

Differential algebraic fast multipole-accelerated boundary element method for nonlinear beam dynamics in arbitrary enclosures

A. J. Tencate¹, A. Gee¹, and B. Erdelyi*Department of Physics, Northern Illinois University, DeKalb, Illinois 60115, USA* (Received 5 December 2020; accepted 4 May 2021; published 24 May 2021)

A novel method is developed to take into account realistic boundary conditions in intense nonlinear beam dynamics. The algorithm consists of three main ingredients: the boundary element method that provides a solution for the discretized reformulation of the Poisson equation as boundary integrals; a novel fast multipole method developed for accurate and efficient computation of Coulomb potentials and forces; and differential algebraic methods, which form the numerical structures that enable and hold together the different components. The fast multipole method, without any modifications, also accelerates the solution of intertwining linear systems of equations for further efficiency enhancements. The resulting algorithm scales linearly with the number of particles N , as $m \log m$ with the number of boundary elements m , and, therefore, establishes an accurate and efficient method for intense beam dynamics simulations in arbitrary enclosures. Its performance is illustrated with three different cases and structures of practical interest.

DOI: [10.1103/PhysRevAccelBeams.24.054601](https://doi.org/10.1103/PhysRevAccelBeams.24.054601)

I. INTRODUCTION

Computational methods applied to nonlinear beam dynamics are diverse [1–3]. The optimal choice of methods, algorithms, and codes are typically chosen based on accuracy, efficiency, complexity of the underlying problem, and computation time. At one end of the spectrum are the tracking codes that assume collections of independent point charges in prescribed external electromagnetic fields. At the other end would be codes that solve the full set of Maxwell equations, including self- and external fields, as boundary and initial value problems, coupled to the dynamical propagation of particle distributions in time. The former are generally fast but include many simplifying assumptions leading to inaccuracies for complicated systems. The latter usually necessitates unrealistic amounts of computing power and time, rendering solutions unfeasible for many practical cases. The spectrum of numerical methods widely varies in approach, assumptions, approximations, and implementations to reduce complexity. In this paper, an intermediate level of complexity is sought, with emphasis on accurate and efficient methods for the inclusion in the dynamics of realistic boundary conditions, while taking into account the discrete and interacting nature of charged particle beams. This approach assumes that there is always a coordinate system in which the motion is

nonrelativistic, and, hence, the magnetic fields due to the relative motion of particles in the beam can be neglected. Additionally, the boundary conditions themselves must be slowly varying with respect to the beam velocity.

Interactions between particles within the beam and feedback effects from beam-wall interactions are called collective effects [2]. As increasingly higher beam intensities and small losses are requested by various applications, effects due to boundaries increase in magnitude as a function of beam sizes relative to vacuum chamber sizes and as a consequence of vacuum chamber asymmetries or inherent beam instabilities [2–4]. Moreover, for bunched beams in synchrotrons, collective effects can compound coherently to resonances that eventually degrade the beam beyond the acceptance of the machine [5]. Even if the collective instabilities do not act coherently, they will still lead to a tune shift whose magnitude is roughly proportional to the current density (and inversely proportional to the beam size) due to the self-fields. The effects of the indirect fields due to the boundary can increase the tune shift further. This tune shift often places an upper limit on the beam brightness. Collective beam instabilities are at the forefront of accelerator design and research, with the Fermilab Integrable Optics Test Accelerator (IOTA) [6–8] and the University of Maryland Electron Ring [9–13], for example, both dedicated to studying nonlinear effects in high-intensity regimes.

Modeling collective effects for intense beams is a challenging problem, in general, and even more so in the complicated geometries of electromagnetic devices in existing and planned machines. With the previously mentioned assumption, the physical problem can be cast mathematically into a Poisson equation with given

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

boundary conditions and a charge density distribution equal to a sum of Dirac delta functions [14]. The Poisson problem, in turn, can be decomposed into two parts: computing the self-fields on the beam with open boundary conditions and computing the modifications of those self-fields due to the presence of nontrivial physical enclosures [15,16].

The first part of the problem is the efficient computation of the Coulomb potentials among the beam particles. This amounts to a pairwise sum of the interaction between all of the particles, which natively requires $\mathcal{O}(N^2)$ operations. High-intensity beams will be comprised of $\mathcal{O}(10^{10})$ (or more) particles [6,13], which makes this part of the problem extremely computationally expensive. The N -body problem is not unique to beam physics, and many methods have been devised to make this computationally tenable. These approaches include approximating the discrete distribution as a smooth continuous distribution (space charge) [17], aggregating neighboring sources into macroparticles (particle in cell) [18], a hierarchical method computing nearby interactions directly and representing distant interactions using macroparticles (Barnes-Hut algorithm) [19], and a related hierarchical method which instead represents far interactions using a far-field expansion of the scalar potential [fast multipole method (FMM)] [20]. The first two methods are inadequate in high-intensity regimes where the stochastic part of the particle interactions is important, since they smooth over the effects of close interactions. The Barnes-Hut algorithm is effectively a first-order multipole expansion (containing only monopole terms), while the FMM includes higher-order terms that give a better representation of the distribution of distant sources. The FMM has effectively been applied to numerous interdisciplinary problems from biology to chemistry, cosmology, and physics [21–25]. The FMM has the additional benefit of providing rigorous error bounds [20] and is shown to exhibit $\mathcal{O}(N)$ scaling for large N [26].

The second part of the problem is a Laplace equation with modified boundary conditions. The conventional approach for its solution is to use a volume discretizing finite element method (FEM) such as COMSOL [27] and MICHELLE [28] or the particle-in-cell codes vsim [18] and WARP [29]. In the intense beam regime, this class of methods suffers both from inaccuracy and from inefficiency. Coulombic self-field forces increase linearly with increasing beam current (for constant beam size), and at least quadratically with decreasing beam size; thus, the representation of the beam as a continuous distribution sampled over the discretized volume in the FEM will lead to larger inaccuracies as the beam brightness increases in this manner. Additionally, the increase in beam charge gives rise to an increasingly nontrivial beam-wall interaction leading to a feedback effect. Increasing beam charge necessitates including the physical boundary in the computation rather than simply representing the boundary

effects by external applied fields. FEMs typically use a regular grid (though some will adaptively modify the element sizes) [18], which can form only a stepwise discrete representation of the boundary. Fully discretizing 3D space and solving becomes increasingly inefficient as smaller elements are required [30].

The boundary element method (BEM) [16] has unique advantages that can be leveraged in the intense beam regime, especially for beams whose size is of similar order to the vacuum chamber size. The BEM forms a 2D discretization of the boundary surface, which is both more efficient and more accurate than the corresponding FEM process [31,32]. Where the FEM uses an approximate representation of the differential equations evaluated throughout the discretized volume, the BEM first transforms them into boundary integral equations and then evaluates the equations exactly, thus only approximating the boundary itself. Additionally, this approximate boundary is more accurate than the boundary representation of the FEM [16,30]. Furthermore, the BEM can be accelerated with the FMM [15], which allows for $\mathcal{O}(N)$ scaling in the computation of the direct particle-particle and particle-wall interactions.

Both the FMM and the BEM are heavily reliant on algorithmic differentiation, which can be easily achieved via a differential algebra (DA) implementation. Incorporating the differentiation operation (and its inverse) to the standard addition and multiplication operations forms a differential algebraic structure. This enables the treatment of functions as continuous entities rather than relying on numerical methods based on discrete function evaluation [33]. In particular, this is achieved by casting the function as a truncated Taylor series to specified order, providing the groundwork for the formulation of both multipole expansions and translation operators, the fundamental structures of the FMM [26]. The DA structure has been successfully implemented at a language level in the Fortran-based COSY INFINITY [34], a general purpose nonlinear-dynamics code in which this algorithm was developed. The reader is referred to Appendix A of Ref. [26] and Chap. 3.2 of Ref. [35] for an introduction to differential algebras; Chap. 2 of Ref. [33] provides an in-depth treatise on the development and properties of differential algebras, and the subsequent chapters develop their use in the computation of Poincaré section maps for the study of the dynamic interactions between charged particle beams and accelerator structures.

This work provides the mathematical and computational foundations for a novel fast multipole-accelerated boundary element method enabled by differential algebraic methods and optimized for nonlinear beam dynamics at the intensity frontier. The novelty of this approach comes from interweaving the FMM and the BEM in the DA framework and leveraging the robust flexibility this provides for the application to 3D problems. Transforming the full 3D beam physics problem into the quasistatic beam frame enables the study of high-intensity beam-enclosure interactions

even within novel electromagnetic structures. The quasi-static assumption excludes modeling structures with high-frequency variations in the boundary conditions and dynamic effects such as traveling waves or trapped magnetic modes. The algorithm is benchmarked using various electromagnetic structures. The outline of the paper is as follows: Sec. II presents the application of a conventional BEM to the Laplace boundary value problem; Sec. III describes and references the adaptive FMM algorithm and sets forth the formalism included in this algorithm; Sec. IV describes the new DA FM-BEM algorithm; Sec. V provides benchmarking analysis of the Poisson solver including boundary element representation and a performance study of both serial and parallel versions. The accuracy of the method as a function of the algorithm parameters is presented for three fundamental electromagnetic structures with known analytic solutions. A brief summary is presented in the conclusion. Some elements of this work were presented in a recent dissertation [35].

II. CONVENTIONAL BOUNDARY ELEMENT METHODS

Boundary element methods are a class of solution methods for boundary integral equations. Many differential equations of interest have an equivalent integral formulation, which yields a unique solution for a given set of boundary conditions. Conventional BEMs consist of three primary steps. First, the boundary value problem (BVP) is reformulated in terms of a boundary integral equation (BIE). A numerical scheme is applied next, which reduces the BIE to a linear system of algebraic equations on the boundary. The solution (inversion) of the dense matrix formed in this step represents the most computationally expensive component of the BEM. Finally, the boundary solution can be applied to obtain the solution at any point in the region of interest. BEMs can be broken up into two main classes: direct and indirect methods [36].

A. Laplace boundary value problem

Consider an electrostatic BVP given by the Laplace equation and relevant boundary conditions

$$\begin{aligned} \nabla^2 \psi(\mathbf{x}) &= 0 \quad \text{for } \mathbf{x} \in \Omega; \\ \left. \begin{aligned} \psi(\mathbf{y}) &= g(\mathbf{y}) \\ \text{or} \\ \frac{\partial \psi}{\partial n_y}(\mathbf{y}) &= h(\mathbf{y}) \end{aligned} \right\} \quad \text{for } \mathbf{y} \in \Gamma, \end{aligned} \quad (1)$$

where $\Omega \subset \mathbb{R}^3$ is a bounded domain with a uniformly continuous boundary Γ . A linear combination of g and h is given such that at least one is specified at each point on the boundary. Unless explicitly stated, Γ is assumed to be smooth; indeed, the following derivation is still valid if

$\Gamma = \Gamma_1 \cup \dots \cup \Gamma_m$, where Γ_i is smooth for $i \in [1, m]$ [30]. This fact will be used to justify the validity of the solution when Γ is discretized in the numerical quadrature scheme. The partial derivative in terms of n_y is shorthand for the normal derivative at \mathbf{y} , namely, the operator $\mathbf{n}(\mathbf{y}) \cdot \nabla$. By convention, all surface normals are taken to be outward-facing normals. A fundamental solution (or Green's function) to Eq. (1) is one that satisfies

$$\nabla^2 G(\mathbf{x}, \mathbf{y}) = -\delta(\mathbf{x} - \mathbf{y}) \quad \text{for } \mathbf{x}, \mathbf{y} \in \Omega. \quad (2)$$

The existence of a fundamental solution is critical in deriving and solving a BVP in the integral representation. The well-known solution for the 3D Laplace equation,

$$G(\mathbf{x}, \mathbf{y}) \equiv \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad (3)$$

can be found in any electrodynamics textbook [14]. Utilizing Green's second identity

$$\int_{\Omega} [u \nabla^2 v - v \nabla^2 u] d\Omega = \int_{\Gamma} \left[u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right] d\Gamma, \quad (4)$$

the direct formulation of the BIE is found by letting $u = \psi$ and $v = G$ and including the relationships defined in Eqs. (1) and (2):

$$\psi(\mathbf{x}) = \int_{\Gamma} \left[G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) - g(\mathbf{y}) \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \right] d\Gamma(\mathbf{y}). \quad (5)$$

The representation formula (5) exists because ψ and G are both harmonic functions on the domain of Ω [37]. If both $g(\mathbf{y})$ (Dirichlet) and $h(\mathbf{y})$ (Neumann) boundary conditions are known, then, in principle, Eq. (5) naturally gives $\psi(\mathbf{x})$ for all $\mathbf{x} \in \Omega$. However, for typical problems, only one set of boundary conditions (g or h) is known *a priori*. More significantly, Eq. (1) has a unique solution (at least up to an arbitrary constant) when either Dirichlet or Neumann conditions are specified on the boundary [14]. This implies that simultaneously specifying both g and h will lead to an overdetermined system which, in general, has no solution.

B. Direct formulation

The regularized form of Eq. (5) for smooth boundary elements (see the Appendix) can be written in the single- or double-layer potential form

$$\begin{aligned} \mathcal{S}(h, \mathbf{x}) &\equiv \int_{\Gamma} h(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) \\ &= \int_{\Gamma} g(\mathbf{y}) \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) + \frac{1}{2} g(\mathbf{x}), \end{aligned} \quad (6)$$

$$\begin{aligned} \mathcal{D}(g, \mathbf{x}) &\equiv \int_{\Gamma} g(\mathbf{y}) \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) d\Gamma(\mathbf{y}) \\ &= \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) - \frac{1}{2} g(\mathbf{x}), \end{aligned} \quad (7)$$

for $\mathbf{x} \in \Gamma$ with known Dirichlet or Neumann boundary conditions, respectively. The factor of 1/2 comes from the regularization of the BIE in the limit that $\mathbf{x} \in \Omega \rightarrow \mathbf{x} \in \Gamma$ assuming a smooth boundary, as discussed in the Appendix. What remains is to solve Eq. (6) for h or Eq. (7) for g and use the result in Eq. (A6) to determine $\psi(\mathbf{x})$ for $\mathbf{x} \in \Omega \setminus \Gamma$. If values on the boundary are desired, then $\psi(\mathbf{x})$ for $\mathbf{x} \in \Gamma$ is either given by the boundary conditions (Dirichlet) or computed as the solution to Eq. (7) (Neumann).

It should be noted that Eq. (6) has a unique solution when g is continuous; however, Eq. (7) is not uniquely solvable even when h is continuous [37]. Moreover, it is necessary that the net flux (given by h) across the boundary be equal to zero. Since ψ is a potential, solving Eq. (7) will return g to only within a constant. Under these conditions, it is possible to modify Eq. (7) such that it is uniquely solvable [38].

