

Implementation of the Integrated Green's Function Method for 3D Poisson's Equation in a Large Aspect Ratio Computational Domain

Ji Qiang, Chad Mitchell, Remi Lehe, Arianna Formenti

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Email: jqiang@lbl.gov

How to cite this paper: Qiang, J., Mitchell, M., Lehe, R. and Formenti, A. (2024) Implementation of the Integrated Green's Function Method for 3D Poisson's Equation in a Large Aspect Ratio Computational Domain. *Journal of Software Engineering and Applications*, 17, 740-749.

<https://doi.org/10.4236/jsea.2024.179039>

Received: August 22, 2024

Accepted: September 17, 2024

Published: September 20, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The solution of Poisson's Equation plays an important role in many areas, including modeling high-intensity and high-brightness beams in particle accelerators. For the computational domain with a large aspect ratio, the integrated Green's function method has been adopted to solve the 3D Poisson equation subject to open boundary conditions. In this paper, we report on the efficient implementation of this method, which can save more than a factor of 50 computing time compared with the direct brute force implementation and its improvement under certain extreme conditions.

Keywords

Green's Function, Poisson Equation, Particle Accelerator

1. Introduction

The solution of the three-dimensional (3D) Poisson equation plays an important role in many fields. In particle accelerator research, the nonlinear space-charge effects due to Coulomb interactions among charged particles play an important role in high-intensity and high-brightness beam physics since such effects can cause beam quality degradation, halo formation, and even particle losses. In order to study the space-charge effects in the charged particle beam self-consistently, one needs to solve the 3D Poisson equation at each time step with the evolving particle density distribution. When the beam (e.g. electron beam) energy increases, in the beam frame computational domain, the longitudinal to transverse aspect ratio becomes larger and larger due to the relativistic effect. In such a case, an integrated Green's function method was developed and showed to be more effective than the standard Green's function method to solve the 3D Poisson

equation subject to open boundary conditions since it does not require resolving the variation of the Green's function through the computational domain [1]. In recent years, this method has been adopted in a number of particle accelerator beam dynamics codes for the self-consistent simulation of the space-charge effects [2]-[11]. However, there is no publication on how this method is implemented. Direct implementation of the mathematical expression of the integrated Green's function method can result in substantially more computational cost. In this paper, we report on an efficient implementation method that reduces the computational cost by more than a factor 50 and alternative expressions that avoid cancellation errors under certain extreme conditions.

2. Integrated Green's Function Solution to 3D Poisson's Equation

The electric potential ϕ of the space-charge fields satisfies the following Poisson's Equation:

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0} \tag{1}$$

where ρ is the charge density distribution function, and ϵ_0 is the permittivity in vacuum. The solution of the above Poisson's Equation subject to the 3D free space open boundary condition can be written as:

$$\phi(x, y, z) = \frac{1}{4\pi\epsilon_0} \iiint G(x-x', y-y', z-z') \rho(x', y', z') dx' dy' dz' \tag{2}$$

where the Green's function G is given by:

$$G(x-x', y-y', z-z') = \frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}} \tag{3}$$

To compute the above integral numerically, we define a computational domain containing the beam with a range of $(0, L_x)$, $(0, L_y)$ and $(0, L_z)$, and discretize each dimension using N_x , N_y and N_z grid points. Then, we decompose this integral in the entire computational domain as a summation of $N_x \times N_y \times N_z$ small cell integrals with each grid point located at the center of the cell.

