

Two-and-a-half dimensional symplectic space-charge solver

Ji Qiang 

Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA



(Received 22 July 2025; accepted 14 October 2025; published 17 November 2025)

The nonlinear space-charge effect plays a significant role in high-intensity accelerators and has been extensively studied using multiparticle tracking methods. In this paper, we present a novel 2.5-dimensional symplectic space-charge solver specifically designed for long beam bunches. We begin by detailing its application to a transverse Gaussian density distribution under open boundary conditions in a straight system, where a semianalytical expression is derived. We then demonstrate the solver's adaptation to arbitrary distributions in open space, as well as within rectangular and round conducting pipes. Finally, we discuss the extension of this solver to circular accelerator systems. This study shows that the fast 2.5-dimensional solver can be a good approximation to the fully three-dimensional solver for long bunches in large circular accelerators.

DOI: [10.1103/npx1-nclt](https://doi.org/10.1103/npx1-nclt)

I. INTRODUCTION

The nonlinear space-charge effect arising from Coulomb interactions among charged particles plays a crucial role in high-intensity, high-brightness accelerators. It can lead to beam emittance growth, halo formation, and even particle losses along the accelerator. To study these space-charge effects, multiparticle tracking is employed to dynamically follow charged particles through the accelerator. In the accelerator community, most multiparticle tracking codes employ a particle-in-cell (PIC) approach to incorporate the space-charge effects self-consistently in simulations [1–10].

The self-consistent PIC method requires solving the three-dimensional Poisson equation in the beam frame at each time step, which is computationally intensive even with parallel computing resources. For a long bunch (in the beam frame) with a smooth density distribution, the variation of the electric potential along the longitudinal direction is much smaller than that along the transverse dimensions. By neglecting the longitudinal variation, the three-dimensional Poisson equation reduces to a two-dimensional one. To account for longitudinal variations in the charge density distribution, the electric potential from the two-dimensional Poisson equation is weighted longitudinally for different longitudinal slices. This approach, known as the two-and-a-half dimensional (2.5D) space-charge model in the

accelerator community [11–15], does not impart a kick to the longitudinal momentum during simulation, leading to a violation of the symplectic condition in multiparticle tracking.

The gridless symplectic three-dimensional particle tracking model was previously proposed for self-consistent space-charge simulations [16]. More recently, a symplectic particle-in-cell model was developed and applied to two-dimensional coasting beam simulations [17]. In this paper, we propose a two-and-a-half dimensional space-charge solver that incorporates not only the longitudinal variation of the transverse momentum kick but also the longitudinal momentum kick. By construction, this solver automatically satisfies the symplectic condition and is well-suited for long-term simulations. The longitudinal variation of the electric potential is treated as a perturbation in the three-dimensional Poisson equation, and the solution can be further refined to include this variation if necessary.

The organization of this paper is as follows: after the Introduction, we discuss the symplectic particle tracking model in Sec. II. Section III presents the two-and-a-half dimensional space-charge solver for a transverse Gaussian density distribution (Sec. III A) and for arbitrary density distributions in a straight accelerator system (Sec. III B). In Sec. IV, we extend the model to a circular accelerator system. Conclusions are drawn in Sec. V.

II. SYMPLECTIC PARTICLE TRACKING MODEL

The charged particle dynamics inside a particle accelerator is governed by Hamilton's equations as

$$\frac{d\mathbf{r}}{ds} = \frac{\partial H}{\partial \mathbf{p}}, \quad (1)$$

*Contact author: jqiang@lbl.gov

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

$$\frac{d\mathbf{p}}{ds} = -\frac{\partial H}{\partial \mathbf{r}}, \quad (2)$$

where $H(\mathbf{r}, \mathbf{p}; s)$ denotes the Hamiltonian of the particle using distance s as an independent variable, $\mathbf{r} = (x, y, z = s - v_0 t)$ denotes spatial coordinates of the particle, and $\mathbf{p} = (p_x/p_0, p_y/p_0, p_z = (p - p_0)/p_0)$ the canonical momentum coordinates of the particle normalized by the reference particle momentum p_0 without acceleration. Let ζ denote a 6-vector of coordinates, the above Hamilton's equation can be rewritten as

$$\frac{d\zeta}{ds} = -[H, \zeta], \quad (3)$$

where $[,]$ is the Poisson bracket. A formal solution for above equation after a single step τ can be written as

$$\zeta(\tau) = \exp[-\tau(:H:)]\zeta(0). \quad (4)$$

Here, we have defined a differential operator $:H:$ as $:H:g = [H, g]$, for arbitrary function g . For a Hamiltonian that can be written as a sum of two terms $H = H_1 + H_2$, an approximate solution to above formal solution can be written as [18]

$$\begin{aligned} \zeta(\tau) &= \exp[-\tau(:H_1: + :H_2:)]\zeta(0) \\ &= \exp\left(-\frac{1}{2}\tau:H_1:\right) \exp(-\tau:H_2:) \\ &\quad \times \exp\left(-\frac{1}{2}\tau:H_1:\right) \zeta(0) + O(\tau^3). \end{aligned} \quad (5)$$

Let $\exp(-\frac{1}{2}\tau:H_1:)$ define a transfer map $\mathcal{M}_1$ and $\exp(-\tau:H_2:)$ a transfer map $\mathcal{M}_2$, for a single step, the above splitting results in a second order numerical integrator for the original Hamilton's equation as

$$\begin{aligned} \zeta(\tau) &= \mathcal{M}(\tau)\zeta(0) \\ &= \mathcal{M}_1(\tau/2)\mathcal{M}_2(\tau)\mathcal{M}_1(\tau/2)\zeta(0) + O(\tau^3). \end{aligned} \quad (6)$$

The above numerical integrator can be extended to 4th order accuracy and arbitrary even-order accuracy following Yoshida's approach [19]. This numerical integrator Eq. (6) will be symplectic if both the transfer map $\mathcal{M}_1$ and the transfer map $\mathcal{M}_2$ are symplectic. A transfer map $\mathcal{M}_i$ is symplectic if and only if the Jacobian matrix M_i of the transfer map $\mathcal{M}_i$ satisfies the following condition:

$$M_i^T J M_i = J, \quad (7)$$

where J denotes the 6×6 matrix given by

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \quad (8)$$

and I is the 3×3 identity matrix.

For the Hamiltonian in Eq. (5), we can choose H_1 as

$$H_1 = \mathbf{p}^2/2 + q\psi(\mathbf{r}), \quad (9)$$

where ψ denotes potential corresponding to external fields. This corresponds to the Hamiltonian of a group of charged particles inside an external field without mutual interaction among themselves. A single-charged particle magnetic optics method can be used to find the symplectic transfer map $\mathcal{M}_1$ for this Hamiltonian with the external fields from most accelerator beam line elements [20].

We can choose H_2 as

$$H_2 = K\phi(\mathbf{r}), \quad (10)$$

where $K = q/(p_0 v_0 \gamma_0^2)$, and ϕ is the space-charge potential from the solution of the Poisson's equation. In this Hamiltonian, the effects of the direct Coulomb electric potential and the longitudinal vector potential are combined together. The electric Coulomb potential ϕ in the Hamiltonian H_2 can be obtained from the solution of the Poisson equation.

For the space-charge Hamiltonian $H_2(\mathbf{r})$, the single step transfer map $\mathcal{M}_2$ can be written as

$$\mathbf{r}(\tau) = \mathbf{r}(0), \quad (11)$$

$$\mathbf{p}(\tau) = \mathbf{p}(0) - \frac{\partial H_2(\mathbf{r})}{\partial \mathbf{r}} \tau. \quad (12)$$

The Jacobian matrix of the above transfer map $\mathcal{M}_2$ is

$$M_2 = \begin{pmatrix} I & 0 \\ L & I \end{pmatrix}, \quad (13)$$

where L is a 3×3 matrix. For M_2 to satisfy the symplectic condition Eq. (7), the matrix L needs to be a symmetric matrix, i.e.,

$$L = L^T. \quad (14)$$

Given the fact that $L_{ij} = \partial p_i(\tau)/\partial r_j = -\frac{\partial^2 H_2(\mathbf{r})}{\partial r_i \partial r_j} \tau$, the matrix L will be symmetric as long as it is *analytically calculated* from the function H_2 . If both the transfer map $\mathcal{M}_1$ and the transfer map $\mathcal{M}_2$ are symplectic, the numerical integrator Eq. (6) for multiparticle tracking will be symplectic.

III. TWO-AND-A-HALF DIMENSIONAL SPACE-CHARGE SOLVER IN A STRAIGHT ACCELERATOR

We first discuss the solution of the Poisson equation in a straight accelerator, where the curvature is zero. This approach can also be applied to circular accelerators when the bending radius is much larger than the transverse aperture size. The effects of finite curvature will be addressed in the next section. In this case, the three-dimensional Poisson equation in the moving coordinate system can be written as

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{1}{\gamma^2} \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho(x, y, z)}{\epsilon_0}, \quad (15)$$

where $\gamma = 1/\sqrt{1 - (v_0/c)^2}$ is the relativistic factor and $z = s - v_0 t$. For a smooth long bunch at high energy $\frac{\partial^2 \phi}{\partial x^2} \gg \frac{1}{\gamma^2} \frac{\partial^2 \phi}{\partial z^2}$ and $\frac{\partial^2 \phi}{\partial y^2} \gg \frac{1}{\gamma^2} \frac{\partial^2 \phi}{\partial z^2}$, the longitudinal partial derivative with respect to z term can be treated as a perturbation and the Poisson equation (15) can be solved using an iterative procedure. For the zeroth-order approximation, the above Poisson equation can be reduced into a two-dimensional Poisson equation:

$$\frac{\partial^2 \phi_0}{\partial x^2} + \frac{\partial^2 \phi_0}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0}. \quad (16)$$

The above solution can be improved in the first-order approximation by including the partial derivative of the zeroth-order potential ϕ_0 with respect to z . This results in the following two-dimensional Poisson equation:

$$\frac{\partial^2 \phi_1}{\partial x^2} + \frac{\partial^2 \phi_1}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0} - \frac{1}{\gamma^2} \frac{\partial^2 \phi_0}{\partial z^2}. \quad (17)$$

This solution can be further refined by replacing ϕ_0 with ϕ_1 on the right-hand side of the equation and solving the two-dimensional Poisson equation again. This iterative process can be repeated multiple times until no further improvement is observed. However, for a long beam bunch with smooth longitudinal density variation, the zeroth-order approximation is often sufficient. In this work, we therefore focus on solving the zeroth-order two-dimensional Poisson equation (16).

