Wavelet-Based Poisson Solver for Use in Particle-in-Cell Simulations

Balša Terzić*
Department of Physics, Northern Illinois University, DeKalb, Illinois 60115

Ilya V. Pogorelov
Accelerator and Fusion Research Division, Lawrence Berkeley National Laboratory
(Dated: May 11, 2005)

We report on a successful implementation of a wavelet-based Poisson solver for use in 3D particle-in-cell simulations. Our method harnesses advantages afforded by the wavelet formulation, such as sparsity of operators and data sets, existence of effective preconditioners, and the ability simultaneously to remove numerical noise and further compress relevant data sets. We present and discuss preliminary results of application of the new solver to test problems in accelerator physics and astrophysics.

KEYWORDS: multiscale dynamics, wavelets, N-body simulations, particle-in-cell method, Poisson equation

I. INTRODUCTION

Gaining insight into the dynamics of multiparticle systems, such as charged particle beams or self-gravitating systems, heavily relies on N-body simulations. The many-body codes can be grouped into three main categories: (*i*) direct summation, (*ii*) tree and (*iii*) particle-in-cell (PIC). The direct summation codes are prohibitively expensive for large systems, since their computational cost scales as N^2. The tree codes use direct summation for nearby particles and evoke statistical arguments for contribution of particles farther away. The PIC codes incorporate a computational grid into which particles are binned, thus resulting in the coarse-grained, discretized particle distribution. The potential associated with such discretized particle distribution is computed by solving the Poisson equation on the grid. Finally, the forces needed to advance each individual particle are computed by interpolation from the discretized potential on the grid. A detailed treatment of computational methods to simulate multiparticle systems is given by Hockney and Eastwood [1]. In this paper we outline a wavelet-based algorithm for solving the Poisson equation for use in PIC codes.

It is important that the algorithms used in solving the Poisson equation:

1. account for *multiscale dynamics*, because even the fluctuations on smallest scales can lead to global instabilities and fine-scale structure formation, as exemplified by halo formation and microbunching instability observed in beam dynamics experiments [2–5],
2. *minimize the numerical noise* due to the fact that the number of particles used to sample the phase-space distribution function in the N-body simulation is several orders of magnitude smaller than the number of particles in the physical system which is being modeled; and
3. be as *efficient* as possible in terms of computational speed and storage requirement, without compromising accuracy.

Furthermore, for some important applications, such as coherent synchrotron radiation (CSR) in beam dynamics, it is necessary to have a compact representation of the system's history [5, 6].

The wavelet-based solver we present here has the potential to satisfy all of the requirements listed above. In Section II we briefly outline the concept of wavelets and wavelet transforms. Formulating the Poisson equation on the grid and solving it using the wavelet-based approach is reported in Section III. In Section IV, we begin by applying the wavelet-based solver to model two density distributions of interest in beam dynamics and astrophysics. We then proceed to replace (for testing purposes) the Green's function-based Poisson solver in IMPACT-T beam dynamics code [7, 8] with the wavelet-based solver, and compare results produced by the two versions of IMPACT-T evolving the same initial distributions. We conclude with a summary of the work presented and a description of the work in progress.

II. WAVELETS AND WAVELET TRANSFORMS

Wavelets and wavelet transforms are a relatively new concept, introduced in the 1980's [9–12]. The discrete wavelet transform can be viewed in two different ways, as:

1. the discrete analog of the *continuous* wavelet trans-

*Corresponding author.
Email addresses: bterzic@nicadd.niu.edu, ivpogorelov@lbl.gov

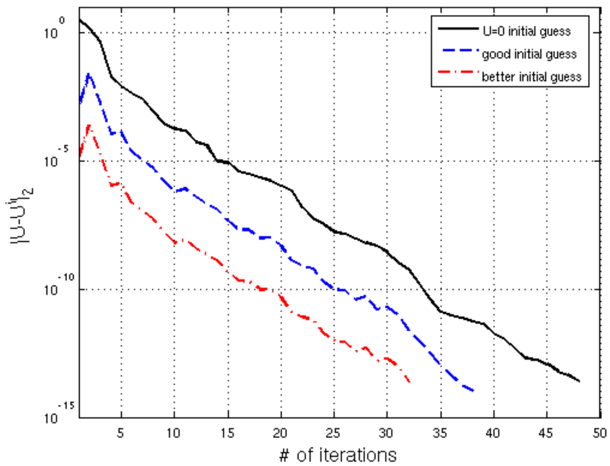

FIG. 1: Affect of the initial guess on the preconditioned conjugate gradient method. The rate of convergence remains the same, given by eq. 5, while it is obvious that a better initial guess leads to convergence to the desired accuracy within significantly fewer steps.

form of a function $f(t)$ given by

$$g(s,x) = \int_{-\infty}^{\infty} f(t) P_{s,x}(t) dt, \tag{1}$$

$$P_{s,x}(t) = \frac{1}{\sqrt{s}} P\left(\frac{t-x}{s}\right), \tag{2}$$

where $P(t)$ is a *mother wavelet*, from which the wavelet basis functions are computed by scaling and/or translation [12].

2. a family of perfect reconstruction high- and low-pass filters, which can extract information from the signal at varying scales. Then the forward and inverse discrete wavelet transforms can be represented by filtering and down- or up-sampling, respectively [10, 11].

There are three main reasons that a wavelet-based Poisson solver is of interest:

1. Solving the problem in wavelet space enables retaining information about the dynamics on different scales spanned by the wavelet expansion.

2. Wavelet formulation also allows for natural removing numerical noise (*denoising*) through thresholding the wavelet coefficients. This also reduces the effective dimensionality of the problem, thereby reducing the computational load.