$$\begin{aligned} \phi(x_i, y_j, z_k) = & \frac{1}{4\pi\epsilon_0} \sum_{i'=1}^{N_x} \sum_{j'=1}^{N_y} \sum_{k'=1}^{N_z} \int_{x_{i'}-h_x/2}^{x_{i'}+h_x/2} dx' \int_{y_{j'}-h_y/2}^{y_{j'}+h_y/2} dy' \int_{z_{k'}-h_z/2}^{z_{k'}+h_z/2} dz' \\ & \times G(x_i-x', y_j-y', z_k-z') \rho(x', y', z') \end{aligned} \tag{4}$$

where $h_x = L_x/(N_x - 1)$, $h_y = L_y/(N_y - 1)$, and $h_z = L_z/(N_z - 1)$. If we assume that the charge density is constant within each cell centered at the grid point (x_i, y_j, z_k) , *i.e.* $\rho(x', y', z') = \rho(x_i, y_j, z_k)$, from the above equation, the electric potential on this grid point can be approximated as:

$$\phi(x_i, y_j, z_k) = \frac{1}{4\pi\epsilon_0} \sum_{i'=1}^{N_x} \sum_{j'=1}^{N_y} \sum_{k'=1}^{N_z} \bar{G}(x_i-x_{i'}, y_j-y_{j'}, z_k-z_{k'}) \rho(x_{i'}, y_{j'}, z_{k'}) \tag{5}$$

where $x_i = (i-1)h_x$, $y_j = (j-1)h_y$, and $z_k = (k-1)h_z$, and the effective Green

function \bar{G} is given as:

$$\begin{aligned} \bar{G}(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'}) &= \int_{x_{i'} - h_x/2}^{x_i + h_x/2} dx' \int_{y_{j'} - h_y/2}^{y_j + h_y/2} dy' \int_{z_{k'} - h_z/2}^{z_k + h_z/2} dz' \times G(x_i - x', y_j - y', z_k - z') \end{aligned} \tag{6}$$

where h_x , h_y , and h_z are cell size in each dimension, respectively. The above integral can be calculated analytically in a closed form for the Green's function given in Equation (3) as [12]:

$$\begin{aligned} \bar{G}(x, y, z) &= f\left(x + \frac{h_x}{2}, y + \frac{h_y}{2}, z + \frac{h_z}{2}\right) - f\left(x + \frac{h_x}{2}, y + \frac{h_y}{2}, z - \frac{h_z}{2}\right) \\ &+ f\left(x - \frac{h_x}{2}, y - \frac{h_y}{2}, z + \frac{h_z}{2}\right) - f\left(x - \frac{h_x}{2}, y - \frac{h_y}{2}, z - \frac{h_z}{2}\right) \\ &+ f\left(x + \frac{h_x}{2}, y - \frac{h_y}{2}, z - \frac{h_z}{2}\right) - f\left(x + \frac{h_x}{2}, y + \frac{h_y}{2}, z - \frac{h_z}{2}\right) \\ &+ f\left(x - \frac{h_x}{2}, y + \frac{h_y}{2}, z - \frac{h_z}{2}\right) - f\left(x - \frac{h_x}{2}, y + \frac{h_y}{2}, z + \frac{h_z}{2}\right) \end{aligned} \tag{7}$$

where

$$\begin{aligned} f(x, y, z) &= yz \ln(x+r) + xz \ln(y+r) + xy \ln(z+r) \\ &- \frac{z^2}{2} \arctan\left(\frac{xy}{zr}\right) - \frac{y^2}{2} \arctan\left(\frac{xz}{yr}\right) - \frac{x^2}{2} \arctan\left(\frac{yz}{xr}\right) \end{aligned} \tag{8}$$

where $r = \sqrt{x^2 + y^2 + z^2}$. With the effective Green's function \bar{G} , the summation of Equation (5) can be computed effectively using the FFT method [1] [13].

3. Efficient Implementation

The effective Green's function in the above equation involves eight f function evaluations. The range of the variables in the function f covers the range from $-L_x$ to L_x , $-L_y$ to L_y , and $-L_z$ to L_z , which suggests two times grid points are needed in each dimension for this function. However, a careful check of Equation (6) suggests that the effective Green's function should have the same symmetry property as the original Green's function, *i.e.*, changing the sign of an individual variable in Equation (7) will not affect the value of the function. This can be seen from Equation (6),

