

## Particle-in-cell beam dynamics simulations with a wavelet-based Poisson solver

Balša Terzić,<sup>1</sup> Ilya V. Pogorelov,<sup>2</sup> and Courtlandt L. Bohn<sup>1</sup>

<sup>1</sup>*Beam Physics and Astrophysics Group, Northern Illinois University, DeKalb, Illinois 60115, USA*

<sup>2</sup>*Accelerator and Fusion Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

(Received 1 September 2006; published 2 March 2007)

We report on a successful implementation of a three-dimensional wavelet-based solver for the Poisson equation with Dirichlet boundary conditions, optimized for use in particle-in-cell (PIC) simulations. The solver is based on the operator formulation of the conjugate gradient algorithm, for which effectively diagonal preconditioners are available in wavelet bases. Because of the recursive nature of PIC simulations, a good initial approximation to the iterative solution is always readily available, which we demonstrate to be a key advantage in terms of overall computational speed. While the Laplacian remains sparse in a wavelet representation, the wavelet-decomposed potential and density can be rendered sparse through a procedure that amounts to simultaneous compression and denoising of the data. We explain how this procedure can be carried out in a controlled and near-optimal way, and show the effect it has on the overall solver performance. After testing the solver in a stand-alone mode, we integrated it into the IMPACT-T beam dynamics particle-in-cell code and extensively benchmarked it against the IMPACT-T with the native FFT-based Poisson solver. We present and discuss these benchmarking results, as well as the results of modeling the Fermi/NICADD photoinjector using IMPACT-T with the wavelet-based solver.

DOI: [10.1103/PhysRevSTAB.10.034201](https://doi.org/10.1103/PhysRevSTAB.10.034201)

PACS numbers: 02.60.Cb, 07.05.Tp, 41.75.-i, 52.65.Rr

### I. INTRODUCTION

Particle-in-cell (PIC) simulation [1,2] is a highly effective computational technique that has been used extensively in application areas as diverse as accelerator physics [3], astrophysics and cosmology [4–10], plasma physics [2,11], heavy-ion-beam-driven inertial fusion [12], hydrodynamics of compressible fluid flows [13], and semiconductor device design [14]. For systems such as charged particle beams, the application domain of this paper, PIC is often the method of choice due to its high speed and memory efficiency.

In the PIC setting, there are several important advantages to using a wavelet-based iterative Poisson solver. One such advantage is the ability to use the solution from the previous time step as the initial approximation used in solving the Poisson equation one time step later: this simple idea was found to have a dramatic effect on the number of iterations to convergence. Another advantage is that, in a variety of wavelet bases, the Laplacian operator remains sparse, while, unlike in the original basis, there also exist in wavelet bases effectively diagonal preconditioners for the Laplacian operator [15,16]. This combination of circumstances favors the use in PIC simulations of a (preconditioned) iterative algorithm such as the conjugate gradient. Additionally, the inherently multiscale wavelet representation provides a natural setting for the study of physical phenomena unfolding simultaneously on many, often widely separated, spatial scales. One example is an onset and growth of the microbunching instability in high-intensity electron beams [17,18]. For problems of this kind, resolution requirements vary considerably across the problem domain, and working in a wavelet basis gives one the ability to use varying levels of resolution in different

regions in phase space, similar to that afforded by adaptive mesh refinement techniques [19,20]. At the same time, to the extent that the unresolved part of the phase-space density distribution can be identified with noise, different parts of the distribution can be, if necessary, denoised according to the local thresholding criteria so as to improve computational efficiency without compromising simulation fidelity.

Finally, sampling the phase-space distribution density by a finite number of test particles, with subsequent mapping of the density onto the computational grid, introduces sampling and discretization errors that can be thought of as “numerical noise.” When density and potential are wavelet decomposed [an  $\mathcal{O}(N)$  operation], a significant reduction in the data size can be achieved by setting to zero all wavelet decomposition coefficients whose magnitudes are below a certain preselected threshold. This thresholding procedure amounts to simultaneous compression and denoising of the potential and density data and can be put on a more rigorous foundation by introducing an entropy-like penalty function that is used to select the best basis out of a sufficiently broad library of bases [21–25]. In so doing, one is also furnished with the “near-optimal” (in the sense that can be made precise) value of the threshold. As described in the body of the paper, we implemented a simplified version of this approach so as to maximize computational speed and minimize the adverse effect on performance of the full library search. One of our goals was to gain a better understanding of the properties and limitations of this compression-and-denoising process, as well as of the properties of sampling-and-deposition noise itself.