Now, let us consider a charge density distribution ρ inside the accelerator that can be expressed as $\rho(x, y, z) = n(x, y)\lambda(z)$. The zeroth-order approximated Poisson equation can then be written as

$$\frac{\partial^2 \phi_0}{\partial x^2} + \frac{\partial^2 \phi_0}{\partial y^2} = -\frac{n(x, y)\lambda(z)}{\epsilon}. \quad (18)$$

The solution of potential from the above equation can be written as $\phi_0(x, y, z) = \phi(x, y)\lambda(z)$, with ϕ satisfying the two-dimensional Poisson equation:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\frac{n(x, y)}{\epsilon_0}. \quad (19)$$

If the charge density distribution function cannot be expressed as the product of transverse and longitudinal density functions, the three-dimensional density function can be approximated as a sum of longitudinal orthogonal modes (e.g., Hermite-Gaussian modes [21]):

$$\rho(x, y, z) = \sum_i n_i(x, y)\lambda_i(z). \quad (20)$$

The zeroth-order potential can be written as

$$\phi_0 = \sum \phi_i \lambda_i \quad (21)$$

with

$$\frac{\partial^2 \phi_i}{\partial x^2} + \frac{\partial^2 \phi_i}{\partial y^2} = -\frac{n_i(x, y)}{\epsilon_0}. \quad (22)$$

A. Space-charge solver with a transverse Gaussian density distribution

First, we consider the case in which the electric potential is subject to three-dimensional open boundary conditions. If we further assume that the transverse density distribution $n(x, y)$ is a bi-Gaussian function with standard deviations σ_x and σ_y , respectively, the solution ϕ to the above Poisson equation after regularization (removing infinite) can be written as [22]:

$$\phi(x, y) = \frac{1}{4\pi\epsilon_0} \int_0^\infty dt \frac{\exp[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}] - 1}{\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}. \quad (23)$$

The above integral is finite and vanishes at $(x = 0, y = 0)$. However, this potential cannot be directly used as the solution for ϕ_0 , since it yields a zero longitudinal electric field on axis $[(x = 0, y = 0)]$. To satisfy the open boundary conditions, the above potential should be shifted by a constant. The resulting zeroth-order solution for the potential ϕ_0 can be written as

$$\phi_0(x, y, z) = (\phi(x, y) + \phi_{00})\lambda(z), \quad (24)$$

where ϕ_{00} is the potential at $(x = 0, y = 0)$ subject to the transverse open boundary conditions. The integral in Eq. (23) is taken over an infinite computational domain, which is not convenient for numerical evaluation. By introducing the variable substitution $w^2 = \frac{1}{1+t/\sigma_x^2}$, the integral can be rewritten as

$$\varphi(x, y) = \frac{2A}{4\pi\epsilon_0} \int_0^1 dw \frac{\exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2)\right] - 1}{w\sqrt{(A^2 + (1-A^2)w^2)}}, \quad (25)$$

where the aspect ratios $A = \sigma_x/\sigma_y$, $x' = x/\sigma_x$, and $y' = y/\sigma_y$. This transformation converts the integral from an infinite domain to a finite range between 0 and 1. Figure 1 shows the integrand function in the above integral for aspect ratios 1, 2, and 3. This integrand function varies smoothly between 0 and 1. The integral can then be approximated as

$$\varphi(x, y) = \frac{2A}{4\pi\epsilon_0} \left(\frac{-\frac{1}{4}\delta^2[x'^2 + y'^2/(A^2 + (1-A^2)\delta^2)]}{\sqrt{(A^2 + (1-A^2)\delta^2)}} + \int_\delta^1 dw \frac{\exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2)\right] - 1}{w\sqrt{(A^2 + (1-A^2)w^2)}} \right), \quad (26)$$

where δ is a small number close to zero (e.g., 0.01). The first term on the right-hand side represents the trapezoidal rule approximation of the original integral between 0 and δ , while the second term can be efficiently computed using a quadrature rule such as Simpson's rule.

The constant φ_{00} can be obtained from the Green's function solution of the two-dimensional Poisson equation (19) as

$$\varphi_{00} = \frac{-1}{4\pi\epsilon_0} \iint \log(x^2 + y^2) n(x, y) dx dy. \quad (27)$$

For a bi-Gaussian distribution, it becomes

$$\varphi_{00} = \frac{-1}{4\pi\epsilon_0} \frac{1}{2\pi\sigma_x\sigma_y} \iint \log(x^2 + y^2) \exp\left[-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right] dx dy. \quad (28)$$

The above integral can be rewritten in the polar coordinate system as

$$\varphi_{00} = \frac{-1}{4\pi\epsilon_0} \frac{1}{2\pi} \int_0^\infty \int_0^{2\pi} \log\{r^2[\sigma_x^2\cos^2(\theta) + \sigma_y^2\sin^2(\theta)]\} \exp\left(-\frac{1}{2}r^2\right) r dr d\theta. \quad (29)$$

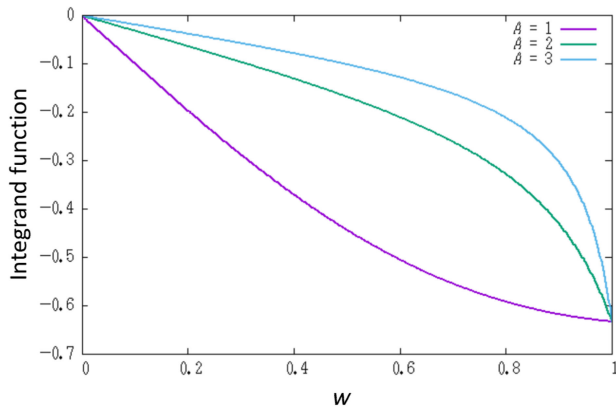


FIG. 1. Integrand function in the above integral with aspect ratios 1 (magenta), 2 (green), and 3 (blue) at $(x' = y' = 1)$.

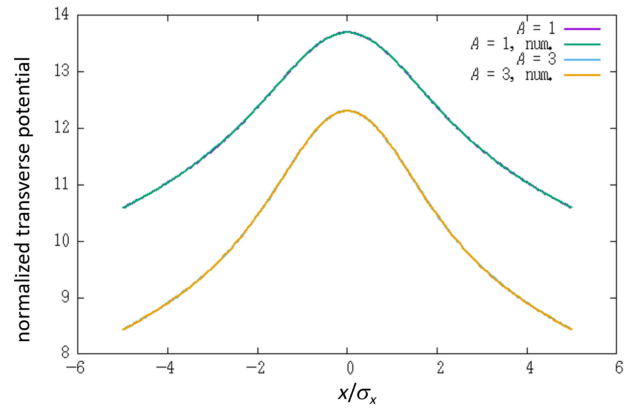


FIG. 2. Normalized transverse potential as a function of normalized horizontal distance with aspect ratios 1 (magenta) and 3 (blue) together with the numerical solutions (green and orange).

This integral admits a closed-form solution:

$$\varphi_{00} = \frac{-1}{4\pi\epsilon_0} \left[\log 2 - \gamma_{em} + 2 \log \left(\frac{\sigma_x + \sigma_y}{2} \right) \right], \quad (30)$$

where $\gamma_{em} = -\int_0^\infty e^{-x} \log(x) dx \simeq 0.577216$ [23] is the Euler-Mascheroni constant.

Figure 2 shows the electric potential, $(\varphi(x, 0) + \varphi_{00})$, normalized by $1/(4\pi\epsilon_0)$, as a function of the normalized horizontal position for aspect ratios of 1 and 3. Results are

$$p_x(\tau) = p_x(0) + \tau K \frac{\lambda(z)}{4\pi\epsilon_0} \int_0^\infty dt \frac{x \exp \left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} \right]}{(\sigma_x^2+t) \sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}, \quad (31)$$

$$p_y(\tau) = p_y(0) + \tau K \frac{\lambda(z)}{4\pi\epsilon_0} \int_0^\infty dt \frac{y \exp \left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} \right]}{(\sigma_y^2+t) \sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}, \quad (32)$$

$$p_z(\tau) = p_z(0) - \tau K (\varphi(x, y) + \varphi_{00}) \frac{\partial \lambda(z)}{\partial z}. \quad (33)$$

The transverse momenta are typically computed using the Bassetti-Erskine complex error function [24], while the update of the longitudinal momentum deviation can be obtained from the integral (26) and φ_{00} . Calculating these two integrals for the transverse momenta may be faster than using the complex error function by employing the above w variable and taking advantage of the similarity among the integrands in all three integrals. A Fortran subroutine and a C++ function for computing the integrals in Eqs. (25), (31), and (32) are provided in Appendixes A and B.

For a longitudinal Gaussian density distribution $\lambda(z) = \frac{Q}{\sqrt{2\pi}\sigma_z} \exp(-\frac{z^2}{2\sigma_z^2})$, the momentum deviation update will be

presented both from the above integral and from the numerical convolution using the integrated Green's function method described in the next subsection. As shown in the figure, the semianalytical and numerical solutions overlap, demonstrating excellent agreement between the two methods.

From the above electric potential, we can compute the symplectic kick in each momentum direction using H_2 . The three momentum components are updated after one step τ as

$$p_z(\tau) = p_z(0) + \tau K (\varphi(x, y) + \varphi_{00}) \frac{Qz}{\sqrt{2\pi}\sigma_z^3} \exp \left(-\frac{z^2}{2\sigma_z^2} \right), \quad (34)$$

where Q is the total charge of the beam bunch.

