but we see the shape is retained even though the final emittance is slightly larger than without space charge. The major difference is the size of the beam envelope. The FMM again shows a slightly stronger transverse kick.

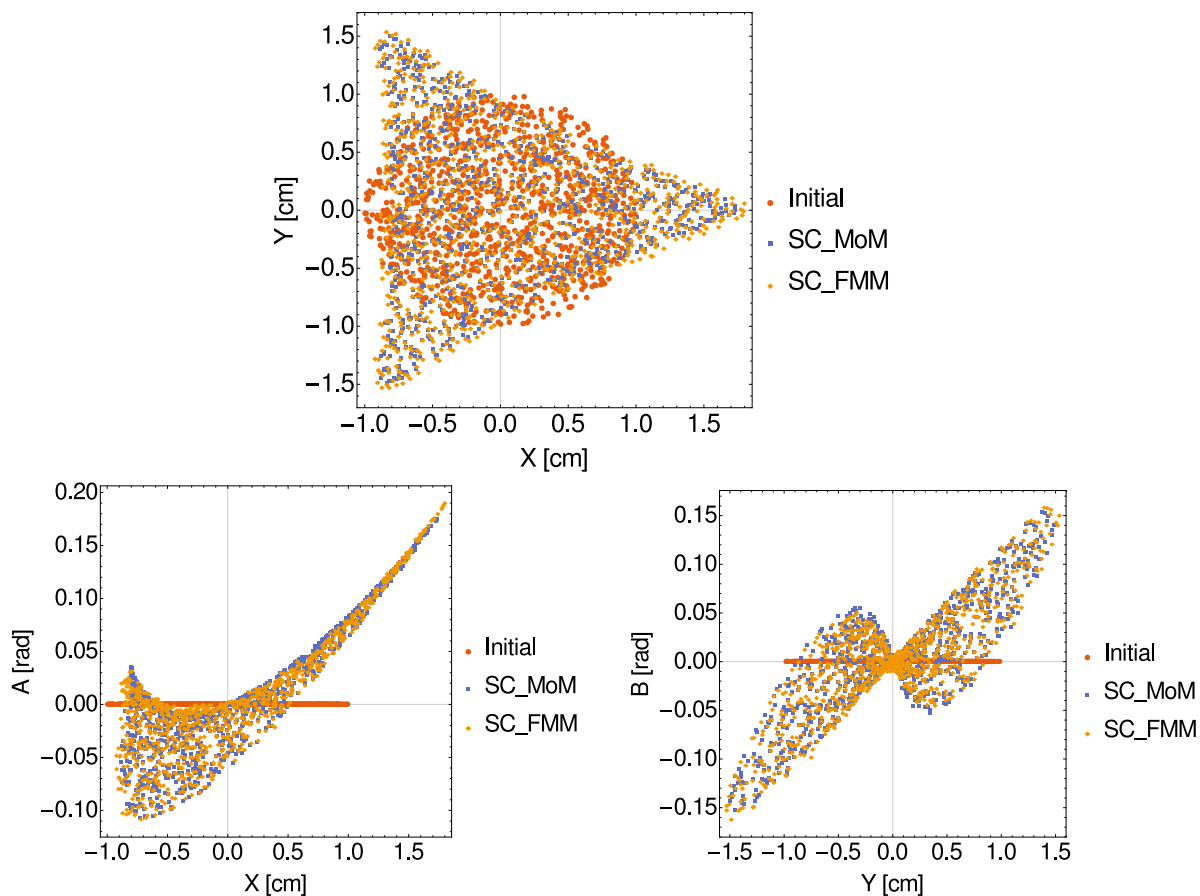


Figure 8.26: Comparison of the space charge kick from method of moments and the fast multipole method. Positions are given in centimeters and a, b in radians. Comparing to COSY, the final distribution retains its shape. The beam envelope is larger due to the space charge. The FMM is also slightly stronger than the MoM.

The results from Figure 8.26 are as expected. The beam-wall interaction adds some new behavior. The final distribution with space charge and the beam-wall effects is shown in Figure 8.27.

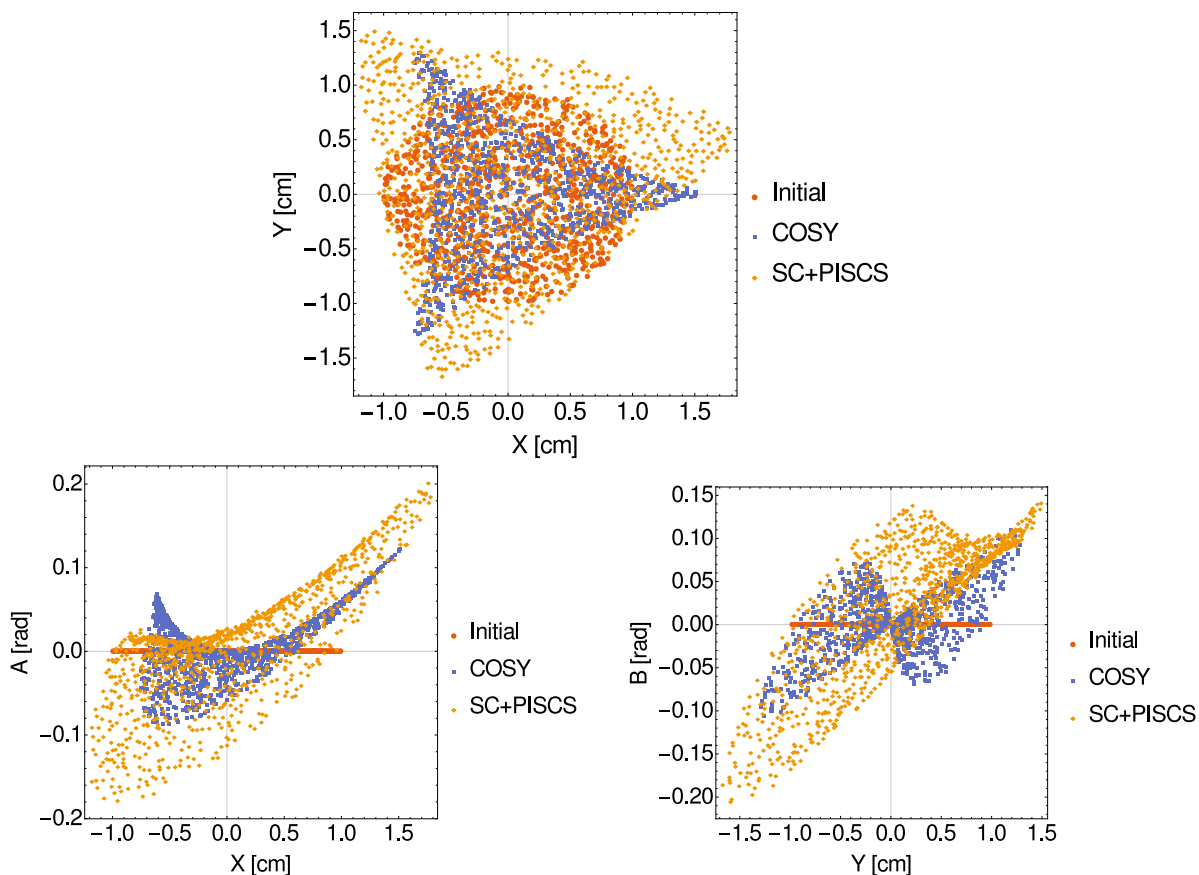


Figure 8.27: Distribution after sextupole with space charge and beam-wall. Positions are given in centimeters and a, b in radians. The shape is no longer retained due to the beam-wall interaction.

Even though the beam-wall interaction is amplified using macroparticles, we may imagine this situation when dealing with a high-intensity beam with extremely high bunch charge. We see that the shape is no longer retained. This suggests the sextupole field is modified by the beam-wall interaction. It is possible this causes longitudinal effects as well. Certainly, the beam envelope is larger than expected. This feedback can lead to significant beam loss over several turns.

We wish to show the beam-wall interaction without any amplification. Generally, the effect is cancelled by cylindrical symmetry in the beamline design. To isolate the beam-wall

interaction, we use a parallel beam ($N = 20000$) centered close to the top pole $X = 0, Y = 2.5$ cm as shown in Figure 8.28.

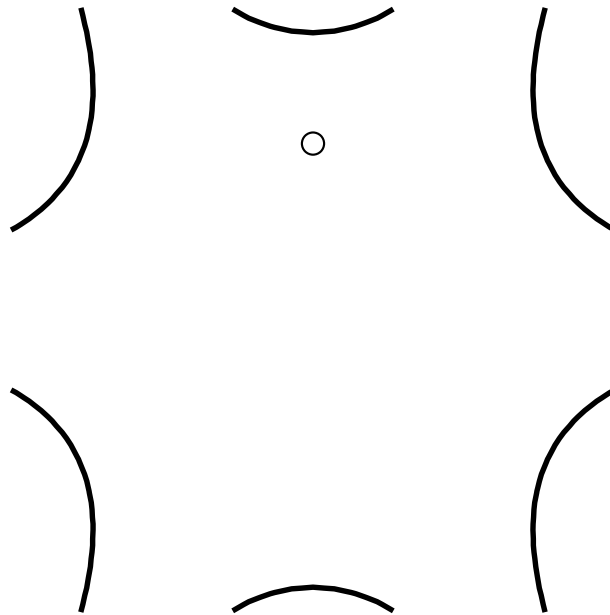


Figure 8.28: Initial position of the beam relative to the sextupole. The beam has initially zero emittance, is uniformly distributed, and centered on the Y -axis.

The beam is uniformly distributed with $R = 0.05$ cm around $X = 0, Y = 1.5$ cm. We remove the other effects by using 7 TeV protons. We use unit charges to match units with COSY and COMFY for this case. We do not use macroparticles. Figure 8.29 shows the beam phase space after the sextupole. XY is not shown since the difference is small for the short 10 cm kick.

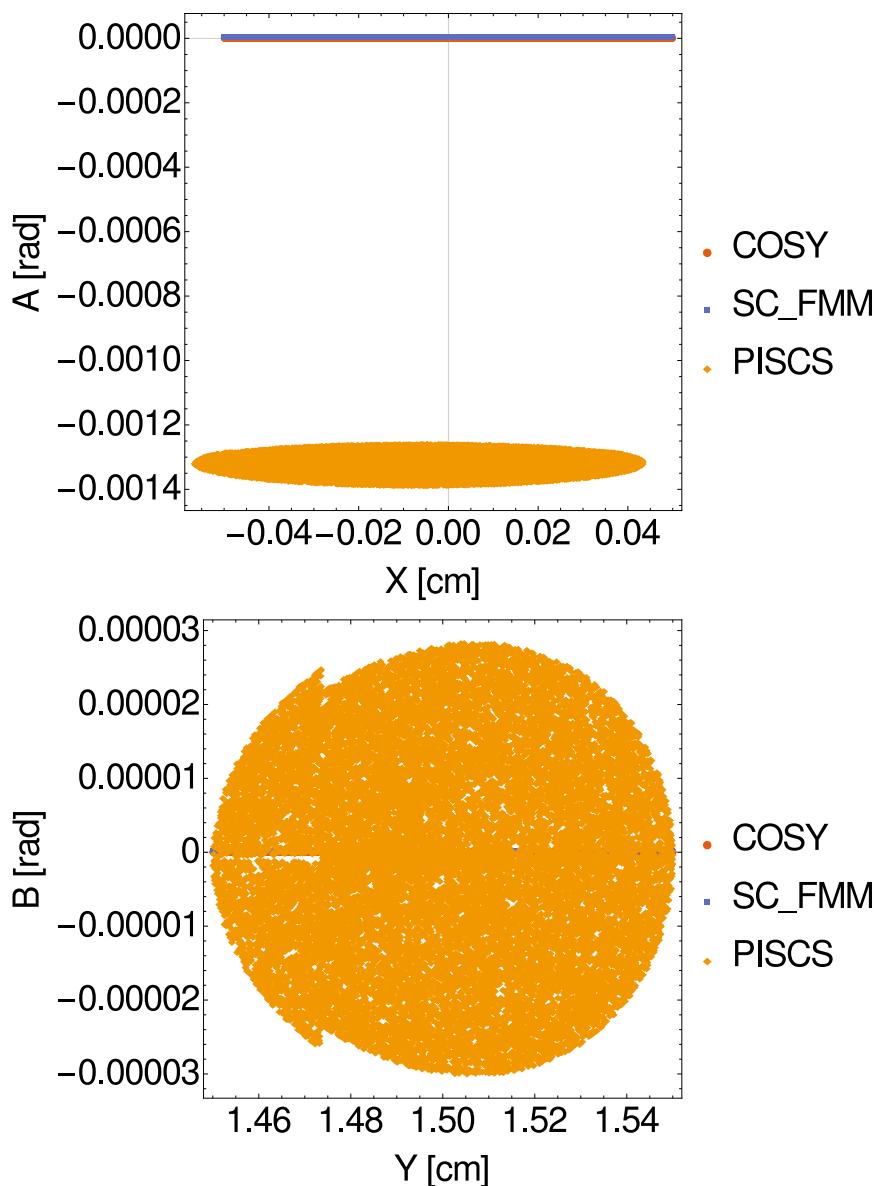


Figure 8.29: 7 TeV proton beam phase space after the sextupole. The initial beam center is at $X = 0, Y = 1.5$ cm near the top pole tip at $X = 0, Y = 2.5$ cm. Initial beam has zero emittance and radius of 0.05 cm. Final beam with only the sextupole (COSY) and with space charge (SC-FMM) are almost the same as initial.