$\bar{G}(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'}) = G(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'}) h_x h_y h_z + O(h_x^3 h_y^3 h_z^3)$ or by using Equation (7) and $f(-x, y, z) = -f(x, y, z) + yz \ln(y^2 + z^2)$ results in $\bar{G}(-x, y, z) = \bar{G}(x, y, z)$ for x (same applies to y and z). Hence, only the first quadrant of the effective Green's function is needed. This saves the computational cost by about a factor of eight. Furthermore, by computing the f function on one corner of the integrated cell for an extended grid (*i.e.*, $N+1$ instead of N) $G^{mp}(i, j, k) = f(x_i - h_x/2, y_j - h_y/2, z_k - h_z/2)$ for $i = 1, \dots, N_x + 1$, $j = 1, \dots, N_y + 1$, and $k = 1, \dots, N_z + 1$, the f function values at the other seven corners of the cell in the Equation (7) can be obtained from the shift of this function on the grid. For example, the f function value at the upper right corner of the cell

will be $f(x_i + h_x/2, y_j + h_y/2, z_k + h_z/2) = G^{mp}(i+1, j+1, k+1)$ for $i=1, \dots, N_x$, $j=1, \dots, N_y$, and $k=1, \dots, N_z$. Only one f function evaluation is needed instead of eight function evaluations in Equation (7). This saves the computational cost by another factor of eight. In total, the efficient implementation can save the computing time of the effective Green's function by more than a factor of 60 compared with the direct brute force implementation. An illustration of this implementation in Fortran90 is given in **Figure 1**.

```

do k0 = 1, nsizez+1
  do j0 = 1, nsizey+1
    do i0 = 1, nx+2
      kkk = k0 - 1
      jjj = j0 - 1
      iii = i0 - 1
      vv(1) = iii*hx-hx/2
      vv(2) = jjj*hy-hy/2
      vv(3) = kkk*hz-hz/2
      rr = sqrt(vv(1)**2+vv(2)**2+vv(3)**2)

      aa = -vv(3)**2*atan(vv(1)*vv(2)/(vv(3)*rr))/2
      bb = -vv(2)**2*atan(vv(1)*vv(3)/(vv(2)*rr))/2
      cc = -vv(1)**2*atan(vv(2)*vv(3)/(vv(1)*rr))/2
      dd = vv(2)*vv(3)*log(vv(1)+rr)
      ee = vv(1)*vv(3)*log(vv(2)+rr)
      ff = vv(1)*vv(2)*log(vv(3)+rr)
      gg2 = aa + bb + cc + dd + ee + ff

      grntmp(i0,j0,k0) = recfourpi*gg2
    enddo
  enddo
enddo

do k0 = 1, nsizez
  do j0 = 1, nsizey
    do i0 = 1, nx+1
      grn(i0,j0,k0) = grntmp(i0+1,j0+1,k0+1)-grntmp(i0,j0+1,k0+1)-grntmp(i0+1,j0,k0+1)+&
        grntmp(i0,j0,k0+1)-grntmp(i0+1,j0+1,k0)+grntmp(i0,j0+1,k0)+&
        grntmp(i0+1,j0,k0)-grntmp(i0,j0,k0)
    enddo
  enddo
enddo

```

Figure 1. The Fortran90 implementation of the effective Green's function calculation.

As a test of the practical performance of the above implementation, we measured the computing time of the above implementation and the computing time of the brute force implementation with a variety of problem sizes $N \times N \times N$. **Figure 2** shows the speedup of the above implementation as a function of one-dimensional grid points in the real application. More than 50 speedup is achieved for problem sizes greater than $64 \times 64 \times 64$.

From the above implementation, it is seen that the variable x, y, z in Equation (8) will be less than zero only when $i0=1$, or $j0=1$, or $k0=1$. The negative value of the variable could cause cancellation error in the evaluation of $x+r$, or $y+r$, or $z+r$. In some extreme applications, e.g., with very high electron beam energy, the longitudinal bunch length in the beam frame can be much larger than the transverse beam size. This results in the $|z| \gg r$ and a large cancellation error in $z+r$ and even overflow of $\ln(z+r)$ for $z < 0$.