3. Formulating and solving the Poisson boundary-value problem in wavelet space provides considerable computational speedup: both the Laplacian and the inverse Laplacian operators are *sparse*; also, the iterative methods can be accelerated using preconditioners that are effectively diagonal in wavelet bases [13–15].

III. WAVELET-BASED POISSON SOLVER

The Poisson equation solved by the PIC codes is defined on a computational grid which contains all the particles used in the simulation. The discretization takes the Poisson equation with Dirichlet boundary conditions (BCs), *i.e.*, for which the value of the function is specified on the boundary, from its continuous form

$$\bigtriangledown^2 u = f, \qquad u_{bnd} = h \tag{3}$$

where $\bigtriangledown^2$ is the continuous Laplacian derivative, to

$$LU = F, \qquad U_{bnd} = H \tag{4}$$

where the Laplacian operator L, potential U, density F are all defined on the computational grid and H is specified on the surface of the grid. Eq. (4) represents a well-known problem in numerical analysis. It can be solved using a number of iterative methods, such as multigrid, successive over-relaxation, steepest descent, or conjugate gradient. For the work presented here, we generalized to 3D the preconditioned conjugate gradient (PCG) method [16, 17]. The PCG method iteratively updates the initial solution along the conjugate directions until the exit requirement

$$|LU - F|_2 \leq \epsilon^2 |F|_2, \tag{5}$$

is satisfied in some norm $||_2$. The convergence rate of the method is dependent on the condition number (k) of the operator L:

$$\left|U - U^i\right|_2 \leq \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |U|_2, \tag{6}$$

where U^i is the approximation to the exact solution U after the i^{th} iteration. The smaller the condition number, the faster the approximation U^i approaches to the exact solution U. The condition number k of the Laplacian operator L on a grid is proportional to the square of the grid resolution, $k(L) \propto O(N^2)$, where N is the number of cells in each coordinate. Such large condition number results in a slowly-converging scheme. However, in wavelet space, there is an effective diagonal preconditioner P for the wavelet-transformed Laplacian operator L_w, which reduces the condition number of the preconditioned operator to $k(PL_wP) \propto O(N)$ [13–15]. This provides a significant computational speedup.

Whereas the rate of convergence is set by the relation given in Eq. (6), the number of iterations needed to attain a certain predefined accuracy also depends on how

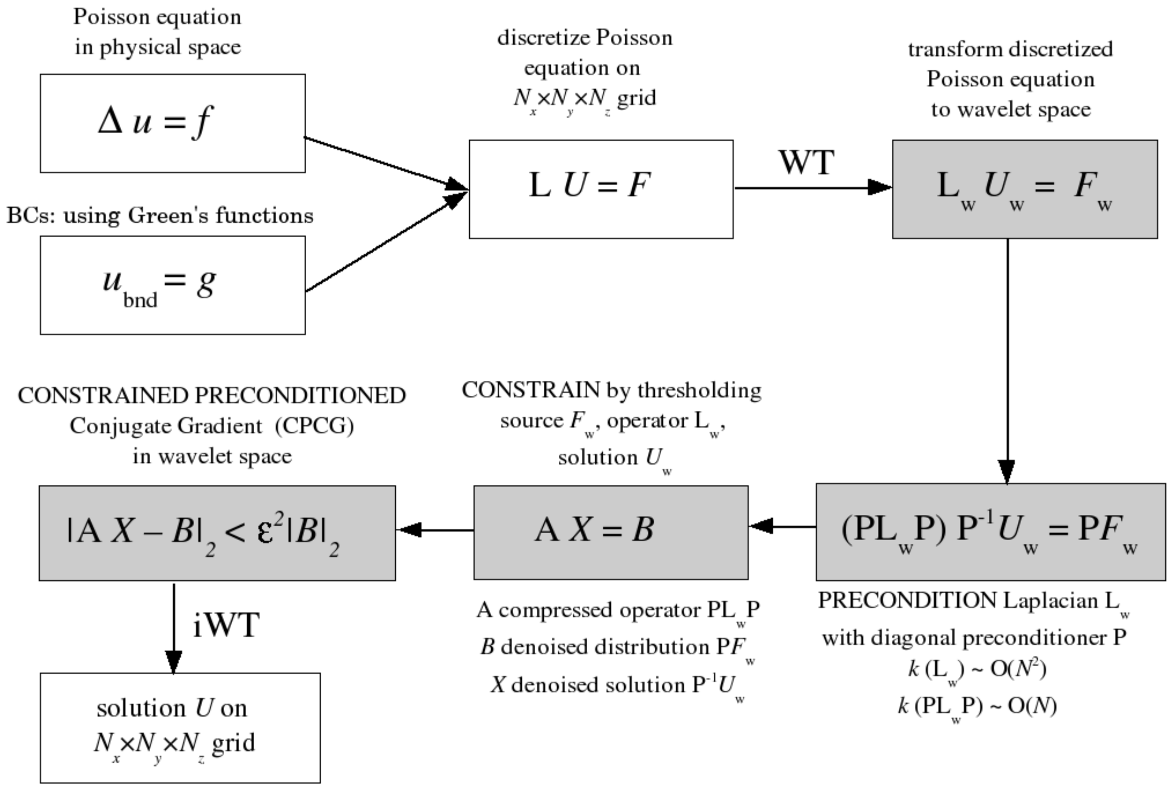

FIG. 2: Flow-chart outline of the wavelet-based Poisson solver using the (constrained) PCG method. The gray boxes represent the wavelet-space; the physical space is in white. Constraining of the PCG method is done in the bottom middle box. The current version of the code does not have this step implemented yet.

close the initial guess is to the solution. Since one does not expect significant changes in the potential from one instant in time $t = t_0$ to the next $t = t_0 + \Delta t$, the potential at $t = t_0$ serves as a good initial guess for the conjugate gradient iteration at $t = t_0 + \Delta t$. The importance of having a "smart" initial guess is illustrated in Figure 1.