The phase space with and without space charge show little change from the parallel beam due to the increased rigidity. The beam-wall effects acts as a sextupole with reversed poles due to the charges. Along the y -axis, the sextupole should only show a vertical field, which

means a horizontal force. For this case, the sextupole field is down, so the force is to the right. The positive charges cause a negative potential change along the pole surface. The negative potential combined with that geometry causes a field like a reversed sextupole. The positioning of the beam means the top pole displays the strongest effect. The change in phase space shows the force due to the beam-wall interaction is mainly towards the left. In the vertical phase space, b changes in a circular fashion with respect to y . These are likely the off-axis particles, where the sextupole causes some vertical force depending on sign of x . However, the vertical force is significantly smaller than in the horizontal, so the change is small. There is a small split in b near the bottom of the beam, $y = 1.46 - 1.47$ cm. The difference from the circle is small enough that it could be numerical error from the calculation order. Otherwise, the split suggests slightly stronger force for the bottom of the beam. The actual change to the average momenta and emittance is given in Table 8.3.

Table 8.3: Average Momenta and Final Emittance after Sextupole.

	Momenta (a, b) [rad]	Emittance $(\varepsilon_x, \varepsilon_y)$ [m·rad]
COSY	$(9.64 \times 10^{-9}, -2.50 \times 10^{-12})$	$(8.04 \times 10^{-14}, 8.04 \times 10^{-14})$
SC-FMM	$(9.64 \times 10^{-9}, -2.50 \times 10^{-12})$	$(8.04 \times 10^{-14}, 8.04 \times 10^{-14})$
PISCS	$(-1.32 \times 10^{-3}, -8.29 \times 10^{-7})$	$(7.91 \times 10^{-9}, 3.60 \times 10^{-9})$

The beamline can be designed to correct for such aberrations. We could track the particles to understand the necessary adjustments, but it is much more efficient to study the transfer map. We show this using a periodic FODO cell, the most common transport system. The FODO cell uses a similar setup as in Ref. [11]. For this study, we set charge in units of e as used in the COSY beam physics package. The parameters are given in Table 8.5. The beam parameters are given in Table 8.4. We use a simple cylinder with $R = 0.0508$ m, $L = 0.167$ m and Neumann BC for the quadrupole. We choose the cylinder for convenience rather than any constraints. The total cell length is 2.084 m.

Table 8.4: Beam Parameters

Energy [MeV]	50
Species	Electrons
Charge [e]	-1
No. of particles	20000
Emittance [m·rad]	9.96×10^{-9}
(α_x, α_y)	0, 0
(β_x, β_y) [m]	(20.07, 10.00)

Table 8.5: FODO Parameters

Quad field [T]	0.0101
Quad aperture [m]	0.0508
Quad length [m]	0.167
Drift [m]	0.875
Calculation order	8
Structure	Cylinder
Element order	5
No. of elements	240

The beam and FODO parameters are selected to accommodate a Gaussian beam with Twiss parameters, $\alpha_{x,y} = 0$, $\beta_{\max} = 20$ m, $\beta_{\min} = 10$ m, $\epsilon_{x,y} = 10^{-8}$ m·rad. As Figure 8.30 shows, our beam without collective effects returns to the same envelope. The Twiss parameters are slightly changed with $\beta_x = 18.73$, $\beta_y = 10.09$ [m].

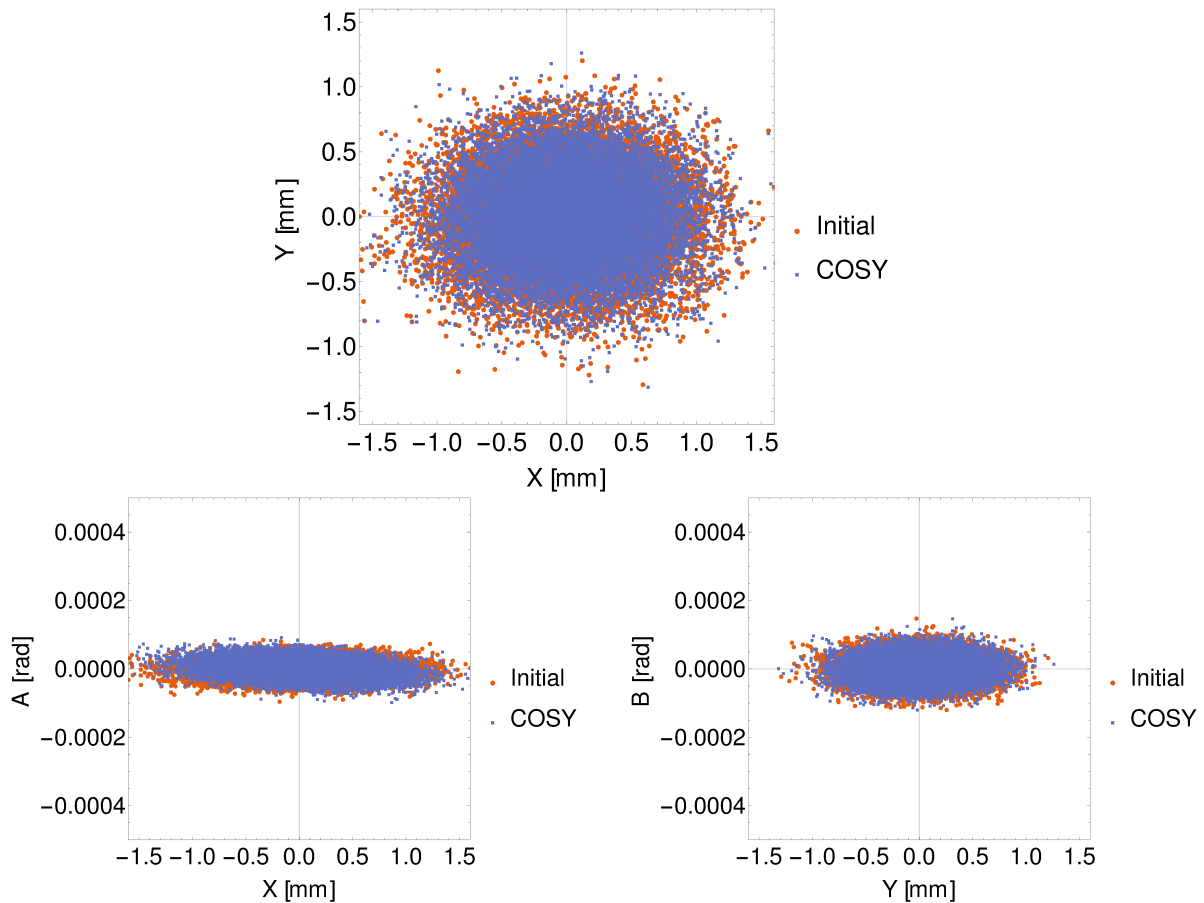


Figure 8.30: Initial and final beam with $N = 20000$ after the periodic FODO cell using calculation order 8. No collective effects are included. Position is given in millimeters and a, b in radians.

We wish to study the stability with respect to space charge and beam-wall interaction. We know there is a point where the collective effects are strong enough to destabilize the beam. We increase the current slowly and found the max stable current around 0.3 A. Figure 8.31 shows the beam with the collective effects at $I = 0.3$ A. Clearly, the beam's Twiss parameters are still similar in all cases, perhaps with a slight correlation in YB .

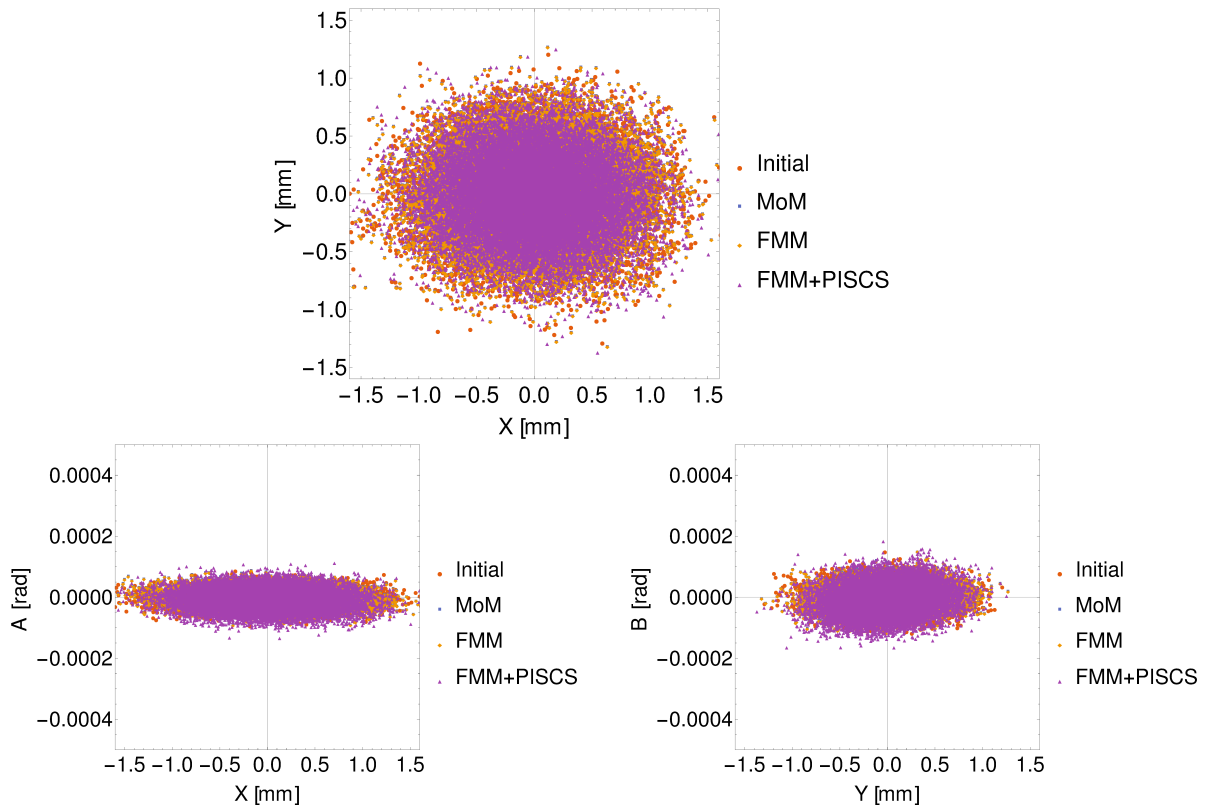


Figure 8.31: Electron beam with $N = 20000$ including collective effects at 0.3 A. MoM and FMM only include space charge and FMM+PISCS indicates space charge plus beam-wall. This is at the maximum current before losing linear stability. Position is given in millimeters and a, b in radians.

We show space charge computed with the MoM and FMM. FMM+PISCS indicates space charge plus beam-wall. While there are slight differences, the beam envelope appears to be the same. We continue increasing the current and see the beam growing divergent in Figure 8.32. The slight differences between space charge and PISCS suggest space charge is the main destabilizing force.