C. Indirect formulation

Alternately, an analogous Laplace problem to Eq. (1) can be solved for the exterior region Ω_e , by utilizing the Kelvin transform $\mathcal{T}_K: \mathbb{R}^3 \setminus \{0\} \rightarrow \mathbb{R}^3 \setminus \{0\}$ by [Eq. (9.1.19) in Ref. [37]]

$$\mathcal{T}_K(x, y, z) \equiv \frac{1}{r^2}(x, y, z), \quad r = \sqrt{x^2 + y^2 + z^2}, \quad (8)$$

which represents a reflection along the radial direction through the unit sphere. Applying Eq. (8) to the exterior Laplace problem results in an equivalent interior problem, i.e., $\nabla^2 \psi_e(\mathbf{x}) = 0$ for $\mathbf{x} \in \Omega_e \Leftrightarrow \nabla^2 \tilde{\psi}(\mathbf{x}) = 0$ for $\mathbf{x} \in \tilde{\Omega}$ (where $\tilde{\Omega} = \Omega \setminus \{0\}$), whose solution has the form of Eq. (5). Since $\mathcal{T}_K^{-1} = \mathcal{T}_K$, ψ_e is given by $\mathcal{T}_K \tilde{\psi}$. This inverse transform is undefined for the origin. Considering the intersection of the interior and exterior solutions at Γ , we can define the charge density η and dipolar density σ functions:

$$\sigma(\mathbf{x}) \equiv \frac{\partial \psi}{\partial n_y}(\mathbf{x}) - \frac{\partial \psi_e}{\partial n_y}(\mathbf{x}), \quad \text{for } x \in \Gamma, \quad (9)$$

$$\eta(\mathbf{x}) \equiv \psi(\mathbf{x}) - \psi_e(\mathbf{x}), \quad \text{for } x \in \Gamma. \quad (10)$$

By this construction, both σ and η are harmonic functions on the boundary.

Analogous to the electrostatic image charge approach, assume the existence of a continuous charge distribution external to the boundary, which exactly reproduces the

boundary conditions on Γ [16]. This charge density can be constructed from monopoles or dipoles without loss of generality. As it is necessary only to match either the potential at (Dirichlet) or the flux through (Neumann) the boundary, only one ansatz (monopoles or dipoles) is required. While the charge density must be exterior to the boundary, i.e., some distance ζ from Γ , the approach remains valid in the limit as $\zeta \rightarrow 0$. This limit is typically taken in practice [39], and with $\zeta = 0^+$, the monopole and dipole densities converge, respectively, to σ and η as defined in Eqs. (9) and (10).

The solution to the Laplace problem (1) can, thus, be represented simply as either the single- or double-layer potentials applied to the ansatz:

$$\left. \begin{aligned} \psi(\mathbf{x}) &= \mathcal{S}(\sigma, \mathbf{x}) \\ \psi(\mathbf{x}) &= \mathcal{D}(\eta, \mathbf{x}) \end{aligned} \right\} \quad \text{for } \mathbf{x} \in \Omega. \quad (11)$$

The solution (11) is called indirect, because it is dependent on a potential density function (either σ or η), which has no physical interpretation, in contrast with the direct solution (5), which is derived directly from the representation formula (4) and depends explicitly on the physical boundary conditions g and h . Given Dirichlet conditions, and considering the limit $\mathbf{x} \rightarrow \Gamma$ for Eq. (11), the unknown densities must satisfy

$$\mathcal{S}(\sigma, \mathbf{x}) = g(\mathbf{x}), \quad (12)$$

$$-\frac{1}{2}\eta(\mathbf{x}) + \mathcal{D}(\eta, \mathbf{x}) = g(\mathbf{x}), \quad (13)$$

from the limits (A3) and (A5). Equations (12) and (13) are Fredholm equations of the first and second kind, respectively. If Neumann conditions are given, taking the normal derivative at \mathbf{x} of Eq. (12) and using arguments analogous to those in the derivation of Eq. (A5) yields

$$-\frac{1}{2}\sigma(\mathbf{x}) + \frac{\partial \mathcal{S}}{\partial n_x}(\sigma, \mathbf{x}) = h(\mathbf{x}). \quad (14)$$

The solution to Eq. (1) is, thus, obtained in the indirect form by solving Eq. (12), (13), or (14) for σ or η and then solving (11) for $\psi(\mathbf{x})$. As in the direct case, the specification of the Neumann problem will give the potential only up to an arbitrary constant.

III. A NOVEL ADAPTIVE FAST MULTIPOLE METHOD

The fast multipole method is used to efficiently compute the Coulombic interactions between a set of discrete charges and can also be used to evaluate the impact those charges would have on a test charge placed at specified locations. In typical FMM parlance (see Ref. [26], especially Appendix B), *sources* are the entities that generate

the potential, while *targets* refer to the locations at which to evaluate that potential. In the case of the particle beam, the sources and targets refer to the same list including each particle. Given a system of N particles, the electric scalar potential at a given target location \mathbf{x} is given by [Eq. (1.17) using Eq. (1.6) in Ref. [14]]

$$\varphi(\mathbf{x}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{q_i}{\|\mathbf{x} - \mathbf{x}_i\|}, \quad (15)$$

where \mathbf{x}_i are the source positions. The FMM approximates this equation by breaking the sum into a set of near and far evaluations. The near evaluations are calculated exactly, while the far evaluations are expressed instead as a sum of multipole expansions, leading to an algorithm which scales asymptotically as $\mathcal{O}(N)$ [20]. Much of Sec. III A follows the presentation in Ref. [26].

A. A Cartesian DA adaptive FMM

1. Domain division and structuring

A critical component of the FMM is the hierarchical subdivision and structuring of the simulation domain. First, the full three-dimensional space that encompasses the particle beam and enclosing structures is scaled to the unit cube, referred to as the root box. The root box is said to be at level 0. The first level of subdivision involves dividing the root box into eight congruent boxes and so on with the ℓ th level consisting of $2^{3\ell}$ congruent boxes [40]. A multilevel FMM implementation utilizing congruent boxes at each level enables efficient error control when incorporated as a part of dynamics simulations.

If the particle distribution is fairly uniform, fully subdividing the system down to a specified level (given by the desired accuracy or efficiency trade-off) would be sufficient. However, most systems of interest are far from uniform, and even a uniform system will evolve complex structure under external (and internal) forces, especially considering nonlinear effects. Thus, a regular FMM will lose efficiency due to empty boxes in low-density regions and overfilled boxes in high-density regions. A better general approach is an adaptive FMM, where, throughout the subdivision process, empty boxes are ignored and overpopulated boxes are further divided. The adaptivity of this process is governed by the clustering parameter q , which delineates the maximum number of sources permitted in the neighborhood of a given target.

The *neighborhood* consists of the target box and every box of the same level that shares a side or vertex with the target box (27 boxes in 3D for an interior target box). Conversely, boxes are said to be *well separated* from the target box if they are not in its neighborhood. An important data structure in the FMM is the *interaction list*. For a given box, its interaction list includes all child boxes of the boxes in its parent's neighborhood excluding its own neighbors

(up to 189 boxes in 3D space). In the adaptive FMM, it is possible for boxes to have an empty interaction list.

Computationally, this process makes use of octree data structures, in which each box (*parent*) is recursively divided into eight *child* boxes [40]. The source list may be substantively different from the target list, so optimal efficiency occurs when structure partitioning is fully adaptive with respect to both the targets and the sources. Thus, in the FMM formalism, two sets of data structures, referred to as *trees*, are generated. A set of boxes constitutes a tree if parent-child relationships can be established among them. The D tree grows from the root box (level 0) and includes each successive child box that contains at least one target up to the finest level specified by the clustering parameter q . Using the structure of the D tree, the C forest is a disconnected set of trees whose root boxes are all level-2 boxes in the D tree which contains sources. Each C tree grows from its root box (a level-2 D-tree box) and includes each successive child box in the D tree that contains at least one source. The C trees direct the collection of multipole expansions given by the sources at the second level, while the D tree directs the translation of the expansions from the second level to each target at the highest level. Put another way, if a box needs a local expansion anywhere in the algorithm, then it is in the D tree; if a box needs a multipole expansion anywhere in the algorithm, then it is in the C forest.

2. Multipole expansion and translation in a differential algebra

Once the data structures have been generated, multipole expansions of Eq. (15) are calculated at the highest-level boxes in the C trees. This is accomplished through the introduction of DA variables [26]

$$\begin{aligned} d_x &\equiv \frac{x - x_0}{r^2}, & d_z &\equiv \frac{z - z_0}{r^2}, \\ d_y &\equiv \frac{y - y_0}{r^2}, & d_r &\equiv \frac{1}{r}, \end{aligned} \quad (16)$$

which represent an expansion about the point $\mathbf{x}_0 = (x_0, y_0, z_0)$ and where $r \equiv \|\mathbf{x} - \mathbf{x}_0\|$. Assuming the box is centered at \mathbf{x}_0 , the potential given by Eq. (15) can be expressed in terms of the DA variables in Eq. (16) leading (after simplification) to a multipole expansion about the center of the box:

$$\begin{aligned} \varphi(\mathbf{d}) &= \frac{d_r}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i}{\sqrt{1 + \|\mathbf{x}_i - \mathbf{x}_0\|^2 d_r^2 - 2(\mathbf{x}_i - \mathbf{x}_0) \cdot \mathbf{d}}} \\ &= kd_r \varphi_m, \end{aligned} \quad (17)$$

where n gives the number of sources in the particular box, k includes the constant terms (typically normalized to 1), φ_m represents the multipole expansion, and the DA vector is

defined $\mathbf{d} \equiv (d_x, d_y, d_z)$. Unlike the typical form of φ_m which sums monopole, dipole, and higher-order terms due to a collection of sources [Eq. (4.10) in Ref. [14]], Eq. (17) instead forms a high-order expansion for each source and sums over all relevant sources. Here, $d_r = \|\mathbf{d}\|$ is the Euclidean norm of the DA vector and cannot be expanded. Rather, it is carried through the following calculations and evaluated in the final step.

The potential φ_m in Eq. (17) is expressed as a truncated Taylor series in the DA framework. It should be noted that the series expansion in the given DA variables will converge only if r is larger than the side length of a box [26], which enforces the requirement that the DA variables in Eq. (16) be small. This establishes the radius of convergence and limits the region of validity for the expansion to beyond the neighborhood of the box in which it originated.

The multipole expansion φ_m is then translated from the child box to the parent box, with center \mathbf{x}'_s . This multipole-to-multipole (M2M) translation forms a transform $\mathcal{T}_{M2M}: \mathbb{R}^3 \mapsto \mathbb{R}^3$ where the translation is expressed as the composition

$$\varphi'_m = \varphi_m \circ \mathcal{T}_{M2M}$$

and where φ'_m is again valid only for observers far from the parent box. \mathcal{T}_{M2M} represents a map from one set of DA variables (d_r, \mathbf{d}) to a new set (d'_r, \mathbf{d}') centered about \mathbf{x}'_s :

$$\begin{aligned} \mathbf{d}' &\equiv \frac{\mathbf{x} - \mathbf{x}'_s}{r'^2}, \\ d'_r &\equiv \frac{1}{r'}, \\ r' &\equiv \|\mathbf{x} - \mathbf{x}'_s\| \end{aligned} \quad (18)$$

and is explicitly defined as

$$\mathcal{T}_{M2M} \equiv \begin{cases} d_r = \sqrt{R} d'_r, \\ \mathbf{d} = R[\mathbf{d}' + (\mathbf{x}'_s - \mathbf{x}_0) d'^2_r], \end{cases} \quad (19)$$

where the radial differential scaling factor R is given by

$$R \equiv [1 + \|\mathbf{x}'_s - \mathbf{x}_0\|^2 d'^2_r + 2(\mathbf{x}'_s - \mathbf{x}_0) \cdot \mathbf{d}']^{-1}. \quad (20)$$

The electric potential (17) can be expressed in terms of the translated multipole expansion via Eq. (19) as

$$\varphi(\mathbf{d}') = k\sqrt{R} d'_r \varphi'_m.$$

The M2M translation is then recursively applied to the potential φ'_m , traversing the C tree from child to parent boxes until the level-2 box in the tree is reached.

From this point, the multipole expansion is translated to a well-separated box at the same level, containing the

observer location (\mathbf{x}), whose center is given by \mathbf{x}'_t . This multipole-to-local (M2L) translation forms another transform $\mathcal{T}_{M2L}: \mathbb{R}^3 \mapsto \mathbb{R}^3$ where the translation is again expressed as the composition

$$\varphi'_l = \varphi'_m \circ \mathcal{T}_{M2L}.$$

Unlike φ_m and φ'_m , the local expansion φ'_l is convergent for targets within a radius that includes the local box (and, by construction, all children of that box) [26]. The variables for the local expansion are simply given by

$$\mathbf{d}'' \equiv \mathbf{x} - \mathbf{x}'_t, \quad (21)$$

where \mathbf{d}'' is guaranteed to be small because the target is near the center, hence a local expansion. The M2L translation is explicitly defined by

$$\mathcal{T}_{M2L} \equiv \begin{cases} d'_r = \sqrt{R'}, \\ \mathbf{d}' = R'(\mathbf{d}'' + (\mathbf{x}'_t - \mathbf{x}'_s)), \end{cases} \quad (22)$$

where the new radial differential scaling factor R' is given by

$$R' \equiv \|\mathbf{d}'' + (\mathbf{x}'_t - \mathbf{x}'_s)\|^{-2}. \quad (23)$$

In terms of the local expansion, the electric potential is given by

$$\varphi(\mathbf{d}'') = k\sqrt{RR'} \varphi'_l.$$

The process of transforming multipole-to-local expansions in 3D becomes increasingly inefficient and represents the greatest computational expense in the FMM for higher expansion orders [26,35,41]. This process can be accelerated by introducing a 3D rotation \mathcal{R} which aligns the z axis of the multipole expansion φ'_m with the translation direction $(\mathbf{x}'_t - \mathbf{x}'_s)$ and applying a 1D translation along that axis. Since \mathcal{R} is an orthogonal matrix, the rotated M2L translation is expressed by the composition

$$\varphi'_l = \varphi'_m \circ \mathcal{R} \circ \mathcal{T}_{M2L}^{1D} \circ \mathcal{R}^T,$$

where \mathcal{T}_{M2L}^{1D} has the same form as Eq. (22) but contains only the z component of the transformation. More details on this process can be found in Appendix C of Ref. [26].

Analogous to the M2M translation but in reverse, a local-to-local (L2L) translation is defined which represents a map from parent to child boxes in the target D tree. This can be expressed as the transform $\mathcal{T}_{L2L}: \mathbb{R}^3 \mapsto \mathbb{R}^3$, and the new potential is again expressed as the composition

$$\varphi_l = \varphi'_l \circ \mathcal{T}_{L2L}.$$

With the potential expressed in terms of a local expansion (φ'_l), \mathcal{T}_{L2L} gives the translation to a child box with center \mathbf{x}_l in terms of the new DA variable $\mathbf{d}''' \equiv \mathbf{x} - \mathbf{x}_l$ and can be defined as the map

$$\mathcal{T}_{L2L} \equiv \{\mathbf{d}'' = \mathbf{d}''' + (\mathbf{x}_l - \mathbf{x}'_l)\}, \quad (24)$$

which is a simple translation. The L2L translation is recursively applied to disseminate φ'_l down the D tree (from level 2) to the highest-level boxes containing targets. For completeness, the final form of the electric potential in terms of the local expansion at each target box will be

$$\varphi(\mathbf{d}''') = k\sqrt{RR'}\varphi_l. \quad (25)$$

3. The fast multipole algorithm

With the preliminaries in place, this section sets forth the order and process of the adaptive FMM algorithm. The algorithm is initialized with the specification of the source and target sets, the source charges, and the clustering parameter q . Following Sec. III A 1, the spatial domain is scaled and recursively subdivided to form the target D tree as constrained by q , and the D tree is traversed in reverse to form the source C forest (whose trees grow from level-2 boxes). Once the data structures have been constructed, the remainder of the process can be subdivided into three stages: the upward pass, the downward pass, and the final summation [26].