The aforementioned cancellation error is due to the subtraction of two close real numbers on finite precision digital computers. This numerical cancellation error can be mitigated by either increasing the precision of each number on a computer or by using the following alternative expressions to replace the original summation in the Equation (8) or the Equation (8) itself.

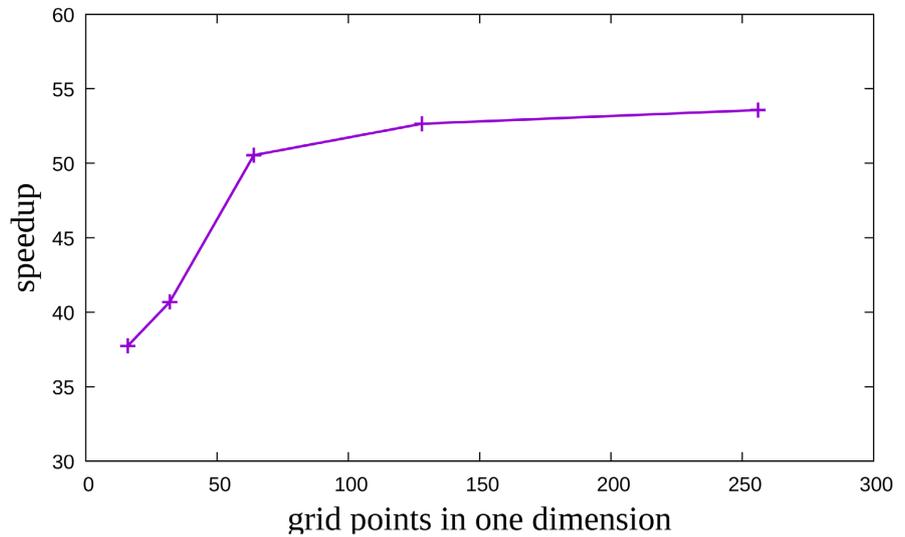


Figure 2. The speedup of the efficient implementation as a function of one-dimensional grid points.

Firstly, we can declare the variables x , y , z , and r in the above implementation as quadruple precision. This substantially increases the accuracy of each variable on a digital computer and reduces the cancellation error due to the double-precision representation of a variable.

Secondly, by making use of the function relationship $\operatorname{arcsinh}(x) = \ln(x + \sqrt{1 + x^2})$, neglecting the terms that do not contribute to the field calculation (*i.e.* $\frac{1}{2}yz \ln(y^2 + z^2) + \frac{1}{2}xz \ln(x^2 + z^2) + \frac{1}{2}xy \ln(x^2 + y^2)$), the Equation (8) can be rewritten as:

$$\begin{aligned}
 f(x, y, z) = & yz \operatorname{arcsinh}\left(\frac{x}{\sqrt{y^2 + z^2}}\right) + xz \operatorname{arcsinh}\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \\
 & + xy \operatorname{arcsinh}\left(\frac{z}{\sqrt{x^2 + y^2}}\right) - \frac{z^2}{2} \arctan\left(\frac{xy}{zr}\right) \\
 & - \frac{y^2}{2} \arctan\left(\frac{xz}{yr}\right) - \frac{x^2}{2} \arctan\left(\frac{yz}{xr}\right)
 \end{aligned} \tag{9}$$

This equation avoids the sum of two close but opposite sign variables in the original equation and the resultant cancellation error.

Thirdly, one can define a small tolerance number ϵ (e.g. 10^{-10}) and use the Taylor expansion to obtain the following approximation:

$$\begin{aligned}
 x + r &= x + |x| + \frac{1}{2} \frac{y^2 + z^2}{|x|}, \quad \text{for } |x + r| < \epsilon \\
 y + r &= y + |y| + \frac{1}{2} \frac{x^2 + z^2}{|y|}, \quad \text{for } |y + r| < \epsilon \\
 z + r &= z + |z| + \frac{1}{2} \frac{x^2 + y^2}{|z|}, \quad \text{for } |z + r| < \epsilon
 \end{aligned} \tag{10}$$

The above approximation is used in the natural logarithm function to mitigate the cancellation error.