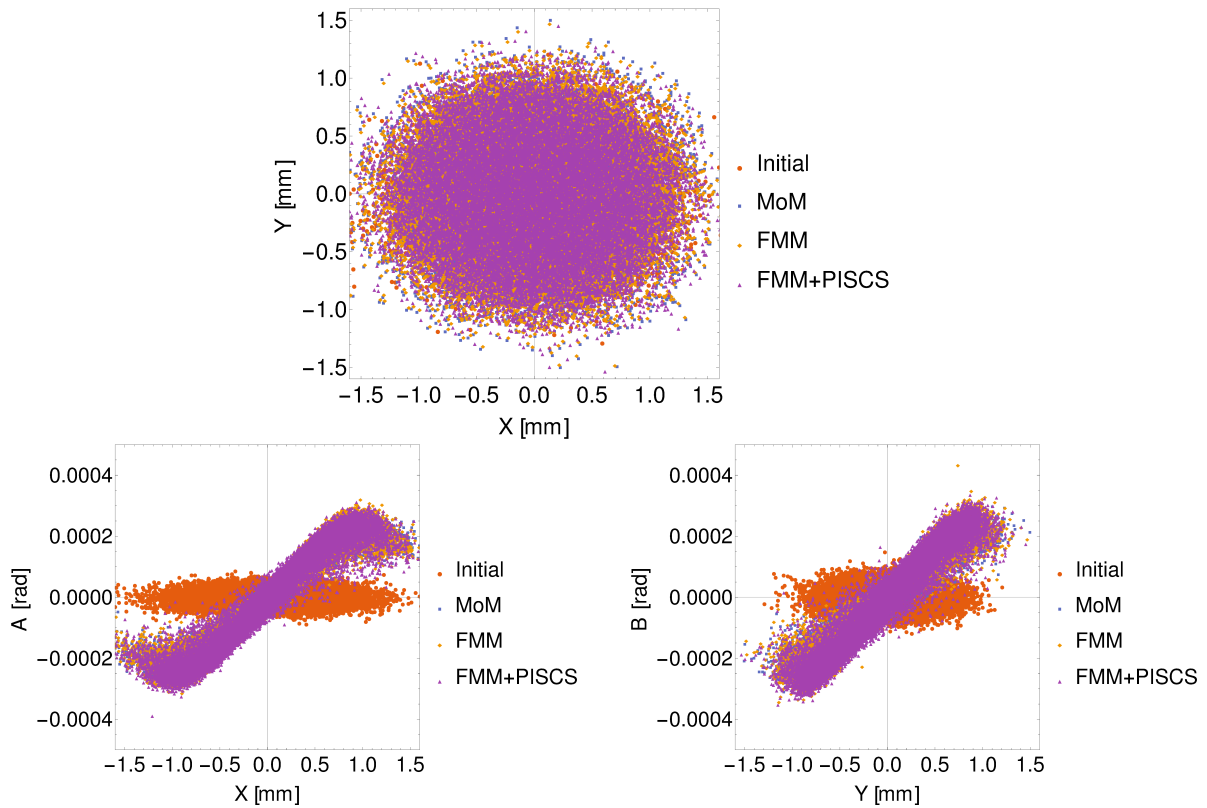


Figure 8.32: Beam with $N = 20000$ at $I = 10$ A with collective effects. The beam is clearly divergent and unstable in both planes. The slight difference suggests space charge is the main destabilizing cause. Position is given in millimeters and a, b in radians.

With 50 MeV, the beam is relativistic, yet space charge is still strong enough to destabilize with < 0.5 A current in the 2D approximation. The beam had no loss up to 20 A for the given aperture radius in Table 8.5. Based on the final distribution, we compute the statistical emittances versus current, shown in Figure 8.33.

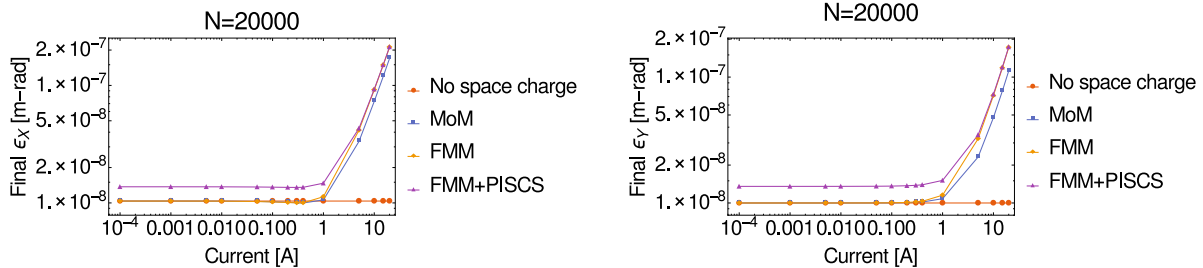


Figure 8.33: Final emittances [m-rad] ϵ_x (left) and ϵ_y (right) versus current. The beam-wall interaction adds a slight amount, but in all cases, the emittance grows up to an order of magnitude once in the unstable region due to collective effects.

The beam-wall interaction increases the final emittance slightly in the stable region. We see the emittances increase an order of magnitude once unstable. Space charge seems to dominate since PISCS does not depend on the current.

The previous statistics come from tracking the beam. By themselves, it is unclear how to increase the stability. We next analyze the transfer maps to better understand the collective effects. We focus on a few aspects: Stability, aberrations, and resonances. First, let's consider how well is the Hamiltonian structure preserved. Directly, we can consider how symplectic is the system. The symplectic condition is [30, 43],

$$\text{Jac}(\mathcal{M})^T \cdot \hat{J} \cdot \text{Jac}(\mathcal{M}) - \hat{J} = 0.$$

where $\text{Jac}(\mathcal{M})$ is the Jacobian of the transfer map and \hat{J} is the antisymmetric tensor. Simply put, if the map is symplectic, our motion must be Hamiltonian. Due to the approximate nature, the map accumulates numerical errors which affect the symplecticity. This is particularly true with the computation of the collective effects. The DA methods in COSY allow for map symplectification, which essentially minimizes the accumulated error. We use COSY's symplectification before computing the symplectic condition. For details on the symplectification, we refer to Refs. [179, 180]. Figure 8.34 shows the symplectic condition versus current after symplectification.

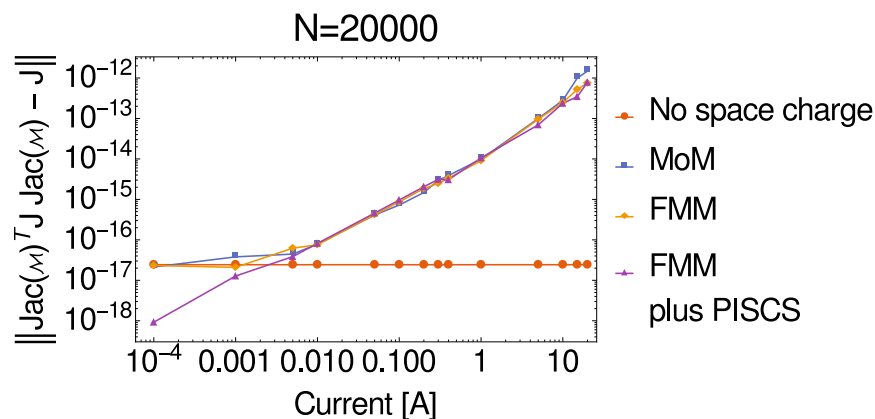


Figure 8.34: Symplecticity of the periodic FODO cell. The map is symplectic to machine precision without collective effects. With collective effects, the map shows increasing symplectic error with current. The error scale suggests remnant approximation error from the moment method. The beam-wall decreases the error for small current, but it is likely an artifact.

Without collective effects, the map is symplectic to machine precision. With symplectification, the collective effects do not quite match the condition to machine precision in Figure 8.34. The symplectic error increases up to 10^{-12} . However, the scale suggests remnant of the approximation error. The increase is likely because the increasing space charge requires higher order to maintain the accuracy and the symplectic error. The beam-wall interaction decreases the symplectic error at $I = 10^{-4}$ A. This is most likely an artifact. Overall, the map is symplectic.

We next consider the system stability. The condition for linear stability requires the trace of the transfer matrix to be between -2 and 2 [43]. If linearly uncoupled, Figure 8.35 shows the stability is maintained until 0.3 A, where space charge causes the trace to increase.

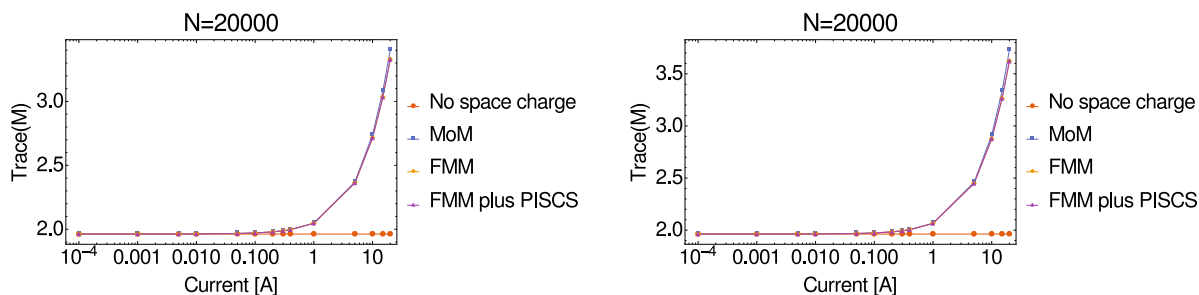


Figure 8.35: Trace of transfer matrix in the XA -plane (left) and YB -plane (right). The FODO starts within the stable region. We see the stability is lost due to mainly space charge.

We can look at what lead to the instability using the transfer map. We know the collective effects are the primary cause but not what exactly changed. One method is simply examining the behavior of the aberrations, or equivalently, the higher order map elements. For instance, spherical aberrations occur from $(x|x^3), (y|y^3)$ [$1/m^2$]. Figure 8.36 shows the spherical aberrations due to the collective effects in each plane. Without collective effects, some aberration is present, but the map element increases rapidly due to space charge. The beam-wall does not seem to affect this aberration. The FODO cell can tolerate this aberration within the stable region.

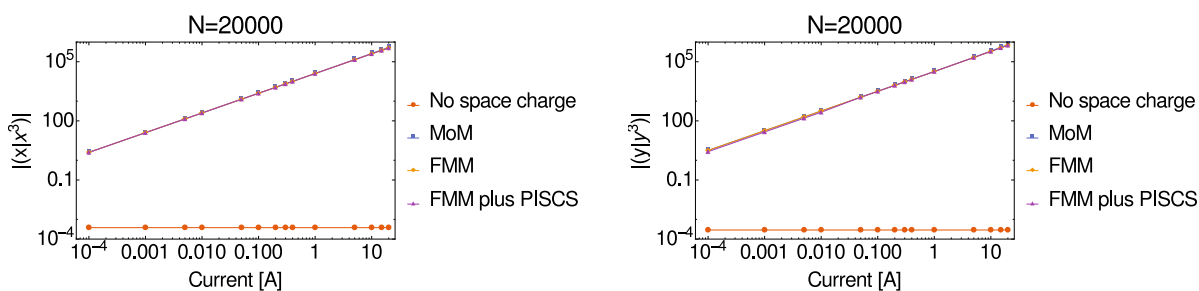


Figure 8.36: Spherical aberration versus current [A] from $(x|x^3), (y|y^3)$ [$1/m^2$] in the transfer map. Contribution is mostly from space charge.

In normal form coordinates, the stable motion are simply rotations with phase advance equal to the tune. The space charge shifts the tune, changing this stable motion. We know the tune shift eventually destabilizes the motion because it reaches a resonance at an integer tune.

We wish to understand the tune depression in the stable region as well as the dominant resonances. We use the DA normal form algorithm [43] in COSY to transform the transfer map and compute the tunes and resonances. Figure 8.37 shows the horizontal and vertical tune shift starts around 0.005 A. The maximum shift is ~ 0.02 in x and ~ 0.018 in y before losing stability, where the tune becomes complex. In this case, the vertical tune becomes complex first around 0.4 A.