During the upward pass, the multipole expansion (17) is computed at the finest level boxes in each of the C trees. These expansions are then translated from child to parent nodes up the C tree via the M2M operator (19). At each parent node, the source contributions from each child node are summed together to form a single composite multipole expansion; this can be done directly in the DA framework. This process is iterated until all source contributions are represented by multipole expansions about the center of level-2 boxes.

The downward pass begins with the transformation of the multipole expansions into local expansions (22) and is guided by the D tree. This transform must be applied from each level-2 box containing a multipole expansion to all well-separated boxes at the same level. More significantly, the multipole expansions of all higher-level boxes with a nonempty interaction list must be distributed via the M2L transform to each box in the interaction list. The sheer number of applications of the transform required in this step leads to its significant computational expense, a factor that is somewhat mitigated in the high-accuracy regime by the inclusion of the rotated M2L. The local expansions are then translated from parent to child down the D tree via the L2L operator (24), summing together respective M2L interaction list contributions.

The final summation phase occurs once all local expansions have been transferred to the highest-level boxes in the D tree. Local expansions are evaluated at each target point, an elementary operation in DA. This operation is often referred to as the local-to-point (L2P) evaluation. Finally, the contributions of sources in the target neighborhood are directly summed via Eq. (15) and added to the result of the L2P evaluation. In some applications, the electric fields may be desired in addition to the electric potential. The fields in the neighborhood are directly given by

$$\mathbf{E} \equiv -\nabla\varphi(\mathbf{x}) = -\frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{q_i(\mathbf{x} - \mathbf{x}_i)}{\|\mathbf{x} - \mathbf{x}_i\|^3}, \quad (26)$$

using Eq. (15). The far-field contributions are obtained by applying the gradient to the local expansions (resulting in an expansion for each field component) and evaluating the results again with the L2P evaluation.

It is worth noting that this formulation uses a straightforward Cartesian representation for all expansions and translations, in contrast with standard algorithms, which rely on kernel-dependent formulations, including spherical representations [20,24,42]. This is achieved through the differential algebraic implementation, in which truncated Taylor expansions, function compositions, and derivation are all elementary operations [33], leading to a general formulation that can easily be applied to general potentials [26]. This algorithm has been shown to exhibit the desired $\mathcal{O}(N)$ scaling and to yield results which converge to the brute force method of Eq. (15) for high-order expansions [26].

IV. THE POISSON INTEGRAL SOLVER WITH CURVED SURFACES

A charged particle beam within a bounded structure is described by the Poisson problem

$$\left. \begin{aligned} \nabla^2 \tilde{\psi}(\mathbf{x}) &= \frac{\rho(\mathbf{x})}{\epsilon_0} & \text{for } \mathbf{x} \in \Omega; \\ \tilde{\psi}(\mathbf{y}) &= \tilde{g}(\mathbf{y}) \\ \text{or} & \\ \frac{\partial \tilde{\psi}}{\partial n_y}(\mathbf{y}) &= \tilde{h}(\mathbf{y}) \end{aligned} \right\} \text{for } \mathbf{y} \in \Gamma, \quad (27)$$

where, as before, $\Omega \subset \mathbb{R}^3$ is a bounded domain with a uniformly continuous boundary Γ and where g or h is specified at every point on Γ . The source term ρ describes a beam of N pointlike particles and can, thus, be represented as a sum:

$$\rho(\mathbf{x}) = \sum_{i=1}^N q_i \delta(\mathbf{x} - \mathbf{x}_i), \quad (28)$$

where q_i denotes the charge and \mathbf{x}_i the position of the i th particle. The linearity of the Laplacian enables the Poisson

BVP (27) to be deconstructed to an equivalent system comprising a boundaryless Poisson problem, whose well-known solution φ is given by Eq. (15), coupled to a Laplace problem for ψ analogous to Eq. (1) but with modified boundary conditions. In order for the coupled solution $\tilde{\psi} \equiv \varphi + \psi$ to satisfy the boundary conditions of Eq. (27), the boundary conditions for Eq. (1) must be

$$\begin{aligned} g(\mathbf{y}) &= \tilde{g}(\mathbf{y}) - \varphi(\mathbf{y}), \\ h(\mathbf{y}) &= \tilde{h}(\mathbf{y}) - \frac{\partial \varphi}{\partial n_y}(\mathbf{y}), \end{aligned} \quad (29)$$

modified by the solution to Eq. (15) on Γ .

The efficient computational solution to Eq. (27) with open boundaries via the FMM has already been discussed in Sec. III. The approach to solving the Laplace BVP, with the boundary conditions modified by Eq. (29), was addressed in Sec. II. The indirect method is chosen, as it involves one fewer integrations over the boundary. What remains is to lay out the numerical approach to evaluating the boundary integral equations (11)–(14). These BIEs are discretized directly using the Nyström quadrature method, and the resulting linear system is iteratively solved by the generalized minimum residual (GMRES) method.

A. Discretization and the Nyström method

Given Dirichlet boundary conditions, there are two possible approaches to solving the Laplace BVP, Eqs. (12) and (13), which form Fredholm equations of the first and second kind, respectively. The Nyström method [37] was originally developed as a numerical solution to Fredholm integral equations of the second kind:

$$y(t) = \lambda x(t) - \int_D G(t, s)x(s)ds \equiv (\lambda - \mathcal{G})x(t), \quad t \in D, \quad (30)$$

which have been well studied and exhibit useful properties. The Nyström method approximates, for example, Eq. (13) as a discrete weighted sum of the kernel evaluated at m locations:

$$-\frac{1}{2}\eta_m(\mathbf{x}_i) + \sum_{\substack{j=1 \\ i \neq j}}^m w_j \eta_m(\mathbf{y}_j) \frac{\partial G}{\partial n_y}(\mathbf{x}_i, \mathbf{y}_j) = g(\mathbf{x}_i). \quad (31)$$

This forms an $m \times m$ linear system of equations for $i \in [1, m]$ of the form $\mathbf{A}_m \boldsymbol{\eta}_m = \mathbf{g}$. Here, the exact solution to Eq. (31), $\boldsymbol{\eta}_m$, is obtained, which is a discrete approximation of $\eta(\mathbf{x})$. The system matrix \mathbf{A}_m for the BEM is typically dense and nonsymmetric [15,16]; thus, it is critical that \mathbf{A}_m be well conditioned and that the numerical quadrature scheme demonstrates rapid convergence.

The condition number of a matrix \mathbf{A}_m is defined as $\text{cond}(\mathbf{A}_m) \equiv \|\mathbf{A}_m\| \|\mathbf{A}_m^{-1}\|$ and gives a relative measure of the sensitivity of a system to perturbative errors. Any numerical method invariably introduces errors due to approximations and numerical noise; thus, it is imperative to construct a well-conditioned linear system. It can be shown that the condition number of the system matrix is bounded by [37]

$$\text{cond}(\mathbf{A}_m) \leq \|\lambda - \mathcal{G}_m\| \|(\lambda - \mathcal{G}_m)^{-1}\|, \quad (32)$$

where $(\lambda - \mathcal{G}_m)$ is the numerical operator of the corresponding linear equation, comparable to Eq. (31). For the continuous system, Theorem 4.1 [37] describes some of the advantages of Fredholm equations of the second kind.

Theorem 4.1 (Fredholm alternative).—Let \mathcal{X} be a Banach space, and let $\mathcal{G}: \mathcal{X} \rightarrow \mathcal{X}$ be a compact operator. Then the equation $(\lambda - \mathcal{G})x = y$, $\lambda \neq 0$ has a unique solution $x \in \mathcal{X}$ if and only if the homogeneous equation $(\lambda - \mathcal{G})z = 0$ has only the trivial solution $z = 0$. In such a case, the operator $\lambda - \mathcal{G}: \mathcal{X} \rightarrow \mathcal{X}$ has a bounded inverse $(\lambda - \mathcal{G})^{-1}$.

Significantly, such a system is guaranteed to have both a unique solution and a bounded inverse. The Appendix has shown that $(\lambda - \mathcal{G})$ is bounded for both Eqs. (13) and (14). Taken with Theorem 4.1, this implies that $\text{cond}(\lambda - \mathcal{G})$ is finite and bounded. Theorem 4.1.2 in Ref. [37] states that there exist finite numbers N and c_s such that $\|(\lambda - \mathcal{G}_m)^{-1}\| \leq c_s$ for $m \geq N$. Assuming that the numerical method is chosen appropriately, the numerical integrals will converge to the true integrals as $m \rightarrow \infty$. Thus, for type-2 integrals, Eq. (32) will be bounded, leading to a numerical solution with low error sensitivity. Fredholm type-1 integrals have a high sensitivity to small errors, as the condition number for type-1 systems is larger and grows more quickly than that of type-2 formulation for the same problem [37]. This property is clearly illustrated in Table I, which gives the matrix condition numbers for a sphere as a function of the size of the linear system m .

The linear system corresponding to the first kind of integral equation with Dirichlet boundary conditions is seen to rapidly degrade as the size of the system is increased, while the second kind of integral equation leads to well-behaved linear systems whose condition number is of $\mathcal{O}(1)$ irrespective of matrix size. Thus, the double-layer

TABLE I. Matrix condition numbers of the linear system from the first (12) and second (13) kind of equations with Dirichlet conditions and second kind of equation (14) with Neumann conditions.

m	Dirichlet (1st)	Dirichlet (2nd)	Neumann (2nd)
80	1.58×10^3	2.891	2.867
320	2.63×10^3	2.912	2.933
1280	2.06×10^4	2.951	2.968
5120	1.60×10^6	2.975	2.985

potential equation (13) will be used for the Dirichlet problem, while the single-layer potential equation (14) is used for the Neumann problem. This selection is equivalent to performing an analytical preconditioning directly in the problem formulation, which obviates the need for a numerical preconditioner in solving the resultant linear systems.

The implementation requires summation of pointlike evaluations on the boundary, and the accuracy of the numerical integration is determined by the quadrature rule [43]. The boundary surface is discretized into a mesh of m flat triangular surface elements, $\Gamma \mapsto \cup_{j=1}^m \Gamma_j$. The nodes \mathbf{x}_j and corresponding normal vectors are chosen to be the center of the corresponding element Γ_j . In this regime, the quadrature weight w_j is simply the area of Γ_j and is calculated by the magnitude of the cross product $w_j = (1/2)\|(\mathbf{a}_j - \mathbf{b}_j) \times (\mathbf{a}_j - \mathbf{c}_j)\|$, where \mathbf{a}_j , \mathbf{b}_j , and \mathbf{c}_j point to the vertices of Γ_j .

This discretization of the indirect method is known to give an unstable evaluation near the boundary due to near-singular integrals [30,35,44,45]. Because of the formation of a discrete density $\boldsymbol{\eta}_m$, the resultant quadrature \mathcal{Q} of integration \mathcal{I} is subject to the error in each term of the sum. The error in the solution is bounded by an expression of the form [46]

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

where C is an undetermined finite constant. If a quadrature term has a large jump, i.e., approaching the singularity, so does the error bound. In this case, the jump comes from the second term when $\mathbf{x} \approx \mathbf{y}_i$, while the first term remains finite and nonzero. Beam physics problems are concerned primarily with sources clustered relatively far from the surface, as surface interactions immediately lead to beam loss. Thus, the near-boundary instability of the indirect numerical method is not a concern for this work. Additionally, results in Sec. V show that, by increasing the number of boundary elements, this instability can be made negligible for the interior domain of interest.

B. GMRES

The linear system $\mathbf{A}_m \boldsymbol{\eta}_m = \mathbf{g}$ resulting from Eq. (31) is iteratively solved using a matrix-free restarted form of the GMRES method. GMRES was developed as a Krylov subspace method for nonsymmetric linear systems [47,48]. GMRES iteratively builds up the Krylov subspace

$$\mathcal{K}_n = \text{span}(\mathbf{r}_0, \mathbf{A}_m \mathbf{r}_0, \dots, \mathbf{A}_m^{n-1} \mathbf{r}_0) \quad \text{for } n \leq m, \quad (33)$$

where $\mathbf{r}_0 = \mathbf{g} - \mathbf{A}_m \boldsymbol{\eta}_0$ is the residual of initial guess $\boldsymbol{\eta}_0$. It is assumed that the exact solution exists in this space, $\boldsymbol{\eta}_m \in \mathcal{K}_m$, and an approximate solution $\tilde{\boldsymbol{\eta}}_k$ is determined through the solution of the least squares problem [35,48]

$$\min_{\tilde{\boldsymbol{\eta}}_n \in \eta_0 + \mathcal{K}_n} \|\mathbf{g} - \mathbf{A}_m \tilde{\boldsymbol{\eta}}_n\|. \quad (34)$$

As a property of the Krylov subspace, GMRES converges monotonically and is mathematically guaranteed to obtain the correct solution in at most m iterations (by Theorem 3.1.2 in Ref. [48]). However, for large m , the high number of matrix vector products in Eq. (33) becomes prohibitively expensive to compute. The process can be accelerated by forming a solution from \mathcal{K}_k for $k < m$ and restarting the iterations taking $\tilde{\boldsymbol{\eta}}_k$ as the new initial guess. Restarted GMRES accelerates the solution process, but at the expense of losing the guarantee for monotonic convergence. Convergence is generally maintained as long as \mathbf{A}_m is well conditioned.

Typically, GMRES is paired with a preconditioner, often in the form of a matrix \mathcal{M} such that $\text{cond}(\mathcal{M}\mathbf{A}_m) < \text{cond}(\mathbf{A}_m)$ (left preconditioning) or $\text{cond}(\mathbf{A}_m\mathcal{M}) < \text{cond}(\mathbf{A}_m)$ (right preconditioning). The disadvantage to this approach is that it requires forming and storing the full matrix \mathbf{A}_m , which is costly in both memory and evaluation time [48]. A consequence of the properties of linear systems formed from Fredholm type-2 equations is that they are and remain well conditioned even for large m (see Table I). Therefore, preconditioning ought not be necessary. Indeed, early tests showed that the use of a preconditioner did not significantly accelerate the solver [35].

The bulk of the computational expense in GMRES comes from the iterative matrix-vector products $\mathbf{A}_m \mathbf{r}_0$. However, the off-diagonal entries of Eq. (31) are proportional to the derivative of the Green's function (3), so it is possible to accelerate the computation via the FMM [49,50]. The formulation of the multipole expansion for GMRES will have a slightly different form from that of Eq. (16) owing to the difference between Eqs. (15) and (31). Explicating the normal derivative in Eq. (31) yields

$$\begin{aligned} \varphi_{\text{gmres}}(\mathbf{x}_i) &= \mathbf{A}_m \boldsymbol{\eta}_m \\ &= -\frac{1}{2} \eta_{m,i} + \sum_{\substack{j=1 \\ i \neq j}}^m \frac{w_j \eta_m(\mathbf{y}_j) \mathbf{n}(\mathbf{y}_j) \cdot (\mathbf{x}_i - \mathbf{y}_j)}{4\pi \|\mathbf{x}_i - \mathbf{y}_j\|^3}. \end{aligned}$$

Defining a new DA vector centered at \mathbf{x}_0 by $r^2 \mathbf{d}_i \equiv \mathbf{x}_i - \mathbf{x}_0$ and simplifying yields the GMRES multipole expansion in DA in a similar form to Eq. (16):

$$\begin{aligned} \varphi_{\text{gmres}}(\mathbf{d}_i) &= -\frac{1}{2} \eta_{m,i} + \frac{d_r}{4\pi} \sum_{\substack{j=1 \\ i \neq j}}^m \frac{w_j \eta_{m,j} \mathbf{n}_j \cdot (\mathbf{d}_i - (\mathbf{y}_j - \mathbf{x}_0) d_r^2)}{\sqrt{[1 + |\mathbf{y}_j - \mathbf{x}_0|^2 d_r^2 - 2(\mathbf{y}_j - \mathbf{x}_0) \cdot \mathbf{d}_i]^3}}. \end{aligned} \quad (35)$$

A major advantage of the kernel independence of the Cartesian FMM being used [26] is that the multipole transfer maps have the same form as with the Coulomb potential in

Sec. III. It is not necessary to know the analytic form of the multipole expansion, as the transformations are automatically performed numerically. This formulation additionally leads to a significant reduction in memory, as the full matrix $(A)_m$ is never stored.