The two-dimensional Poisson equation solution discussed above assumes that the longitudinal variation within a long beam bunch is much smaller than the transverse variation. In the following, we assess the accuracy of this approximation by considering a three-dimensional Gaussian density distribution, for which a semianalytical solution can be obtained. For a three-dimensional Gaussian density distribution,

$$\rho(x, y, z) = \frac{Q}{(2\pi)^{3/2} \sigma_x \sigma_y \sigma_z} \exp \left[-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{z^2}{2\sigma_z^2} \right) \right],$$

the electric potential in the Poisson equation (15) can be determined using the same Fourier transform method as in the two-dimensional case.

$$\phi_{3D}(x, y, z) = \frac{Q}{4\pi\epsilon_0 \sqrt{2\pi}} \int_0^\infty dt \frac{\exp \left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} - \frac{z^2}{2(\sigma_z^2+t/\gamma^2)} \right]}{\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)(\sigma_z^2+t/\gamma^2)}}. \quad (35)$$

The resulting electric fields, given by $\mathbf{E} = -\nabla \phi_{3d}$, in the three spatial directions are:

$$E_x = \frac{Qx}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} - \frac{z^2}{2(\sigma_z^2+t/\gamma^2)}\right]}{(\sigma_x^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)(\sigma_z^2+t/\gamma^2)}}, \quad (36)$$

$$E_y = \frac{Qy}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} - \frac{z^2}{2(\sigma_z^2+t/\gamma^2)}\right]}{(\sigma_y^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)(\sigma_z^2+t/\gamma^2)}}, \quad (37)$$

$$E_z = \frac{Qz}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)} - \frac{z^2}{2(\sigma_z^2+t/\gamma^2)}\right]}{(\sigma_z^2+t/\gamma^2)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)(\sigma_z^2+t/\gamma^2)}}. \quad (38)$$

Equation (35) can be rewritten using w as the independent variable in the integrand as follows:

$$\phi_{3D}(x, y, z) = \frac{Q2A}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z} \int_0^1 dw \frac{\exp\left\{-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/[A^2 + (1-A^2)w^2] - \frac{1}{2}z'^2\left[1 + \frac{B^2}{w^2}(1-w^2)\right]\right\}}{w\sqrt{(A^2 + (1-A^2)w^2)\left[1 + \frac{B^2}{w^2}(1-w^2)\right]}}, \quad (39)$$

where $z' = z/\sigma_z$ and $B = \sigma_x/(\gamma\sigma_z)$. If $(B/w) \ll 1$, i.e., $1 + \frac{B^2}{w^2}(1-w^2) \approx 1$, the above equation is reduced to:

$$\phi_{3D}(x, y, z) \approx \frac{Q2A \exp\left(-\frac{1}{2}z'^2\right)}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z} \int_0^1 dw \frac{\exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2)\right]}{w\sqrt{(A^2 + (1-A^2)w^2)}}. \quad (40)$$

From the definition of $\lambda(z)$ for a longitudinal Gaussian distribution, it can be seen that the above equation corresponds to the zeroth-order two-dimensional potential solution Eq. (25), without regularization. The error introduced by this zeroth-order two-dimensional approximation can be estimated by the relative difference between the two integrands of potential integrals, given by

$$\text{Err}(w, B) = \left| \exp\left(-\frac{1}{2} \frac{B^2(1-w^2)z'^2}{w^2 + B^2(1-w^2)}\right) \sqrt{1 + \frac{B^2}{w^2}(1-w^2)} - 1 \right|. \quad (41)$$

Here, w is between 0 and 1. The above function becomes infinite at $w = 0$ except with $B = 0$, which suggests a potential large error in the two-dimensional approximation.

Figure 3 shows the above error function as a function of w with two B values. It can be seen that the maximum error occurs at the minimum w values and can be small as long as B/w is small. For $B/w \ll 1$, the above equation can be rewritten as

$$\text{Err}(w, B) \approx \left| \frac{1}{2}(1-w^2)(1-z'^2) \frac{B^2}{w^2} \right|. \quad (42)$$

The maximum error will be

$$\text{Err}(w, B)_{\max} \approx \left| \frac{1}{2}(1-w_{\min}^2)(1-z'^2) \frac{B^2}{w_{\min}^2} \right|. \quad (43)$$

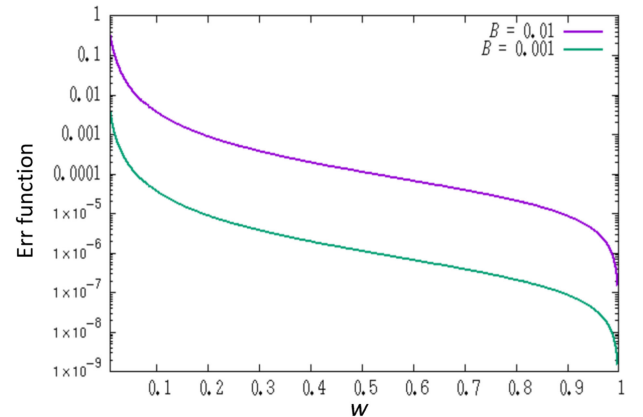


FIG. 3. Error defined in Eq. (41) as a function of w with aspect ratios 0.01 (magenta) and 0.001 (green) at $(z' = 0.5)$.

Assuming the maximum longitudinal position $z'_{\max} = 5$, and small w value $1 - w^2 \approx 1$, the maximum error becomes

$$\text{Err}(w, B)_{\max} \approx 12 \frac{B^2}{w_{\min}^2}. \quad (44)$$

The choice of $w_{\min}$ and B affects the accuracy of the zeroth-order two-dimensional approximation. Although

the minimum value of w is zero, its contribution to the electric fields, as shown below, is also zero. This suggests that the integration over w can begin from a small finite value rather than from zero.

The three-dimensional electric fields can be rewritten using the variable w as

$$E_x(x, y, z) = \frac{Q2Ax}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z\sigma_x^2} \int_0^1 dw \frac{w^2 \exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2) - \frac{1}{2}w^2z'^2/(B^2 + (1-B^2)w^2)\right]}{\sqrt{(A^2 + (1-A^2)w^2)(B^2 + (1-B^2)w^2)}}, \quad (45)$$

$$E_y(x, y, z) = \frac{Q2Ay}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z\sigma_y^2} \int_0^1 dw \frac{w^2 \exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2) - \frac{1}{2}w^2z'^2/(B^2 + (1-B^2)w^2)\right]}{(A^2 + (1-A^2)w^2)\sqrt{(A^2 + (1-A^2)w^2)(B^2 + (1-B^2)w^2)}}, \quad (46)$$

$$E_z(x, y, z) = \frac{Q2Az}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z\sigma_z^2} \int_0^1 dw \frac{w^2 \exp\left[-\frac{1}{2}w^2x'^2 - \frac{1}{2}w^2y'^2/(A^2 + (1-A^2)w^2) - \frac{1}{2}w^2z'^2/(B^2 + (1-B^2)w^2)\right]}{(B^2 + (1-B^2)w^2)\sqrt{(A^2 + (1-A^2)w^2)(B^2 + (1-B^2)w^2)}}. \quad (47)$$

Figure 4 shows the integrand of the above integral (45) as a function of w for $B = 0.01$ and $B = 0.001$ (with $x' = y' = z' = 1, A = 1$). It can be seen that the maximum value of the integrand is approximately 0.7, and the integral itself is on the order of $O(0.1)$. For small w , the contribution of the integral between 0 and $w_{\min}$ in the above equations scales as $O(w_{\min}^3)$. For $w_{\min} = 0.01$, this results in an integral of order $O(10^{-6})$. Consequently, setting $w_{\min} = 0.01$ leads to a loss of accuracy in the field calculation of $O(10^{-5})$. As an illustration, we consider a three-dimensional Gaussian bunch with $\sigma_x = \sigma_y = \sigma_z = 1$ mm and various values of γ . Figure 5 shows the

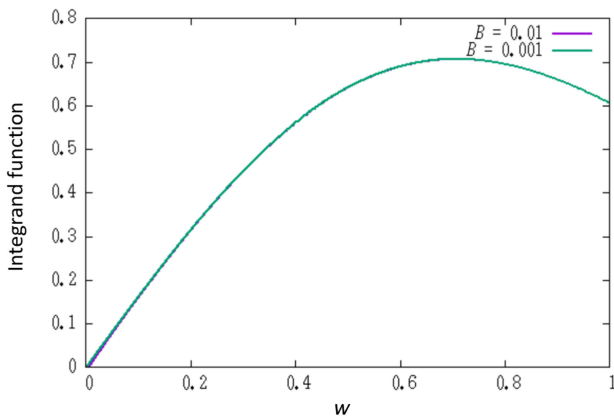


FIG. 4. Integrand function in the above integral (45) with aspect ratios 0.01 (magenta) and 0.001 (green) at $(x' = y' = z' = 1)$.

horizontal electric field (normalized by $\frac{Q}{4\pi\epsilon_0\sqrt{2\pi}\sigma_z}$) as a function of x/σ_x at $y = z = 0$ for $\gamma = 1, 10$, and $\gamma = 100$, comparing the results from the above two-dimensional zeroth-order solution and the full three-dimensional solution.

Figure 6 shows the normalized longitudinal electric field as a function of z/σ_z at $x = y = 0$ for $\gamma = 1, 10, 100$, and 1000, comparing the results from the above two-dimensional

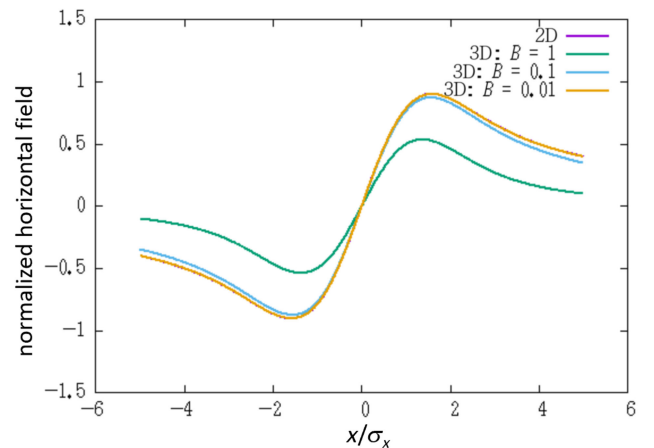


FIG. 5. Normalized horizontal electric field as a function of x/σ_x from the two-dimensional solution with longitudinal aspect ratio 1 (magenta) and from the three-dimensional solution with aspect ratios 1 (green), 0.1 (blue), and 0.01 (orange) at $(y = z = 0)$.

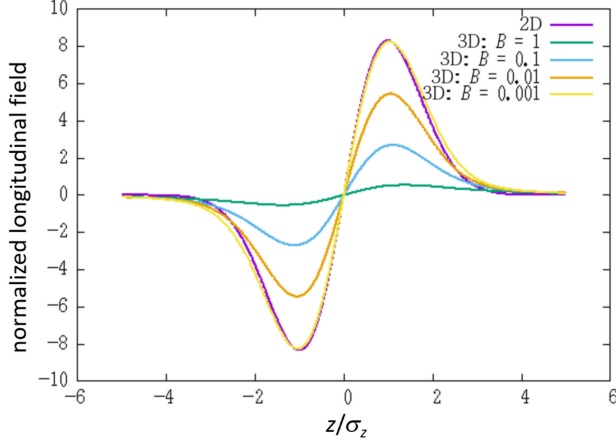


FIG. 6. Normalized longitudinal electric field as a function of z/σ_z from the two-dimensional solution with longitudinal aspect ratio 1 (magenta) and from the three-dimensional solution with aspect ratios 1 (green), 0.1 (blue), 0.01 (orange), and 0.001 (yellow) at $(x = y = 0)$.

zeroth-order solution and the three-dimensional solution. It can be seen that as the longitudinal aspect ratio B decreases, the three-dimensional solution approaches the two-dimensional solution (independent of γ) for both the

horizontal and longitudinal electric fields. This indicates that the two-dimensional solution provides a good approximation to the three-dimensional solution when the transverse-to-longitudinal aspect ratio is small, corresponding to a very long bunch or a high-energy beam with a large γ factor.