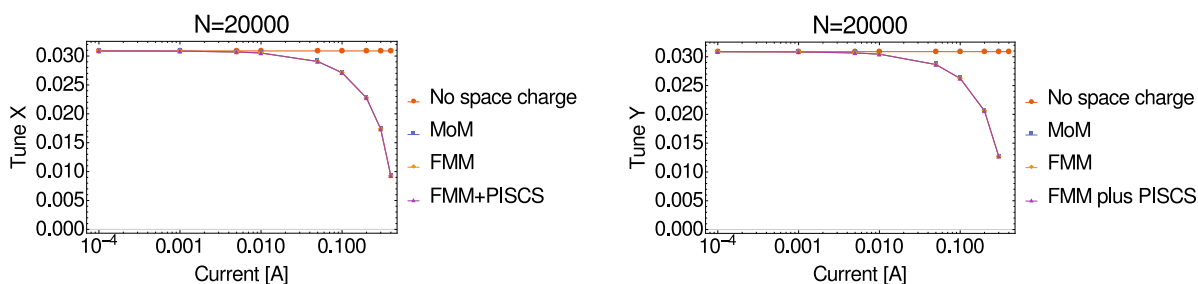


Figure 8.37: Fractional tune shift versus current [A] in the linearly stable region. The space charge begin depressing the tune around $I = 0.005$ A, reaching a maximum shift of 0.02 in x and 0.018 in y before stability is lost.

Due to the short cell length, there is no integer part of the tune, so Figure 8.37 essentially shows the only shift. Clearly the tune is approaching a resonance due to space charge.

We next look at the resonances. The resonances are particularly important for beam lifetime in repetitive systems. The resonances indicate the number of periods for stable motion to return to the same point in phase space. A destabilizing factor at this point adds up. Lower order means fewer periods are required for the destabilizing factor to become significant. The lower order resonances reduce the beam lifetime more than higher order resonances. If the resonance is strong enough, any small kick eventually causes large divergence for a particle. This can cause effects such as halo formation [30]. Without collective effects, we only saw weak 4th, 6th, and 8th order resonances in Figures 8.38, 8.39 and 8.40. The dominant strengths are approximately 0.024, 0.005, and 0.002 respectively.

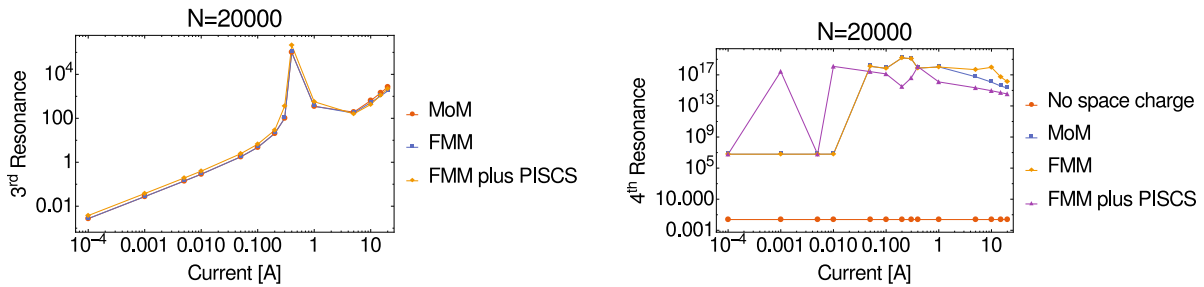


Figure 8.38: Dominant 3rd (left) and 4th (right) order resonances versus current [A] in the FODO cell. 3rd order shows the space charge and SC plus beam-wall behavior is comparable. 4th order shows a weak resonance without collective effects and large jump due to collective effects.

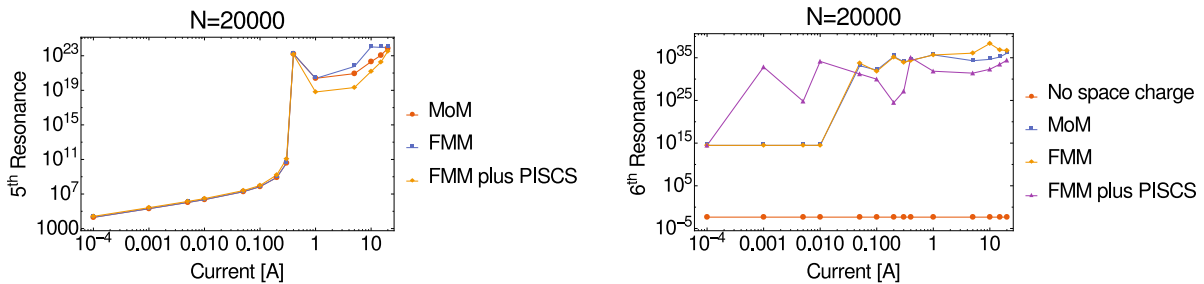


Figure 8.39: Dominant 5th (left) and 6th (right) order resonances versus current [A] in the FODO cell. Similarly, there is a large jump for the 6th order resonance due to collective effects. The 5th order resonance is comparable between collective effects until $\sim 0.5A$.

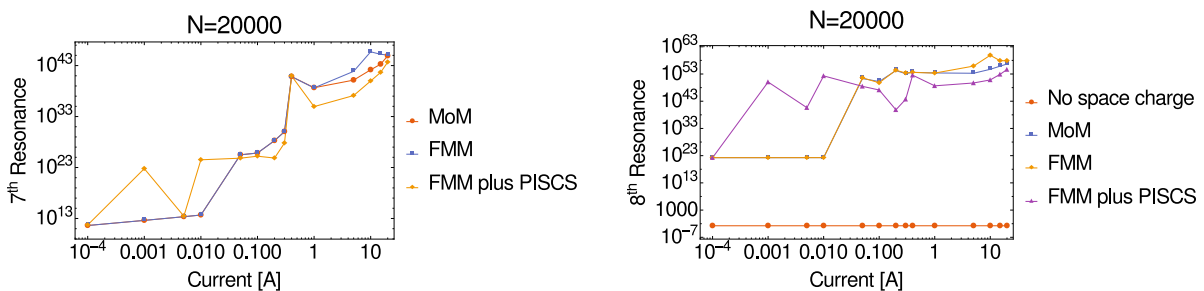


Figure 8.40: Dominant 7th (left) and 8th (right) order resonances versus current [A] in the FODO cell. The dominant 7th order resonance varies significantly with the beam-wall interaction.

Including the collective effects, the FODO gains 3rd, 5th, and 7th order resonances shown in Figures 8.38, 8.39 and 8.40. The even order resonances also became several orders of

magnitude stronger. We can see a significant effect due to the beam-wall interaction for the dominant 4th, 6th, 7th, and 8th order resonances. The large spike for the 3rd order resonance around 0.5 A is particularly interesting as it occurs near where linear stability is lost. The spike suggests the resonance is excited by the collective effects. The space charge effect from the MoM and FMM are comparable in almost all cases except when $I > 1$ A. The dominant resonance strengths at all orders appear to flatten out somewhat once the current is high enough. The beam is no longer stable, so the FODO would not be used for these currents. Considering the spike in the 3rd order resonance, this suggests the stability is lost due to the excited resonance. The beam-wall interaction changes the motion enough to affect the space charge map. This leads to the erratic behavior in the resonances. It is possible to continue analyzing the transfer map and select improved FODO parameters, but the quadrupole is mostly limited to modifying the first order map elements. To compensate for collective effects and improve the beamline design, we must consider transport elements affecting the nonlinear terms, which is what leads to integrable optics and IOTA. We have shown with this work that PISCS can be combined with tracking codes for beamline analysis including the beam-wall interaction with arbitrary structures. We have also demonstrated some aspects of transfer map analysis. The beam-wall interaction is generally weaker than space charge for actual beamlines but must be considered for high-intensity beams.

CHAPTER 9

SUMMARY AND CONCLUSIONS

Beam physics plays an important role in science and society. Several current and future applications require high-intensity beams. UMER and IOTA are two current machines designed to study high-intensity beams to minimize the loss using integrable systems and nonlinear elements. The collective effects due to the charged-particle beams limited the stability and beam lifetime of many past and current rings. Several codes exist that accurately simulate and track the beam dynamics including space charge. However, as of this work, there is no method that accurately models the space charge and beam-wall interaction inside arbitrary 3-D structures for use in studying intense beam dynamics. This work summarizes our research and development of the Poisson Integral Solver with Curved Structures or PISCS.

To facilitate the future beamline designs, we implement the fast multipole accelerated boundary element method for beam physics studies and create PISCS. PISCS solves the Poisson equation with boundary conditions in the presence of millions of charged particles. In creating PISCS, we develop and implement a high-order interpolation and quadrature for surface integrals over curved elements. For the first time, we combine the BEM and FMM in the DA framework. We show PISCS can handle on the order of a hundred thousand elements. With a high memory system, PISCS may handle a million elements or more with intense beams. We show PISCS can handle on the order of a million particles. We have parallelized and successfully tested PISCS with up to 32 MPI processes using a high-performance computing cluster. We have verified PISCS using various geometries and given a measure of its performance. We demonstrate the practicality of high-order meshes of

CAD structures. We significantly improve the convergence of our results using high-order elements. We have demonstrated PISCS with several arbitrary structures that cannot be generally simplified. We accurately model an IOTA sextupole using its CAD file. We depict the beam feedback within an elliptical vacuum pipe without symmetry requirements. We show a closeup view of the nanoemitter and form some qualitative conclusions. We have demonstrated PISCS may be used for a wide range of structure sizes and geometries with ease.

PISCS can be combined with tracking codes for in-depth analysis of the beam dynamics including collective effects. We show the beam-wall feedback affects the beam dynamics despite extreme beam rigidity. For the first time, we combine the Taylor transfer map with space charge and the beam-wall interaction using an arbitrary structure. The transfer map including space charge and the beam-wall interaction is a necessary tool for high-intensity beams. We demonstrate the efficiency of the normal form analysis of the collective transfer map, extracting several important quantities for system analysis. We show the beam-wall interaction affects the beam behavior at various levels. We successfully accomplish the implementation of a high-order, fast Poisson solver including boundary conditions inside arbitrary structures. We combine our solver with a beam dynamics code in the DA framework and, for the first time, accurately and efficiently model the beam dynamics in a general environment.

REFERENCES

- [1] Jie Wei. The very high intensity future. In *Proc. IPAC2014*, pages 17–22, Dresden, Germany, 2014.
- [2] Robert J. Wilson. Deep underground neutrino experiment. In *Particle Physics at the Year of Light*, pages 19–26. World Scientific, 2017.
- [3] S. Antipov, D. Broemmelsiek, D. Bruhwiler, D. Edstrom, E. Harms, V. Lebedev, J. Leibfritz, S. Nagaitsev, C. S. Park, H. Piekarz, P. Piot, E. Prebys, A. Romanov, J. Ruan, T. Sen, G. Stancari, C. Thangaraj, R. Thurman-Keup, A. Valishev, and V. Shiltsev. IOTA (Integrable Optics Test Accelerator): Facility and experimental beam physics program. *Journal of Instrumentation*, 12(03):T03002, 2017.
- [4] R.A. Kishek, B.L. Beaudoin, S. Bernal, M. Cornacchia, D. Feldman, R. Fiorito, I. Haber, T. Koeth, Y.C. Mo, P.G. O’Shea, K. Poor Rezaei, D. Sutter, and H.D. Zhang. The University of Maryland electron ring program. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 733(Supplement C):233–237, 2014. 19th International Symposium on Heavy Ion Inertial Fusion.
- [5] S. Bernal, B. Beaudoin, I. Haber, T. Koeth, Y. Mo, E. Montgomery, K. P. Rezaei, K. Ruisard, W. Stem, D. Sutter, H. Zhang, and R. A. Kishek. Nonlinear dynamics with space-charge in a small electron recirculator. *AIP Conference Proceedings*, 1777(1):100003, 2016.

- [6] Helmut Wiedemann. *Particle Accelerator Physics*. Springer Berlin Heidelberg, 3 edition, 2007.
- [7] Pulukkuttige D. Sumana Abeyratne. *New computational approaches to the N-body problem with applications to electron cooling of heavy ion beams*. Ph.D. dissertation, Northern Illinois University, 2016.
- [8] Carlos Alberto Brebbia and J. Dominguez. *Boundary Elements: An Introductory Course*. WIT Press / Computational Mechanics, 1998.
- [9] Anthony Gee and Bela Erdelyi. Implementing the fast multipole boundary element method with high-order elements. In *Proc. of North American Particle Accelerator Conference (NAPAC'16), Chicago, IL, USA, October 9-14, 2016*, number 3 in North American Particle Accelerator Conference, pages 518–520, Geneva, Switzerland, January 2017. JACoW.
- [10] Abtin Rahimian, Alex Barnett, and Denis Zorin. Ubiquitous evaluation of layer potentials using quadrature by kernel-independent expansion. *eprint arXiv:1612.00977 [math.NA]*, 2016. arXiv:1612.00977.
- [11] Anthony Gee and Bela Erdelyi. Space charge map extraction and analysis in a differential algebraic framework. In *Proc. HB2014*, pages 103–107, East Lansing, MI, USA, 2014.
- [12] Division of Physics of Beams, APS. Accelerators and beams, tools of discovery and innovation. http://www.aps.org/units/dpb/upload/accel_beams_2013.pdf, 2013. 4th ed.