Fourthly, one can rewrite the expression in the natural logarithm function of Equation (8) as:

$$\begin{aligned} x+r &= \frac{y^2+z^2}{r-x}, \quad \text{for } x < 0 \\ y+r &= \frac{x^2+z^2}{r-y}, \quad \text{for } y < 0 \\ z+r &= \frac{x^2+y^2}{r-z}, \quad \text{for } z < 0 \end{aligned} \quad (11)$$

The above expression turns the original summation of two opposite sign variables into an expression that includes subtraction of these variables and avoids the cancellation error.

4. A Benchmark Example

As a test, we used a 100 pC electron beam with a 3D Gaussian density distribution and computed the electric fields in the beam frame from all of the above schemes and from a semi-analytical solution. The semi-analytical electric potential in the rest beam frame for a Gaussian density distribution is given by [14]:

$$\phi(x, y, z) = \frac{Q}{4\pi\epsilon_0} \sqrt{\frac{2}{\pi}} \int_0^\infty \frac{e^{\frac{-\lambda^2 x^2}{2(\lambda^2 \sigma_x^2 + 1)}} e^{\frac{-\lambda^2 y^2}{2(\lambda^2 \sigma_y^2 + 1)}} e^{\frac{-\lambda^2 z^2}{2(\lambda^2 \sigma_z^2 + 1)}}}{\sqrt{(\lambda^2 \sigma_x^2 + 1)(\lambda^2 \sigma_y^2 + 1)(\lambda^2 \sigma_z^2 + 1)}} d\lambda \quad (12)$$

where Q is the total charge of the beam, and σ_x , σ_y , σ_z are the RMS sizes of the beam in each dimension. In this test, we assumed that $\sigma_x = \sigma_y = 0.5$ mm and $\sigma_z = 1$ mm. We varied electron beam kinetic energy so that the electron beam longitudinal bunch length $\gamma\sigma_z$ in the beam frame increased with the increase of beam energy. This increases the longitudinal-to-transverse aspect ratio with the increase of beam energy in the beam frame. We used $129 \times 129 \times 257$ grid points to solve the Poisson equation using the above integrated Green's function method in the rest beam frame. Electric fields are numerically computed from $E = -\nabla\phi$ in this frame using a second-order finite difference approximation.

Figure 3 shows horizontal and longitudinal electric fields as a function of longitudinal coordinate z in the beam frame from the semi-analytical solution and from the original integrated Green's function method at 100 GeV electron beam energy. It is seen that the electric fields from the numerical integrated Green's function solution agree with those from the semi-analytical solution very well.

In order to check the valid regime of the integrated Green's function method, we varied the electron beam kinetic energy from 100 GeV to 100 TeV. This results in about 2×10^5 to 2×10^8 longitudinal-to-transverse aspect ratio for the computational cells in the rest beam frame. **Figure 4** shows the maximum horizontal relative electric field error and the maximum longitudinal relative electric field error as a function of the electron beam kinetic energy using the original double