Thresholding the wavelet coefficients (setting them equal to zero if their magnitudes are below a certain predefined threshold) can be used to remove the smallest scale fluctuations usually associated with numerical noise. It is worth remembering, however, that essential physics that must be captured in a typical PIC simulation includes various instabilities and fine structure/substructure formation. These processes owe their existence to the coupling between multiple spatial scales on which the system's dynamics unfolds. Uncontrolled denoising carries with it the obvious danger of "smoothing out" the fine-scale details that serve as seeds for the onset of these processes. Still, there are numerous indications that properly implemented adaptive denoising can enable significant reduction in the size of the relevant data sets without compromising the solver's ability to resolve the physically important multiscale aspects of the system's dynamics. Denoising-by-thresholding effectively reduces (constrains) the search space for the iterative PCG method. Our implementation of the constrained PCG (CPCG) will be reported elsewhere.

A schematic representation of the solver is given in Figure 2.

A. Implementing Boundary Conditions

In the current implementation, we take the beam to pass through a grounded rectangular pipe. Over the four walls of the pipe, $U = 0$, and the two open ends through which the beam passes have open BCs. We choose the computational grid to have dimensions 3-10 times smaller than those of the pipe, and we compute the potential over the six surfaces of this grid using a Green function while satisfying the constraints on U that the pipe imposes.

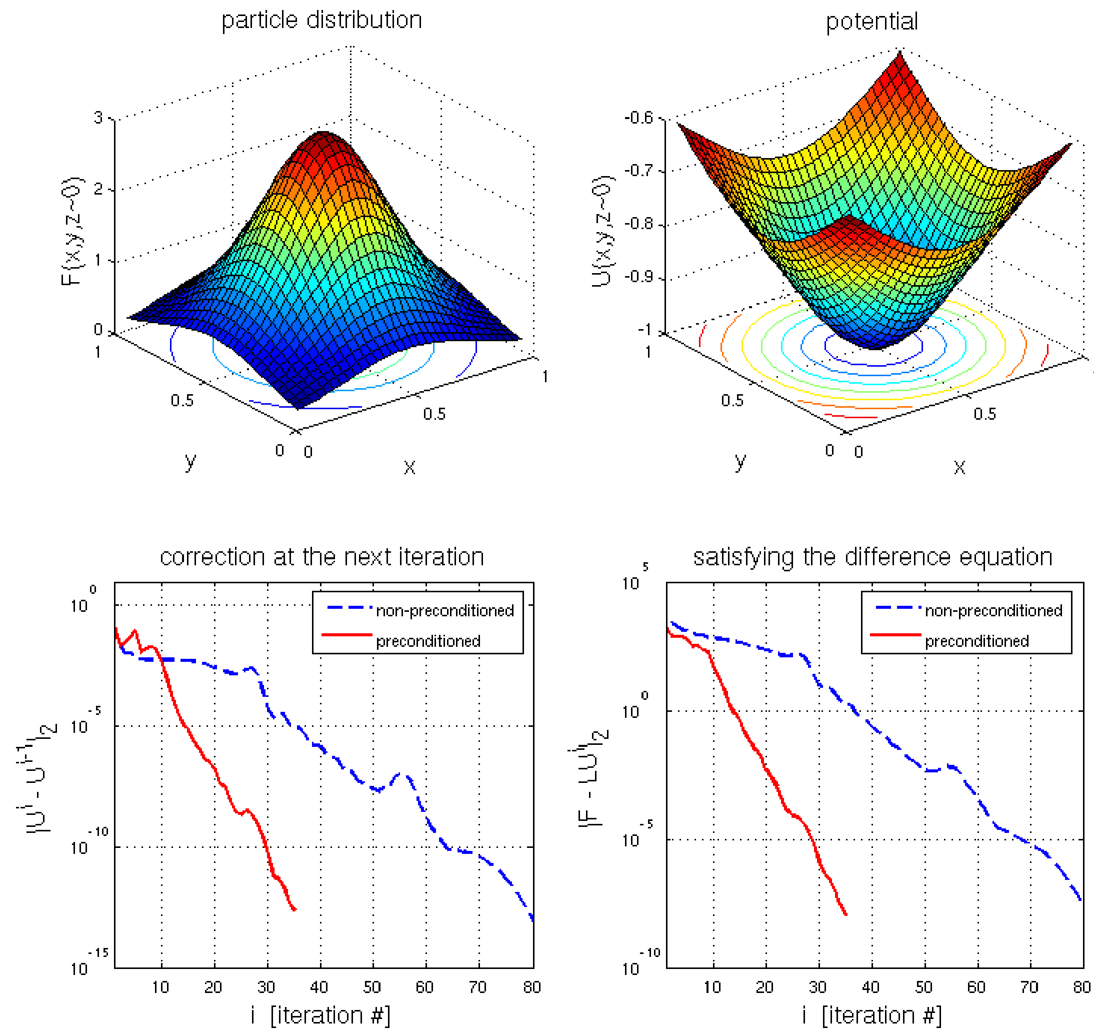

FIG. 3: Plummer sphere particle distribution (top left) and corresponding potential (top right) obtained using the PCG. The lower panels show two convergence criteria correction at the next iteration (bottom left) and how well the difference equation is satisfied (bottom right) with (solid line) and without the preconditioner (dashed line).

