THREE-DIMENSIONAL INTEGRATED GREEN FUNCTIONS FOR THE POISSON EQUATION*

D. T. Abell, P. J. Mullowney, K. Paul, V. H. Ranjbar, Tech-X Corp.
J. Qiang, R. D. Ryne, LBNL

Abstract

The standard implementation of using FFTs to solve the Poisson equation with open boundary conditions on a Cartesian grid loses accuracy when the change in ρG (the product of the charge density with the Green function) over a mesh cell becomes nonlinear; this is commonly encountered in high aspect ratio situations and results in poor efficiency due to the need for a very large number of grid points. A modification which solves this problem, the integrated Green function (IGF), has been implemented in two dimensions using linear basis functions and in three dimensions using constant basis functions. But, until recently, it has proved to be very difficult to implement IGF in three dimensions using linear basis functions. Recently significant progress has been made. We present both the implementation and test results for the three-dimensional extension.

INTRODUCTION

Poisson solvers used in quasi-static electric and gravitational particle-in-cell simulations generally fail when the grid aspect ratio differs significantly from unity [1]. Important problems that involve extreme aspect ratios include long beams in rf accelerators, beams in induction linacs, and galactic collisions. For these applications, standard grid-based approaches [2] require large numbers of grid points, leading to prohibitive computational effort. As a result, the accurate modeling of high-aspect-ratio systems is extremely challenging.

Difficulties with the standard approach arise when variations in the Green function and the charge density occur on very different scales. If it is the charge density that varies on the larger scale, then one can take advantage of the fact that the Green function is known *a priori*. Integrating the Green function with appropriate basis functions allows one to make full use of that *a priori* knowledge. This has been done in very successfully in two dimensions [3], see Figure 1. But three dimensions, as always, presents a greater challenge. The three-dimensional case has been addressed using constant basis functions [4], but never before with linear basis functions.

THE INTEGRATED GREEN FUNCTION

The scalar potential can be written as a convolution of the charge (or particle) density with the appropriate Green function:

$$\phi(\mathbf{r}) = \int \rho(\mathbf{r}) G(\mathbf{r} - \mathbf{r}') d\mathbf{r}'. \qquad (1)$$

Discretizing this integral over a grid, one obtains the discrete convolution

$$\phi_{ijk} = V_h \sum_{i'j'k'} \rho_{i'j'k'} G_{i-i',j-j',k-k'}, \qquad (2)$$

where V_h denotes the volume of a grid cell. This sum can be computed efficiently using FFTs [2].

The discretization (2) is equivalent (modulo boundary terms) to using the trapezoidal rule to approximate the convolution integral (1). To yield accurate results, this approach requires that the product ρG vary slowly over a grid cell; but for large aspect ratios, G can vary much more rapidly than ρ. Rather than discretize the domain on the finer scale, one may take advantage of the known form of the Green function G: Write (1) as a sum of integrals over each cell; and within each cell approximate the charge density as

$$\begin{aligned}\rho(\mathbf{r}_{i'j'k'} + \boldsymbol{\epsilon}) = \rho_{i'j'k'} &\left(1 - \frac{\epsilon_x}{h_x}\right)\left(1 - \frac{\epsilon_y}{h_y}\right)\left(1 - \frac{\epsilon_z}{h_z}\right) \\ &+ \rho_{i'+1,j'k'} \left(\frac{\epsilon_x}{h_x}\right)\left(1 - \frac{\epsilon_y}{h_y}\right)\left(1 - \frac{\epsilon_z}{h_z}\right) \\ &+ \cdots + \rho_{i'+1,j'+1,k'+1} \left(\frac{\epsilon_x}{h_x}\right)\left(\frac{\epsilon_y}{h_y}\right)\left(\frac{\epsilon_z}{h_z}\right), \qquad (3)\end{aligned}$$