- [13] Anussara Prayongrat, Cai Xu, Heng Li, and Steven H. Lin. Clinical outcomes of intensity-modulated proton therapy and concurrent chemotherapy in esophageal carcinoma: a single institutional experience. *Advances in Radiation Oncology*, 2017.
- [14] Suresh D. Pillai. Introduction to electron-beam food irradiation. *Chemical Engineering Progress*, 112(11):36–44, November 2016.
- [15] Robert W. Hamm and Marianne E. Hamm. The beam business: Accelerators in industry. *Physics Today*, 64(6):46, 2011.
- [16] World Nuclear Association. Accelerator-driven nuclear energy. <http://www.world-nuclear.org/information-library/current-and-future-generation/accelerator-driven-nuclear-energy.aspx>, 2017. Accessed: 2017-10-11.
- [17] Robert W. Garnett and Richard L. Sheffield. Overview of accelerator applications in energy. *Reviews of Accelerator Science and Technology*, 08:1–25, 2015.
- [18] Arlyn J. Antolak. Overview of accelerator applications for security and defense. *Reviews of Accelerator Science and Technology*, 08:27–36, 2015.
- [19] Chuanxiang Tang. Low energy accelerators for cargo inspection. *Reviews of Accelerator Science and Technology*, 08:143–163, 2015.
- [20] Andrew Sessler and Edmund Wilson. *Engines of Discovery: A Century of Particle Accelerators*. World Scientific Publishing Company Pte Limited, 2014.
- [21] Fermilab and SLAC. Symmetry. <https://www.symmetrymagazine.org/>. Accessed: Oct 10, 2017.
- [22] TIARA. Accelerators for society. <http://www.accelerators-for-society.org>. Accessed: Oct 10, 2017.

- [23] C.-M. Charlie Ma and Tony Lomax. *Proton and Carbon Ion Therapy*. Taylor & Francis Inc, Boca Raton, Florida, USA, 2012.
- [24] Sophia Chen. Mini particle accelerators make cancer treatment safer for everyone. <https://www.wired.com/story/mini-particle-accelerators-make-cancer-treatment-safer-for-everyone/>, June 2017. Accessed: Oct 10, 2017.
- [25] Sueo Machi. Trends for electron beam accelerator applications in industry. *Reviews of Accelerator Science and Technology*, 04(01):1–10, 2011.
- [26] C. M. Bhat. Proton beam intensity upgrades for the neutrino program at Fermilab. *eprint arXiv:1705.01499*, pages 1–6, 2017.
- [27] A. Deshpande. Science and status of the electron ion collider. *International Journal of Modern Physics E*, 26(01n02):1740007, 2017.
- [28] Kenichi Imai. Intensity frontier of accelerators for nuclear physics. *Reviews of Accelerator Science and Technology*, 06:19–36, 2013.
- [29] Stuart D. Henderson. Spallation neutron sources and accelerator-driven systems. *Reviews of Accelerator Science and Technology*, 06:59–83, 2013.
- [30] Martin Reiser. *Theory and Design of Charged Particle Beams*. Wiley-VCH, 2nd edition, 2008.
- [31] Andrei Seryi. *Unifying Physics of Accelerators, Lasers and Plasma*. Apple Academic Press Inc., Boca Raton, Florida, USA, 2015.
- [32] Ernest D Courant and Hartland S Snyder. Theory of the alternating-gradient synchrotron. *Annals of physics*, 3(1):1–48, 1958.

- [33] Hong Qin, Ronald C Davidson, Moses Chung, and Joshua W Burby. Generalized Courant-Snyder theory for charged-particle dynamics in general focusing lattices. *Physical review letters*, 111(10):104801, 2013.
- [34] P.a. P. Nghiem, N. Chauvin, M. Comunian, C. Oliver, W. Simeoni, D. Uriot, and M. Valette. Advanced concepts and methods for very high intensity accelerators. *Laser and Particle Beams*, 32(04):639–649, 2014.
- [35] John David Jackson. *Classical Electrodynamics*. John Wiley & Sons, 3rd edition, 1998.
- [36] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method Volume 1 : The Basis*, volume 1. Butterworth-Heinemann, fifth edition, 2000.
- [37] Andrew Ronald Mitchell and David Francis Griffiths. *The Finite Difference Method in Partial Differential Equations*. John Wiley, 1980.
- [38] David Tskhakaya. *The particle-in-cell method*, pages 161–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [39] Haruo Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5-7):262–268, 1990.
- [40] Paul J Channell and Clint Scovel. Symplectic integration of Hamiltonian systems. *Nonlinearity*, 3(2):231, 1990.
- [41] Rubén Flores Espinoza, Martín Gildardo García Alvarado, and Georgii Omel’yanov. *Differential Equations of Mathematical Physics: Theory and Numerical Simulations*. Universidad de Sonora, 2005.
- [42] Louise Olsen-Kettle. Numerical solution of partial differential equations. http://svmoore.pbworks.com/w/file/fetch/78638864/PDE_Numerical_Book_Olsen-Kettle.pdf, 2011. Lecture Notes.

- [43] Martin Berz. Modern map methods in particle beam physics. *Advances in Imaging and Electron Physics*, 108:1–318, 1999.
- [44] Hong Qin, Jian Liu, Jianyuan Xiao, Ruili Zhang, Yang He, Yulei Wang, Yajuan Sun, Joshua W. Burby, Leland Ellison, and Yao Zhou. Canonical symplectic particle-in-cell method for long-term large-scale simulations of the vlasov–maxwell equations. *Nuclear Fusion*, 56(1):014001, 2016.
- [45] Xiaobiao Huang, SY Lee, KY Ng, and Y Su. Emittance measurement and modeling for the fermilab booster. *Physical Review Special Topics-Accelerators and Beams*, 9(1):014202, 2006.
- [46] Edward Nissen. *Differential algebraic methods for space charge modeling and applications to the University of Maryland Electron Ring*. Ph.D. dissertation, Northern Illinois University, 2011.
- [47] Jean-Luc Vay and Rémi Lehe. Simulations for plasma and laser acceleration. *Reviews of Accelerator Science and Technology*, 09:165–186, 2016.
- [48] Alexander Wu Chao. *Physics of Collective Beam Instabilities in High Energy Accelerators*. John Wiley & Sons, New York, NY, USA, 1993.
- [49] L J Laslett. On intensity limitations imposed by transverse space-charge effects in circular particle accelerators. *eConf*, C630610:324, 1963.
- [50] King Yuen Ng. *Physics of Intensity Dependent Beam Instabilities*. FERMILAB-FN-0713. World Scientific, 2006.
- [51] Emmanuel Branlard. On the implementation of a 3D space charge algorithm to understand and further study the physics of linear accelerators. Technical Report August, Fermilab, 2009.

- [52] N. Chauvin. Space charge effects. <https://cas.web.cern.ch/cas/Slovakia-2012/Lectures/Chauvin.pdf>, 2006. Presentation.
- [53] Anthony Gee and Bela Erdelyi. A differential algebraic framework for the fast indirect boundary element method. In *7th International Particle Accelerator Conference (IPAC'16), Busan, Korea, May 8-13, 2016*, pages 3107–3109, Busan, Korea, 2016. JACOW, Geneva, Switzerland.
- [54] Steve Ritz, Hiroaki Aihara, Martin Breidenbach, Bob Cousins, Andre de Gouvea, Marcel Demarteau, Scott Dodelson, Jonathan L. Feng, Bonnie Fleming, Fabiola Gianotti, Francis Halzen, JoAnne Hewett, Wim Leemans, Joe Lykken, Dan McKinsey, Lia Meringa, Toshinori Mori, Tatsuya Nakada, Steve Peggs, Saul Perlmutter, Kevin Pitts, Kate Scholberg, Rick van Kooten, and Mark Wise. Building for discovery: Strategic plan for u.s. particle physics in the global context. *Report of the Particle Physics Project Prioritization Panel*, 2014. http://inspirehep.net/record/1299183/files/FINAL_P5_Report_053014.pdf.
- [55] Pushpalatha Bhat and Vladimir Shiltsev. Future HEP accelerators : the US perspective. *eprint arXiv:1511.00390*, pages 1–14, 2015.
- [56] C. R. Prior and Stfc Rutherford. Overview of high intensity accelerator projects. In *Proc. HB2010*, pages 6–10, Morschach, Switzerland, 2010.
- [57] P Adamson, I Anghel, C Backhouse, G Barr, M Bishai, A Blake, GJ Bock, D Bogert, SV Cao, CM Castromonte, et al. Measurement of neutrino and antineutrino oscillations using beam and atmospheric data in MINOS. *Physical Review Letters*, 110(25):251801, 2013.
- [58] M. C. Gonzalez-garcia, Michele Maltoni, and Thomas Schwetz. Global analyses of neutrino oscillation experiments. *Nuclear Physics B*, 908:199–217, 2016.

- [59] Serguei Chatrchyan, V Khachatryan, AM Sirunyan, A Tumasyan, W Adam, T Bergauer, M Dragicevic, J Erö, C Fabjan, M Friedl, et al. Combined results of searches for the standard model Higgs boson in pp collisions at. *Physics Letters B*, 710(1):26–48, 2012.
- [60] Serguei Chatrchyan, V Khachatryan, AM Sirunyan, A Tumasyan, Wolfgang Adam, T Bergauer, M Dragicevic, J Erö, C Fabjan, M Friedl, et al. Search for the standard model Higgs boson produced in association with a w or a z boson and decaying to bottom quarks. *Physical Review D*, 89(1):012003, 2014.
- [61] L.B.N.E. Collaboration. The Long-Baseline Neutrino Experiment: Exploring fundamental symmetries of the universe. *eprint arXiv:1307.7335*, page 264, 2013.
- [62] W. Pellico, K. Domann, F. G. Garcia, K. Gollwitzer, K. Seyia, and B. Zwaska. FNAL - The proton improvement plan (PIP). In *Proc. IPAC2014*, pages 3409–3411, Dresden, Germany, 2014.
- [63] M. Ball, A. Burov, B. Chase, A. Chakravarty, A. Chen, S. Dixon, J. Edelen, A. Grassellino, D. Johnson, S. Holmes, S. Kazakov, A. Klebaner, I. Kourbanis, A. Leveling, O. Melnychuk, D. Neuffer, T. Nicol, J. Ostiguy, R. Pasquinelli, D. Passarelli, L. Ristori, W. Pellico, J. Patrick, L. Prost, I. Rakhno, A. Saini, W. Schappert, A. Shemyakin, J. Steimel, V. Scarpine, A. Vivoli, and Others. *The PIP-II Conceptual Design Report*. OSTI, 2017.
- [64] Vladimir Shiltsev. Fermilab proton accelerator complex status and improvement plans. *Modern Physics Letters A*, 32(16):1730012, 2017.
- [65] Alex Chao. Beam dynamics with high intensity. <https://cas.web.cern.ch/cas/Intensity-Limitations-2015/Lectures/Tuesday3/ChaoBD1.pdf>, 2015. Lecture Notes.

- [66] Vladimir Shiltsev and Moses Chung. Space-charge compensation experiments at IOTA ring. *AIP Conference Proceedings*, 1777(1):1–5, 2016.
- [67] S. Nagaitsev, A. Valishev, V. V. Danilov, and D. N. Shatilov. Beam physics of integrable optics test accelerator at Fermilab. *FERMILAB-CONF-12-209-AD-APC*, pages 1371–1373, January 2013.
- [68] M. Reiser, P. G. O’Shea, R.a. Kishek, S. Bernal, P. Chin, S. Guharay, Y. Li, M. Venturini, J. G. Wang, V. Yun, W. Zhang, Y. Zou, M. Pruessner, T. Godlove, D. Kehne, P. Haldemann, R. York, D. Lawton, L. G. Vorobiev, I. Haber, and H. Nishimura. The Maryland electron ring for investigating space-charge dominated beams in a circular FODO system. *Proc. 1999 Part. Accel. Conf.*, 1:234–236, 1999.
- [69] S. Bernal, B. Beaudoin, Y. Cui, D. Feldman, R. Feldman, M. Glanzer, T. Godlove, I. Haber, J. Harris, M. Holloway, Y. Huo, R. A. Kishek, D. Lamb, W.-T. Lee, H. Li, B. Quinn, M. Reiser, A. Valfells, M. Walter, M. Wilson, R. Yun, Y. Zou, and P. G. O’Shea. Beam transport experiments over half-turn at the University of Maryland electron ring (UMER). *Proc. IEEE Part. Accel. Conf.*, 1:426–428, 2003.
- [70] S. Bernal, B. Beaudoin, H. Baumgartner, S. Ehrenstein, I. Haber, T. Koeth, E. Montgomery, K. Ruisard, D. Sutter, D. Yun, and R. A. Kishek. Ultra-low current beams in UMER to model space-charge effects in high-energy proton and ion machines. *AIP Conference Proceedings*, 1812(1):110006, 2017.
- [71] B. L. Qian, J. Zhou, and C. Chen. Image-charge effects on the envelope dynamics of an unbunched intense charged-particle beam. *Phys. Rev. Spec. Top. - Accel. Beams*, 6(1):47–53, 2003.
- [72] Stanley Humphries. *Charged Particle Beams*. Wiley, 1990.