Accordingly, the computation of BCs reduces to solving the following system of equations [8]:

$$\rho^{lm}(z) = \frac{4}{ab}\int_0^a\int_0^b \rho(x,y,z)\sin(\alpha_l x)\sin(\beta_m y)dydx, \quad (7)$$

$$\frac{\partial^2\phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2\phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\epsilon_0}, \quad (8)$$

$$\phi(x,y,z) = \sum_{l=1}^{N_x}\sum_{l=1}^{N_x}\phi^{lm}(z)\sin(\alpha_l x)\sin(\beta_m y), \quad (9)$$

where ρ is the charge distribution, ϕ is the potential, $\alpha_l = l\pi/a$, $\beta_m = m\pi/b$, $\gamma_{lm}^2 = \alpha_l^2 + \beta_m^2$, ϵ_0 is the permittivity of vacuum and the geometry of the pipe is given by $0 \le x \le a$ and $0 \le y \le b$ [8]. Eq. (9) is evaluated only on the surface of the computational grid, and for the predefined number of expansion coefficients N_x and N_y, thus yielding U_{bnd} from Eq. (4). This is only one of the ways to compute the potential on the surface of the grid. Others, more efficient and computationally cheaper, will be a implemented in the future editions of the code.

The PCG solves Eq. (5) assuming $U = 0$ outside the computational grid. The inhomogeneous Dirichlet boundary-value problem in Eq. (4) has been made equivalent to the homogeneous one by transferring the inhomogeneous boundary value terms to the source. In the

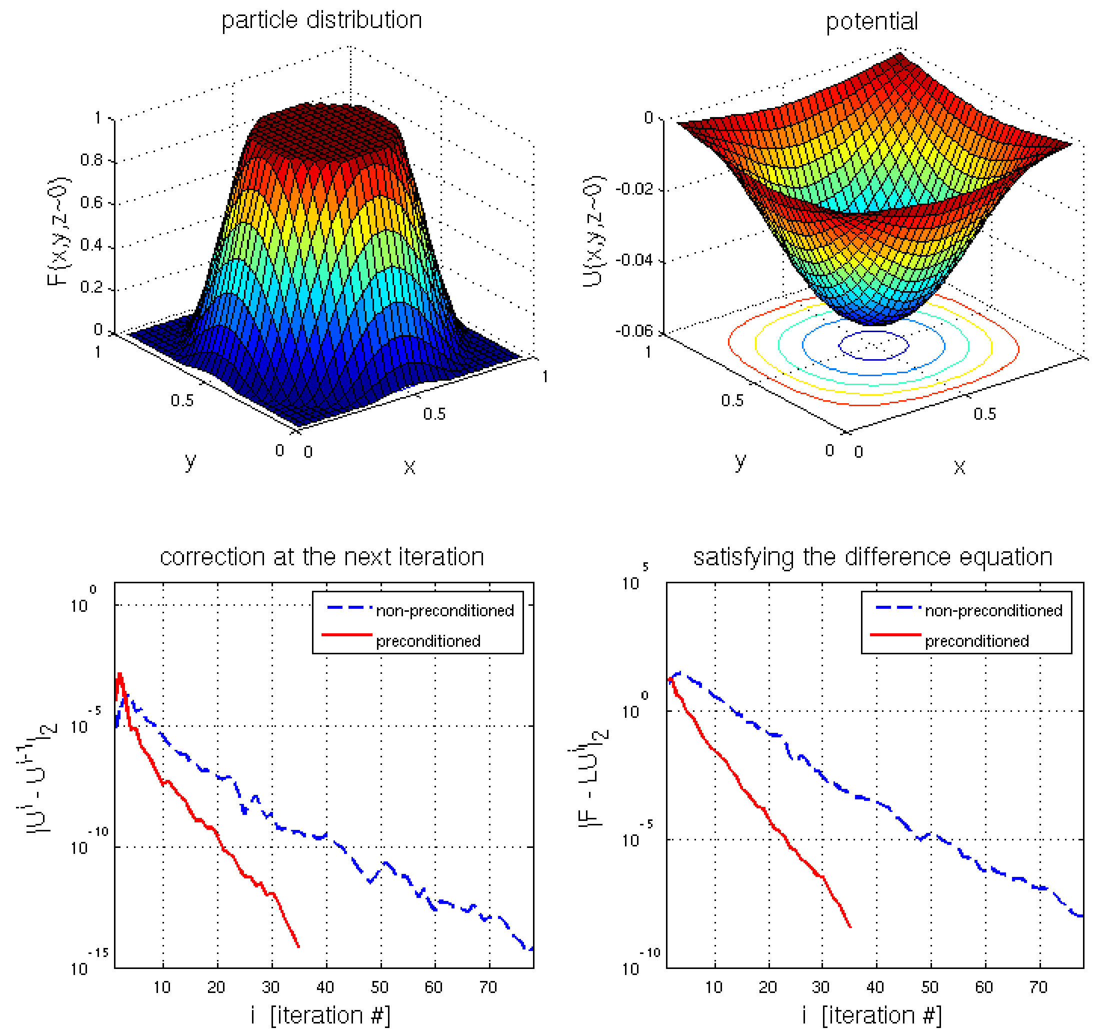

FIG. 4: Particle distribution (top left) and corresponding potential (top right) for the "fuzzy cigar" obtained using the PCG. The lower panels show two convergence criteria correction at the next iteration (bottom left) and how well the difference equation is satisfied (bottom right) with (solid line) and without the preconditioner (dashed line).

simplest case where the spacing of the computational grid Δ is the same in x-, y- and z-directions, the altered source becomes $F \to F - H/\Delta^2$ [18].