For reference, Fortran and C++ implementations of the above integrals for computing the three-dimensional electric fields from the three-dimensional Gaussian distribution are provided in Appendixes C and D.

If the longitudinal density distribution $\lambda(z)$ is not a Gaussian function, it can be decomposed as a sum of Gaussian wavelets, i.e.,

$$\lambda(z) = \frac{Q}{\sqrt{2\pi}} \sum_{i=1}^N \frac{1}{\sigma_{z_i}} \exp\left(-\frac{1}{2} \frac{(z - z_i)^2}{\sigma_{z_i}^2}\right), \quad (48)$$

where z_i and σ_{z_i} are the centroid and standard deviation of the i th wavelet, respectively, and N is the total number of wavelets.

The electric potential solution to the three-dimensional Poisson equation, for a transverse Gaussian distribution and the longitudinal distribution given above, can be expressed as

$$\phi_{3D}(x, y, z) = \frac{Q}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right]}{\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}} \left(\sum_{i=1}^N \frac{\exp\left[-\frac{(z-z_i)^2}{2(\sigma_{z_i}^2+t/\gamma^2)}\right]}{\sqrt{(\sigma_{z_i}^2+t/\gamma^2)}} \right). \quad (49)$$

The resulting electric fields can be written as

$$E_x(x, y, z) = \frac{Qx}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right]}{(\sigma_x^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}} \left(\sum_{i=1}^N \frac{\exp\left[-\frac{(z-z_i)^2}{2(\sigma_{z_i}^2+t/\gamma^2)}\right]}{\sqrt{(\sigma_{z_i}^2+t/\gamma^2)}} \right), \quad (50)$$

$$E_y(x, y, z) = \frac{Qy}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right]}{(\sigma_y^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}} \left(\sum_{i=1}^N \frac{\exp\left[-\frac{(z-z_i)^2}{2(\sigma_{z_i}^2+t/\gamma^2)}\right]}{\sqrt{(\sigma_{z_i}^2+t/\gamma^2)}} \right), \quad (51)$$

$$E_z(x, y, z) = \frac{Q}{4\pi\epsilon_0\sqrt{2\pi}} \int_0^\infty dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right]}{\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}} \left(\sum_{i=1}^N \frac{(z - z_i) \exp\left[-\frac{(z-z_i)^2}{2(\sigma_{z_i}^2+t/\gamma^2)}\right]}{(\sigma_{z_i}^2+t/\gamma^2)\sqrt{(\sigma_{z_i}^2+t/\gamma^2)}} \right). \quad (52)$$

B. Space-charge solver with an arbitrary density distribution

The semianalytical expression for space-charge fields derived from a Gaussian distribution can facilitate rapid tracking and parameter space exploration [25,26]. However, for accurate predictions of emittance growth or beam loss, particularly in regions near resonances or in the presence of

machine errors, the particle density distribution may deviate significantly from a Gaussian function, necessitating the use of a direct numerical space-charge solver.

For an arbitrary transverse charge density distribution, numerical methods must be employed. Various numerical techniques can effectively solve the two-dimensional Poisson equation under different boundary conditions.

For transverse open boundary conditions, an efficient approach is to use the Green's function method combined with the fast Fourier transform (FFT) to perform cyclic summation on a two-dimensional grid. The Green's function solution to two-dimensional Poisson's equation (19) can be expressed as

$$\varphi(x, y) = \frac{1}{2\pi\epsilon_0} \int \int G(x - x', y - y') n(x', y') dx' dy', \quad (53)$$

where

$$G(x, y) = -\frac{1}{2} \log(x^2 + y^2). \quad (54)$$

The above convolution can be computed numerically on a grid using an integrated Green's function method [27].

$$\varphi(x_i, y_j) = \frac{1}{2\pi\epsilon_0} \sum_{i'=1}^{N_x} \sum_{j'=1}^{N_y} \bar{G}(x_i - x_{i'}, y_j - y_{j'}) n(x_{i'}, y_{j'}), \quad (55)$$

where

$$\bar{G}(x_i - x_{i'}, y_j - y_{j'}) = \int_{x_{i'} - h_x/2}^{x_{i'} + h_x/2} dx' \int_{y_{j'} - h_y/2}^{y_{j'} + h_y/2} dy' G(x_i - x', y_j - y'). \quad (56)$$

This integration can be done analytically using the indefinite integral:

$$\int \int \ln(x^2 + y^2) dx dy = -3xy + x^2 \arctan(y/x) + y^2 \arctan(x/y) + xy \ln(x^2 + y^2). \quad (57)$$

The cyclic summation for the electric potential described above can be computed efficiently using an FFT [28]. From the two-dimensional potential computed on the grid, a spline shape function S can be used to obtain the space-charge potential and its derivatives with respect to x and y . The one-step momentum update from the space-charge map can then be expressed concisely as

$$p_x(\tau) = p_x(0) - \tau K \lambda(z) \sum_i \sum_j \frac{\partial S(x_i - x)}{\partial x} S(y_j - y) \varphi(x_i, y_j),$$

$$p_y(\tau) = p_y(0) - \tau K \lambda(z) \sum_i \sum_j S(x_i - x) \frac{\partial S(y_j - y)}{\partial y} \varphi(x_i, y_j), \quad (58)$$

$$p_z(\tau) = p_z(0) - \tau K \frac{\partial \lambda(z)}{\partial z} \sum_i \sum_j S(x_i - x) S(y_j - y) \varphi(x_i, y_j). \quad (59)$$

Next, we consider the case where the electric potential is confined within a rectangular, perfectly conducting pipe. The charge density distribution, $\rho(x, y, z)$, is still assumed to be separable into a product of a transverse distribution, $n(x, y)$, and a longitudinal distribution, $\lambda(z)$. The boundary conditions for the electric potential $\varphi(x, y)$ inside the rectangular perfectly conducting pipe are:

$$\varphi(x = 0, y) = 0, \quad (60)$$

$$\varphi(x = a, y) = 0, \quad (61)$$

$$\varphi(x, y = 0) = 0, \quad (62)$$

$$\varphi(x, y = b) = 0, \quad (63)$$

where a is the horizontal width of the pipe and b is the vertical width of the pipe.

Given the boundary conditions in Eqs. (60)–(63), the electric potential φ and the source term $n(x, y)$ can be approximated using two sine functions as

$$n(x, y) = \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} n^{lm} \sin(\alpha_l x) \sin(\beta_m y), \quad (64)$$

$$\varphi(x, y) = \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \varphi^{lm} \sin(\alpha_l x) \sin(\beta_m y), \quad (65)$$

where

$$n^{lm} = \frac{4}{ab} \int_0^a \int_0^b n(x, y) \sin(\alpha_l x) \sin(\beta_m y) dx dy, \quad (66)$$

$$\varphi^{lm} = \frac{4}{ab} \int_0^a \int_0^b \varphi(x, y) \sin(\alpha_l x) \sin(\beta_m y) dx dy, \quad (67)$$

where $\alpha_l = l\pi/a$ and $\beta_m = m\pi/b$. The above approximation follows the numerical spectral Galerkin method since each basis function satisfies the boundary conditions on the wall. For a smooth function, this spectral approximation has an accuracy whose numerical error scales as $O[\exp(-cN)]$ with $c > 0$, where N is the number of the basis function (i.e., mode number in each dimension) used in the approximation [29,30]. By substituting above expansions into the Poisson equation (19) and making use of the orthonormal condition of the sine functions, we obtain

$$\varphi^{lm} = \frac{n^{lm}}{\epsilon_0 \gamma_{lm}^2}, \quad (68)$$

where $\gamma_{lm}^2 = \alpha_l^2 + \beta_m^2$. The zeroth-order approximate solution of potential ϕ_0 can be written as

$$\phi_0(x, y, z) = \varphi(x, y)\lambda(z). \quad (69)$$

From the above potential, the symplectic momentum kicks after each step due to the space-charge effects are:

$$p_x(\tau) = p_x(0) - \tau K \lambda(z) \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \varphi^{lm} \alpha_l \cos(\alpha_l x) \sin(\beta_m y), \quad (70)$$

$$p_y(\tau) = p_y(0) - \tau K \lambda(z) \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \varphi^{lm} \beta_m \sin(\alpha_l x) \cos(\beta_m y), \quad (71)$$

$$p_z(\tau) = p_z(0) - \tau K \frac{\partial \lambda(z)}{\partial z} \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \varphi^{lm} \sin(\alpha_l x) \sin(\beta_m y). \quad (72)$$

As a test of the above space-charge solver, we assumed a two-dimensional Gaussian density distribution with $\sigma_x = \sigma_y = 1$ mm and computed the electric potential using both the Green's function method with open boundary conditions and the spectral method with a perfectly conducting rectangular pipe of 6×6 mm aperture size. Figure 7 shows the normalized electric potential (normalized by $\frac{\lambda(z)}{4\pi\epsilon_0}$) as a function of horizontal position, calculated using both the Green's function method and the spectral method. Here, the electric potential from the Green's function method is shifted to match the maximum of the spectral solution

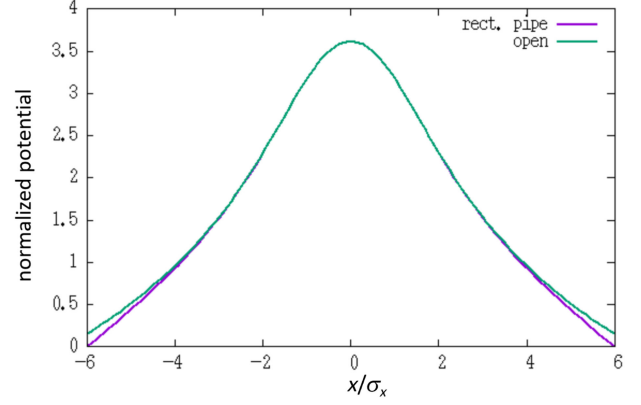


FIG. 7. Normalized electric potential as a function horizontal location from the spectral solver inside a rectangular pipe and from the Green's function solver with open boundary conditions.

inside the pipe. A Gaussian distribution extends to infinity, the spectral approach necessarily truncates the distribution at its boundary. The potential will be shielded from the hypothetical charges beyond the boundary resulting in different maximum potential values from the open boundary solution. After shifting, the two solutions agree well within the core of the distribution. The slight difference near the edge of the computational domain is noticeable and is due to the different boundary conditions used. For the spectral solution, the electric potential must be zero on the boundary as required by the boundary condition. For the Green's function solution, there is no such constraint, so the electric potential at the edge of the computational domain is not necessarily zero.