Utilizing the FMM to compute the matrix-vector product introduces an approximation error that can interfere with the convergence properties of the algorithm [51–54]. However, this deviation is bounded by the product of the magnitude of the error vector and the norm of \mathcal{K}_n [51]. Additionally, the FMM error has been shown to scale with expansion order p like $p \sim \log_{10}(\epsilon)$ [26]; thus, it is possible to constrain the system such that GMRES converges with a prescribed accuracy [54]. This provides an additional benefit, namely, that the FMM order may be relaxed in subsequent iterations as the residual shrinks [35]. Letting $p_{\min} = 2$, the relaxation process is implemented using the strategy from Ref. [54] to select the iteration order with ϵ determined by the ratio of absolute tolerance τ_{res} with the estimated residual. The initial evaluation each time GMRES is restarted remains set at p_{\max} , the prescribed FMM order.

The definition of τ_{res} has a substantial impact on the rate of convergence in GMRES. If τ_{res} is too large, then GMRES becomes the dominant source of error in the BEM, and if too small, then the process stagnates, performing numerically redundant iterations. The final residual, given by the norm of the residual vector, scales as a function of the boundary condition value φ , number of boundary elements m (matrix size), and the FMM polynomial order p . Optimization tests with a variety of examples has lead to the following definition:

$$\tau_{\text{res}} \equiv \mathcal{C}_\tau \mathcal{N}_e(m) \Phi_{bc}(\varphi) f(p), \quad (36)$$

where

$$\begin{aligned} \mathcal{C}_\tau &= 10^{-3}, \\ \mathcal{N}_e(m) &= \frac{\sqrt{m}}{20}, \\ \Phi_{bc}(\varphi) &= \left(\max \left[\|\varphi\| \quad \text{or} \quad \left\| \frac{\partial \varphi}{\partial n} \right\| \right] \right)^{4/5}, \\ f(p) &= 10^{-(p+1)/5}. \end{aligned}$$

This definition leads to consistent convergence in a moderate number of iterations without significantly impacting the overall error scaling of the BEM for each of the structures evaluated. It is possible to improve performance further for certain problems by trial and error on a case by case basis.

C. A fast multipole-accelerated solution to the Poisson problem

The composition of the fast multipole method with the Nyström boundary element method lead to the

development of the Poisson integral solver with curved surfaces (PISCS). The approach in this algorithm involves breaking the Poisson problem (27) into an open boundary Poisson problem solved via the FMM and a Laplace problem with modified boundary conditions. The Laplace boundary integral equations are discretized through the Nyström quadrature method, leading to the formation of a linear system which is solved by the GMRES method incorporating FMM accelerated matrix-vector products. The final results are obtained through a final FMM application by casting Eq. (11) in terms of the multipole expansion given in Eq. (35) with the final potential density function and no constant term.

PISCS is written in COSYScript for COSY INFINITY v10.0 [55], a Fortran-based scripting language developed at Michigan State University (MSU) [34]. COSY incorporates a language-level differential algebra implementation and includes a robust suite of beam physics routines and procedures. COSY utilizes the DA to create and compose high-order transfer maps to generate and analyze particle accelerator beam lines. A package including the PISCS code and relevant utilities can be found on the beam physics code repository [56]; COSY INFINITY must be obtained through MSU [55]. A user manual and examples for PISCS included in the package provide more details on running the algorithm. PISCS uses a Cartesian coordinate system based on the boundary structure, not the beam. This must be considered when incorporating the results of PISCS analysis with standard beam physics tracking codes. It is assumed that the structure longitudinal axis is aligned with the beam longitudinal axis, which is comparable to the cylindrically symmetrical beam transport designs [5]. PISCS can generally be broken up into five sections: reading and initializing the data files, three FMM calls, and the final summation of the results. Figure 1 outlines the major procedures and progression in the implementation of PISCS. PISCS saves interim results throughout in order to preserve dynamic memory for cases with a high number of particles and boundary elements.

First, the simulation is initialized by reading the boundary structure and particle beam files and setting algorithmic parameters including DA order and boundary condition type. Point files formatted for the FMM are written from the input structure and beam files. The first FMM call assigns the beam particles as *sources* and both particles and boundary points as *targets*. This call is used to calculate the particle self-fields and the modifications to the boundary conditions due to the charged particle beam given by Eq. (15). The second FMM call assigns the boundary points as both sources and targets. Here, the FMM algorithm actually runs many times as GMRES iteratively solves the system matrix for the layer-potential density (9) and (10), a process involving the computation of numerous matrix-vector products. The final FMM call assigns the boundary points as sources and the particles

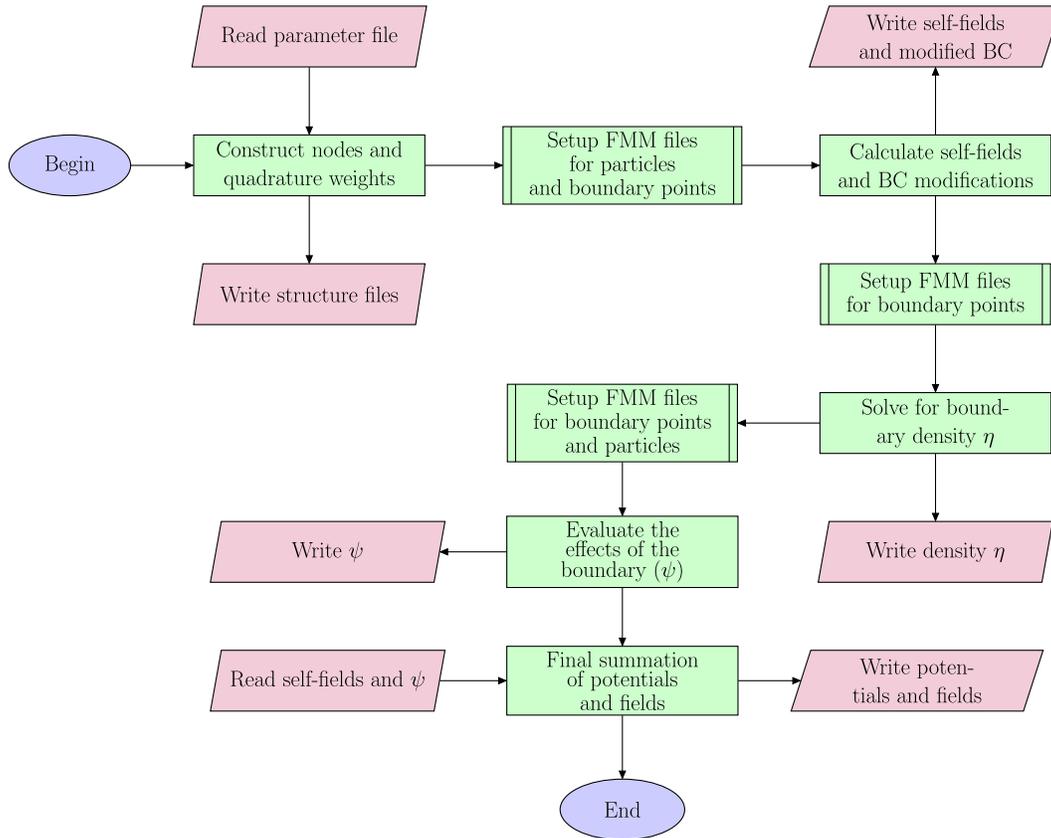


FIG. 1. Flowchart for the implementation of PISCS. Green nodes give the major internal modules, with the double-bounded nodes representing calls to the external fmmcpp structuring script, while red nodes represent read-write steps.

as targets. This call evaluates (13) and (14) using the previously determined density to determine the effect of the structure on the particle beam. Lastly, PISCS sums up the contribution of the self- and the boundary components for each particle and writes the resulting scalar potential and field at each particle position.

Using the DA framework throughout the calculations enables PISCS to simultaneously compute both the scalar potentials and the fields. This eliminates the additional errors that are incurred when calculating only the potential and attempting to interpolate the fields or vice versa. It is worth noting that PISCS can be used to solve a multitude of problems beyond those applicable to beam physics applications. Any system that can be cast as a quasistatic electromagnetic BVP could be solved, with or without the presence of sources within the boundary region.

COSY INFINITY supports an message passing interface (MPI) parallelization scheme which utilizes a high-performance vector data type to facilitate the sharing of information between tasks distributed via a parallel loop structure. The bulk of the parallelization (and the overall computation time) in PISCS is accomplished in the FMM, which has been extensively optimized. The FMM distributes the tasks using a dynamic load-balancing approach by first sorting the independent entities (i.e.,

C-forest trees or D-tree nodes) and then distributing them evenly among the available processes [57].

1. Units and boundary conditions in the beam frame

PISCS assumes that the boundary conditions are specified in SI units. Thus, the electrostatic boundary conditions should be specified in units of volts (Dirichlet) or volts per meter (Neumann). For problems that can be described in magnetostatic terms, the boundary conditions should be specified in units of Tesla-meters (Dirichlet) or Tesla (Neumann). These values are converted to the normalized working units of the simulation:

$$\varphi_{\text{units}} = \begin{cases} \frac{\epsilon_0}{e} \varphi_{\text{SI}} & \text{for } \varphi_e \text{ or } \mathbf{E}, \\ \frac{1}{\mu_0 e} \varphi_{\text{SI}} & \text{for } \varphi_m \text{ or } \mathbf{B}. \end{cases}$$

The charge value(s) for sources in the volume Ω must, therefore, be specified in terms of the unit charge (i.e., -1 for electrons). This normalization serves to mitigate the risk of numerical instabilities arising due to large order of magnitude difference in values used in the numerical equations.

Relevant problems in beam dynamics consist of the beam, a collection of copropagating charged particles, and the surrounding enclosure, which often consists of

electromagnetic elements. The driving electromagnetic forces in this system are in full generality described by Maxwell's equations [14]. The imposing system of equations can be substantially reduced in complexity by a clever shift in frame of reference. This is achieved by prescribing a reference particle following an “ideal” trajectory through the accelerating structure. The problem can then be transformed to the reference particle frame, which leads to a quasistatic system.

Taking the normalized reference particle velocity to be given by $c\beta$, the longitudinal and transverse velocities of the particles comprising the beam must be transformed from the lab frame S to the reference particle frame \tilde{S} . The resulting velocity components are given by the Lorentz transform [from Eq. (11.31) in Ref. [14]]

$$\begin{aligned}\frac{\tilde{v}_{\parallel}}{c} &= \frac{v_{\parallel}/c - \beta}{1 - \beta v_{\parallel}/c} \approx \gamma^2 \left(\frac{v_{\parallel}}{c} - \beta \right), \\ \frac{\tilde{\mathbf{v}}_{\perp}}{c} &= \frac{\mathbf{v}_{\perp}/c}{\gamma(1 - \beta v_{\parallel}/c)} \approx \gamma \frac{\mathbf{v}_{\perp}}{c},\end{aligned}\quad (37)$$

where γ is the usual relativistic factor for \tilde{S} and the parallel component is with respect to the direction of β . Assuming that deviations from the reference velocity are small (true by definition for a particle beam [1]), then $v_{\parallel}/c \approx \beta$ and $\mathbf{v}_{\perp}/c \ll \beta$ hold, yielding the approximation in Eq. (37). Particle deviations from the reference trajectory are many orders of magnitude smaller than β ; thus, the velocities in S' will be small, even for highly relativistic ($\gamma > 10$) cases.

Taking the reference trajectory to define the \hat{z} direction, the external electromagnetic fields in \tilde{S} are given by the Lorentz transform [simplifying Eq. (11.149) in Ref. [14]]

$$\begin{aligned}\tilde{\mathbf{E}} &= \gamma[(E_x - c\beta B_y)\hat{x} + (E_y + c\beta B_x)\hat{y}] + E_z\hat{z}, \\ \tilde{\mathbf{B}} &= \gamma\left[\left(B_x + \frac{\beta}{c}E_y\right)\hat{x} + \left(B_y - \frac{\beta}{c}E_x\right)\hat{y}\right] + B_z\hat{z}.\end{aligned}\quad (38)$$

For even marginally relativistic beams ($\beta > 0.01$), it follows from Eqs. (38) that $\tilde{\mathbf{E}} \gg \tilde{\mathbf{B}}$ whether the conditions in S are electrostatic, magnetostatic, or a combination of the two. It is assumed that the external boundary conditions are not time dependent or at least are slowly varying. The force in \tilde{S} on the particle beam in the presence of external fields is given by $\tilde{\mathbf{F}} = q(\tilde{\mathbf{E}} + \tilde{\mathbf{v}} \times \tilde{\mathbf{B}})$ [14]. Since intrabeam motion ($\tilde{\mathbf{v}}$) is negligible, the beam only “feels” the effects of $\tilde{\mathbf{E}}$ in Eq. (38). Additionally, the magnetic self-force of the beam is of second order in $\tilde{\mathbf{v}}$ and, thus, can also be neglected.

These simplifying approximations reduce the system of Maxwell equations to the Poisson problem specified by Eq. (27) but with all values given for \tilde{S} . If Dirichlet conditions (potentials) are given in S , then the full Lorentz transformation for the 4-potential should be used instead. Beginning in the lab frame, the boundary conditions are

transformed according to Eq. (38), and the corresponding scalar potential can be obtained by the solution of $\mathbf{E} = -\nabla\phi_e$. This method is valid only if the electromagnetic boundary conditions are slowly varying with respect to the beam velocity. This is not the case for resonant frequency accelerating structures; however, considering the problem in Fourier space, the associated Helmholtz equation can be solved via a modification of this method. Evaluation with PISCS will yield the electrostatic fields in \tilde{S} . In order to propagate the particles in time, the fields must be transformed back to S using the inverse of Eqs. (38) and supplied to the numerical integrator of choice. This final step is not an aspect of the dynamics problem addressed in this work.

V. BENCHMARKING AND ANALYSIS OF THE ALGORITHM

Inherent in scientific computing is the trade-off between accuracy and efficiency [58]. This trade-off is especially pronounced when considering problems whose exact numerical representation is already untenable. Electrostatic problems fall into this category when the boundaries cannot be represented analytically, and the inclusion of particle beams (a classic n-body problem) only exacerbates the difficulty. Thus, the two primary thrusts of this section will be to analyze the run-time characteristics of each system and the accuracy of the computed potentials and fields.