Certainly one principal motivation for developing a wavelet-based Poisson solver is to take advantage of wave-

let compression. Wavelet compression enables compact storage and easy recall of the beam history. In turn, it facilitates simulations wherein the beam history is important.

An example is simulating the influence of coherent synchrotron radiation (CSR) on the beam as it transits magnetic bends in the accelerator lattice. Such bends are unavoidable in, for example, recirculating linear accelerators and bunch compressors. To compute the scalar and vector potentials  $\Phi(\mathbf{r}, t)$  and  $\mathbf{A}(\mathbf{r}, t)$ , respectively, and hence the force acting on each particle, requires integrating over the beam history to account for retardation [26]:

$$\begin{aligned}\Phi(\mathbf{r}, t) &= e \int d\mathbf{r}' dt' \frac{\delta(t - t' - \frac{1}{c}|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} n(\mathbf{r}, t'), \\ \mathbf{A}(\mathbf{r}, t) &= e \int d\mathbf{r}' dt' \frac{\delta(t - t' - \frac{1}{c}|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} \frac{\mathbf{v}'}{c} n(\mathbf{r}, t').\end{aligned}\quad (1)$$

Here,  $\mathbf{v}'$  is the particle velocity evaluated at the retarded time  $t'$  and  $c$  is the speed of light. The idea is to evaluate these potentials from the evolutionary history of the charge density  $n(\mathbf{r}, t)$  as the beam transits a magnetic bend. The end result would be a fully three-dimensional (3D) model of CSR, something that has been notoriously difficult to achieve (for a survey of CSR codes see [27], and references therein).

It is with such simulations in mind, combined with a desire to preserve accurately the influence of the hierarchy of spatial scales on the space-charge force, and hence on the overarching beam dynamics, that we proceed.

We formulated and implemented a 3D wavelet-based solver for the Poisson equation with general (inhomogeneous:  $U \neq 0$ ) Dirichlet boundary conditions (BCs), optimized for use in PIC simulations. The solver is based on the preconditioned conjugate gradient algorithm. We built on previous implementations of wavelet-based solvers for the Poisson equation with homogeneous ( $U = 0$ ) Dirichlet BCs in 1D [15] and periodic BCs in 1D, 2D, and 3D [16,28]. However, our formulation of the discretized 3D problem, which includes the treatment of the inhomogeneous BCs and the Laplacian operator, differs significantly from the periodized and homogeneous problem.

In Sec. II, we describe formulating the Poisson equation on the grid and solving it using the wavelet-based approach. Section III is devoted to the detailed treatment of noise in PIC simulations and wavelet-based methods for its removal. In Sec. IV, we test the wavelet-based solver by applying it to two analytic potential-density pairs of interest in beam dynamics and astrophysics. We then proceed to replace the Green-function-based Poisson solver in the IMPACT-T beam dynamics code [3,29,30], designed and maintained at Lawrence Berkeley National Laboratory, with the wavelet-based solver, and compare results produced by the two Poisson solvers evolving the same initial distributions through a real photoinjector. Finally, we summarize the main results and discuss possible applications

of the wavelet-based approach to problems in beam dynamics and astrophysics.

## II. WAVELET-BASED POISSON SOLVER

Wavelets and wavelet transforms are a relatively new concept, introduced in the 1980s [31–35]. The discrete wavelet transform (DWT), like the discrete Fourier transform (DFT), is a fast, linear operation on data sets with the size of integer power of 2 in each dimension, resulting in a transformed data set of the same size. Just like DFT, DWT is also invertible and orthogonal, with the inverse transform in 1D being the transpose of the transform. The most important difference between DFT and DWT is that the individual wavelet functions are localized in both frequency space (like DFT) *and* configuration space (similar to what windowed DFT attempts to do). This kind of dual localization makes a number of operators and functions sparse in the wavelet space. Essential background on wavelets and wavelet transforms is available in the literature [31,34,35].