For a pipe with a circular cross section, the two-dimensional Poisson equation (19) can be written in cylindrical coordinates as

$$\frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \varphi}{\partial \theta^2} = -\frac{n(r, \theta)}{\epsilon_0}. \quad (73)$$

The boundary conditions for the potential are:

$$\varphi(r = a, \theta, z) = 0, \quad (74)$$

$$\varphi(r, \theta + 2\pi, z) = \varphi(r, \theta, z), \quad (75)$$

where a is the pipe radius.

The periodic boundary condition for the potential along the θ direction suggests the use of complex exponential eigenfunctions in that direction. In the radial direction, Bessel functions serve as appropriate eigenfunctions for a round conducting pipe. Hence, we can approximate the potential φ and the source term $n(r, \theta)$ as follows [31]:

$$n(r, \theta) = \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} n^{lm} J_m(\gamma_{lm} r) \exp(-im\theta), \quad (76)$$

$$\varphi(r, \theta) = \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \varphi^{lm} J_m(\gamma_{lm} r) \exp(-im\theta), \quad (77)$$

where γ_{lm} is a solution of

$$J_m(\gamma_{lm} a) = 0. \quad (78)$$

The n^{lm} and φ^{lm} are determined from

$$n^{lm} = \frac{1}{\pi J_m'^2(\gamma_{lm})} \int_0^{2\pi} \int_0^a n(r, \theta) \exp(im\theta) r J_m(\gamma_{lm} r) dr d\theta, \quad (79)$$

$$\varphi^{lm} = \frac{1}{\pi J_m'^2(\gamma_{lm})} \int_0^{2\pi} \int_0^a \varphi(r, \theta) \exp(im\theta) r J_m(\gamma_{lm} r) dr d\theta. \quad (80)$$

Substituting the above expansion into the Poisson equation and multiplying by $\exp(im\theta) r J_m(\gamma_{lm} r)$ and integrating from 0 to 2π and 0 to a , we obtain

$$\varphi^{lm} = \frac{n^{lm}}{\epsilon_0 \gamma_{lm}^2}. \quad (81)$$

From the φ^{lm} , the zeroth-order approximation of the potential ϕ_0 can be obtained using Eq. (77). Given the electric potential $\phi_0(r, \theta, z) = \varphi(r, \theta)\lambda(z)$, the symplectic momentum kicks after one step τ due to the space-charge effects are:

$$p_x(\tau) = p_x(0) - \tau K \lambda(z) \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \varphi^{lm} \left[\gamma_{lm} J_m'(\gamma_{lm} r) \frac{x}{r} + im \frac{y}{r^2} J_m(\gamma_{lm} r) \right] \exp(-im\theta), \quad (82)$$

$$p_y(\tau) = p_y(0) - \tau K \lambda(z) \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \varphi^{lm} \left[\gamma_{lm} J_m'(\gamma_{lm} r) \frac{y}{r} - im \frac{x}{r^2} J_m(\gamma_{lm} r) \right] \exp(-im\theta), \quad (83)$$

$$p_z(\tau) = p_z(0) - \tau K \frac{\partial \lambda(z)}{\partial z} \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \varphi^{lm} J_m(\gamma_{lm} r) \exp(-im\theta). \quad (84)$$

IV. TWO-AND-A-HALF DIMENSIONAL SPACE-CHARGE SOLVER IN A CIRCULAR ACCELERATOR

In this section, we consider the space-charge solver for a long bunch in a circular accelerator. In this case, we employ the Frenet-Serret coordinate system to formulate Poisson's equation, assuming a curvature h in the horizontal plane.

$$\frac{1}{1+hx} \frac{\partial}{\partial x} \left[(1+hx) \frac{\partial \phi}{\partial x} \right] + \frac{\partial^2 \phi}{\partial y^2} + \frac{1}{\gamma^2} \frac{1}{1+hx} \frac{\partial}{\partial z} \left(\frac{1}{1+hx} \frac{\partial \phi}{\partial z} \right) = -\frac{\rho}{\epsilon_0}. \quad (85)$$

For a constant curvature, the above equation can be rewritten as

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{h}{1+hx} \frac{\partial \phi}{\partial x} + \frac{\partial^2 \phi}{\partial y^2} + \frac{1}{\gamma^2 (1+hx)^2} \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0}. \quad (86)$$

In most circular accelerators, the horizontal beam size is much smaller than the bending radius, i.e., $hx \ll 1$. By noting that $\frac{h}{1+hx} \frac{\partial \phi}{\partial x} \ll \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}$, and $\frac{1}{\gamma^2(1+hx)^2} \frac{\partial^2 \phi}{\partial z^2} \ll \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}$, the equation above can be solved using an iterative procedure, which is equivalent to a perturbative approach.

$$\frac{\partial^2 \phi_0}{\partial x^2} + \frac{\partial^2 \phi_0}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0}, \quad (87)$$

$$\frac{\partial^2 \phi_1}{\partial x^2} + \frac{\partial^2 \phi_1}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0} - \frac{h}{1+hx} \frac{\partial \phi_0}{\partial x} - \frac{1}{\gamma^2(1+hx)^2} \frac{\partial^2 \phi_0}{\partial z^2}, \quad (88)$$

$$\frac{\partial^2 \phi_2}{\partial x^2} + \frac{\partial^2 \phi_2}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0} - \frac{h}{1+hx} \frac{\partial \phi_1}{\partial x} - \frac{1}{\gamma^2(1+hx)^2} \frac{\partial^2 \phi_1}{\partial z^2} \quad (89)$$

$$\vdots = \vdots$$

The above equations provide the correction to the space-charge potential due to the finite curvature in a circular accelerator. For the first-order correction, Eq. (88) can be written as

$$\frac{\partial^2 \phi_1}{\partial x^2} + \frac{\partial^2 \phi_1}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0} - h \frac{\partial \phi_0}{\partial x}. \quad (90)$$

For a transverse Gaussian density distribution, and using the zeroth-order solution ϕ_0 of Eq. (24) from Sec. III A, the solution to the first-order equation above can be obtained as

$$\phi_1(x, y, z) = \phi_0(x, y, z) - h \frac{\lambda(z)x}{8\pi\epsilon_0} \int_0^\infty t dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right] - 1}{(\sigma_x^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}. \quad (91)$$

Here, we have included a term in the potential that depends linearly on x to eliminate the divergence present in the original integral. A derivation of the solution above is provided in Appendix E.

As an illustration of the effect of curvature correction, we used a Gaussian density distribution with $\sigma_x = \sigma_y = 1$ mm, and considered several curvature values: 1.0, 0.1, and 0.01/m. Figure 8 shows the normalized horizontal electric potential correction from Eq. (91) as a function of the horizontal position at $y = z = 0$. For comparison, the electric potential in the straight system, $\phi_0(x, 0, 0)$, is shown in Fig. 2 with a maximum value of about 13.7. Compared with the solution for the straight system, this correction is several orders of magnitude smaller and is proportional to $h\sigma_x$. For a typical circular accelerator with a radius greater than 100 m and a beam size on the order of millimeters, this correction will be $O(10^{-5})$. Therefore, the effect of curvature is likely negligible.

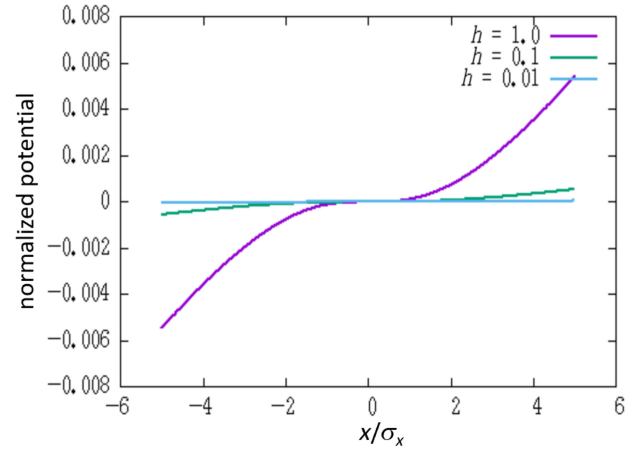


FIG. 8. Normalized potential as a function of normalized horizontal distance with curvatures 1/m (magenta), 0.1/m (green), and 0.01/m (blue).

Next, we consider the case of a pipe with a circular cross section. The three-dimensional Poisson equation can then be rewritten in transverse polar coordinates as

$$\nabla_{\perp}^2 \phi + h \frac{1}{1 + hr \cos(\theta)} \left(\cos(\theta) \frac{\partial \phi}{\partial r} - \frac{\sin(\theta)}{r} \frac{\partial \phi}{\partial \theta} \right) + \frac{1}{\gamma^2 [1 + hr \cos(\theta)]^2} \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0}. \quad (92)$$

This equation can be solved using the same iterative procedure:

$$\nabla_{\perp}^2 \phi_0 = -\frac{\rho}{\epsilon_0}, \quad (93)$$

$$\nabla_{\perp}^2 \phi_1 = -\frac{\rho}{\epsilon_0} - h \frac{1}{1 + hr \cos(\theta)} \left(\cos(\theta) \frac{\partial \phi_0}{\partial r} - \frac{\sin(\theta)}{r} \frac{\partial \phi_0}{\partial \theta} \right) + \frac{1}{\gamma^2 [1 + hr \cos(\theta)]^2} \frac{\partial^2 \phi_0}{\partial z^2}, \quad (94)$$

$$\nabla_{\perp}^2 \phi_2 = -\frac{\rho}{\epsilon_0} - h \frac{1}{1 + hr \cos(\theta)} \left(\cos(\theta) \frac{\partial \phi_1}{\partial r} - \frac{\sin(\theta)}{r} \frac{\partial \phi_1}{\partial \theta} \right) + \frac{1}{\gamma^2 [1 + hr \cos(\theta)]^2} \frac{\partial^2 \phi_1}{\partial z^2} \quad (95)$$

$\vdots = \vdots$

V. CONCLUSIONS

In this paper, we present a two-and-a-half dimensional symplectic space-charge solver for a long charged particle beam bunch (in the beam frame). Instead of solving the full three-dimensional Poisson equation, this solver addresses the two-dimensional Poisson equation by exploiting the slow longitudinal variation of the beam. This approach significantly improves computational speed in multiparticle tracking simulations where a fast space-charge solver is needed. Meanwhile, the longitudinal momentum kick is included to account for longitudinal density variation, in addition to the transverse momentum kicks. As a result, a symplectic space-charge solver is achieved for particle tracking.