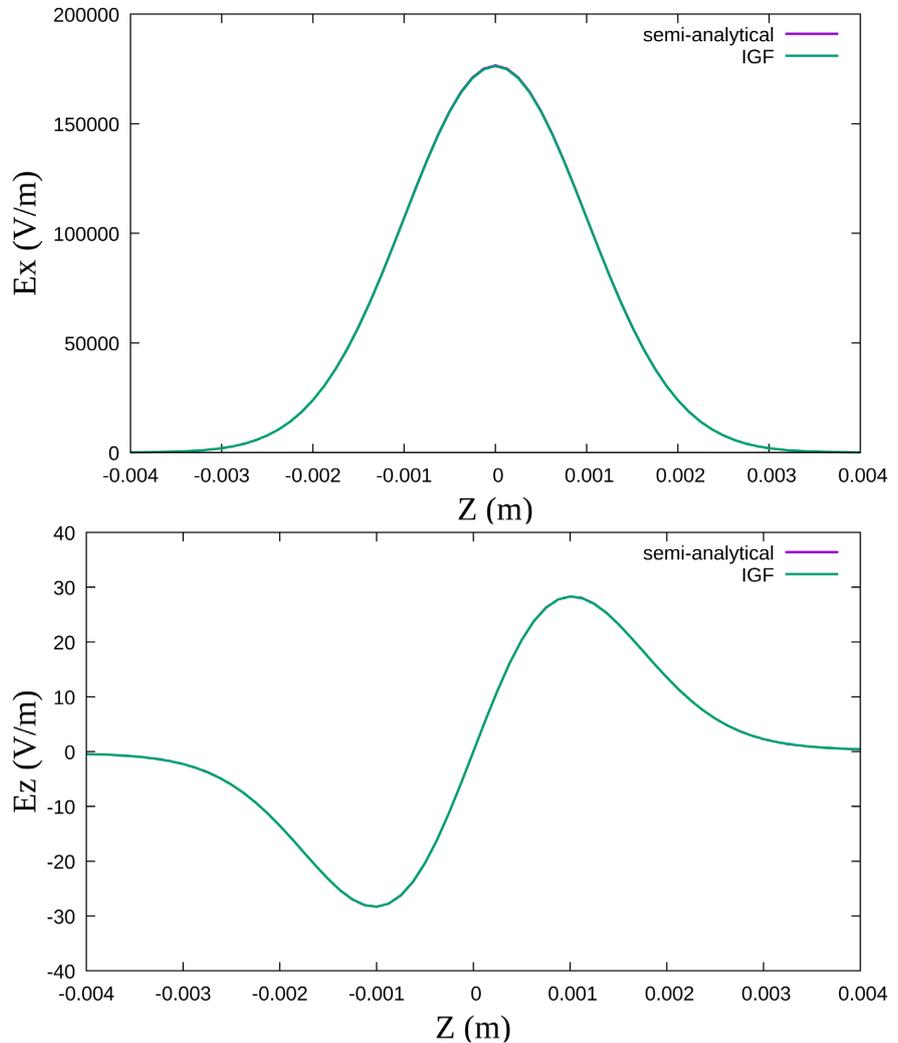


Figure 3. Horizontal electric field E_x (top) and longitudinal electric field E_z (bottom) as a function of longitudinal coordinate z in the beam from the semi-analytical solution (magenta) and from the integrated Green’s function method (green) at 100 GeV electron beam energy. There are two lines in each plot sitting on top of each other.

precision logarithm integrated Green’s function (IGF), the quadruple precision logarithm IGF, the arcsinh IGF, the approximated logarithm IGF, and the rewritten logarithm IGF. Here, the differences between the numerical solutions from the integrated Green’s function method and the semi-analytical solutions were calculated on the three-dimensional $33 \times 33 \times 65$ grid points for both the horizontal electric field and the longitudinal electric field. These differences are normalized by the maximum values of the horizontal electric field and the longitudinal field from the semi-analytical model, respectively, to attain the 3D relative errors. The maximum relative errors are attained from the relative errors on the 3D grid. It is seen that all five integrated Green’s function implementations yield nearly the same less than 0.1% relative errors up to 50 TeV beam energy. This is probably due to the fact that the cancellation error occurs only $z = -\frac{h_z}{2}$ and