This performance is evaluated by considering three different boundary structures: a perfect electrically conducting spherical shell held at a constant potential, an electric dipole with a constant applied electric field, and a section of beam pipe whose boundary potential is due to an external magnetic quadrupole. The first two are electrostatic problems, while the third is magnetic. The first and third utilize Dirichlet boundary conditions, while the second is specified by Neumann conditions. The performance of the boundaryless Poisson solver has already been well studied [26], so the following tests will evaluate the Laplace solver (with no sources) first, before considering the combined effects of all the components (sources and boundaries). Understanding the complete system is essential for many beam dynamics problems.

First, Sec. VA discusses the accuracy of the discretized boundary representation method used in this work, which gives an indication of the underlying degree of accuracy that can be expected. Efficiency and resulting accuracy for the three structures are presented in Sec. VB. Finally, in Sec. VC, the same quadrupole system is analyzed in the presence of a proton beam to illustrate the impact of the beam-wall interaction for intense beams.

A. Accuracy of the structure representation

The boundary structure is specified to PISCS through a file containing a list of points and their respective surface normals which define the triangularization of the boundary

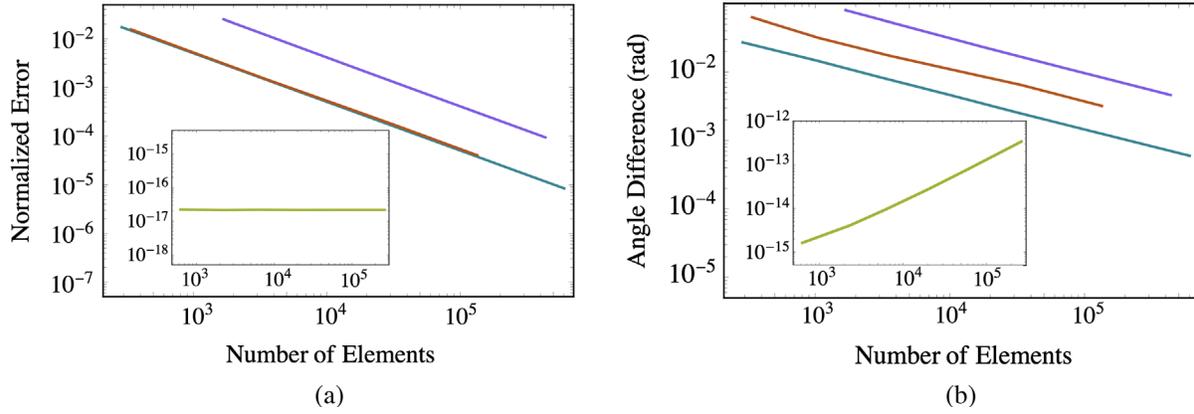


FIG. 2. Plotted is the normalized deviation of the surface nodes from the physical surface (a) and the deviation of normal vector angle (b). Results for the sphere (blue line), cylinder (orange line), rectangular prism (green line), and ellipsoid (purple line).

surface, as described in Sec. IV C. The structure can be generated using any computer-aided design (CAD) software with mesh generation capabilities. All of the models studied in this work were defined and discretized using Gmsh, an open source three-dimensional high-order finite element mesh generator [59]. Gmsh can import standard CAD files but also includes a built-in CAD engine and offers a high degree of control over mesh refinement and adaptivity.

Flat element structures are generated for a sphere, a cylinder, a rectangular prism, and a prolate ellipsoid, with the number of triangular boundary elements ranging from $\mathcal{O}(10^2)$ to $\mathcal{O}(10^5)$. The sphere has a radius of $r = 1 \mu\text{m}$ corresponding to a surface area of $12.6 \mu\text{m}^2$. The cylinder has a radius of $r = 25 \text{ mm}$ and length of $\ell = 0.1 \text{ m}$, corresponding to a surface area of about 196 cm^2 . The rectangular prism has an equal length and width of $\ell = w = 0.9 \text{ mm}$ and a height of $h = 0.3 \text{ mm}$ leading to a surface area of 2.7 mm^2 . The prolate ellipsoid is meant to test the limits of the mesh generation process and, as such, has a large aspect ratio in dimensions. The major radius is $c = 0.5 \mu\text{m}$, and the minor radius is $a = 0.05 \mu\text{m}$, corresponding to a high eccentricity of 0.99 and a surface area of approximately $0.243 \mu\text{m}^2$.

Each boundary element is represented by three quantities: a node at the center of the element (position), the normal vector at that point, and the element area. Each of these quantities enter into the numerical solution of the integral equations as discussed in Sec. IV A.

Figure 2(a) shows the average deviation of the element centers from the physical surface, normalized to unit scale. Clearly, flat boundary elements accurately represent flat surfaces at the order of machine precision, irrespective of the total number of elements. The representation of the sphere and the cylinder exhibit nearly identical behavior in terms of central node accuracy. While the trends for the ellipse are similar, they are offset by roughly an order of magnitude.

Noting the linear behavior of the results in the log-log scale in Fig. 2, a linear fit is performed, and the results are converted to linear scale and given in Table II. The node position exhibits excellent scaling with the number of boundary elements and implies that a normalized error of $\epsilon_c = 1 \times 10^{-6}$ would be achieved with approximately 6 million boundary elements for the sphere and cylinder and 41 million elements for the ellipsoid. The flat elements of the cube map to the physical surface to machine precision.

Next, the orientation of surface normal vectors is considered. The *correct* normal orientation is defined by the analytical representation for each structure at the respective center node discussed in the previous section. Deviations in the angle of computed normals for each structure are plotted in Fig. 2(b). Here, as with the node position, the orientation of the normal vectors is accurate to machine precision for the rectangular prism and scales at similar rates for the other three structures. In this case, the sphere and cylinder results are not identical. The general functional form of the scaling of the difference in angles is given in Table II.

The error in the normal vector orientation converges much more slowly than the error in the node position.

TABLE II. Error scaling in the model geometries, electric potentials, and fields. The terms in parentheses correspond to the results using Neumann boundary conditions.

	Cube	Cylinder	Sphere	Ellipsoid
Node	$e^{-38}n^0$	$e^{-5.3}n^{-1.0}$	$e^{-5.3}n^{-1.0}$	$e^{-3.2}n^{-1.0}$
Normal	$e^{-34}n^{0.9}$	$e^{-3.4}n^{-0.5}$	$e^{-4.2}n^{-0.5}$	$e^{-2.3}n^{-0.5}$
Area	$e^{-37}n^{-0.9}$	$e^{-5.8}n^{-1.3}$	$e^{-4.7}n^{-1.3}$	$e^{-4.2}n^{-0.9}$
φ_e	$e^{-6.0}n^{-1.1}$	$e^{-6.2}n^{-0.4}$	$e^{-3.9}n^{-0.5}$	
E	$(e^{-10}n^{-0.3})$			
	$e^{-6.0}n^{-1.1}$	$e^{-4.9}n^{-0.4}$	$e^{-6.3}n^{-0.8}$	
	$(e^{-8.4}n^{-0.4})$			

This is due, at least in part, to the increased sensitivity of the cross product (used to calculate the normal orientation) to small deviations in the ideal node positions. An additional contributing factor is the fact that the boundary mesh is generated with spatial constraints based on the CAD model. While the normal orientation ought to converge as the number of elements increases, that will always be a secondary effect for curved surfaces caused by improving the resolution with ever-smaller elements.

Interestingly, the error in the normal vector angle increases for the rectangular prism almost linearly ($\sim n^{0.9}$) with an increasing number of elements. This can be understood by recognizing that the errors in triangular element node positions for the rectangular prism are dominated by numerical *noise*. While such errors average out when considering the central node position via averaging [see Fig. 2(a), inset], the same is not true of the normal vector angles obtained via the cross product. Assuming that the difference from true normal is given by a vector with magnitude η , the corresponding angular difference ϑ will be given by

$$\vartheta = \tan^{-1}\left(\frac{\eta}{\|\mathbf{A}\|}\right),$$

where $\|\mathbf{A}\|$ is the area of the triangular element. In this case, η is on the order of machine precision, while the element areas (which scale inversely with number of elements) are many orders of magnitude larger, thus implying a linear scaling of ϑ with number of elements.

The final metric to be considered is the element area, which enters the calculation in the form of the weight in the Nyström method (see Sec. IV A). Average errors in element area are shown in Fig. 3(a), where each is normalized by the theoretical area of a single element if all are equal in size. This error scaling is not as clearly linear as the previous cases, but the results of the least squares linear fit are shown in Table II.

It is also worthwhile to consider the manner in which the total area for each structure deviates from the analytic

value. From Fig. 3(b), it is clear that, as with the node positions and normals, the total area for the rectangular prism is correct to within machine precision. While the sphere and cylinder areas both converge to within 0.25%, the error in total summed area for the ellipse increases with an increasing number of elements, converging to around 2.2%. This is indicative of the ultimate limits in representing a highly curved structure solely using planar elements. The inflection point in the curve for the total area of the cylinder is due to the summation overshooting the analytic value at around $n = 3.5 \times 10^3$.

Despite the limitations addressed in this section, the general trends show acceptable scaling in the quantities that are required for PISCS related to the boundary representation. Additionally, the power law scaling can be used to facilitate the interpretation of the accuracy of the results from the following section.

B. Performance study versus number of flat panel elements

All of the computations presented in this section are performed on the Gaea Cluster maintained by the Center for Research Computing and Data at Northern Illinois University [60]. Gaea is a hybrid CPU-GPU cluster running Red Hat Enterprise Linux 7 with 60 nodes connected via full 1:1 nonblocking Infiniband and ethernet switch connectors. Each node has two Intel Xeon X5650 2.66 GHz 6-core processors with a total of 72 GB of RAM. Parallel executions are performed using the Intel MPI library.

1. Perfect electric conductor sphere

The first structure considered is a $1 \mu\text{m}$ radius perfect electrically conducting spherical shell [Fig. 4(a)]. Dirichlet boundary conditions are utilized, with a uniform applied potential of $\Phi_e = 5 \text{ kV}$. The analytic behavior of the electric potential and fields is given by

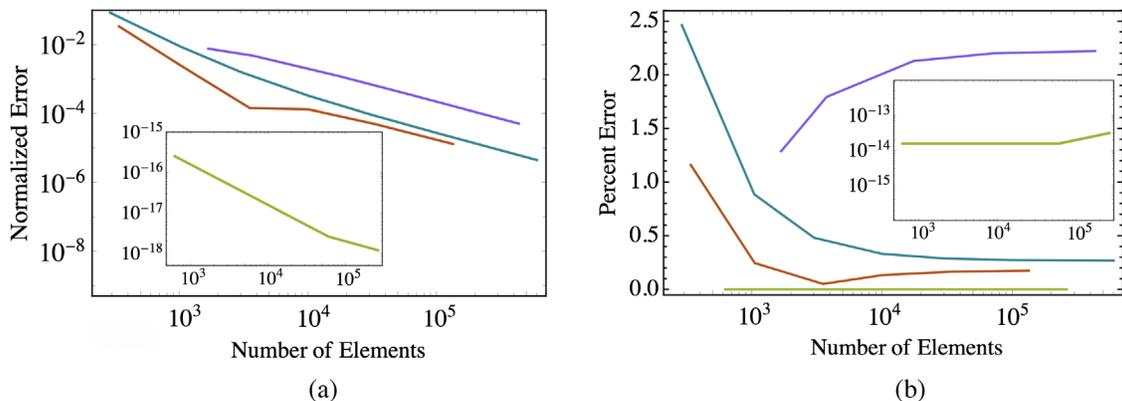


FIG. 3. Plots of the mean normalized error per element (a) and percent error in total surface area (b) for the sphere (blue line), cylinder (orange line), rectangular prism (green line), and ellipsoid (purple line).

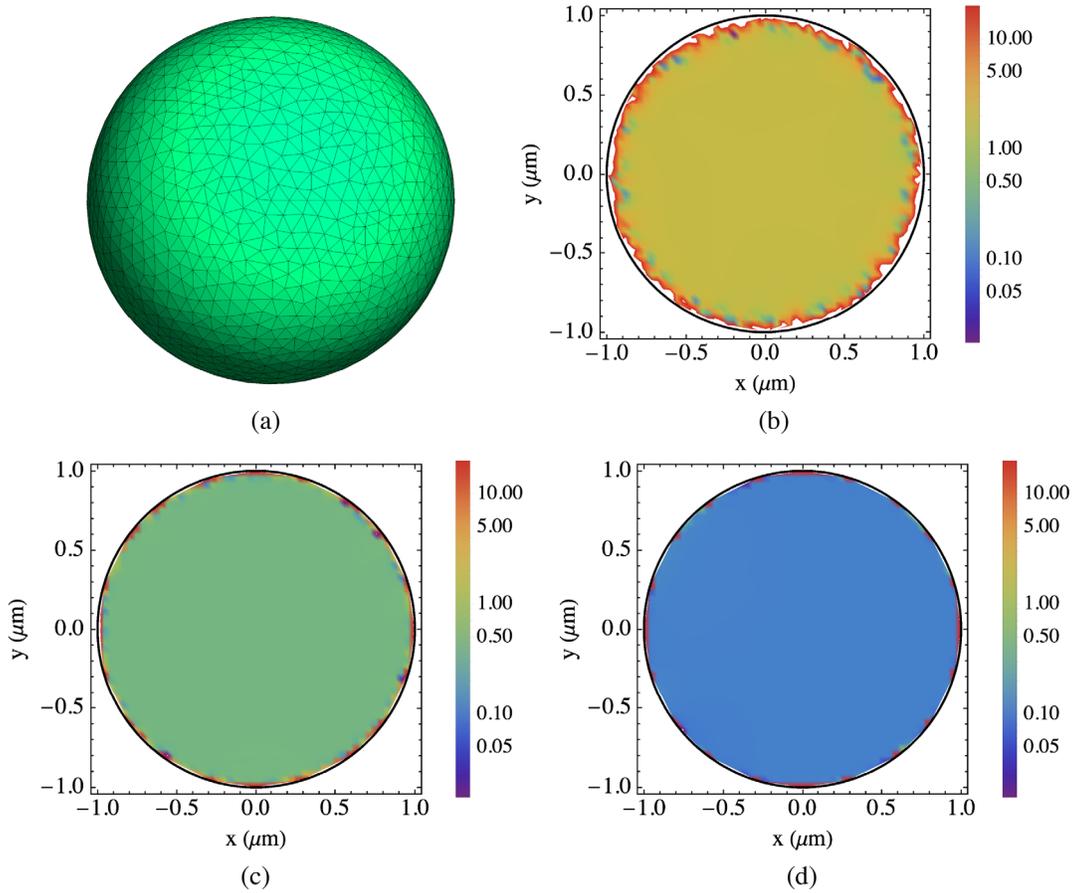


FIG. 4. Visualization of the discretized boundary for the sphere (a) with 3×10^3 elements. Density plots of the percent error in the electric potential in a cross section of the sphere using 1×10^3 (b), 3×10^4 (c), and 6×10^5 (d) boundary elements.

$$\varphi_e(r) = \begin{cases} \Phi_e & r \leq R, \\ \frac{\Phi_e R}{r} & r \geq R \end{cases} \quad \text{and} \quad \mathbf{E}(r) \equiv -\nabla\varphi_e(r) = \begin{cases} 0 & r < R, \\ -\frac{\Phi_e R}{r^2} & r > R, \end{cases} \quad (39)$$

where R is the radius of the spherical shell.