For a transverse Gaussian density distribution, a semi-analytical solution is obtained. By comparing this solution with that from a fully three-dimensional Gaussian density distribution, we observe that for a longitudinal-to-transverse aspect ratio (in the beam frame) greater than 1000, the zeroth-order two-and-a-half dimensional space-charge solver provides a good approximation to the full three-dimensional solution.

For an arbitrary transverse density distribution, several efficient numerical methods are presented under different boundary conditions. These include an FFT-based integrated Green's function method and two spectral methods. The computational cost of the Green's function method scales as $O(N \log N)$, where N is the total number of computational grid points. The computational cost of the spectral method for a rectangular perfectly conducting pipe also scales as $O(N \log N)$, due to the use of the fast sine transform. The computational cost of the spectral method for a round conducting pipe is somewhat higher due to the

use of Bessel functions. However, given the rapid convergence of the spectral method, only a small number of spectral modes are needed.

In addition to straight accelerator systems, the two-and-a-half dimensional space-charge solver is extended to circular accelerators. Using a transverse Gaussian density distribution, we show that the correction to the electric potential due to the finite curvature of a circular accelerator is small and is proportional to the ratio of the beam size to the bending radius. For a typical circular accelerator with a bending radius greater than 100 m, this correction can be as small as $O(10^{-5})$. Thus, the finite curvature effect in the space-charge solver may be negligible in large accelerators.

ACKNOWLEDGMENTS

We thank Yue Hao and Helena Alamprez for their valuable discussions. This work was supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics, under Contract No. DE-AC02-05CH11231. Computational resources were provided by the National Energy Research Scientific Computing Center (NERSC), which is supported by the DOE Office of Science under the same contract number.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: Fortran IMPLEMENTATION OF INTEGRAL IN EQS. (25), (31), AND (32)

In the appendix, we show Fortran and C++ implementations of the integrals in the space-charge solver for a Gaussian density distribution and present the derivation of the first-order correction to the solution of the two-dimensional Poisson equation, accounting for the effect of the finite curvature of the circular accelerator system for a transverse Gaussian density distribution.

```

!-----
!The following subroutine calculates three integrals used to determine the
!3 space-charge kicks for a transverse Gaussian distribution in the paper.
!Note: This implementation might be most efficient for a small aspect ratio beam (e.g., A<5).
!
!Inputs: delta - small region near 0 (e.g., 0.01), nint - # of grid points (odd number).
!        xin, yin - x and y coordinates, sigx, sigy - x RMS size and y RMS size.
!Outputs: pintex - integral for Ex in Eq.(31); pintey - integral for Ey in Eq.(32);
!        pintezy - integral for Ez in Eq.(25). (including *2A)
!Author:: Ji Qiang
!-----

subroutine potInt(delta,nint,xin,yin,sigx,sigy,pintex,pintey,pintezy)
implicit none
integer, intent(inout) :: nint
real*8,intent(in) :: delta,xin,yin,sigx,sigy
real*8,intent(out) :: pintex,pintey,pintezy
!xp is x/sigma_x, yp is y/sigma_y
!asp = sigma_x/sigma_y
real*8 :: asp,xp,yp,asp2,asp2m
real*8 :: sum0,xmin,xmax,x,h,t2,f1,f2,fx,yp2,sum0ex,sum0ey
real*8, dimension(nint+1) :: xvec,t2vec,t2sqrt
real*8, dimension(nint+1) :: f2vec,f2yvec,f1vec,f1xvec,fxvec
integer :: i
real*8 :: exparg,sqrt2,f1x
!enforce an odd grid points
if(mod(nint,2).eq.0) nint = nint+1

!xmin = 0.01d0
xmin = delta
xmax = 1.0d0
h = (xmax-xmin)/(nint-1)

xp = xin/sigx
yp = yin/sigy
asp = sigx/sigy
!-----
! The following does integral Eq.(25) without division of 4pi\epsilon0
!
!trapezoidal rule approximation integral from 0 to delta
x = xmin
t2 = asp**2+(1-asp**2)*x**2
sqrt2 = sqrt(t2)
exparg = -(xp**2+yp**2/t2)/2
f1 = x*exparg
sum0 = x*f1/sqrt2/2
!Ex
f1 = x*exp(x**2*exparg)
sum0ex = x*f1/sqrt2/2
!Ey
sum0ey = x*f1/(t2*sqrt2)/2
!Simpson's rule for two end points
!-----
!Ez
f1 = exp(x**2*exparg)
f2 = x*sqrt2

```

```

      fx = (f1-1.0d0)/f2
      sum0 = sum0-fx*h

      f1x = x*f1
      f2 = sqrt2
      fx = f1x/f2
      sum0ex = sum0ex-fx*h

      f2 = t2*sqrt2
      fx = f1x/f2
      sum0ey = sum0ey-fx*h

      !Ez
      x = xmax
      t2 = asp**2+(1-asp**2)*x**2
      sqrt2 = sqrt(t2)
      exparg = -(xp**2+yp**2/t2)/2
      f1 = exp(x**2*exparg)
      f2 = x*sqrt2
      fx = (f1-1.0d0)/f2
      sum0 = sum0-fx*h
      !Ex
      f1x = x*f1
      f2 = sqrt2
      fx = f1x/f2
      sum0ex = sum0ex-fx*h
      !Ex
      f2 = t2*sqrt2
      fx = f1x/f2
      sum0ey = sum0ey-fx*h
!-----

      do i = 1, nint
         xvec(i) = xmin + (i-1)*h
      enddo
      asp2 = asp**2
      asp2m = 1.0d0-asp**2
      do i = 1, nint
         t2vec(i) = asp2+asp2m*xvec(i)**2
      enddo
      do i = 1, nint
         t2sqrt(i) = sqrt(t2vec(i))
      enddo
      !for Ey error function
      do i = 1, nint
         f2yvec(i) = t2vec(i)*t2sqrt(i)
      enddo
      do i = 1, nint
         f2vec(i) = xvec(i)*t2sqrt(i)
      enddo

      xp2 = xp**2
      yp2 = yp**2
      do i = 1, nint
         f1vec(i) = exp(-xvec(i)**2*(xp2+yp2/t2vec(i))/2)
      enddo
      !for Ex error function
      do i = 1, nint

```

```

        f1xvec(i) = xvec(i)*f1vec(i)
    enddo

!-----
!for potential used in Ez
do i = 1, nint
    fxvec(i) = (f1vec(i)-1.0d0)/f2vec(i)
enddo
do i = 1, nint, 2
    sum0 = sum0 + 2*fxvec(i)*h
enddo
do i = 2, nint, 2
    sum0 = sum0 + 4*fxvec(i)*h
enddo
pintez = 2*asp*sum0/3.0d0
!-----
! The following does integral in Eq.(31) for Ex
!
    !for Ex
do i = 1, nint
    fxvec(i) = f1xvec(i)/t2sqrt(i)
enddo
do i = 1, nint, 2
    sum0ex = sum0ex + 2*fxvec(i)*h
enddo
do i = 2, nint, 2
    sum0ex = sum0ex + 4*fxvec(i)*h
enddo
pintex = 2*asp*xp*sum0ex/3.0d0/sigx
!-----
! The following does integral in Eq.(32) for Ey
!
    !for Ey
do i = 1, nint
    fxvec(i) = f1xvec(i)/f2yvec(i)
enddo
do i = 1, nint, 2
    sum0ey = sum0ey + 2*fxvec(i)*h
enddo
do i = 2, nint, 2
    sum0ey = sum0ey + 4*fxvec(i)*h
enddo
pintey = 2*asp*yp*sum0ey/3.0d0/sigy

end subroutine potInt

```

APPENDIX B: C++ IMPLEMENTATION OF INTEGRAL IN EQS. (25), (31), AND (32)

```

/*-----
!The following subroutine calculates three integrals used to determine the
!3 space-charge kicks for a transverse Gaussian distribution in the paper.
!Inputs: delta - small region near 0 (e.g., 0.01), nint - # of grid points (odd number).
!        xin, yin - x and y coordinates, sigx, sigy - x RMS size and y RMS size.
!Outputs: pintex - integral for Ex in Eq.(31); pintey - integral for Ey in Eq.(32);
!        pinteZ - integral for Ez in Eq.(25). (including *2A)

!Author:: Ji Qiang with the help of ChatGPT
!-----
*/
void potInt(double delta, int& nint, double xin, double yin, double sigx, double sigy,
           double& pintex, double& pintey, double& pinteZ)
{
    // enforce odd grid points
    if (nint % 2 == 0) nint += 1;

    double xmin = delta;
    double xmax = 1.0;
    double h = (xmax - xmin) / (nint - 1);

    double xp = xin / sigx;
    double yp = yin / sigy;
    double asp = sigx / sigy;

    double sum0, sum0ex, sum0ey;
    sum0 = sum0ex = sum0ey = 0.0;

    // Trapezoidal rule approximation integral from 0 to delta
    double x = xmin;
    double t2 = asp*asp + (1 - asp*asp)*x*x;
    double sqrt2 = std::sqrt(t2);
    double exparg = -(xp*xp + yp*yp / t2) / 2.0;
    double f1 = x*exparg;

    sum0 = x*f1/sqrt2/2.0;
    f1 = x*std::exp(x*x*exparg);
    sum0ex = x*f1 / sqrt2 / 2.0;
    sum0ey = x*f1 / (t2*sqrt2) / 2.0;

    // Simpson's rule for two end points
    // Ez
    f1 = std::exp(x*x*exparg);
    double f2 = x*sqrt2;
    double fx = (f1 - 1.0) / f2;
    sum0 = sum0 - fx*h;

    double f1x = x*f1;
    f2 = sqrt2;
    fx = f1x / f2;
    sum0ex = sum0ex - fx*h;

    f2 = t2*sqrt2;
    fx = f1x / f2;
    sum0ey = sum0ey - fx*h;
}