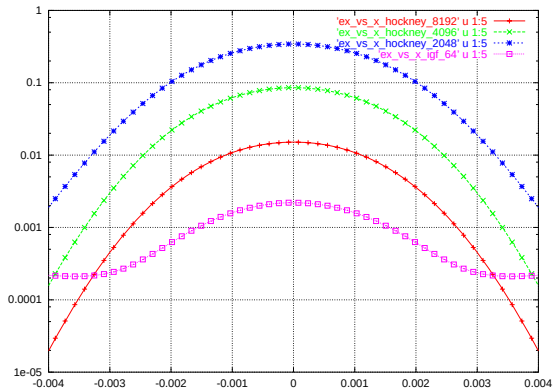

Figure 1: Errors in computed electric field values E_x as a function of x for a two-dimensional system [3]. The blue, green, and red data, all based on a standard FFT-based Poisson solver [2], use increasingly more grid points along the x axis. That algorithm requires more than 8000 points to achieve an accuracy that is roughly 1% or better. By contrast, the two-dimensional IGF algorithm (light violet) does substantially better with just 64 points.

* Work supported by the U.S. DOE Office of Science, Office of Nuclear Physics under grant DE-FG02-03ER83796

where (h_x, h_y, h_z) denote the dimensions of each grid cell, and $\epsilon_j \in [0, h_j]$. This approximation reproduces the values on the grid nodes, and within a cell it interpolates linearly along each axis. Pulling the nodal charge density values out of the integrals, shifting indices, and rearranging terms, we can recover the convolution sum of (2), but with the Green function replaced by an *effective* Green function G^{eff}:

$$\phi_{ijk} = V_h \sum_{i'j'k'} \rho_{i'j'k'} G^{\text{eff}}_{i-i',j-j',k-k'}. \qquad (4)$$

This G^{eff}, the integrated Green function (IGF), is a sum of eight terms—one for each of the terms in (3):

$$G^{\text{eff}}_{\Delta i,\Delta j,\Delta k} = \frac{1}{V_h^2} \sum_{r,s,t=0}^{1} \int_{h_x(\Delta i+r-1)}^{h_x(\Delta i+r)} du \int_{h_y(\Delta j+s-1)}^{h_y(\Delta j+s)} dv \int_{h_z(\Delta k+t-1)}^{h_z(\Delta k+t)} dw\,(-1)^{r+s+t} \times [u - h_x(\Delta i - 1 + 2r)][v - h_y(\Delta j - 1 + 2s)] \times [w - h_z(\Delta k - 1 + 2t)]\, G(u, v, w). \qquad (5)$$

Given a grid indexed by $i \in [0, N_x]$, $j \in [0, N_y]$, and $k \in [0, N_z]$, we shall need G^{eff}_{ijk} at values $i \in [-N_x, N_x]$, $j \in [-N_y, N_y]$, and $k \in [-N_z, N_z]$. Using the fact that $G(u, v, w)$ is even in all its arguments, one can show that G^{eff}_{ijk} is even in all its indices. Even so, with eight terms, and eight limits per term, there is a lot of work to be done. The generic indefinite integral was computed using Mathematica [5] and *lots* of algebraic simplifications:

$$\begin{aligned}
&\int\int\int \frac{(u-a)(v-b)(w-c)}{\sqrt{u^2+v^2+w^2}} dw\, dv\, du = \\
&\frac{1}{288}\Bigg\{\frac{12}{5} r\Big[8r^4 + 40(abuv + acuw + bcvw) \\
&- 5\Big(au(2u^2+5v^2+5w^2) + bv(5u^2+2v^2+5w^2) \\
&+ cw(5u^2+5v^2+2w^2)\Big)\Big] + u\Big[3u\Big(3(b+3c)u^2 \\
&+ 2c(9v-28b)v - 24bcw + 6(b+2c)w^2\Big) \\
&-8a\Big(2b(u^2-27cv-18cw+6w^2)+c(5u^2+21v^2+9w^2)\Big)\Big] \\
&- 144abcw^2 \arctan\Big(\frac{u}{w}\Big) - 48bc(3a-2u)u^2\Big[\arctan\Big(\frac{v}{u}\Big) \\
&- \arctan\Big(\frac{vw}{ur}\Big)\Big] - 48ac(3b-2v)v^2\Big[\arctan\Big(\frac{u}{v}\Big) \\
&- \arctan\Big(\frac{4au(3b-2v)w - (4b-3v)v^2 r}{v(u(4b-3v)w + 4a(3b-2v)r)}\Big)\Big] \\
&- 48ab(3c-2w)w^2\Big[\arctan\Big(\frac{u}{w}\Big) \\
&- \arctan\Big(\frac{4auv(3c-2w) - (4c-3w)w^2 r}{w(uv(4c-3w) + 4a(3c-2w)r)}\Big)\Big] \\
&+ 12a\Big[4bv\big(v^2+3w(w-c)\big) + 4cw\big(w^2+3v(v-b)\big) \\
&- 3(v^2+w^2)^2\Big]\log(u+r) - 12bu\Big[3u\big(u^2+2w(w-2c)\big) \\
&- 4a\big(u^2+3w(w-2c)\big)\Big]\log(v+r) \\
&- 12cu\Big[3u\big(u^2+2v(v-2b)\big) - 4a\big(u^2+3v(v-2b)\big)\Big] \\
&\times \log(w+r) + 48bcw^3\log(u^2+w^2) + 6bw(4c-3w)w^2 \\
&\times \log\Big[\frac{v^2+w^2}{16a^2(3c-2w)^2 + (4c-3w)^2w^2}\Big(\frac{4(v+r)}{bv^2w^2}\Big)^2\Big] \\
&+ 6cv(4b-3v)v^2 \\
&\times\log\Big[\frac{v^2+w^2}{16a^2(3b-2v)^2 + (4b-3v)^2v^2}\Big(\frac{4(w+r)}{cv^2w^2}\Big)^2\Big]\Bigg\}, \qquad (6)
\end{aligned}$$

where $r = \sqrt{u^2+v^2+w^2}$. Special cases of this integral occur when any one or more of a, b, or c vanish, and when any one or more of the limits vanish. These spacial cases require some care to compute. Moreover, combining the values of (6) at the limits leads, in some cases to cancelations that result in a significant loss of precision. We found the use of quad-precision arithmetic a necessity for the accurate computation of G^{eff}.

We have coded (6), and all its special cases, in a FORTRAN90 module to compute 3D space-charge forces using the IGF, and we have implemented this as one of the options available for computing space-charge effects in MARYLIE/IMPACT. Because the integrals that compose G^{eff} are complicated and require quad precision arithmetic, the 3D IGF calculation of space-charge for large-aspect-ratio beams still demands considerable computational effort. To speed these computations, we parallelized the calculation of the IGFs over the domain grid, and we recompute the IGF array only when the change in the bunch size is sufficient to require it.

VALIDATION

To test our implementation, we used it to compute the scalar potential produced by a homogeneous ellipsoid, for which the result is known analytically [6, 7]. Figure 2 shows some typical results. In this case, the ellipsoid has semi-axes $a_x = 10^{-1}$, $a_y = 3 \times 10^{-3}$, and $a_z = 10^{-4}$, yielding aspect-ratios $a_x/a_y \approx a_y/a_z \approx 30$, and $a_x/a_z = 10^3$. In this case, the benefit of using IGFs is dramatic. With a grid of just $32 \times 32 \times 32$ points, IGF reproduces the analytic results to an accuracy better than 1% everywhere except at the very edge, where it rises to about 2%. The standard approach needs eight times more points along the x-axis to even *approach* 10% accuracy.