IV. APPLICATIONS

Our goal has been to develop a wavelet-based Poisson solver which can be easily merged into existing PIC codes designed for multiparticle dynamics simulations. As the first step towards that goal, we tested the solver on two idealized particle distributions, one from astrophysics and the other from beam dynamics. We used the PCG solver to compute the potential associated with the Plummer spherical stellar distribution (Figure 3). Both the potential and density are analytically known and are given by

$$F(r) = \frac{3}{(1+r^2)^{\frac{5}{2}}}, \qquad U(r) = -\frac{1}{\sqrt{1+r^2}}, \tag{10}$$

where $r = \sqrt{x^2+y^2+z^2}$. The potential on the surface of the computational grid is specified analytically. The bottom panels of the Figure 3 demonstrate the substantial computational speedup gained by preconditioning. Here we applied open BCs, $U(r \to \infty) \to 0$, which is the natural choice for self-gravitating systems.

We then applied the algorithm to a more realistic setting in which only the particle distribution is analytically

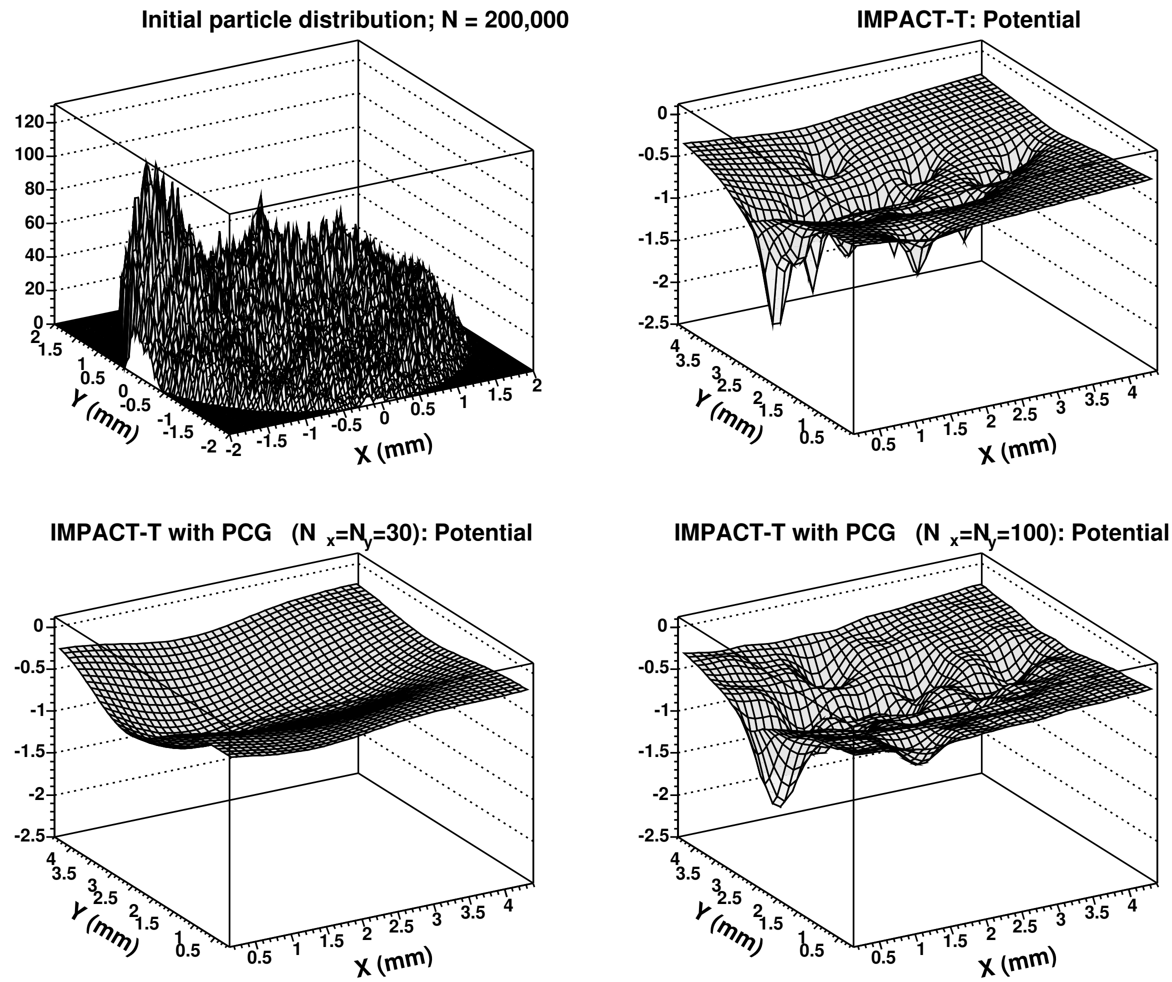

FIG. 5: z-integrated non-axisymmetric particle distribution (top left) for a beam. The other panels show the corresponding potential computed using the Green's function method used in the standard IMPACT-T (top right), IMPACT-T with PCG and $N_x = N_y = 30$, *i.e.* 900 coefficients in the Green's function expansion of the potential on the surface of the computational grid (bottom left) and IMPACT-T with PCG and $N_x = N_y = 100$ (bottom right).