Almost from their inception, wavelets have been used to solve partial differential equations (PDEs), elliptic in particular [15,16,28,34,36]. An introduction to solving PDEs, including Poisson's equation, using wavelet formalism is provided in [34]. In the context of PIC solvers, it is necessary to solve the 3D Poisson equation with general (inhomogeneous) Dirichlet BCs, which we do herein.

The Poisson equation with Dirichlet BCs is given by

$$\Delta u = f, \quad u_{\text{bnd}} = g, \quad (2)$$

where  $\Delta$  is the continuous Laplacian operator, and  $u_{\text{bnd}}$  is defined only on the boundary. The problem is then discretized and can be solved using a number of conceptually different approaches. A wavelet-based approach that we present in this paper possesses a number of important advantages.

There are four main reasons that a wavelet-based Poisson solver is of interest: (i) Solving the problem in wavelet space enables retaining information about the dynamics over the hierarchy of scales spanned by the wavelet expansion. (ii) Wavelet formulation also allows for natural removal of numerical noise (*denoising*) by thresholding of the wavelet coefficients. (iii) By the same token, relevant data sets (the particle distribution and potential) can be represented compactly using only a fraction of the wavelet coefficients. (iv) Finally, there are three significant advantages to carrying out the inversion of the Laplacian in the transform space, as opposed to the original coordinate space of the problem. The first one is that the wavelet-decomposed Laplacian remains symmetric and sparse, so that an iterative method such as conjugate gradient (CG), which references the sparse  $L$  only through multiplication, immediately becomes attractive. The second advantage is that preconditioners exist that are effectively diagonal in the transform space. The third advantage