DISCUSSION

The complexity of the calculation, and the need for quad-precision arithmetic both mean that the 3D IGF algorithm described here will not be the first choice for the calculation of space-charge forces in routine (garden-variety) charge distributions. On the other hand, in situations where aspect ratios do differ significantly from unity, the 3D IGF tech-

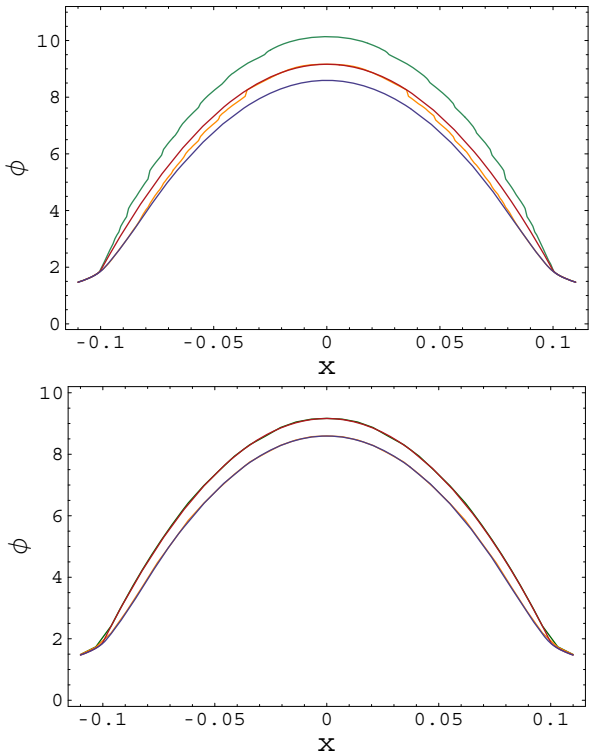

Figure 2: Comparisons of different methods for computing the scalar potential ϕ of a homogeneous ellipsoid. These plots show ϕ as a function of x. The red and blue curves in each plot show the analytic results: on-axis in red, and $(y, z) = (a_y/2, a_z/2)$ in blue. The green and orange curves give the corresponding results obtained using different techniques: standard Hockney algorithm on a $32 \times 32 \times 32$ grid (top), and a $256 \times 32 \times 32$ grid (middle); 3D IGF algorithm on a $32 \times 32 \times 32$ grid (bottom).

nique dramatically reduces the number of mesh points required for an accurate computation of space-charge forces.

We see two ways in which the IGF computation might be improved. First, one might use numerical cubature [8, 9] to evaluate (5). While 3D numerical cubature may be expensive, so are the many transcendental function calls in (6). Moreover, numerical cubature will not suffer from the cancellations that render quad-precision arithmetic necessary. The singularity at the origin will require some care, but that obstacle is surmountable. Second, we chose (3) to approximate the charge density because it is simple and continuous. But it may be that different basis functions exist which are continuous and have the property that their convolution with the Green function yields a result much simpler than (6).

REFERENCES

[1] R. D. Ryne, *et al.*, J. Physics: Conf. Series **16**, 210–214 (2005).

[2] R. W. Hockney, J. W. Eastwood, *Computer Simulation Using Particles*, Institute of Physics Publishing, London (1988).

[3] R. D. Ryne, ICFA Beam Dynamics Mini Workshop on Space-Charge Simulation, Trinity College, Oxford, 2–4 April 2003, `http://www.isis.rl.ac.uk/AcceleratorTheory/workshop/talks/R_Ryne2.pdf` .

[4] J. Qiang, S. Lidia, R. D. Ryne, C. Limborg-Deprey, *Phys. Rev. ST Accel. Beams* **9**, 044204 (2006).

[5] Wolfram Research, Inc., *Mathematica, V.5.0*, Champaign, IL (2003).

[6] S. Chandrasekhar, *Ellipsoidal Figures of Equilibrium*, Dover Pub., New York (1987).

[7] J. Binney, S. Tremaine, *Galactic Dynamics*, Princeton University Press, Princeton, NJ (1988).

[8] A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, NJ (1971).

[9] I. H. Sloan, S. Joe, *Lattice Methods for Multiple Integration*, Oxford Science Publications, Oxford University Press, New York, NY (1994).