known, and where the potential on the surface of the computational grid is computed using the analytically known Green's function (Figure 4). It is an axially symmetric "fuzzy cigar"-shaped configuration of charged particles (a "beam bunch") given by

$$F(x, y, z) = d_1(R)d_2(z), \tag{11}$$

$$d_1(R) = \begin{cases} 1 & 0 \le R \le R_1, \\ \frac{(R-R_2)^2(R-(3R_1-2R_2))^2}{4(R_1-R_2)^4} & R_1 \le R \le R_2, \\ 0 & \text{otherwise,} \end{cases} . \tag{12}$$

$$d_2(z) = \begin{cases} 1 & z_{1,2} \le z \le z_{2,1}, \\ \frac{(z-z_1)^2(z-(3z_{1,2}-2z_1))^2}{4(z_{1,2}-z_2)^4} & z_1 \le z \le z_{1,2}, \\ \frac{(z-z_2)^2(z-(3z_{2,1}-2z_2))^2}{4(z_{2,1}-z_2)^4} & z_{2,1} \le z \le z_2, \\ 0 & \text{otherwise,} \end{cases} . \tag{13}$$

where and the beam parameters R_1, R_2, z_1, z_2, $z_{1,2}$, $z_{2,1}$ are chosen so that $0 \le R_1 \le R_2$ and $z_1 \le z_{1,2} \le z_{2,1} \le z_2$. We applied BCs of a grounded rectangular pipe in the transverse direction (*i.e.*, $U = 0$ on the pipe walls), and open in the longitudinal (z) direction. Similarly to the case of the Plummer sphere, a high accuracy solution is obtained in about 30 iterations of the algorithm with a preconditioner, or about 60 without.

Upon successfully testing the PCG as a stand-alone Poisson solver, we replaced the standard Green's function-based Poisson solver in the PIC code IMPACT-

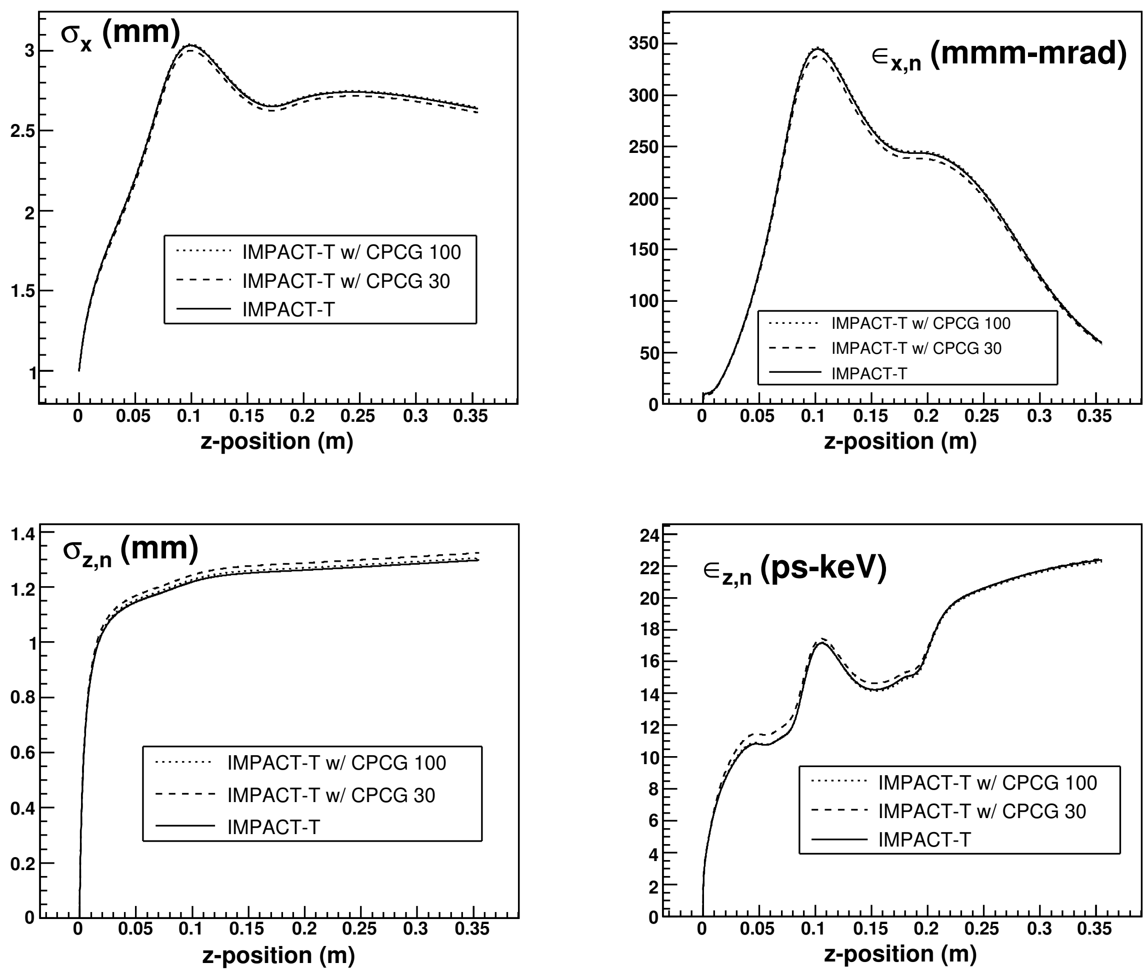

FIG. 6: Simulation results for the radio frequency gun of the Fermilab/NICADD photoinjector done with the standard version of IMPACT-T (solid lines), IMPACT-T with PCG with $N_x = N_y = 30$ (dashed line) and IMPACT-T with PCG with $N_x = N_y = 100$ (dotted line): rms beam radius (top left), rms normalized transverse emittance (top right), rms bunch length (bottom left), rms normalized longitudinal emittance (bottom right). The close agreement indicates that both codes launch the beam in essentially identical fashion; this is critically important because the output of the full photoinjector depends sensitively on the "initial conditions".