The three final plots in Fig. 4 illustrate two predominant features of the BEM. First, the near-boundary instabilities discussed in Sec. IV A are clearly present in the electric potential (this behavior is analogous in the electric field results), especially when larger boundary elements are used. The instability amounts to an exponential deviation in computed results from the analytic solution when approaching the boundary due to the singularity in the kernel [coming from Eq. (3) and its derivative]. As noted by Ref. [61], the region of instability decreases as a function of decreasing element size. Second, the accuracy of the solution is improved markedly by increasing the number of boundary elements. Methods to address these near-boundary instabilities are being implemented and will be addressed in a subsequent publication covering PISCS' application to a different physical problem ("the study and

design of novel electron sources") where frequent evaluations arbitrarily close to the boundary are required.

The algorithm is evaluated in terms of both serial performance and parallelization efficiency for up to 36 processors. The total computation time, as seen in Fig. 5(a), exhibits fairly consistent behavior for fewer than 1×10^4 boundary elements and then increases rapidly for higher numbers of elements. Doubling the number of processors leads to a speedup factor of almost 2, and increasing to six processors accelerates the evaluation by another factor of 2 for fewer than 1×10^5 boundary elements. The efficiency gains for further increasing the number of processors are not significant, with negligible differences between 18 and 36 processors. The exception is the case with 1×10^5 boundary elements, where the 36 processor run exhibited a tenfold increase decrease in run-time. These results generally show good strong scaling on a single node (up to six processors) with more inconsistencies for three (18 processors) and six (36 processors) nodes. These inconsistencies are not thought to be fundamental but rather due to peculiarities of the available hardware. In particular, inter-node communication seems to be a significant limiting factor. For very large numbers of elements, the saturation of

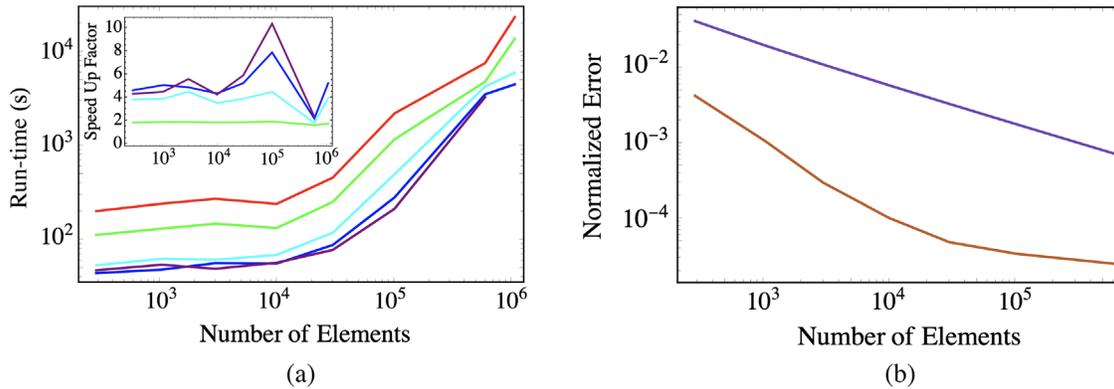


FIG. 5. The left plot shows the computation time for PISCS in serial (red line) and parallel operation with two (green line), six (cyan line), 18 (blue line), and 36 (violet line) processors. The inset shows the speedup factor relative to serial performance. The right plot relates the normalized error in the electric potentials (purple line) and fields (orange line) as a function of the number of boundary elements for the sphere.

the speedup factor is likely due to the increase in memory sharing for the MPI loops. The parallel structure involves infrequent transfers of large arrays; thus, bandwidth (rather than latency) is likely the limiting factor. The shared arrays are minimized to limit the memory scaling per processor; thus, the results must be collated, adding an increasingly large serial component as the number of elements increases. Another fundamental limit to parallel efficiency is the intrinsically serial nature of the iterations within GMRES. The remainder of the trials simply illustrate the serial performance compared to the performance with 18 processors.

An analogous accuracy metric is determined and evaluated for each structure, leading to the results in Fig. 5(b). The error in the computed electric potential and field values far from the surface ($r < 0.6R$) are averaged and normalized. The potentials are normalized by $\Phi_e = 5$ kV, while the fields are normalized by the magnitude of the electric field in the limit as the surface is approached from the exterior, $\|\mathbf{E}(R^+)\| = 5$ GV/m. The scaling of the potential and field error are given in Table II. The electric field error shows linear scaling early but falls off for higher numbers of elements, possibly because the analytic value is zero, and, for larger n , small errors per element begin to dominate, where again n gives the number of boundary elements in thousands. The scaling for the potentials is similar to that of the normal vector angles, suggesting that this factor is likely the dominating source of error for this geometry. Given Dirichlet boundary conditions, the normal derivative enters into both the system matrix calculation (7) and the final evaluation of the potentials (11), leading to a greater impact on the result due to deviations in the normal angles.

2. Electric dipole

An electric dipole is considered next. This is represented in its simplest case by two parallel rectangular plates of

equal size. The plates are square with side length $w = 900$ μm and separation $d = 300$ μm , where the plates are assumed to lie in the horizontal plane. A potential difference is applied between the two plates, leading to a constant gradient, whose analytic representations are given by

$$\varphi_e(z) = \Phi_c + \frac{\Phi_a - \Phi_c}{d} z \quad \text{and} \quad \mathbf{E}(z) = -\frac{\Phi_a - \Phi_c}{d} \hat{z}. \quad (40)$$

Here, the electric potential on the cathode and anode are defined to be $\Phi_c = -3$ kV and $\Phi_a = 1.5$ kV, respectively, leading to a constant field of $\mathbf{E} = -5$ kV/m.

The BEM requires the volume of interest to be fully bounded, so four pseudoboundary walls are introduced far from the region of interest to form a rectangular prism. The boundary conditions on the walls are given by $\phi_e(z)$ in Eq. (40) in the Dirichlet case or by

$$\frac{\partial \varphi_e}{\partial n} = 0 \text{ kV/m}$$

in the case of Neumann boundary conditions. The Neumann conditions on the plates will be

$$\frac{\partial \varphi_e}{\partial n_c} = -5 \text{ kV/m} \quad \text{and} \quad \frac{\partial \varphi_e}{\partial n_a} = 5 \text{ kV/m}$$

for the cathode and anode, respectively.

The results for a slice through the center of the dipole are shown in Fig. 6 for the structure comprising 6×10^4 boundary elements. The Neumann case [Fig. 6(a)] contrasts the Dirichlet case [Fig. 6(b)] in two notable respects. First, both the electric field components and the potential are far smoother than the corresponding Dirichlet case. Second, the magnitude of the error due to the near-boundary instability is 2–3 orders of magnitude lower. Matching

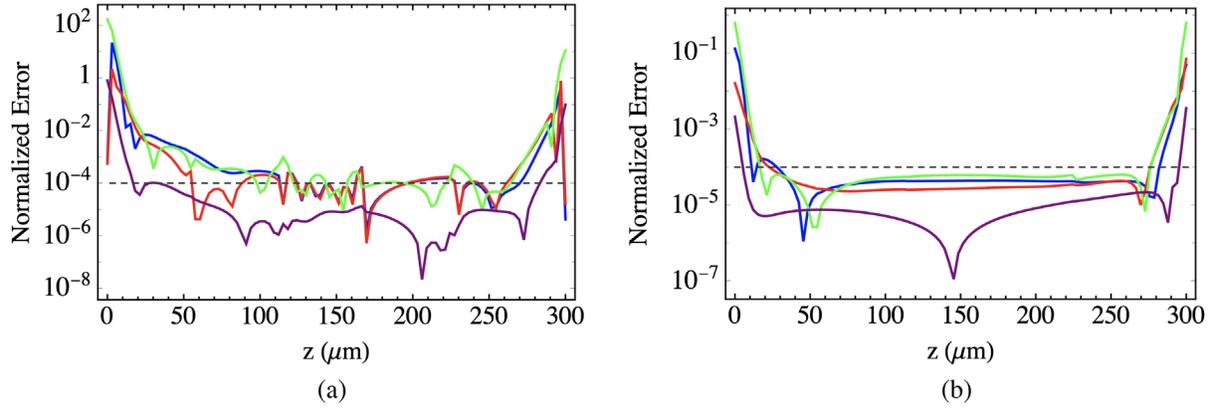


FIG. 6. Normalized error in electric potential (purple line) and field components (blue, red, and green line, respectively) for the electric dipole using 6×10^4 boundary elements with Dirichlet (a) and Neumann (b) conditions. The dashed black line represents an error of 10^{-4} for comparison.

the derivative at the boundaries is advantageous, as integration is functionally a smoothing process.

The unusual dip in the error of the electric potential in Fig. 6(b) is not natural but is rather a result of the matching process required in this case. Specifying Neumann boundary conditions will solve for φ_e only up to an additive constant. Since the near-boundary instabilities make matching the potential at the boundary itself problematic, the final result is instead scaled to fit based on an average from a linear fit in the central region, $z \in [90, 210 \mu\text{m}]$. The downward spike in error marks the point at which the smooth potential function is matched to the correct scale. The fact that the error increases (mostly) uniformly moving away from this point indicates that there is some small error in the slope of φ_e , leading to the one unique crossing point. This is indeed observed when plotting the scaled potential values directly, in contrast with the Dirichlet example, where the potential varies randomly around the analytic values.

Much like for the sphere, Fig. 7(a) shows that running PISCS with 18 processors in parallel leads to a speedup of

roughly an order of magnitude compared to the serial case. The Neumann problem is consistently slower than the Dirichlet problem, due to a more complex system matrix requiring more GMRES iterations. Additionally, a comparison to Fig. 5(a) reveals that the run-time for the rectangular geometry is roughly an order of magnitude longer than for the sphere with a corresponding number of boundary elements.

Considering the accuracy of the electric potentials and fields, Fig. 7(b) shows the resulting errors normalized by the analytic values given by Eq. (39). There is a clear distinction between the two BVP regimes, and the corresponding error scaling is listed in Table II. Notably, the Neumann results are much more accurate at low element numbers, while the Dirichlet results overtake and outperform when considering greater than 3×10^4 elements.

3. Magnetic quadrupole

Having considered cases with constant and linearly varying potentials, the final example, a magnetic quadrupole,

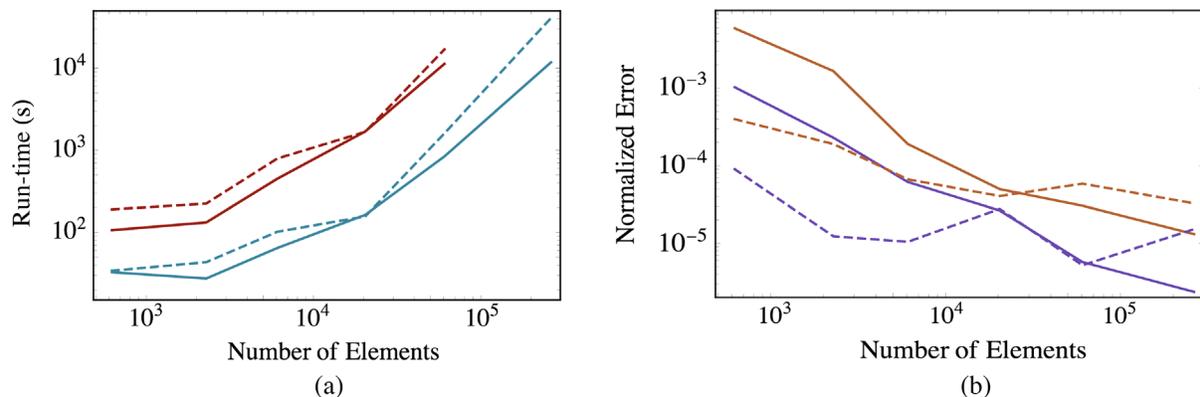


FIG. 7. The left plot shows the computation time for PISCS in serial (red lines) and parallel (blue lines) operation, and the right plot relates the normalized error in the electric potentials (purple lines) and fields (orange lines) as a function of the number of boundary elements for the electric dipole. The solid and dashed lines correspond to the results when Dirichlet and Neumann boundary conditions, respectively, are specified.

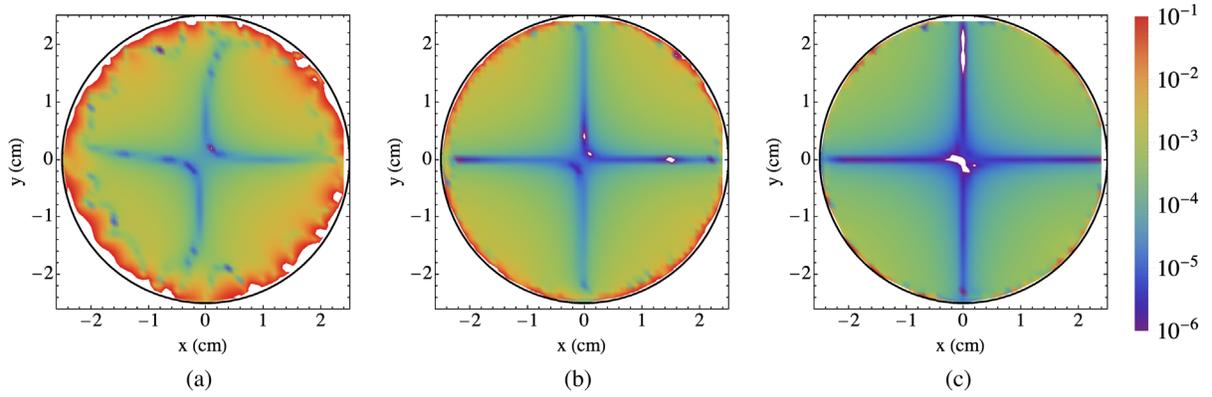


FIG. 8. Log-scale density plots of the absolute value of the difference in the computed electric potential (Tmm) from Eq. (41) in a cross section at the center of the beam pipe using 1×10^3 (a), 1×10^4 (b), and 1.3×10^5 (c) boundary elements.

exhibits a quadratic potential. Since the magnetic fields within a quadrupole can be described as the negative gradient of a scalar potential, it can be mathematically represented as a homogeneous (sourceless) Poisson problem and solved using the methodology developed in this work. This problem has the simple analytic representation given by

$$\begin{aligned} \varphi_m(x, y) &= B'xy, \\ \mathbf{B}(x, y) &= -B'(y\hat{x} + x\hat{y}), \end{aligned} \quad (41)$$

where B' is the magnetic gradient and x and y represent the horizontal and vertical components, respectively, of the transverse plane. For this problem, a section of a cylindrical beam pipe is used for the boundary with a radius of $r = 25$ mm and length of $\ell = 0.1$ m, and the gradient is set to $B' = 1$ T/m. Similar to the previous case, pseudo-boundary caps are placed on either end of the cylindrical shell with boundary conditions given by Eq. (41), ensuring that the region is fully bounded.

Clearly, from Fig. 8(a), higher quadrature errors (corresponding to fewer boundary elements) lead not only to more significant boundary errors, but also to an interior error distribution that is spatially dependent, in contrast with the previous two examples. Structures with 1×10^4 boundary elements or greater exhibit the correct spatial potential profile, with deviations from the analytic results simply scaling as a function of the magnitude of the potential.

The trends in evaluation time for the quadrupole also differ somewhat from the previous cases, as seen in Fig. 9(a). Overall, the evaluation time was of the same magnitude as for the sphere. However, for structures with fewer than 5×10^3 elements, the increased efficiency from parallel execution diminishes from an order of magnitude to less than a factor of 2.