```

```

// Ez at xmax
x = xmax;
t2 = asp*asp + (1 - asp*asp)*x*x;
sqrt2 = std::sqrt(t2);
exparg = -(xp*xp + yp*yp / t2) / 2.0;
f1 = std::exp(x*x*exparg);
f2 = x*sqrt2;
fx = (f1 - 1.0) / f2;
sum0 = sum0 - fx*h;
f1x = x*f1;
f2 = sqrt2;
fx = f1x / f2;
sum0ex = sum0ex - fx*h;
f2 = t2*sqrt2;
fx = f1x / f2;
sum0ey = sum0ey - fx*h;

// Arrays
std::vector<double> xvec(nint), t2vec(nint), t2sqrt(nint);
std::vector<double> f2vec(nint), f2yvec(nint), f1vec(nint), f1xvec(nint), fxvec(nint);

for (int i = 0; i < nint; ++i)
    xvec[i] = xmin + i*h;
double asp2 = asp*asp;
double asp2m = 1.0 - asp*asp;
for (int i = 0; i < nint; ++i)
    t2vec[i] = asp2 + asp2m * xvec[i]*xvec[i];
for (int i = 0; i < nint; ++i)
    t2sqrt[i] = std::sqrt(t2vec[i]);
for (int i = 0; i < nint; ++i)
    f2yvec[i] = t2vec[i]*t2sqrt[i];
for (int i = 0; i < nint; ++i)
    f2vec[i] = xvec[i]*t2sqrt[i];

double xp2 = xp*xp;
double yp2 = yp*yp;
for (int i = 0; i < nint; ++i)
    f1vec[i] = std::exp(-xvec[i]*xvec[i]*(xp2 + yp2 / t2vec[i]) / 2.0);
for (int i = 0; i < nint; ++i)
    f1xvec[i] = xvec[i]*f1vec[i];

// Ez
for (int i = 0; i < nint; ++i)
    fxvec[i] = (f1vec[i] - 1.0) / f2vec[i];
for (int i = 0; i < nint; i += 2)
    sum0 += 2 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0 += 4 * fxvec[i] * h;
pintez = 2 * asp * sum0 / 3.0;

// Ex
for (int i = 0; i < nint; ++i)
    fxvec[i] = f1xvec[i] / t2sqrt[i];
for (int i = 0; i < nint; i += 2)
    sum0ex += 2 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0ex += 4 * fxvec[i] * h;
pintex = 2 * asp * xp * sum0ex / 3.0 / sigx;

```

```

// Ey
for (int i = 0; i < nint; ++i)
    fxvec[i] = f1xvec[i] / f2yvec[i];
for (int i = 0; i < nint; i += 2)
    sum0ey += 2 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0ey += 4 * fxvec[i] * h;
pintey = 2 * asp * yp * sum0ey / 3.0 / sigy;
}

```

APPENDIX C: Fortran IMPLEMENTATION OF 3D FIELD INTEGRALS

```

!-----
!The following subroutine calculates three integrals used to calculate the
!3 space-charge kicks for a 3D Gaussian distribution in the paper.
!
!Inputs: nint - # of grid points (odd number),
!        xin, yin, zin - x, y and z coordinates, sigx, sigy, sigz -
!        x RMS size, y RMS size, and z RMS size, gamma - relativistic factor.
!Outputs: pintex - integral for Ex in Eq. (45); pintey - integral for Ey in Eq. (46);
!        pintezy - integral for Ez in Eq. (47).
!Author:: Ji Qiang
!-----

subroutine efldgauss(nint,xin,yin,zin,sigx,sigy,sigz,gamma,pintex,pintey,pintezy)
implicit none
integer, intent(inout) :: nint
real*8,intent(in) :: xin,yin,zin,sigx,sigy,sigz,gamma
real*8,intent(out) :: pintex,pintey,pintezy
!xp is x/sigma_x, yp is y/sigma_y, zp is z/sigma_z
!asp = sigma_x/sigma_y
!basp = sigma_x/(sigma_z*gamma)
real*8 :: asp,xp,yp,asp2,asp2m,basp,zp
real*8 :: xmin,xmax,x,h,t2,bt2,f1,f2,fx,yp2,zp2,sum0ex,sum0ey,sum0ez
real*8, dimension(nint+1) :: xvec,t2vec,t2sqrt,bt2vec,bt2sqrt
real*8, dimension(nint+1) :: f2xvec,f2yvec,f2zvec,f1vec,fxvec
integer :: i
real*8 :: exparg,sqrt2,sqrbt2,f1x

!enforce an odd grid points
if(mod(nint,2).eq.0) nint = nint+1

xmin = 0.0d0
xmax = 1.0d0
h = (xmax-xmin)/(nint-1)
xp = xin/sigx
yp = yin/sigy
zp = zin/sigz
asp = sigx/sigy
basp = sigx/(sigz*gamma)

x = xmax
t2 = asp**2+(1-asp**2)*x**2
sqrt2 = sqrt(t2)
bt2 = basp**2+(1-basp**2)*x**2
sqrbt2 = sqrt(bt2)

sum0ex = 0.0d0
sum0ey = 0.0d0
sum0ez = 0.0d0

```

```

!end point correction
!Ex
exparg = -(xp**2+yp**2/t2+zp**2/bt2)/2
f1 = x**2*exp(x**2*exparg)
f2 = sqrt2*sqrbt2
fx = f1/f2
sum0ex = sum0ex-fx*h

!Ey
f2 = t2*sqrt2*sqrbt2
fx = f1/f2
sum0ey = sum0ey-fx*h

!Ez
f2 = bt2*sqrt2*sqrbt2
fx = f1/f2
sum0ez = sum0ez-fx*h
!-----

do i = 1, nint
  xvec(i) = xmin + (i-1)*h
enddo
do i = 1, nint
  t2vec(i) = asp**2+(1.0d0-asp**2)*xvec(i)**2
enddo
do i = 1, nint
  t2sqrt(i) = sqrt(t2vec(i))
enddo
do i = 1, nint
  bt2vec(i) = basp**2+(1.0d0-basp**2)*xvec(i)**2
enddo
do i = 1, nint
  bt2sqrt(i) = sqrt(bt2vec(i))
enddo

do i = 1, nint
  f2xvec(i) = t2sqrt(i)*bt2sqrt(i)
enddo
do i = 1, nint
  f2yvec(i) = t2vec(i)*t2sqrt(i)*bt2sqrt(i)
enddo
do i = 1, nint
  f2zvec(i) = bt2vec(i)*t2sqrt(i)*bt2sqrt(i)
enddo

xp2 = xp**2
yp2 = yp**2
zp2 = zp**2
do i = 1, nint
  f1vec(i) = xvec(i)**2*exp(-xvec(i)**2*(xp2+yp2/t2vec(i)+zp2/bt2vec(i))/2)
enddo

```

```

!-----
      !for Ex
      do i = 1, nint
        fxvec(i) = f1vec(i)/f2xvec(i)
      enddo
      do i = 1, nint, 2
        sum0ex = sum0ex + 2*fxvec(i)*h
      enddo
      do i = 2, nint, 2
        sum0ex = sum0ex + 4*fxvec(i)*h
      enddo
      pintex = sum0ex/3.0d0

      !for Ey
      do i = 1, nint
        fxvec(i) = f1vec(i)/f2yvec(i)
      enddo
      do i = 1, nint, 2
        sum0ey = sum0ey + 2*fxvec(i)*h
      enddo
      do i = 2, nint, 2
        sum0ey = sum0ey + 4*fxvec(i)*h
      enddo
      pintey = sum0ey/3.0d0

      !for Ez
      do i = 1, nint
        fxvec(i) = f1vec(i)/f2zvec(i)
      enddo
      do i = 1, nint, 2
        sum0ez = sum0ez + 2*fxvec(i)*h
      enddo
      do i = 2, nint, 2
        sum0ez = sum0ez + 4*fxvec(i)*h
      enddo
      pinte = sum0ez/3.0d0

!-----
      end subroutine efldgauss

```

APPENDIX D: C++ IMPLEMENTATION OF 3D FIELD INTEGRALS

```

/*-----
!The following subroutine calculates three integrals used to calculate the
!3 space-charge kicks for a 3D Gaussian distribution in the paper.
!
!Inputs: nint - # of grid points (odd number),
!        xin, yin, zin - x, y and z coordinates, sigx, sigy, sigz -
!        x RMS size, y RMS size, and z RMS size, gamma - relativistic factor.
!Outputs: pintex - integral for Ex in Eq.(45); pintey - integral for Ey in Eq.(46);
!        pinteze - integral for Ez in Eq.(47).
!Author:: Ji Qiang with help of ChatGPT
!-----
*/
void efldgauss(int &nint, double xin, double yin, double zin,
              double sigx, double sigy, double sigz, double gamma,
              double &pintex, double &pintey, double &pinteze)
{
    // enforce an odd grid points
    if (nint % 2 == 0) nint += 1;

    double xmin = 0.0, xmax = 1.0;
    double h = (xmax - xmin) / (nint - 1);
    double xp = xin / sigx;
    double yp = yin / sigy;
    double zp = zin / sigz;
    double asp = sigx / sigy;
    double basp = sigx / (sigz * gamma);

    std::vector<double> xvec(nint), t2vec(nint), t2sqrt(nint), bt2vec(nint), bt2sqrt(nint);
    std::vector<double> f2xvec(nint), f2yvec(nint), f2zvec(nint), f1vec(nint), fxvec(nint);
    for (int i = 0; i < nint; ++i)
        xvec[i] = xmin + i * h;

    for (int i = 0; i < nint; ++i)
        t2vec[i] = asp * asp + (1.0 - asp * asp) * xvec[i] * xvec[i];

    for (int i = 0; i < nint; ++i)
        t2sqrt[i] = std::sqrt(t2vec[i]);

    for (int i = 0; i < nint; ++i)
        bt2vec[i] = basp * basp + (1.0 - basp * basp) * xvec[i] * xvec[i];

    for (int i = 0; i < nint; ++i)
        bt2sqrt[i] = std::sqrt(bt2vec[i]);

    for (int i = 0; i < nint; ++i)
        f2xvec[i] = t2sqrt[i] * bt2sqrt[i];

    for (int i = 0; i < nint; ++i)
        f2yvec[i] = t2vec[i] * t2sqrt[i] * bt2sqrt[i];

    for (int i = 0; i < nint; ++i)
        f2zvec[i] = bt2vec[i] * t2sqrt[i] * bt2sqrt[i];

    double xp2 = xp * xp, yp2 = yp * yp, zp2 = zp * zp;
    for (int i = 0; i < nint; ++i)
        f1vec[i] = xvec[i] * xvec[i] * std::exp(-xvec[i] * xvec[i] * (xp2 + yp2 / t2vec[i] + zp2 / bt2vec[i]
    ]) / 2.0);