T [7, 8] with the PCG. For algorithm testing purposes, our solver uses Green's functions to evaluate the potential only on the surface of the computational grid, and then proceeds with the PCG algorithm to compute the potential on the interior. This introduces a certain computational inefficiency that will be eliminated, at the stage of optimizing the solver for performance, by using a different approach for computing BCs. The details of this optimization will be reported elsewhere. The parameters N_x and N_y specify the number of Green's function expansion coefficients in x- and y-directions. The BCs, again, correspond to a grounded rectangular pipe in the transverse directions, and open in the longitudinal direction. In a typical simulation, the cross-section of the pipe is larger than the cross-section of the computational grid by a factor of 3-10.

In Figure 5, we compare the Green's functions-based Poisson solver used by IMPACT-T with the PCG by plotting the potential each algorithm computes from the same initial particle distribution. For the simulations done with the wavelet-based Poisson solver, no wavelet coefficient thresholding was done, *i.e.*, the full wavelet expansion is retained. The thresholding and the resulting denoising will be developed and implemented in the final version of the code.

We tested our wavelet-based code in the context of the Fermilab/NICADD photoinjector using 200,000 simulation particles and a nonuniform initial particle distribution at the cathode. It appears that not specifying the potential on the surface of the grid accurately enough

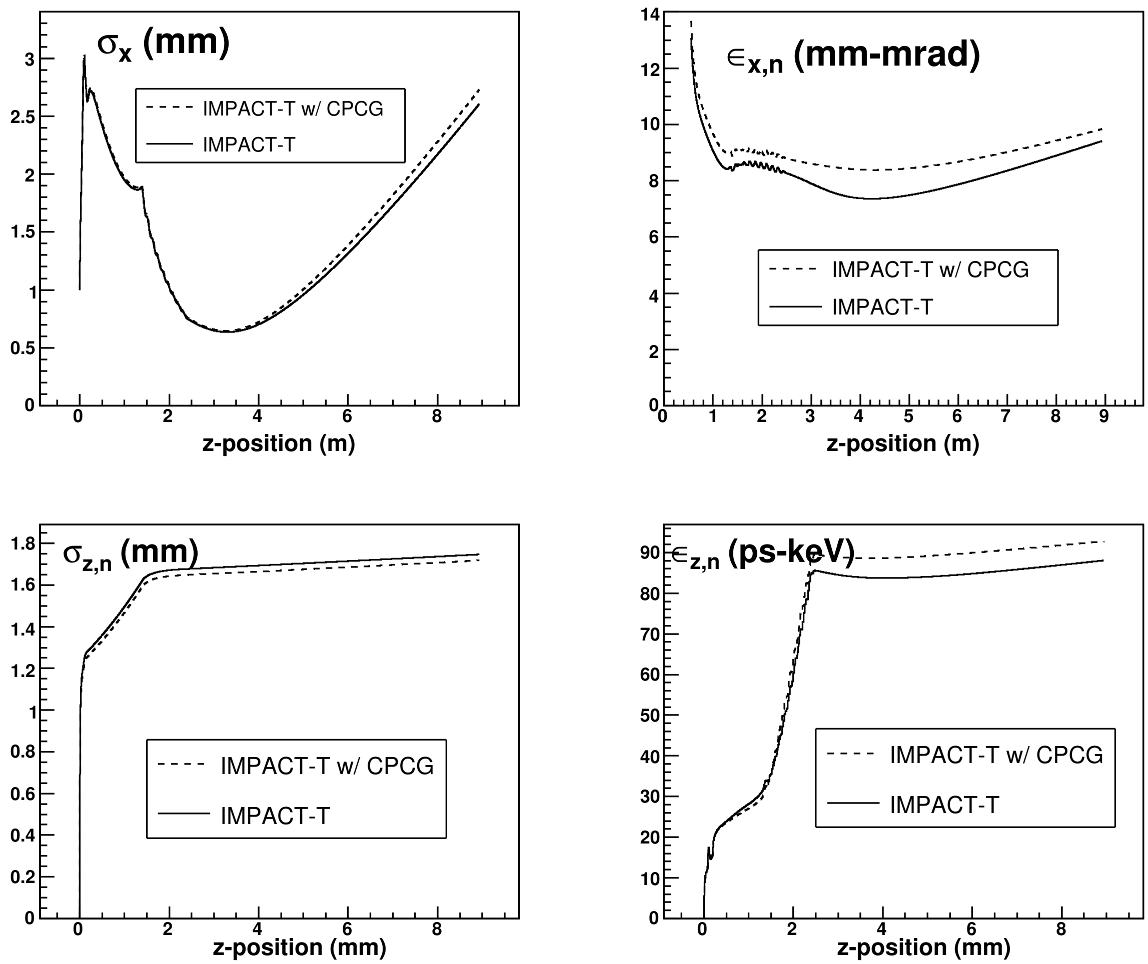

FIG. 7: Simulation results for the full Fermilab/NICADD photoinjector done with the standard version of IMPACT-T (solid lines) and IMPACT-T with PCG with $N_x = N_y = 30$ (dashed line): rms beam radius (top left), rms normalized transverse emittance (top right), rms bunch length (bottom left), rms normalized longitudinal emittance (bottom right).

causes considerable smoothing of the smaller scale features as computed by the wavelet algorithm. This, however, does not significantly affect the root mean square (rms) properties of the beam (Figures 6 and 7), as confirmed by the excellent agreement between simulation runs done with standard IMPACT-T (solid line) and IMPACT-T with a wavelet-based Poisson solver with $N_x = N_y = 30$, *i.e.*, 900 expansion coefficients (dashed line). The difference between the simulations with the standard IMPACT-T (solid line) and IMPACT-T with a wavelet-based Poisson solver with $N_x = N_y = 100$ is almost imperceptible (Figure 6).