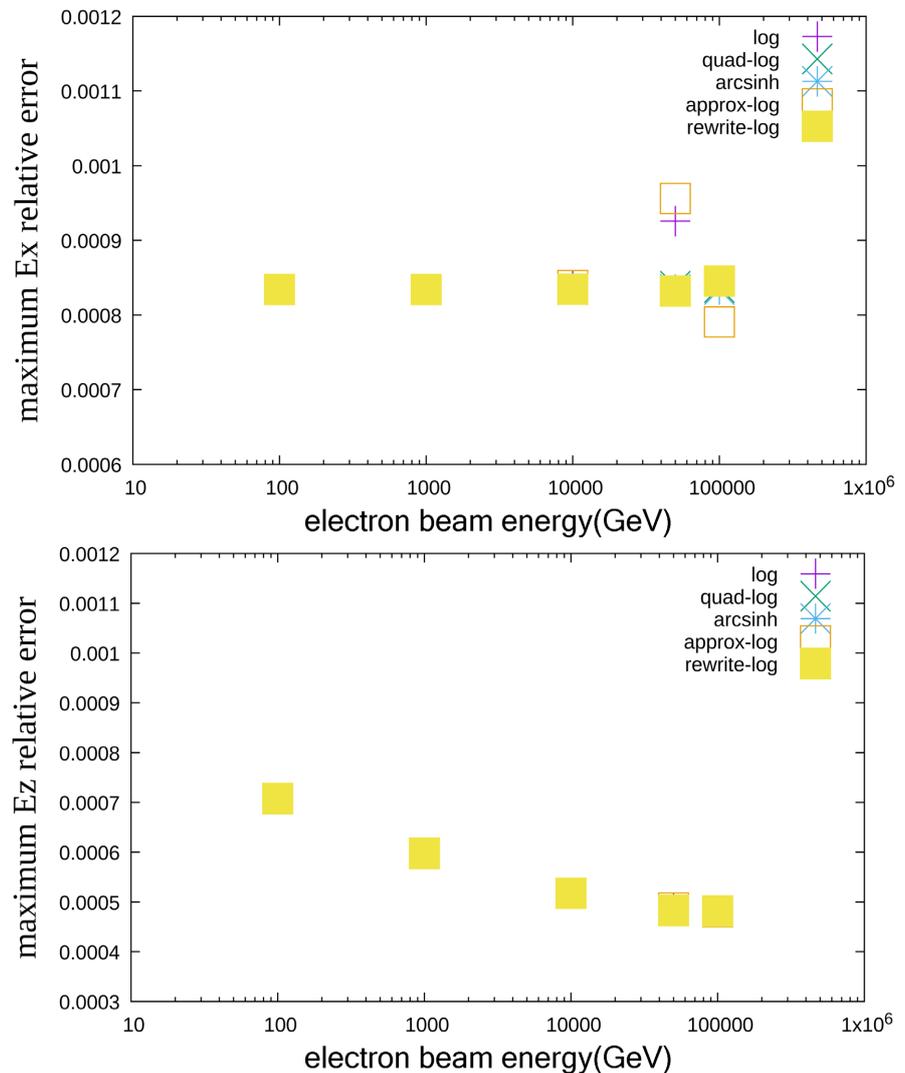


Figure 4. The maximum horizontal relative electric field error (top) and the maximum longitudinal relative electric field error (bottom) as a function of the electron beam kinetic energy using the original double precision logarithm IGF (plus), the quadruple precision logarithm IGF (cross), the arcsinh IGF (star), the approximated logarithm IGF (empty square), and the rewritten logarithm IGF (solid square).

has a small contribution to the total 3D summation. At the 100 TeV electron beam energy, the original logarithm integrated Green's function fails due to the cancellation error in the evaluation of $z+r$ and the resultant overflow of the logarithm function. The other four mitigation schemes all work well and yield less than 0.1% relative errors. The computational cost of the quadruple precision implementation of the IGF is the highest (more than 10 times the original IGF) due to the lack of direct hardware support for such operations. The computational cost of the arcsinh implementation of the IGF is about a factor of two of that of the original IGF. The computational costs of the approximated IGF and the rewritten IGF implementations are close to that of the original IGF, while the rewritten IGF implementation does not need to specify any tolerance number.