```

```

// Simpson's rule integration
double sum0ex = 0.0, sum0ey = 0.0, sum0ez = 0.0;

// Ex
for (int i = 0; i < nint; ++i)
    fxvec[i] = f1vec[i] / f2xvec[i];
for (int i = 0; i < nint; i += 2)
    sum0ex += 2.0 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0ex += 4.0 * fxvec[i] * h;
//end point correction
sum0ex -= fxvec[nint-1] * h;
pintex = sum0ex / 3.0;

// Ey
for (int i = 0; i < nint; ++i)
    fxvec[i] = f1vec[i] / f2yvec[i];
for (int i = 0; i < nint; i += 2)
    sum0ey += 2.0 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0ey += 4.0 * fxvec[i] * h;
sum0ey -= fxvec[nint-1] * h;
pintey = sum0ey / 3.0;

// Ez
for (int i = 0; i < nint; ++i)
    fxvec[i] = f1vec[i] / f2zvec[i];
for (int i = 0; i < nint; i += 2)
    sum0ez += 2.0 * fxvec[i] * h;
for (int i = 1; i < nint; i += 2)
    sum0ez += 4.0 * fxvec[i] * h;
sum0ez -= fxvec[nint-1] * h;
pintez = sum0ez / 3.0;
}

```

APPENDIX E: DERIVATION OF THE FIRST-ORDER CURVATURE CORRECTION

The first-order correction to the two-dimensional Poisson equation in a circular accelerator system can be written as

$$\frac{\partial^2 \phi_1}{\partial x^2} + \frac{\partial^2 \phi_1}{\partial y^2} = -\frac{\rho(x, y, z)}{\epsilon_0} - h \frac{\partial \phi_0}{\partial x}. \quad (\text{E1})$$

Letting $\phi_1 = \phi_0 + \tilde{\phi}_1$, the above equation can be reduced to:

$$\frac{\partial^2 \phi_0}{\partial x^2} + \frac{\partial^2 \phi_0}{\partial y^2} = -\frac{n(x, y)\lambda(z)}{\epsilon_0}, \quad (\text{E2})$$

$$\frac{\partial^2 \tilde{\phi}_1}{\partial x^2} + \frac{\partial^2 \tilde{\phi}_1}{\partial y^2} = -h \frac{\partial \phi_0}{\partial x}. \quad (\text{E3})$$

The above equations can be solved by applying the Fourier transform for a transverse bi-Gaussian density distribution, which yields:

$$\tilde{\Phi}_1(k_x, k_y) = h \frac{ik_x}{\epsilon_0(k_x^2 + k_y^2)^2} \exp\left(-\frac{\sigma_x^2 k_x^2}{2} - \frac{\sigma_y^2 k_y^2}{2}\right). \quad (\text{E4})$$

Here, we have used

$$\tilde{\phi}_1(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} dk_x \int_{-\infty}^{\infty} dk_y \tilde{\Phi}_1(k_x, k_y) \exp(ik_x x + ik_y y) \quad (\text{E5})$$

and

$$\Phi_0(k_x, k_y) = h \frac{\lambda(z)}{\epsilon_0(k_x^2 + k_y^2)} \exp\left(-\frac{\sigma_x^2 k_x^2}{2} - \frac{\sigma_y^2 k_y^2}{2}\right). \quad (\text{E6})$$

Taking the inverse Fourier transform and making use of the following identities:

$$\frac{1}{(k_x^2 + k_y^2)^2} = \int_0^\infty t \exp[-t(k_x^2 + k_y^2)] dt, \quad (\text{E7})$$

$$\int_{-\infty}^\infty \exp(-Ax^2 + Bx) dx = \sqrt{\frac{\pi}{A}} \exp\left(\frac{B^2}{4A}\right), \quad (\text{E8})$$

$$\int_{-\infty}^\infty x \exp(-Ax^2 + Bx) dx = \frac{B}{2A} \sqrt{\frac{\pi}{A}} \exp\left(\frac{B^2}{4A}\right), \quad (\text{E9})$$

we obtain

$$\tilde{\phi}_1(x, y, z) = -h \frac{\lambda(z)x}{8\pi\epsilon_0} \int_0^\infty t dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right]}{(\sigma_x^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}. \quad (\text{E10})$$

By introducing a linear x term, the regularized solution (91) is obtained:

$$\tilde{\phi}_1(x, y, z) = -h \frac{\lambda(z)x}{8\pi\epsilon_0} \int_0^\infty t dt \frac{\exp\left[-\frac{x^2}{2(\sigma_x^2+t)} - \frac{y^2}{2(\sigma_y^2+t)}\right] - 1}{(\sigma_x^2+t)\sqrt{(\sigma_x^2+t)(\sigma_y^2+t)}}. \quad (\text{E11})$$

-
- [1] A. Friedman, D. P. Grote, and I. Haber, Three-dimensional particle simulation of heavy-ion fusion beams, *Phys. Fluids B* **4**, 2203 (1992).
 - [2] S. Machida and M. Ikegami, Simulation of space charge effects in a synchrotron, *AIP Conf. Proc.* **448**, 73 (1998).
 - [3] J. Qiang, R. D. Ryne, S. Habib, and V. Decyk, An object oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators, *J. Comput. Phys.* **163**, 434 (2000).
 - [4] P. N. Ostroumov and K. W. Shepard, Correction of beamsteering effects in low-velocity superconducting quarterwave cavities, *Phys. Rev. ST Accel. Beams* **11**, 030101 (2001).
 - [5] R. Duperrier, A radio frequency quadrupole code, *Phys. Rev. ST Accel. Beams* **3**, 124201 (2000).
 - [6] J. D. Galambos, S. Danilov, D. Jeon, J. A. Holmes, D. K. Olsen, F. Neri, and M. Plum, Comparison of simulated and observed beam profile broadening in the proton storage ring and the role of space charge, *Phys. Rev. ST Accel. Beams* **3**, 034201 (2000).
 - [7] G. Franchetti, I. Hofmann, M. Giovannozzi, M. Martini, and E. Metral, Space charge and octupole driven resonance trapping observed at the CERN Proton Synchrotron, *Phys. Rev. ST Accel. Beams* **6**, 124201 (2003).
 - [8] J. Qiang, S. Lidia, R. D. Ryne, and C. Limborg-Deprey, Three-dimensional quasistatic model for high brightness beam dynamics simulation, *Phys. Rev. ST Accel. Beams* **9**, 044204 (2006).
 - [9] J. Amundson, P. Spentzouris, J. Qiang, and R. Ryne, Synergia: An accelerator modeling tool with 3-D space charge, *J. Comput. Phys.* **211**, 229 (2006).
 - [10] https://amas.web.psi.ch/docs/opal/opal_user_guide-1.6.0.pdf.
 - [11] <https://synergia.fnal.gov/>.
 - [12] <https://github.com/PyORBIT-Collaboration/py-orbit>.
 - [13] A. Oeftiger and S. Hegglin, Space charge modules for PyHEADTAIL, in *Proceedings of the HB2016* (2016), p. 124, <https://proceedings.jacow.org/hb2016/papers/mopr025.pdf>.
 - [14] M. Schwinzerl, R. De Maria, K. Paraschou, H. Bartosik, G. Iadarola, and A. Oeftiger, Optimising and extending a single-particle tracking library for high parallel performance, in *Proceedings of the 12th International Particle Accelerator Conference (IPAC2021)*, Campinas, SP, Brazil (JACoW, Geneva, Switzerland, 2021), 10.18429/JACoW-IPAC2021-THPAB190.
 - [15] <https://github.com/xsuite/xsuite>.
 - [16] J. Qiang, Symplectic multiparticle tracking model for self-consistent space-charge simulation, *Phys. Rev. Accel. Beams* **20**, 014203 (2017).
 - [17] J. Qiang, Symplectic particle-in-cell model for space-charge beam dynamics simulation, *Phys. Rev. Accel. Beams* **21**, 054201 (2018).
 - [18] E. Forest and R. D. Ruth, Fourth-order symplectic integration, *Physica (Amsterdam)* **43D**, 105 (1990).
 - [19] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A* **150**, 262 (1990).
 - [20] R. D. Ryne, Computational methods in accelerator physics, US particle accelerator class note (2012), <https://uspas.fnal.gov/materials/09UNM/ComputationalMethods.pdf>.
 - [21] J. Qiang and R. Gluckstern, Three-dimensional Poisson solver for a charged beam with large aspect ratio in a conducting pipe, *Comput. Phys. Commun.* **160**, 120 (2004).
 - [22] A. W. Chao, *Special Topics in Accelerator Physics* (World Scientific Publishing Co. Pte. Ltd., Singapore, 2022).
 - [23] <https://mathworld.wolfram.com/Euler-MascheroniConstant.html>.
 - [24] M. Bassetti and G. A. Erskine, Closed expression for the electrical field of a two-dimensional Gaussian charge, Report No. CERN-ISR-TH-80-06, CERN, Switzerland, 1980.
 - [25] I. Hofmann, A. Oeftiger, and O. Boine-Frankenheim, Self-consistent long-term dynamics of space charge driven resonances in 2D and 3D, *Phys. Rev. Accel. Beams* **24**, 024201 (2021).
 - [26] A. Oeftiger, O. Boine-Frankenheim, V. Chetvertkova, V. Kornilov, D. Rabusov, and S. Sorge, Simulation study of the space charge limit in heavy-ion synchrotrons, *Phys. Rev. Accel. Beams* **25**, 054402 (2022).
 - [27] J. Qiang, M. Furman, and R. Ryne, A parallel particle-in-cell model for beam-beam interaction in high energy ring colliders, *J. Comput. Phys.* **198**, 278 (2004).
 - [28] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Adam Hilger, New York, 1988).

-
- [29] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, Society for Industrial and Applied Mathematics (1977), [10.1137/1.9781611970425](#).
- [30] B. Fornberg, *A Practical Guide to Pseudospectral Methods* (Cambridge University Press, Cambridge, England, 1998).
- [31] J. Qiang and R. D. Ryne, Parallel 3D Poisson solver for a charged beam in a conducting pipe, *Comput. Phys. Commun.* **138**, 18 (2001).