These results clearly demonstrate that the simulations using wavelet-based Poisson solver and the standard IMPACT-T are in excellent agreement in regard to the compuation of beam moments. This establishes the present study as an important proof-of-concept.

V. DISCUSSION AND CONCLUSION

We formulated and implemented a prototype, 3D, wavelet-based Poisson solver that uses the preconditioned conjugate gradient method. The idea of combining the wavelet formulation and PCG to solve the Poisson equation is not new: there have been pioneering implementations of wavelet-based solvers for the Poisson equation with homogeneous ($U = 0$) Dirichlet BCs in 1D [13] and periodic BCs in 1D, 2D and 3D [14, 15]. We built on this earlier work to design and implement a solver for the three-dimensional Poisson equation with general inhomogeneous Dirichlet boundary conditions. This constitutes an original contribution on our part, since the formulation of the discretized problem, which includes the treatment of the boundary conditions and the Laplacian operator, differs significantly from the periodized problem.

Having first tested our method as a stand-alone solver on two model problems, we then merged it into IMPACT-T to obtain a fully functional serial PIC code. We found that simulations performed using IMPACT-T with the "native" Poisson solver (based on Green's functions and fast Fourier transforms) and IMPACT-T with the PCG solver described in this paper produce essentially equivalent outcomes (in terms of a standard set of rms diagnostics). This result enables us to move from the proof-of-concept stage to the advanced optimization and application-specific algorithm design. To our knowledge, the work reported here constitutes the first application of the wavelet-based multiscale methodology to 3D computer simulations in beam dynamics.

Our current efforts are focused on several areas that encompass both algorithm optimization and applications work. On the optimization side, the top priority is to enable efficient computation of the potential over the boundary of the computational grid (as distinct from the physical boundaries of the system). Another priority is incorporation into the solver of state-of-the-art routines for efficient storage and multiplication of multidimensional sparse arrays. Next, adaptive denoising (and simultaneous compression) in the context of PIC modeling presents us with a unique set of technical challenges and a wealth of complex and engaging multiscale physics. Finally, we have not yet addressed the complex issues of solver parallelization for use with the parallel version of IMPACT-T on multiprocessor machines.

On the side of applications, we are working on leveraging the advantages afforded by the multiscale wavelet formulation to tackle the previously all but intractable – in the sense of being prohibitively expensive computationally – problem of high-precision 3D modeling of CSR and its effects on the dynamics of beams in a variety of accelerator systems. The details of our approach will be reported, together with the first results, in the near future.

Acknowlegements. Both authors are thankful to Henry Emil Kandrup for the role he played in their lives. His thoughtful guidance and patient mentorship greatly influenced their scientific careers. It is through him that this collaboration was established in the first place.

The authors are grateful to Daniel Mihalcea for running the numerical simulations and generating the Figures 5-7. Ji Qiang provided valuable help in integrating the solver into the IMPACT-T suite. We are thankful to Courtlandt Bohn for useful and stimulating discussions and comments. Balša Terzić was supported by the Air Force contract FA9471-040C-0199. The work of Ilya V. Pogorelov was supported by the Director, Office of Science, of the U. S. Department of Energy under contract No. DE-AC03-76SF00098.

[1] R. Hockney and J. Eastwood, *Computer Simulations Using Particles* (Institute of Physics Publishing, London, 1988).
[2] C. A. et al., Phys. Rev. Lett. **89**, 214802 (2003).
[3] C. Bohn and I. Sideris, Phys. Rev. Lett. **91**, 264801 (2003).
[4] J. Qiang, R. Ryne, and I. Hofmann, Phys. Rev. Lett. **92**, 174801 (2004).
[5] Z. Huang, M. Borland, P. Emma, J. Wu, C. Limborg, G. Stupakov, and J. Welch, Phys. Rev. ST Accel. Beams **7**, 074401 (2004).
[6] C. Bohn, AIP Conference Proceedings **647**, 81 (2002).
[7] J. Qiang, R. Ryne, S. Habib, and V. Decyk, J. Comp. Phys. **163**, 434 (2000).
[8] J. Qiang and R. Ryne, Comp. Phys. Comm. **138**, 18 (2001).
[9] S. Goedecker, *Wavelets and Their Applications* (Presses polytechniques et universitaires romandes, Lausanne, 1998).
[10] M. Misiti, Y. Misiti, G. Oppenheim, and J. Poggi, *Wavelet Toolbox for Use with MATLAB* (The MathWorks, Inc., Natick, Massachusetts, 1997).
[11] M. Wickerhauser, *Adaptive Wavelet Analysis From Theory to Software* (A K Peters, Wellesley, Massachusetts, 1994).
[12] I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992).
[13] G. Beylkin, *In Wavelets: Mathematics and Applications, eds. J. Benedetto and M. Frazier* (CRC Press LLC, Boca Raton, Florida, 1993).
[14] A. Averbuch, G. Beylkin, R. Coifman, P. Fischer, and M. Israeli (2003), http://www.cs.tau.ac.il/∼amir1/PS/poisson.pdf.
[15] A. Averbuch, G. Beylkin, R. Coifman, P. Fischer, and M. Israeli, *In Signal and Image Representation in Combined Spaces, eds. Y. Zeevi and R. Coifman* (Academic Press, San Diego, 1998).
[16] G. Golub and C. V. Loan, *Matrix Computations* (Johns Hopkins University, Baltimore and London, 1996).
[17] M. Hestines and E. Stiefel, J. Res. Nat. Bur. Stand. 49, 409. **49**, 409 (1996).
[18] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing, 2nd edition* (Press Syndicate of the University of Cambridge, Cambridge, 1992).