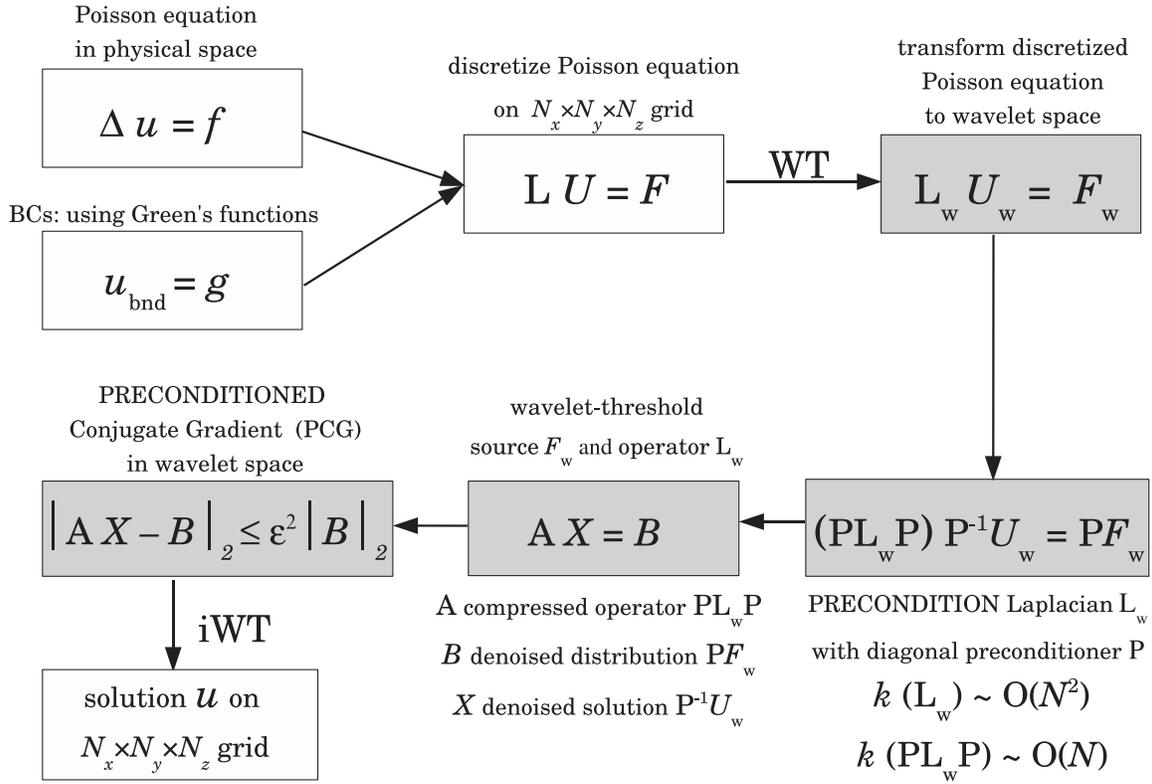


FIG. 1. Flow chart outlining the wavelet-based Poisson equation solver. The physical space is shown in white boxes, and the wavelet space in gray. First, the continuous Poisson equation with Dirichlet BCs is discretized on the finite grid in physical space, thus reducing it to a discrete linear algebra problem. Next, the problem is transformed to wavelet space using discrete wavelet transforms (WT), where the efficient, diagonal preconditioner (P) was applied to the wavelet-transformed Laplacian operator  $L_w$ . Then, still in wavelet space, the wavelet-thresholding is applied to the source and the preconditioned Laplacian operator. The resulting linear algebra problem is then solved in wavelet space using a standard preconditioned conjugate gradient (PCG) method and a “smart guess” provided by the solution to the Poisson equation at the previous step in the PIC simulation (see Sec. II B). Finally, the solution on the grid in wavelet space is transformed using inverse wavelet transform (iWT) to yield the solution on the grid in physical space.

of employing an iterative solver in a PIC simulation comes from the nature of the simulation process itself: the Poisson equation is solved repeatedly, once per simulation time step, with the source term changing only slightly from one solve to the next. This means that every time the Poisson equation is solved, the solution from the previous time step can serve as a reasonable initial approximation to the solution, and the number of iterations necessary to converge to the solution will be significantly reduced.

The new solver is outlined in Fig. 1.

### A. Discretizing the physical problem

The Poisson equation solved by 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 BCs from its continuous form given in Eq. (2) to

$$L U = F, \quad U_{\text{bnd}} = G, \quad (3)$$

where the Laplacian operator  $L$ , potential  $U$ , and density  $F$

are all defined on the computational grid, and  $G$  is specified on the surface of the grid.

In this paper, the discretized Laplacian operator  $L$  is given in terms of the finite difference scheme involving a 3-point stencil:

$$\begin{aligned} L &\equiv L_x + L_y + L_z \\ &\equiv h_x^{-2}(\delta_x^- - 2 + \delta_x^+) + h_y^{-2}(\delta_y^- - 2 + \delta_y^+) \\ &\quad + h_z^{-2}(\delta_z^- - 2 + \delta_z^+), \end{aligned} \quad (4)$$

where  $\delta^-$  and  $\delta^+$  are backward and forward shift operators, respectively, in the coordinate noted in the subscript.  $h_x$ ,  $h_y$ , and  $h_z$  denote grid spacing in  $x$ -,  $y$ -, and  $z$ -coordinates, respectively. The approximation to the second derivative, say  $D_{xx}$ , provided by the above finite difference 3-point stencil is a second-order approximation:

$$D_{xx} = h_x^{-2}(\delta_x^- - 2 + \delta_x^+) + \mathcal{O}(h_x^2). \quad (5)$$

Since  $\exp(i\omega)$  and  $\exp(-i\omega)$  are the eigenvalues of  $\delta^+$  and  $\delta^-$ , respectively, the spectral response of the 3-point discretized Laplacian is given by the transfer function  $S(\omega)$ :



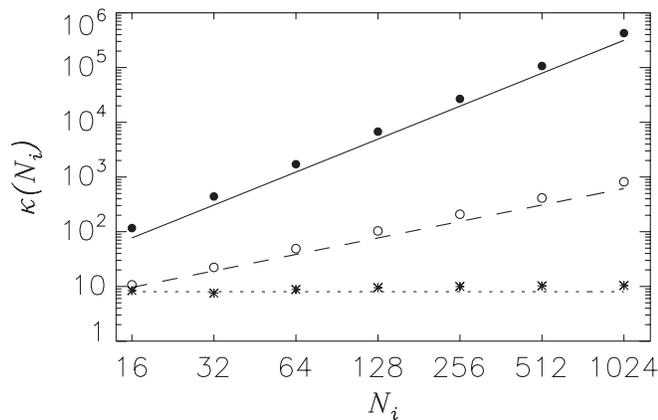


FIG. 4. Condition number  $\kappa$  for the discretized Laplacian operator with a 3-point stencil as a function of the resolution  $N_i$ : for the nonpreconditioned operator (solid circles; superimposed against a solid line  $\propto N_i^2$ ); for the operator preconditioned in wavelet space of Daubechies family of order 2 (empty circles; superimposed against a dotted line  $\propto N_i$ ). Asterisks denote the ratio of the second largest to the smallest eigenvalues after preconditioning in wavelet space of Daubechies family of order 2 (dashed line is  $\propto \text{const}$ ).