5. Conclusion

In this paper, we present an efficient implementation of the integrated Green's function method to solve the 3D Poisson's Equation in a large aspect ratio computational domain subject to the open boundary conditions. Our implementation suggests more than a factor of 50 reduction of computational cost compared with the direct brute force implementation. Furthermore, several alternative expressions are proposed to avoid cancellation errors and work well under extreme conditions. This implementation can have applications in many fields, such as high-brightness beam physics in particle accelerators, plasma physics, and micromagnetics, where the solution of 3D Poisson equation in the large aspect ratio computational domain is needed.

Acknowledgements

This work was supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and used computer resources at the National Energy Research Scientific Computing Center.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Qiang, J., Lidia, S., Ryne, R.D. and Limborg-Deprey, C. (2006) Three-Dimensional Quasistatic Model for High Brightness Beam Dynamics Simulation. *Physical Review Special Topics—Accelerators and Beams*, **9**, Article 044204. <https://doi.org/10.1103/physrevstab.9.044204>
- [2] Qiang, J., Ryne, R.D., Venturini, M., Zholents, A.A. and Pogorelov, I.V. (2009) High Resolution Simulation of Beam Dynamics in Electron Linacs for X-Ray Free Electron Lasers. *Physical Review Special Topics—Accelerators and Beams*, **12**, Article 100702. <https://doi.org/10.1103/physrevstab.12.100702>
- [3] Floettmann, K. (2017) ASTRA: A Space-Charge TRacking Algorithm. https://www.desy.de/~mpyflo/Astra_manual/Astra-Manual_V3.2.pdf
- [4] Adelman, A., *et al.* (2018) The OPAL (Object Oriented Parallel Accelerator Library) Framework. Paul Scherrer Institut PSI-PR-08-02. <https://gitlab.psi.ch/OPAL/src/wikis/home>
- [5] Tomin, S., *et al.* (2017) OCELOT as a Framework for Beam Dynamics Simulations of X-Ray Sources. *Proceedings of the 8th International Particle Accelerator Conference*, Copenhagen, 14-19 May 2017, 2642.
- [6] Mayes, C.E., Ryne, R.D. and Sagan, D.C. (2018) 3D Space Charge in BMAD. *Proceedings of the 9th International Particle Accelerator Conference*, Vancouver, 29 April-4 May 2018, 3428.
- [7] Oeftiger, A. and Hegglin, S. (2016) Space Charge Modules for PyHEADTAIL. *Proceedings of HB 2016*, Malmo, 3-8 July 2016, 124.
- [8] Latina, A. (2020) RF-Track Reference Manual. Tech. <https://zenodo.org/record/3887085>
- [9] Iadarola, G., *et al.* (2023) Xsuite: An Integrated Beam Physics Simulation Framework.

Proceedings of HB 2023, Geneva, 9-13 October 2023, 73.

- [10] Fedeli, L., Huebl, A., Boillod-Cerneux, F., Clark, T., Gott, K., Hillairet, C., et al. (2022) Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-in-Cell Simulations on Exascale-Class Supercomputers. *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, Dallas, 13-18 November 2022, 1-12.
<https://doi.org/10.1109/sc41404.2022.00008>
- [11] Huebl, A., Lehe, R., Mitchell, C.E., Qiang, J., Ryne, R.D., Sandberg, R.T. and Vay, J. (2022) Next Generation Computational Tools for the Modeling and Design of Particle Accelerators at Exascale. *Proceedings of 2022 North American Particle Accelerator Conference*, Albuquerque, 7-12 August 2022, 302-306.
- [12] Qiang, J., Lidia, S., Ryne, R.D. and Limborg-Deprey, C. (2007) Erratum: Three-Dimensional Quasistatic Model for High Brightness Beam Dynamics Simulation. *Physical Review Special Topics—Accelerators and Beams*, **10**, Article 129901.
<https://doi.org/10.1103/physrevstab.10.129901>
- [13] Hockney, R. and Eastwood, J. (1988) *Computer Simulation Using Particles*. Taylor & Francis. <https://doi.org/10.1201/9781439822050>
- [14] Stupakov, G. and Penn, G. (2018) *Classical Mechanics and Electromagnetism in Accelerator Physics*. Springer.