- [73] Yijun Liu. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, Cambridge, New York, USA, 2009.
- [74] Stefan A. Sauter and Christoph Schwab. *Boundary Element Methods*, volume 39 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [75] L. Greengard, V. Rokhlin, L. Greengard Rokhlin, and V. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.
- [76] Leslie Greengard and Vladimir Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.
- [77] N. Chauvin. Space-charge effects. *eprint arXiv:1410.7991*, page 21, 2014.
- [78] Giovanni Lapenta. Particle in cell method: A brief description of the PIC method. <https://perswww.kuleuven.be/~u0052182/weather/pic.pdf>, 2009. Lecture Notes.
- [79] L. Gurel and O. Ergul. Comparisons of FMM implementations employing different formulations and iterative solvers. *IEEE Antennas Propag. Soc. Int. Symp. Dig. Held conjunction with Usn. North Am. Radio Sci. Meet. (Cat. No.03CH37450)*, 1:19–22, 2003.
- [80] Ramani Duraiswami. An introduction to fast multipole methods. *IEEE Trans. Mob. Comput.*, 2(4):275–287, 2003.
- [81] Liang Shen and Yijun J. Liu. An adaptive fast multipole boundary element method for three-dimensional potential problems. *Computational Mechanics*, 39(6):681–691, 2007.
- [82] Andrea Mostacci. *Beam-Wall interaction in the LHC liner*. Ph.D. dissertation, Università deli Studi di Roma, 2001.

- [83] Giovanni Rumolo. Beam instabilities. In *CAS - CERN Accelerator School: Advanced Accelerator Physics Course: Trondheim, Norway, August 18-29, 2013*, volume 009, pages 19–29, 2013.
- [84] James Bremer and Zydrunas Gimbutas. A Nyström method for weakly singular integral operators on surfaces. *Journal of Computational Physics*, 231(14):4885–4903, 2012.
- [85] Gerald Farin. *Curves and Surfaces for Computer-aided Geometric Design: A Practical Guide*. Elsevier, 2014.
- [86] Shengtun Hao, Alex H. Barnett, Per-Gunnar Martinsson, and Patrick Young. High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane. *Advances in Computational Mathematics*, 40(1):245–272, 2014.
- [87] Joseph Fels Ritt. Differential algebra. *Math. Gaz.*, 33:137, 1950.
- [88] Martin Berz. Differential algebraic description of beam dynamics to very high orders. *Part. Accel.*, 24(SSC-152):109–124, 1988.
- [89] MSU. COSY Infinity, 2017. http://bt.pa.msu.edu/index_cosy.htm.
- [90] W. T. Weng. Space charge effects - tune shifts and resonances. *AIP Conference Proceedings*, 153(1):348–389, 1987.
- [91] A. Burov. Head-tail modes for strong space charge. *Physical Review Special Topics - Accelerators and Beams*, 12(4), 2009.
- [92] Vladimir N. Litvinenko and Gang Wang. On compensating tune spread induced by space charge in bunched beams. *Physical Review Special Topics - Accelerators and Beams*, 17(11):114401–114441, 2011.

- [93] J. Kurzak and B. M. Pettitt. Fast multipole methods for particle dynamics. *Molecular Simulation*, 32(10-11):775–790, 2006.
- [94] J. Carrier, Leslie Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comput.*, 9(4):669–686, 1988.
- [95] Sivaram Ambikasaran and Eric Darve. The inverse fast multipole method. *eprint arXiv:1407.1572*, pages 1–25, 2014.
- [96] Rick Beatson and Leslie Greengard. A short course on fast multipole methods. *New York*, pages 1–37, 1997.
- [97] Barry Cipra. The best of the 20th century: Editors name top 10 algorithms. *SIAM News*, 33(4):2, May 2000.
- [98] He Zhang and Martin Berz. The fast multipole method in the differential algebra framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 645(1):338–344, 2011. The Eighth International Conference on Charged Particle Optics.
- [99] Sumana Abeyratne and Bela Erdelyi. N-body code to demonstrate electron cooling. In *Proc. of International Workshop on Beam Cooling and Related Topics (COOL’15)*, Newport News, VA, USA, September 28 - October 2, 2015, number 10 in International Workshop on Beam Cooling and Related Topics, pages 59–62, Geneva, Switzerland, November 2016. JACoW. doi:10.18429/JACoW-COOL2015-MOPF12.
- [100] Olivier Coulaud, Pierre Fortin, and Jean Roman. High performance BLAS formulation of the multipole-to-local operator in the fast multipole method. *Journal of Computational Physics*, 227(3):1836–1862, 2008.

- [101] Toru Takahashi, Cris Cecka, William Fong, and Eric Darve. Optimizing the multipole-to-local operator in the fast multipole method for graphical processing units. *International Journal for Numerical Methods in Engineering*, 89(1):105–133, 2012.
- [102] Sumana Abeyratne and Bela Erdelyi. Optimization of the multipole to local translation operator in the adaptive fast multipole method. In *Proceedings of PAC2013*, volume 13, pages 201–203, 2013.
- [103] Walter Dehnen. Towards optimal softening in 3D N-body codes. *Monthly Notices of the Royal Astronomical Society*, 324(February):20, 2001.
- [104] Joshua E. Barnes. Gravitational softening as a smoothing operation. *Monthly Notices of the Royal Astronomical Society*, 425(2):1104–1120, 2012.
- [105] Anthony Gee and Bela Erdelyi. Smooth fast multipole method for space charge tracking : An alternate to particle-in-cell. In *Proc. IPAC2015*, pages 663–665, Richmond, VA, USA, 2015. JaCoW.
- [106] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. STAT. Comput. Vo—*, 7(3):856–869, 1986.
- [107] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995.
- [108] Xiaobai Sun and N. P. Pitsianis. A matrix version of the fast multipole method. *Math. Appl. Rev. SIAM*, 43(2):289–300, 2001.
- [109] Rio Yokota, Huda Ibeid, and David Keyes. Fast multipole method as a matrix-free hierarchical low-rank approximation. *eprint arXiv:1602.02244*, pages 2–12, February 2016.

- [110] Valeria Simoncini and Daniel B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25(2):454–477, 2003.
- [111] Amina Bouras and Valerie Fraysse. Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy. *SIAM Journal on Scientific Computing*, 26(3):660–678, 2005.
- [112] Xiuhong Du and Daniel B. Szyld. Inexact GMRES for singular linear systems. *BIT Numer. Math.*, 48(3):511–531, 2007.
- [113] Simon K. Layton and Lorena A. Barba. Inexact Krylov iterations and relaxation strategies with fast-multipole boundary element method. *eprint arXiv:1506.05957*, 02215, 2015.
- [114] Bruno Carpentieri. Preconditioning for large-scale boundary integral equations in electromagnetics. *IEEE Antennas and Propagation Magazine*, 56(6):338–345, 2014.
- [115] Jurjen Duintjer Tebbens and Miroslav Tuma. Preconditioner updates for solving sequences of linear systems in matrix-free environment. *Numerical Linear Algebra with Applications*, 17:997–1019, 2010.
- [116] Jianlin Xia. Matrix-free sparse direct solvers. In *Proceedings of the Project Review, Geo-Mathematical Imaging Group*, volume 1, pages 171–183, West Lafayette, IN, 2013. Purdue University.
- [117] Pieter Coulier, Hadi Pouransari, and Eric Darve. The inverse fast multipole method: Using a fast approximate direct solver as a preconditioner for dense linear systems. *eprint arXiv:1508.01835*, pages 1–32, 2015.

- [118] Anastasis Polycarpou. *Introduction to the Finite Element Method in Electromagnetics*. Online access: IEEE (Institute of Electrical and Electronics Engineers) IEEE Morgan & Claypool Synthesis eBooks Library. Morgan & Claypool Publishers, 2006.
- [119] Peter Hunter and Andrew Pullan. *FEM/BEM Notes*. Department of Engineering Science, University of Auckland, New Zealand, 2001.
- [120] Stephen Kirkup and Javad Yazdani. A gentle introduction to the boundary element method in MATLAB/FREEMAT. In *Math. Methods, Comput. Tech. Non-Linear Syst. Intell. Syst.*, number 10, pages 46–52, 2008.
- [121] Nicola Giuliani. *Towards exascale BEM simulations: Hybrid parallelisation strategies for boundary element methods*. M.S. thesis, Scuola Internazionale Superiore di Studi Avanzati, 2014.
- [122] Jeng Tzong Chen, Yu Lung Chang, Shing Kai Kao, and Jie Jian. Revisit of the indirect boundary element method: Necessary and sufficient formulation. *Journal of Scientific Computing*, 65(2):467–485, 2015.
- [123] M. Tanaka, V. Sladek, and J. Sladek. Regularization techniques applied to boundary element methods. *Applied Mechanics Reviews*, 47:457–500, 1994.
- [124] Heinz Antes. A short course on boundary element methods. <https://cel.archives-ouvertes.fr/cel-01352830/document>, 2010. Lecture Notes.
- [125] Kendall E. Atkinson. A survey of boundary integral equation methods for the numerical solution of Laplace’s equation in three dimensions. In *Numer. Solut. Integr. Equations*, chapter 1, pages 1–34. University of NSW, 1988.
- [126] Dan Rosen and Donald E. Cormack. Singular and near singular integrals in the BEM: a global approach. *SIAM Journal on Applied Mathematics*, 53(2):340–357, 1993.

- [127] Stephen D. Gedney. On deriving a locally corrected Nyström scheme from a quadrature sampled moment method. *IEEE Trans. Antennas Propag.*, 51(9):2402–2412, 2003.
- [128] Ken Hayami. Variable transformations for nearly singular integrals in the boundary element method. *NII Tech. Reports*, 2005(10):1–21, 2005.
- [129] Yijun Liu and F. J. Rizzo. A weakly singular form of the hypersingular boundary integral equation applied to 3-D acoustic wave problems. *Computer Methods in Applied Mechanics and Engineering*, 96(2):271–287, 1992.
- [130] Kendall E. Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, 1997.
- [131] W. Dijkstra and R. M. M. Mattheij. The condition number of the BEM-matrix arising from Laplace’s equation. *Computing*, 4(2):67–81, 2006.
- [132] M. A. Golberg. Improved convergence rates for some discrete Galerkin methods. *Trans. Model. Simul.*, 10:91–98, 1970.
- [133] Kendall E. Atkinson. The numerical solution of boundary integral equations. *Inst. Math. ITS Appl. Conf. Ser.*, 63:223–260, 1997.
- [134] Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, 2nd edition, 1989.
- [135] A. Rathsfeld. Quadrature methods for 2D and 3D problems. *Journal of Computational and Applied Mathematics*, 125(1-2):439–460, 2000.
- [136] Concetta Laurita and Giuseppe Mastroianni. Condition numbers in numerical methods for Fredholm integral equations of the second kind. *Journal of Integral Equations and Applications*, 14(3):311–341, Fall 2002.