The error in Fig. 9(b) is normalized by the maximum surface potential $\max(\varphi_m|_{\Gamma}) = 0.44$ Tmm and surface field magnitude $\max(\|\mathbf{B}|_{\Gamma}\|) = 25$ mT. The error scaling for the quadrupole, presented in Table II, resembles that of the sphere potentials, indicating again that the curved cylindrical surface leads to normal angle deviations which

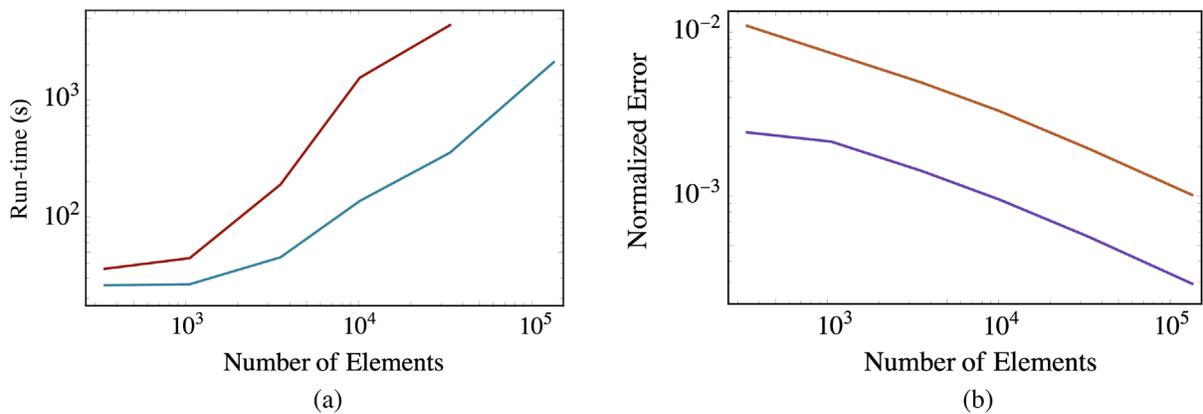


FIG. 9. The left plot shows the computation time for PISCS in serial (red line) and parallel (blue line) operation, and the right plot relates the normalized error in the magnetic potentials (purple line) and fields (orange line) as a function of the number of boundary elements for the magnetic quadrupole.

TABLE III. Proposed IOTA proton beam parameters.

	Variable	Value
Momentum	p	70 MeV/c
Velocity	β	0.075
Particles	N	9×10^{10}
Current	I	8 mA
Emittance (equil.)	$\epsilon_{x,y}$	$0.3 \mu\text{m}$
Bunch length (FWHM)	L	3.6 m

introduce errors that dominate the scaling of the overall algorithm. However, in this case, the exponent of -0.4 is even lower than -0.5 for the sphere, which implies that the quadratically varying boundary conditions impose additional challenges on the method than the simpler constant boundary condition case.

C. Beam-enclosure interaction

Given the push to higher-intensity beams, it is of interest to consider a practical example. Planned proton runs at IOTA are likely to encounter nontrivial manifestations of collective effects. Proposed beam parameters for proton beams in IOTA compiled from Refs. [6,62] are presented in Table III.

The boundary conditions are the same as those in Sec. VB 3; however, the system must be transformed to the beam frame as discussed in Sec. IV C 1. Assuming that the protons are traveling in the \hat{z} direction, this results in an electrostatic system with the analytic representation given by

$$\begin{aligned} \varphi_e(x, y) &= \frac{1}{2} E' (x^2 - y^2), \\ \mathbf{E}(x, y) &= -E' (x\hat{x} - y\hat{y}), \end{aligned} \quad (42)$$

with the electric field gradient $E' \equiv c\gamma\beta B' = 22.55 \text{ MV/m}^2$, where γ is the usual relativistic factor. Dirichlet boundary conditions are used for this study.

The beam distribution is assumed to be Gaussian in each component, with $\sigma_{x,y} = 0.3 \mu\text{m}$ and $\sigma_z = L/\sqrt{8 \ln(2)} = 1.53 \text{ m}$. The design bunch length is so much greater than the magnet length that it is more interesting to consider a short bunch with a comparable longitudinal charge density. The peak density of the beam is centered in the quadrupole, and the longitudinal component of the beam has $\sigma_z = 1 \text{ cm}$. This bunch length is short enough to ensure that it remains a sufficient distance from the pseudoboundary caps. Assuming a Gaussian distribution, the total number of protons in this bunch will be 1×10^9 . This is represented for computational efficiency by 1×10^4 macroparticles each with an effective charge of $1 \times 10^5 e$.

The structure has been designed to showcase the capacity of PISCS. It is based around a 2.5 cm radius cylindrical pipe and includes both a radial discontinuity and a small

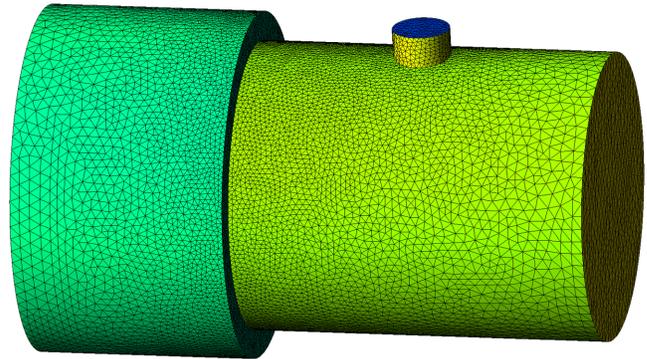


FIG. 10. The beam pipe discretized boundary with two segments of differing radius and a small circular opening. The boundary mesh scale is adaptively reduced near the geometric features to improve accuracy.

opening, as shown in Fig. 10. The structure generated has 2.7×10^4 boundary elements, and the resulting electric potentials and fields were computed for a cross section of the beam pipe at its center and longitudinally along the length of the structure. Figure 11 illustrates both the transverse and longitudinal potential distribution due to the presence of the proton beam, with charge scaled by a factor of 10 to emphasize the modulation. The most substantial impact, by approximately 10%, occurs in the center of the region near the beam itself. The presence of the protons leads to a defocusing effect (Coulomb expansion or heating) which reduces the focusing power of the quadrupole (in the horizontal plane, while increasing the defocusing effect in the vertical plane). This becomes especially clear when considering the distribution of the electric fields, plotted in Fig. 12. When the high current beam is present, the overall focusing of the fields is diminished, with two unstable islands forming in the horizontal plane, and the defocusing of the fields is noticeably amplified.

Finally, it is interesting to note the impact of the proton beam on the boundary conditions themselves. This modification is calculated in the first pass of the FMM and written to a temporary file. Figure 13 shows the difference in boundary conditions due to the presence of the proton beam (plotted in red) as a function of the length of the beam pipe. These values are normalized by the maximum boundary condition value in the absence of a beam. The effect on the boundary near its center is nearly 1%, with many of the quadrature node values being altered more substantially. For comparison, it was noted that increasing the bunch charge by orders of magnitude (blue and green, respectively) led to a comparable increase in the magnitude of the boundary modifications. Figure 13 represents the minimum boundary potential modification, as an off-axis beam leads to higher deviations and will introduce dipolar and higher-order modes in the external fields.

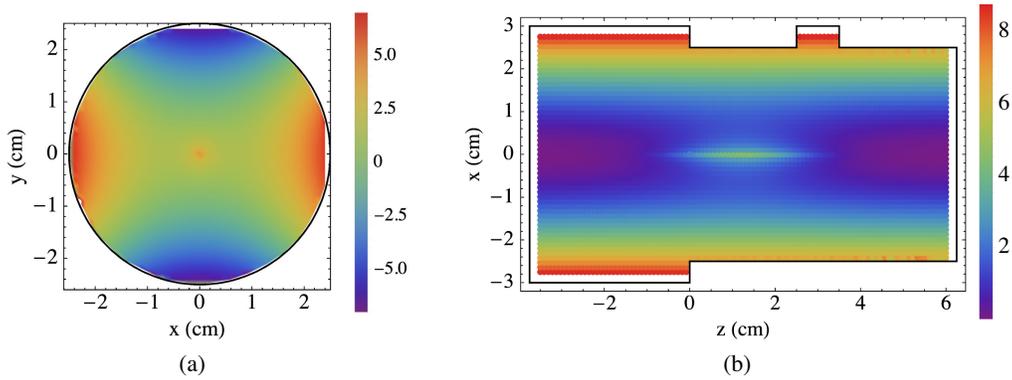


FIG. 11. Plots of the electrostatic potential (kV) in a cross section of the narrow beam pipe (a) and along the longitudinal shot of the beam pipe (b) with the Gaussian beam for 10 times the number of particles.

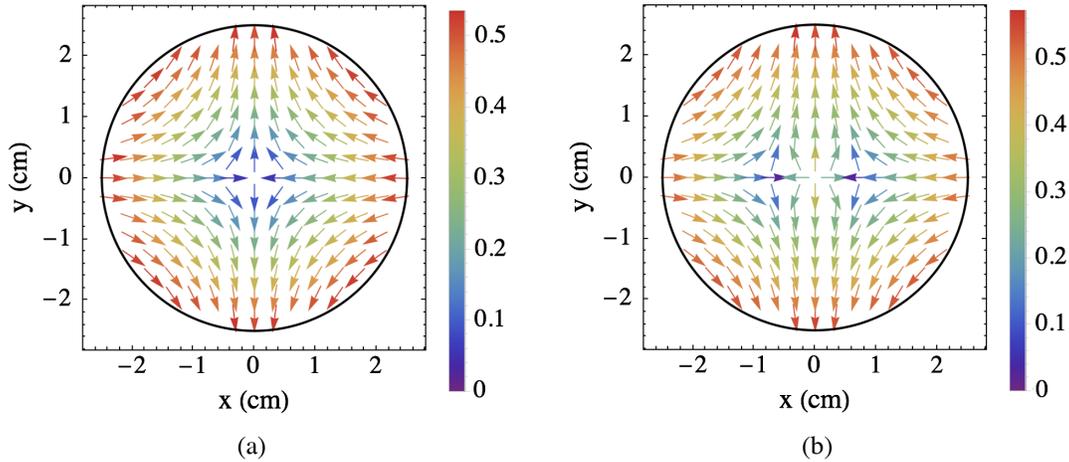


FIG. 12. Vector plots of the transverse electric field (MV/m) in a cross section of the beam pipe without the presence of a beam (a) and a plot of the percent difference in that potential when the proton beam (again scaled by 10) is present (b).

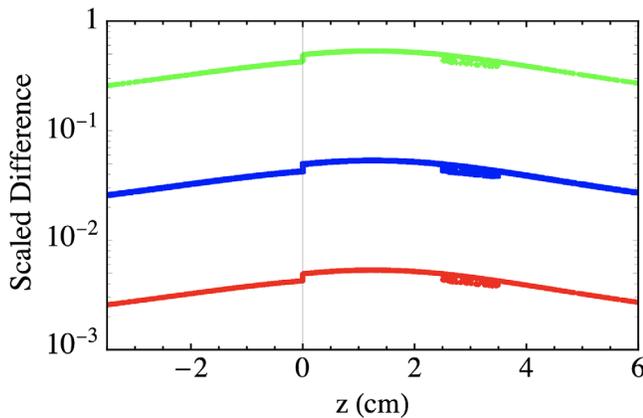


FIG. 13. Modifications in the boundary conditions (scaled by the maximum boundary potential) due to the presence of the proton beam (red line). Additional results shown for 10 (blue line) and 100 (green line) times the number of particles.

VI. CONCLUSION

Driven by continuously increasing intensity needs for charged particle beams and their applications, the fast multipole-accelerated boundary element method enabled by differential algebraic methods, PISCS, was developed. PISCS solves the Poisson equation with boundary conditions in the presence of millions of charged particles. For the first time, the BEM and FMM are combined in the DA framework. PISCS has been shown to handle hundreds of thousands of boundary elements with ease. With a high memory system, PISCS may handle millions of elements with intense beams. PISCS is parallelized and has superlinear weak scaling as a function of the number of boundary elements for at least 36 MPI processes and good strong scaling on a single node (up to 12 processors) using a high-performance computing cluster. It is likely that the strong scaling would extend to more processors on a high performance computing (HPC) cluster with optimized internode communication. The N -body FMM

has already been shown to exhibit good scaling as a function of the number of particles.

The accuracy of PISCS has been verified using various geometries, and the resulting error scales acceptably with an increasing number of elements. This errors scales roughly as $m^{-0.5}$ for the curved enclosures. Based on structure tests, this scaling appears to be dominated by errors in the normal vector angle for curved geometries. When these errors are limited (electric dipole test), the error in the Nyström BEM is seen to scale as m^{-1} . Many advancements have been implemented and optimized to ensure that PISCS is not only accurate but also efficient in terms of run-time. Using the GMRES optimized parameters, the run-time is relatively consistent for few elements, with overall scaling as $m \log m$ or better up to $\mathcal{O}(10^5)$ elements for each of the examples shown in Sec. V B. Since PISCS is dominated by GMRES, which is well known to have $m \log m$ scaling, this behavior is expected. PISCS scales with $\mathcal{O}(N)$ for number of sources, as there is very little overhead beyond the base FMM algorithm (in contrast to the number of elements). There is a consistent speedup of roughly an order of magnitude when running PISCS in parallel with 18 processors.

This method is the first of its kind in beam physics which enables the accurate and efficient treatment of large-scale intense beam dynamics with pairwise forces in realistic bounded environments. The only limitation on the bounded environment is that the electromagnetic boundary conditions are slowly varying. This method cannot directly incorporate electrodynamic effects that fail to satisfy the quasistatic assumption in the beam frame, which includes the mechanisms of certain known collective instabilities. The precision demanded by the next generation of high-energy physics experiments necessitates the inclusion of particle-to-particle effects in the presence of realistic enclosures, including edges and corners.

Further advancements are in progress and will be the subject of a future work. These include the implementation of a high-order interpolation and quadrature scheme for surface integrals over curved elements, to further improve the convergence rate of the results, and an advanced numerical method to mitigate and even eliminate the effects of near-boundary instabilities. The first enables substantially more accurate representations of curved surfaces while utilizing fewer surface elements in the numerical discretization. This decreases the computation time by reducing the iterations and matrix vector product evaluations, since the size of the linear system is proportional to the square of the number of boundary nodes. The second becomes relevant when considering electron sources where the precise potential distribution at the cathode boundary contributes to the barrier potential. The study and design of novel electron sources is a burgeoning field which promises many potential applications once the boundary instabilities are resolved.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Department of Energy, Office of High Energy Physics, under Contract No. DE-SC0011831, and by the National Science Foundation (NSF) under Grant No. PHY-1535401 with Northern Illinois University. This work used resources of the Center for Research Computing and Data at Northern Illinois University.

APPENDIX: REGULARIZATION OF THE BOUNDARY INTEGRAL EQUATION

The boundary integral equation (5) is, in fact, valid only for the interior region $\mathbf{x} \in \Omega \setminus \Gamma$. Considering the functional form of G (3), the integral in Eq. (5) contains both strongly and weakly singular terms, respectively, for $\mathbf{x} \in \Gamma$. In both cases, the approach will be to extend a small region of the boundary about the singular point and consider the limits as that region is contracted. For simplicity of integration, let \mathcal{B}_ξ be a ball centered at \mathbf{x} with radius ξ , and define Γ_B as the subset of the surface of \mathcal{B}_ξ that is exterior to the boundary Γ . Define $\Gamma_\xi \subset \Gamma$ to be the surface including all points $\mathbf{z} \in \Gamma \cap \mathcal{B}_\xi$.

Thus, \mathbf{x} is in the interior of the extended volume $\Omega \cup \mathcal{B}_\xi$, and the kernels of Eq. (5) are finite everywhere. The form of Eq. (5) for $\mathbf{x} \in \Gamma$ is then given by evaluating the integrals over the surface of the extended region $\Omega \cup \mathcal{B}_\xi$ and taking the limit $\xi \rightarrow 0$. The weakly singular kernel is therefore expressed as the limit

$$\int_{\Gamma} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) = \lim_{\xi \rightarrow 0} \left[\int_{\Gamma \setminus \Gamma_\xi} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) + \int_{\Gamma_B} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) \right], \quad (\text{A1})$$

where neither integrand is explicitly singular. The Green's function is constant over \mathcal{B}_ξ by construction, so the only challenge is the behavior of the boundary condition function h . For a smooth surface, the gradient over the boundary $\nabla_\Gamma h$ will both exist and be bounded, so that h satisfies the Hölder condition

$$\|h(\mathbf{x}) - h(\mathbf{y})\| < \mathcal{C}_h \|\mathbf{x} - \mathbf{y}\|^{\lambda_h}, \quad (\text{A2})$$

with constants $\mathcal{C}_h > 0$ and $0 < \lambda_h \leq 1$. Given Eq. (A2), the magnitude of the second integral of Eq. (A1) is bounded by

$$\left\| \int_{\Gamma_B} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) \right\| < \left(1 - \frac{\Lambda_i(\mathbf{x})}{4\pi} \right) (\mathcal{C}_h \xi^{\lambda_h+1} + h(\mathbf{x}) \xi),$$

where $\Lambda_i(\mathbf{x})$ gives the interior solid angle subtended by the surface at \mathbf{x} . Thus, the weak singularity is removable and Eq. (A1) converges to

$$\int_{\Gamma} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}) = \lim_{\xi \rightarrow 0} \int_{\Gamma \setminus \Gamma_{\xi}} G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) d\Gamma(\mathbf{y}). \quad (\text{A3})$$

A similar approach can be taken for the strongly singular integral; however, that singularity is not removable and must instead be evaluated via its Cauchy principle value. Considering again the extended spherical surface about the singularity yields the limit

$$\int_{\Gamma} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) = \lim_{\xi \rightarrow 0} \left[\int_{\Gamma \setminus \Gamma_{\xi}} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) + \int_{\Gamma_B} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) \right], \quad (\text{A4})$$

where the precise form of the kernel is not as trivial as the previous example. The normal derivative of G can alternatively be written using $\mathbf{r} \equiv (\mathbf{y} - \mathbf{x})$, $r \equiv |\mathbf{y} - \mathbf{x}|$, and $\hat{\mathbf{r}} \equiv \mathbf{r}/r$ as

$$\frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \equiv \mathbf{n}(\mathbf{y}) \cdot \nabla G(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{n}(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3}.$$

The integral over the spherical surface Γ_B is evaluated, noting that $\mathbf{n}(\mathbf{y}) = \hat{\mathbf{r}}$ and $r = \xi$ for all $\mathbf{y} \in \mathcal{B}_{\xi}$. Assuming that g satisfies the Hölder condition analogous to Eq. (A2), this integral is bounded by

$$\left\| \int_{\mathcal{B}_{\xi}} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) \right\| < \left(1 - \frac{\Lambda_i(\mathbf{x})}{4\pi} \right) (\xi^{\lambda_g} C_g + g(\mathbf{x})).$$

The inequality is, in fact, solely due to the first term, which is vanishing. The principle value itself will be negative, since $\mathbf{n}(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})$ is negative over \mathcal{B}_{ξ} . Evaluating the limit from Eq. (A4) results in the relation

$$\int_{\Gamma} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) = \lim_{\xi \rightarrow 0} \left[\int_{\Gamma \setminus \Gamma_{\xi}} g(\mathbf{y}) \frac{\partial G}{\partial n_y} d\Gamma(\mathbf{y}) - \left(1 - \frac{\Lambda_i(\mathbf{x})}{4\pi} \right) g(\mathbf{x}) \right]. \quad (\text{A5})$$

Taking both singularity limits into account, the fully generalized representation formula becomes

$$\int_{\Gamma} \left[G(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) - g(\mathbf{y}) \frac{\partial G}{\partial n_y}(\mathbf{x}, \mathbf{y}) \right] d\Gamma(\mathbf{y}) = \begin{cases} \psi(\mathbf{x}), & \mathbf{x} \in \Omega \setminus \Gamma, \\ \frac{\Lambda_i(\mathbf{x})}{4\pi} g(\mathbf{x}), & \mathbf{x} \in \Gamma, \end{cases} \quad (\text{A6})$$

where the integrals are taken in the limiting values excluding the singularity for the second case ($\mathbf{x} \in \Gamma$). For this work, it is assumed that Γ is smooth (or at least piecewise smooth), and, thus, the interior solid angle is

$\Lambda_i(\mathbf{x}) = 2\pi$ for $\mathbf{x} \in \Gamma$. This assumption does not represent a fundamental limitation in this derivation but is made for simplicity. The boundary discretization method utilized in PISCS ensures that Γ is piecewise smooth.

-
- [1] M. Reiser, *Theory and Design of Charged Particle Beams*, 2nd ed. (Wiley, New York, 2008).
 - [2] A. Chao, Beam dynamics with high intensity, 2015 (unpublished).
 - [3] K. Y. Ng, Physics of intensity dependent beam instabilities, in *Proceedings of the U.S. Particle Accelerator School (USPAS 2002), Long Beach, California, 2002*, Report No. FERMILAB-FN-0713.
 - [4] L. J. Laslett, On intensity limitations imposed by transverse space-charge effects in circular particle accelerators, eConf **C630610**, 324 (1963).
 - [5] A. Chao, in *Physics of Collective Beam Instabilities in High Energy Accelerators*, edited by M. Month, Wiley Series in Beam Physics and Accelerator Technology (John Wiley & Sons, New York, 1993).
 - [6] 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, *J. Instrum.* **12**, T03002 (2017).
 - [7] A. Valishev, S. Nagaitsev, D. N. Shatilov, and V. V. Danilov, Beam physics of integrable optics test accelerator at Fermilab, in *Proceedings of the 3rd International Particle Accelerator Conference, New Orleans, LA, 2012* (IEEE, Piscataway, NJ, 2012), TUPPC090, pp. 1371–1373.
 - [8] V. Shiltsev and M. Chung, Space-charge compensation experiments at IOTA ring, *AIP Conf. Proc.* **1777**, 100006 (2016).
 - [9] R. Kishek, B. Beaudoin, S. Bernal, M. Cornacchia, D. Feldman, R. Fiorito, I. Haber, T. Koeth, Y. Mo, P. O’Shea, K. P. Rezaei, D. Sutter, and H. Zhang, The University of Maryland electron ring program, *Nucl. Instrum. Methods Phys. Res., Sect. A* **733**, 233 (2014).
 - [10] 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 Conf. Proc.* **1777** (2016).
 - [11] M. Reiser *et al.*, The Maryland electron ring for investigating space-charge dominated beams in a circular FODO system, in *Proceedings of the 18th Particle Accelerator Conference, New York, 1999* (IEEE, New York, 1999), Cat. No. 99CH36366, Vol. 1, pp. 234–236.
 - [12] S. Bernal *et al.*, Beam transport experiments over half-turn at the University of Maryland electron ring (UMER), in *Proceedings of the 2003 Particle Accelerator Conference, Portland, OR* (IEEE, New York, 2003), Vol. 1, pp. 426–428.
 - [13] 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 Conf. Proc.* **1812**, 110006 (2017).
- [14] J. D. Jackson, *Classical Electrodynamics*, 3rd ed. (John Wiley & Sons, New York, 1999).
- [15] Y. Liu, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* (Cambridge University Press, Cambridge, England, 2009).
- [16] S. A. Sauter and C. Schwab, *Boundary Element Methods*, Springer Series in Computational Mathematics (Springer, New York, 2011), Vol. 39.
- [17] M. Ferrario, M. Migliorati, and L. Palumbo, Space charge effects, in *CERN Accelerator School: Advanced Accelerator Physics Course* (2014), <https://doi.org/10.5170/CERN-2014-009.331>.
- [18] VSim User Guide, Technical Report No. 10.0.1rc1-r2685, Tech-X Corporation, 2019.
- [19] J. Barnes and P. Hut, A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm, *Nature (London)* **324**, 446 (1986).
- [20] L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, 325 (1987).
- [21] J. A. Lupo, Z. Wang, A. M. McKenney, R. Pachter, and W. Mattson, A large scale molecular dynamics simulation code using the fast multipole algorithm (FMD): Performance and application, *J. Mol. Graphics Modell.* **21**, 89 (2002).
- [22] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Linear scaling density functional calculations via the continuous fast multipole method, *Chem. Phys. Lett.* **253**, 268 (1996).
- [23] M. Kawata and M. Mikami, Computationally efficient canonical molecular dynamics simulations by using a multiple timestep integrator algorithm combined with the particle mesh Ewald method and with the fast multipole method, *J. Comput. Chem.* **21**, 201 (2000).
- [24] W. A. Dehnen, A fast multipole method for stellar dynamics, *Comput. Astrophys.* **1**, 1 (2014).
- [25] H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, A wideband fast multipole method for the Helmholtz equation in three dimensions, *J. Comput. Phys.* **216**, 300 (2006).
- [26] S. Abeyratne, A. Gee, and B. Erdelyi, An adaptive fast multipole method in cartesian basis, enabled by algorithmic differentiation, *Commun. Nonlinear Sci. Numer. Simul.* **72**, 294 (2019).
- [27] Introduction to COMSOL Multiphysics, COMSOL, 2019.
- [28] J. Petillo, E. Nelson, J. DeFord, N. Dionne, and B. Levush, Recent developments to the MICHELLE 2-D/3-D electron gun and collector modeling code, *IEEE Trans. Electron Devices* **52**, 742 (2005).
- [29] A. Friedman, R. H. Cohen, D. P. Grote, S. M. Lund, W. M. Sharp, J.-L. Vay, I. Haber, and R. A. Kishek, Computational methods in the Warp code framework for kinetic simulations of particle beams and plasmas, *IEEE Trans. Plasma Sci.* **42**, 1321 (2014).
- [30] P. Hunter and A. Pullan, FEM/BEM notes, 2003 (unpublished).
- [31] K. Hayami, Variable transformations for nearly singular integrals in the boundary element method, *Publ. RIMS Kyoto Univ.* **41**, 821 (2005).
- [32] M. Harmel, M. P. Rajsiki, and R. A. Sauer, Desingularization in boundary element analysis of three-dimensional Stokes flow, *Proc. Appl. Math. Mech.* **18**, e201800479 (2018).
- [33] M. Berz, in *Modern Map Methods in Particle Beam Physics*, edited by P. W. Hawkes, Advances in Imaging and Electron Physics Vol. 108 (Academic, New York, 1999).
- [34] K. Makino and M. Berz, COSY INFINITY version 9, *Nucl. Instrum. Methods Phys. Res., Sect. A* **558**, 346 (2006).
- [35] A. Gee, Intense Beam Dynamics in Arbitrary Structures, Ph.D. dissertation, Northern Illinois University, 2018.
- [36] S. Rjasanow and O. Steinbach, *The Fast Solution of Boundary Integral Equations* (Springer, New York, 2007).
- [37] K. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind* (Cambridge University Press, Cambridge, England, 1997).
- [38] K. Atkinson, The solution of non-unique linear integral equations, *Numer. Math.* **10**, 117 (1967).
- [39] H. Antes, A short course on boundary element methods, 2010.
- [40] N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, New York, 2004).
- [41] O. Coulaud, P. Fortin, and J. Roman, High performance BLAS formulation of the multipole-to-local operator in the fast multipole method, *J. Comput. Phys.* **227**, 1836 (2008).
- [42] J. Kurzak and B. M. Pettitt, Fast multipole methods for particle dynamics, *Mol. Simul.* **32**, 775 (2006).
- [43] A. Rathsfeld, Quadrature methods for 2d and 3d problems, *J. Comput. Appl. Math.* **125**, 439 (2000).
- [44] X. Qin, J. Zhang, G. Xie, F. Zhou, and G. Li, A general algorithm for the numerical evaluation of nearly singular integrals on 3d boundary element, *J. Comput. Appl. Math.* **235**, 4174 (2011).
- [45] J. T. Beale and M. C. Lai, A method for computing nearly singular integrals, *SIAM J. Sci. Comput.* **238**, 1902 (2001).
- [46] K. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. (John Wiley & Sons, New York, 1989).
- [47] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
- [48] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995).
- [49] X. Sun and N. P. Pitsianis, A matrix version of the fast multipole method, *SIAM Rev.* **43**, 289 (2001).
- [50] R. Yokota, H. Ibeid, and D. Keyes, Fast multipole method as a matrix-free hierarchical low-rank approximation, in *Eigenvalue Problems: Algorithms, Software and Applications in Petascale Computing*, edited by T. Sakurai, S.-L. Zhang, T. Imamura, Y. Yamamoto, Y. Kuramashi, and T. Hoshi (Springer International Publishing, Cham, 2017), pp. 267–286.
- [51] V. Simoncini and D. B. Szyld, Theory of inexact Krylov subspace methods and applications to scientific computing, *SIAM J. Sci. Comput.* **25**, 454 (2003).
- [52] A. Bouras and V. Frayssé, Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy, *SIAM J. Matrix Anal. Appl.* **26**, 660 (2005).

- [53] X. Du and D. B. Szyld, Inexact GMRES for singular linear systems, *BIT* **48**, 511 (2008).
- [54] T. Wang, S. K. Layton, and L. A. Barba, Inexact Krylov iterations and relaxation strategies with fast-multipole boundary element method, [arXiv:1506.05957](https://arxiv.org/abs/1506.05957).
- [55] M. Berz and K. Makino, *COSY INFINITY Version 10.0 Programmer's Manual* (Michigan State University, East Lansing, 2017).
- [56] Beam physics code repository, <https://www.niu.edu/beam-physics-code/projects/index.shtml>.
- [57] P. D. S. Abeyratne, New Computational Approaches to the N-Body Problem with Applications to the Electron Cooling of Heavy Ion Beams, Ph.D. dissertation, Northern Illinois University, 2016.
- [58] R. Corless and N. Fillion, *A Graduate Introduction to Numerical Methods* (Springer, New York, 2013).
- [59] C. Geuzaine and J.-F. Remacle, Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, *Int. J. Numer. Methods Eng.* **79**, 1309 (2009).
- [60] Center for research and computing and data, <https://www.niu.edu/crcd/index.shtml>.
- [61] A. Klöckner, A. Barnett, L. Greengard, and M. O'Neil, Quadrature by expansion: A new method for the evaluation of layer potentials, *J. Comput. Phys.* **252**, 332 (2013).
- [62] G. Kafka, Lattice design of the integrable optics test accelerator and optical stochastic cooling experiment at Fermilab, Ph.D. thesis, Illinois Institute of Technology, 2015.