- [137] Numerical integration methods to evaluate double integrals using generalized Gaussian quadrature. http://shodhganga.inflibnet.ac.in/bitstream/10603/36937/10/10_chapter3.pdf, 2016.
- [138] E. M. Garau and R. Vázquez. Algorithms for the implementation of adaptive isogeometric methods using hierarchical splines. Technical report, IMATI-CNR, Pavia, 2016.
- [139] Michael S. Floater and Kai Hormann. Surface parameterization: A tutorial and survey. *Advances in multiresolution for geometric modelling*, pages 157–186, 2005.
- [140] J. Zechner, B. Marussig, G. Beer, and T. P. Fries. The isogeometric Nyström method. *eprint arXiv:1506.03914*, 2015.
- [141] Rafael Vázquez. A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0. *Computers & Mathematics with Applications*, 72(3):523–554, 2016.
- [142] G. Farin, J. Hoschek, and M.-S. Kim. *Handbook of Computer Aided Geometric Design*. Elsevier, 2002.
- [143] M.M.M. Sarcar, K. Mallikarjuna Rao, and K. Lalit Narayan. *Computer Aided Design and Manufacturing*. PHI Learning, 2008.
- [144] Neil Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, 1998.
- [145] Takashi Nagata. Smooth local interpolation of surfaces using normal vectors. *Journal of Applied Mathematics*, 2010(952420):24, 2010.
- [146] Takashi Nagata. Simple local interpolation of surfaces using normal vectors. *Computer Aided Geometric Design*, 22(4):327–347, 2005.

- [147] Weihua Geng. Parallel higher-order boundary integral electrostatics computation on molecular surfaces with curved triangulation. *Journal of Computational Physics*, 241(205):253–265, 2013.
- [148] R. J. Zauhar and R. S. Morgan. The rigorous computation of the molecular electric potential. *Journal of Computational Chemistry*, 9(2):171–187, 1988.
- [149] B. Jüttler and M. L. Sampoli. Hermite interpolation by piecewise polynomial surfaces with rational offsets. *Computer Aided Geometric Design*, 17(4):361–385, 2016.
- [150] Miroslav Lávička, Zbyněk Šír, and Jan Višek. Smooth surface interpolation using patches with rational offsets. *Computer Aided Geometric Design*, 48:75–85, 2016.
- [151] Helmut Pottmann. Rational curves and surfaces with rational offsets. *Computer Aided Geometric Design*, 12(2):175–192, 1995.
- [152] M. A. Taylor and B. A. Wingate. The Fekete collocation points for triangular spectral elements. *SIAM Journal on Numerical Analysis*, pages 1–23, 1999.
- [153] Mark a. Taylor, Beth a. Wingate, and Len P. Bos. A cardinal function algorithm for computing multivariate quadrature points. *SIAM Journal on Numerical Analysis*, 45(1):193–205, 2007.
- [154] Ling Guo, Akil Narayan, Liang Yan, and Tao Zhou. Weighted approximate Fekete points: Sampling for least-squares polynomial approximation. *eprint arXiv:1708.01296*, August 2017.
- [155] Mohammad Shafieipour, Jonatan Aronsson, Ian Jeffrey, Chen Nui, and Vladimir I. Okhmatovski. On new triangle quadrature rules for the locally corrected Nyström method formulated on NURBS-generated Bézier surfaces in 3-D. *IEEE Transactions On Antennas And Propagation*, 64(7):3027–3038, 2016.

- [156] P. Silvester. Symmetric quadrature formulae for simplexes. *Mathematics of Computation*, 24(109):95–95, 1970.
- [157] MG Blyth and C Pozrikidis. A Lobatto interpolation grid over the triangle. *IMA journal of applied mathematics*, 71(1):153–169, 2006.
- [158] John C Mason and David C Handscomb. *Chebyshev Polynomials*. CRC Press, 2002.
- [159] Christophe Geuzaine and Jean-Francois Remacle. *Gmsh manual*, 2017.
- [160] J.-F. Remacle, C. Geuzaine, G. Com Ere, and E. Marchandise. High quality surface remeshing using harmonic maps. *International Journal for Numerical Methods in Engineering*, 83(4):403–425, 2010.
- [161] Lawrence F. Canino, John J. Ottusch, Mark A. Stalzer, John L. Visher, and Stephen M. Wandzura. Numerical solution of the Helmholtz equation in 2D and 3D using a high-order Nyström discretization. *Journal of Computational Physics*, 146(2):627–663, 1998.
- [162] Andrew F. Peterson and Malcolm M. Bibby. An introduction to the locally-corrected Nyström method. *Synthesis Lectures on Computational Electromagnetics*, 4(1):1–115, 2009.
- [163] Andreas Klöckner, Alexander Barnett, Leslie Greengard, and Michael O’Neil. Quadrature by expansion: a new method for the evaluation of layer potentials. *Journal of Computational Physics*, 252(3):332–349, 2013.
- [164] Manas Rachh, Andreas Klöckner, and Michael O’Neil. Fast algorithms for quadrature by expansion I: Globally valid expansions. *eprint arXiv:1602.05301*, 2016.
- [165] Charles L Epstein, Leslie Greengard, and Andreas Klöckner. On the convergence of local expansion of layer potentials. *SIAM Journal on Numerical Analysis*, 51(5):2660–2679, 2013.

- [166] Ludvig af Klinteberg and Anna-Karin Tornberg. Error estimation for quadrature by expansion in layer potential evaluation. *Advances in Computational Mathematics*, 43(1):195–234, February 2016.
- [167] John Edward Campbell. On the theory of simultaneous partial differential equations. *Proceedings of the London Mathematical Society*, s1-31(1):235–263, 1899.
- [168] Henry Frederick Baker. On the integration of linear differential equations. *Proceedings of the London Mathematical Society*, s1-35(1):333–378, 1902.
- [169] István Faragó and Ágnes Havasi. On the convergence and local splitting error of different splitting schemes. *Progress in Computational Fluid Dynamics*, 5(8):495–504, 2005.
- [170] Bela Erdelyi, Ed Nissen, and Shashikant Manikonda. A differential algebraic method for the solution of the Poisson equation for charged particle beams. *Communications in Computational Physics*, 17(1):47–78, 2015.
- [171] Anthony Gee and Bela Erdelyi. Study and comparison of the method of moments and the single level fast multipole method for 2D space charge tracking. In *Proc. PAC2013*, pages 207–209, Pasadena, CA, USA, 2013. JaCoW.
- [172] Walton C. Gibson. *The Method of Moments in Electromagnetics*. Chapman & Hall/CRC, 2010.
- [173] Michael George Bulmer. *Principles of Statistics*. Courier Corporation, 1979.
- [174] Herman Schaumburg and Bela Erdelyi. A Picard iteration based integrator. In *Proceedings of NA-PAC*, volume 13, pages 210–212, 2013.

- [175] Sumana Abeyratne, Bela Erdelyi, and Herman Schaumburg. A Novel Code with High-order Adaptive Dynamics to Solve the N-body Problem. In *Proceedings, 54th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams (HB2014): East Lansing, MI, November 10-14, 2014*, page MOPAB21, 2015.
- [176] M. Ferrario, M. Migliorati, and L. Palumbo. Space charge effects. *eprint arXiv:1601.05214*, 2016.
- [177] G Xia, M Harvey, A J Murray, L Bellan, W Bertsche, R B Appleby, O Mete, and S Chattopadhyay. An ultracold low emittance electron source. *Journal of Instrumentation*, 9(06):P06011, 2014.
- [178] Hans-Werner Fink. Point source for ions and electrons. *Physica Scripta*, 38(2):260, 1988.
- [179] Béla Erdélyi and Martin Berz. Optimal symplectic approximation of hamiltonian flows. *Physical Review Letters*, 87(11):114302, 2001.
- [180] Béla Erdélyi and Martin Berz. Local theory and applications of extended generating functions. *International Journal of Pure and Applied Mathematics*, 11:241–282, 2004.
- [181] Per Christian Hansen. The truncated SVD as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, December 1987.

APPENDIX A

DERIVATIONS

A.1 Regularization of boundary integral equations

We consider the Laplace BVP below when $\mathbf{x} \rightarrow \Gamma$.

$$\begin{aligned}\nabla^2\psi(\mathbf{x}) &= 0, & \mathbf{x} \in \Omega, \\ \psi(\mathbf{x}) &= g(\mathbf{x}), & \mathbf{x} \in \Gamma.\end{aligned}$$

We represent the indirect forms of the solution using the Green's function (5.6).

$$\begin{aligned}\int_{\Gamma} G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y})d\Gamma(\mathbf{y}), & \quad \mathbf{x} \in \Omega, \mathbf{y} \in \Gamma, \\ \int_{\Gamma} \frac{\partial G}{\partial n_y}\eta(\mathbf{y})d\Gamma(\mathbf{y}), & \quad \mathbf{x} \in \Omega, \mathbf{y} \in \Gamma.\end{aligned}$$

Recall the spherical surface Γ_ε centered at fixed \mathbf{x} with some radius ε with $\mathbf{y} \in \Gamma \cap \Gamma_\varepsilon$ as shown in figure 5.1 [8]. We can define $\mathbf{n}(\mathbf{x})$ to choose one dimension, that is, $\mathbf{n}(\mathbf{x}) \cdot \mathbf{y} = y_i$, for $i = 1, 2, 3$. In addition, $\mathbf{n}(\mathbf{x}) \leq C\|\mathbf{x}\|$. When $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon$, there is some \mathbf{y} such that $\lim_{\varepsilon \rightarrow 0} \mathbf{x} = \mathbf{y}$. For chosen coefficients $C(\varepsilon)$,

$$\begin{aligned} |(\mathbf{x} - \mathbf{y}) \cdot \mathbf{n}(\mathbf{y})| &= |(\mathbf{x} - \mathbf{y}) \cdot \mathbf{n}(\mathbf{x})| + |(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{n}(\mathbf{y}) - \mathbf{n}(\mathbf{x}))|, \\ &= |y_i| + |(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{n}(\mathbf{y}) - \mathbf{n}(\mathbf{x}))|, \\ &\leq C_1\|\mathbf{x} - \mathbf{y}\|^2 + C_2|(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})|, \\ &\leq C\|\mathbf{x} - \mathbf{y}\|^2.\end{aligned}$$

Now, $G(\mathbf{x}, \mathbf{y})$ and its derivative are of the form, $\frac{1}{\|\mathbf{x}-\mathbf{y}\|^d}$, where $d = 1, 3$. Thus,

$$\begin{aligned} \left| \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) d\Gamma(\mathbf{y}) \right| &\leq |\sigma(\mathbf{y}_c)| \left| \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) \right|, \\ &\leq C \int_{\Gamma} \frac{1}{\|\mathbf{x}-\mathbf{y}\|} d\Gamma(\mathbf{y}), \\ &\leq C \frac{1}{\|\mathbf{x}-\mathbf{y}\|} \|\mathbf{x}-\mathbf{y}\|^2, \\ &\leq C \|\mathbf{x}-\mathbf{y}\|. \end{aligned}$$

$$\begin{aligned} \left| \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) d\Gamma(\mathbf{y}) \right| &\leq C \int_{\Gamma} \frac{(\mathbf{x}-\mathbf{y}) \cdot \mathbf{n}(\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|^3} d\Gamma(\mathbf{y}), \\ &\leq C \frac{1}{\|\mathbf{x}-\mathbf{y}\|} \|\mathbf{x}-\mathbf{y}\|^2, \\ &\leq C \|\mathbf{x}-\mathbf{y}\|. \end{aligned}$$

Gauss's law tells us the following with $\mathbf{y} \in \Gamma$,

$$\begin{aligned} \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) &= 1, & \mathbf{x} \in \Omega, \\ \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) &= 1 - \frac{\phi}{4\pi} (1 - \cos \theta), & \mathbf{x} \in \Gamma. \end{aligned}$$

When $\mathbf{x} \in \Omega$, we can see that,

$$\begin{aligned} \int_{\Omega} \delta(\mathbf{x}-\mathbf{y}) d\Omega(\mathbf{y}) &= \int_{\Omega} \nabla^2 G(\mathbf{x}, \mathbf{y}) d\Omega(\mathbf{y}), \\ &\Leftrightarrow 1 = \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}). \end{aligned}$$

When $\mathbf{x}, \mathbf{y} \in \Gamma$, we form the sphere $\Gamma_\varepsilon(\mathbf{x}, \varepsilon)$ for fixed \mathbf{x} . We can define $\mathbf{n}(\mathbf{y}) = \frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}-\mathbf{y}\|}$ for $\mathbf{y} \in \Gamma_\varepsilon$. The integral becomes

$$\begin{aligned} \int_{\Omega} \delta(\mathbf{x}, \mathbf{y}) \, d\Omega(\mathbf{y}) &= \int_{\Omega} \nabla^2 G(\mathbf{x}, \mathbf{y}) \, d\Omega(\mathbf{y}), \\ \Leftrightarrow 1 &= \int_{\Gamma - (\Gamma \cap \Gamma_\varepsilon)} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}) + \int_{\Gamma \cap \Gamma_\varepsilon} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}). \end{aligned}$$

Calculating the second integral, we get

$$\begin{aligned} &\int_{\Gamma \cap \Gamma_\varepsilon} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}), \\ &= \int_{\Gamma \cap \Gamma_\varepsilon} \frac{(\mathbf{x} - \mathbf{y}) \cdot \mathbf{n}(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3} \, d\Gamma(\mathbf{y}), \\ &= \int_{\Gamma \cap \Gamma_\varepsilon} \frac{1}{4\pi \|\mathbf{x} - \mathbf{y}\|^2} \, d\Gamma(\mathbf{y}), \\ &= \int_0^\phi \int_0^\theta \frac{1}{4\pi \varepsilon^2} \varepsilon^2 \sin \theta \, d\theta \, d\phi, \\ &= \frac{\phi}{4\pi} (1 - \cos \theta), \\ \Rightarrow 1 &= \int_{\Gamma - \Gamma \cap \Gamma_\varepsilon} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}) + \frac{\phi}{4\pi} (1 - \cos \theta). \end{aligned}$$

Thus as $\varepsilon \rightarrow 0$, we get as shown above. For smooth surfaces, we form a hemisphere and

$$\int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}) = \frac{1}{2}, \quad \mathbf{x} \in \Gamma.$$

Now we can evaluate the single and double layer on Γ . Let $\mathbf{x} \in \Omega$, $\mathbf{x}_0 \in \Gamma$ and $\mathbf{y} \in \Gamma$.

$$\begin{aligned}
\psi(\mathbf{x}) &= \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \eta(\mathbf{y}) \, d\Gamma(\mathbf{y}), \\
&= \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \eta(\mathbf{y}) \, d\Gamma(\mathbf{y}) + \eta(\mathbf{x}_0) \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}) - \eta(\mathbf{x}_0) \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}), \\
&= \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) [\eta(\mathbf{y}) - \eta(\mathbf{x}_0)] \, d\Gamma(\mathbf{y}) + \eta(\mathbf{x}_0) \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y}), \\
\Rightarrow \lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \psi(\mathbf{x}) - \psi(\mathbf{x}_0), \\
&= \int_{\Gamma} \left[\frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) - \frac{\partial G}{\partial n_y}(\mathbf{x}_0, \mathbf{y}) \right] [\eta(\mathbf{y}) - \eta(\mathbf{x}_0)] \, d\Gamma(\mathbf{y}) \\
&\quad + \eta(\mathbf{x}_0) \int_{\Gamma} \left[\frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) - \frac{\partial G}{\partial n_y}(\mathbf{x}_0, \mathbf{y}) \right] \, d\Gamma(\mathbf{y}).
\end{aligned}$$

Since $\int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \, d\Gamma(\mathbf{y})$ is finite, we know these terms are bounded. That is,

$$\begin{aligned}
&\left| \int_{\Gamma} \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) - \frac{\partial G}{\partial n_y}(\mathbf{x}_0, \mathbf{y}) \, d\Gamma(\mathbf{y}) \right| \leq C, \\
\Rightarrow &\left| \int_{\Gamma} \left[\frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) - \frac{\partial G}{\partial n_y}(\mathbf{x}_0, \mathbf{y}) \right] [\eta(\mathbf{y}) - \eta(\mathbf{x}_0)] \, d\Gamma(\mathbf{y}) \right| \leq C \\
&|\eta(\mathbf{y}_c) - \eta(\mathbf{x}_0)| \left| \int_{\Gamma} \left[\frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) - \frac{\partial G}{\partial n_y}(\mathbf{x}_0, \mathbf{y}) \right] \, d\Gamma(\mathbf{y}) \right| \leq C.
\end{aligned}$$

A.2 Smoothed FMM operators

Let $r^2 = x^2 + y^2 + z^2 + \lambda^2$, where λ is a smoothing parameter. We express ϕ as follows.

$$\begin{aligned}
\phi(\mathbf{x}) &= \sum_{i=1}^N \frac{1}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 + \lambda^2}}, \\
&= \frac{1}{r} \sum_{i=1}^N \left[1 + \frac{r_i^2}{r^2} - \frac{2x_i x}{r^2} - \frac{2y_i y}{r^2} - \frac{2z_i z}{r^2} \right]^{-\frac{1}{2}}.
\end{aligned}$$

For the multipole expansion with DA variables d_x, d_y, d_z, d_λ ,

$$\begin{aligned}
d_r &= \frac{1}{r}, & d_x &= \frac{x}{r^2}, \\
d_y &= \frac{y}{r^2}, & d_z &= \frac{z}{r^2}, \\
d_\lambda &= \frac{\lambda}{r^2}, \\
\Rightarrow \phi(d_x, d_y, d_z) &= d_r \sum_{i=1}^N [1 + r_i^2 d_r^2 - 2x_i d_x - 2y_i d_y - 2z_i d_z]^{-\frac{1}{2}}, \\
&= d_r \cdot \phi_M.
\end{aligned}$$

Let the new multipole center be \mathbf{x}'_0 . We translate a multipole expansion using \mathcal{M}_1 .

$$\begin{aligned}
d'_x &= \frac{x - x'_0}{r'^2} = \frac{x'}{r'^2} \Rightarrow d'_y = \frac{y'}{r'^2}, & d'_z &= \frac{z'}{r'^2}, & d'_\lambda &= \frac{\lambda}{r'^2}, \\
r' &= \sqrt{x'^2 + y'^2 + z'^2 + \lambda^2}, \\
R &\equiv (1 + r'^2 d_r'^2 + 2x'_0 d'_x + 2y'_0 d'_y + 2z'_0 d'_z)^{-1}, \\
d_r &= d'_r \cdot \sqrt{R}, \\
\mathcal{M}_1 &\equiv \begin{cases} d_x = (d'_x + x'_0 d_r'^2) \cdot R \\ d_y = (d'_y + y'_0 d_r'^2) \cdot R \\ d_z = (d'_z + z'_0 d_r'^2) \cdot R \\ d_\lambda = d'_\lambda \cdot R \end{cases}, \\
\Rightarrow \phi'_M &= \phi_M \circ \mathcal{M}_1, \\
\Rightarrow \phi' &= \phi'_M \cdot d'_r \cdot \sqrt{R}.
\end{aligned}$$

Let the new local center be \mathbf{x}'_0 . Note $d'_\lambda = \lambda$. We translate a multipole expansion to a local expansion with \mathcal{M}_2 .

$$\mathcal{M}_2 \equiv \begin{cases} d_x = (x'_0 + d'_x) \cdot R \\ d_y = (y'_0 + d'_y) \cdot R \\ d_z = (z'_0 + d'_z) \cdot R \\ d_\lambda = \lambda \cdot R \end{cases},$$

$$R \equiv ((x'_0 + d'_x)^2 + (y'_0 + d'_y)^2 + (z'_0 + d'_z)^2 + \lambda^2)^{-1/2},$$

$$\Rightarrow \phi_L = \phi_M \circ \mathcal{M}_2,$$

$$\Rightarrow \phi = \phi_L \cdot \sqrt{R}.$$

Since λ has already been evaluated, it is no longer translated. The L2L operator is unchanged.

APPENDIX B
NUMERICAL METHODS

B.1 Singular value decomposition

This section is based on [134, 181]. The singular value decomposition (SVD) is a matrix factorization related to eigendecomposition. Given the $m \times n$ matrix \mathbf{M} , there is a factorization such that,

$$\mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{V}^*,$$

where \mathbf{W} is a $m \times n$ diagonal matrix containing the singular values of \mathbf{M} . \mathbf{U} and \mathbf{V} are unitary matrices of dimensions $m \times m$ and $n \times n$ respectively. Many order the singular values in \mathbf{W} from largest to smallest by swapping rows or columns accordingly. The factorization implies the pseudoinverse, if it exists, may be constructed by,

$$\mathbf{M}^\dagger = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^*,$$

where \mathbf{W}^{-1} is the reciprocal of the diagonal. Since the condition number of a matrix is given by,

$$\begin{aligned} \kappa(\mathbf{M}) &= \|\mathbf{M}\| \|\mathbf{M}^\dagger\|, \\ &= \frac{\sigma_{\max}}{\sigma_{\min}}, \end{aligned}$$

where σ denotes the singular values, we may control the conditioning of \mathbf{M} by dropping the smallest singular values in \mathbf{W} and related basis vectors in \mathbf{U}, \mathbf{V} depending on some user criteria. Different algorithms exist to construct the SVD and may be found in a variety of libraries and languages (Ex. <https://www.gnu.org/software/gsl/>).

B.2 Runge-Kutta integration

This section is based on [134]. The so-called single step integrator solves ODEs of the form,

$$y' = f(x, y),$$

using some initial guess y_0 to iteratively reach a solution. Each iteration requires knowing a single past solution y_n to reach y_{n+1} . Runge-Kutta methods are the best known and often used single step integrators. The Runge-Kutta (RK) methods do not require explicit derivatives of $f(x, y)$. They are of the form,

$$y_{n+1} = y_n + h \sum_i^p a_i k_i,$$

where $k_p = f(x_n + c_p h, y_n + h(b_{p,1}k_1 + b_{p,2}k_2 + \dots + b_{p,p-1}k_{p-1}))$ for order p . For some order, the derivation of the RK method is closely linked to the truncation error, namely,

$$\varepsilon_n = y(x_{n+1}) - y(x_n) - h \sum_i^p a_i k_i.$$

By letting the coefficients \mathbf{b} be unknown with chosen a_i, c_i and solving the least squares system such that the error is minimized, it is possible to derive the RK method for order p . This can become very complicated for higher orders and is not terribly insightful. The

widely used RK4 method is known to be $\mathcal{O}(h^5)$ for 4 evaluations of $f(x, y)$ and the algorithm is given below.

$$\begin{aligned}
 y_{n+1} &= y_n + \frac{h}{6}(k_1) + 2k_2 + 2k_3 + k_4, \\
 x_{n+1} &= x_n + h, \\
 k_1 &= f(x_n, y_n), \\
 k_2 &= f\left(x_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right), \\
 k_3 &= f\left(x_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right), \\
 k_4 &= f(x_n + h, y_n + hk_3).
 \end{aligned}$$

An adaptive RK7(8) DA integrator [89] uses the precomputed coefficients and integrates the optical equations of motion. This algorithm was partially parallelized and adapted into PISCS to integrate the Lagrange basis polynomials and derive the quadrature weights.

B.3 Krylov solvers

This section is based on [107]. For matrix equation, $\mathbf{Ax} = \mathbf{b}$ with \mathbf{A} of size N , Krylov solvers rely on \mathbf{x} existing within the k th Krylov subspace,

$$\mathcal{K}_k = \text{span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0),$$

where \mathbf{r}_0 is the residual of initial guess \mathbf{x}_0 , $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. The generalized minimal residual (GMRES) minimizes the least squares,

$$\min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_k} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|.$$

For nonsingular matrix \mathbf{A} with the above iterate, GMRES is guaranteed to converge because the residual must decrease with every iteration. This also guarantees convergence in N iterations. The GMRES algorithm has some variations, but generally depends on the Arnoldi iteration to form the orthonormal basis of \mathcal{K}_k .

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{H}_k.$$

The columns of \mathbf{V}_k are the basis vectors of \mathcal{K}_k . Then,

$$\mathbf{r}_k = \mathbf{V}_{k+1}(\beta\mathbf{e}_1 - \mathbf{H}_k\mathbf{y}_k),$$

where $\beta\mathbf{e}_1^T = (\|\mathbf{r}_0\|, 0, \dots, 0)$ and \mathbf{y}_k minimizes $\|\beta\mathbf{e}_1 - \mathbf{H}_k\mathbf{y}_k\|$. Then,

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{V}_k\mathbf{y}_k.$$

This essentially gives the inner loop, where the k iterations build the orthonormal basis. Restarted GMRES stops the inner loop at some tolerance or after a certain number of iterations and gets \mathbf{x}_k . It then inputs \mathbf{x}_k as the new initial guess. This outer loop continues until a final tolerance is reached. Thus, there are two matrix vector products with \mathbf{A} in restarted GMRES, at the beginning of the outer loop and in the inner loop. Note that restarted GMRES is no longer guaranteed to converge in N iterations because we lost the information of the previous \mathcal{K}_k subspace.