problem in Eq. (3) then becomes

$$(PL_w P)(P^{-1}U_w) = PF_w. \quad (11)$$

The preconditioner  $P$  reduces the condition number of the Laplacian operator with inhomogeneous Dirichlet BCs in wavelet space to  $\kappa(PL_w P) \propto \mathcal{O}(N_i)$ , thereby greatly improving the convergence rate. We also observe that, whereas after preconditioning the condition number becomes  $\propto \mathcal{O}(N_i)$ , the ratio between the second largest and the smallest eigenvalue is roughly constant, indicating that all but the largest eigenvalue are of the same order. Figure 4 shows the condition numbers, including the ratio between the second largest and the smallest eigenvalues, as a function of grid resolution  $N_i$ .

The number of iterations needed to attain a certain predefined accuracy also depends on how close the initial approximation is to the solution. With the possible exception of the very first time step, one does not expect significant changes in the potential from one instant in time  $t = t_i$  to the next  $t = t_i + \Delta t$ . Thus, the potential at  $t = t_i$  serves as a good initial approximation for the conjugate gradient iteration at the next time step  $t = t_i + \Delta t$ .

The computational speedup due to operator preconditioning and a good initial approximation is illustrated in Fig. 5. Using the distribution at the previous step of the simulation as the initial (smart) approximation at the next step greatly reduces the number of iterations needed for convergence. Figure 5 shows the number of iterations for the first 2000 steps of the simulations of the Fermilab/NICADD photoinjector using: no preconditioning and zero initial approximation (green line); preconditioning

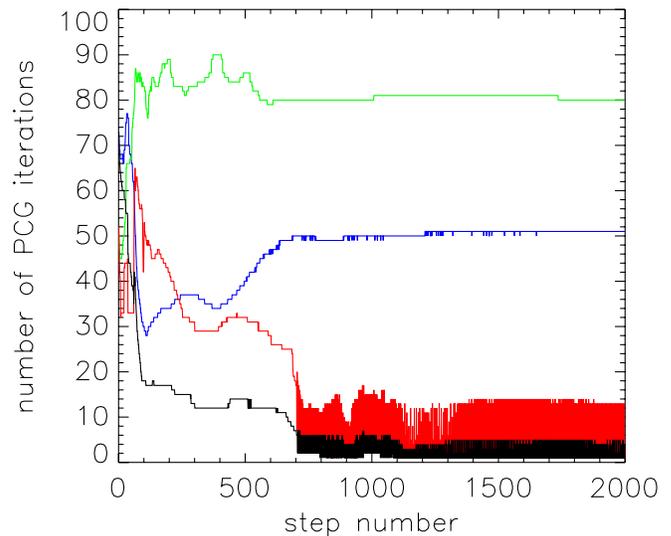


FIG. 5. (Color) Number of iterations of the PCG algorithm for the first 2000 steps of a realistic simulation: no preconditioning and zero initial approximation (green line), preconditioning and zero initial approximation (blue line), no preconditioning and “smart initial approximation” (red line), and preconditioning and “smart initial approximation” (black line). The simulation is done using IMPACT-T PIC code and the *Distribution 1* on the Fermilab/NICADD photoinjector (as described later in the text), with  $N = 125\,000$  particles, resolution  $N_i = 32$  and Daubechies wavelets of order 2. The computational speedup is quite similar when wavelet thresholding is performed (with and without) the Anscombe transformation. Other wavelet families exhibit the same qualitative behavior. Averages for the entire 30 000-step run are 75.2 for the green line, 60.7 for the blue line, 4.8 for the red line, and 2.4 for the black line.

and zero initial approximation (blue line); no preconditioning and “smart initial approximation” (red line); and preconditioning and “smart initial approximation” (black line). Taking the potential at the previous step as the initial approximation at the next step causes the PCG to compute only the (small) difference between two consecutive steps, usually taking only a few iterations. This is a significant improvement over the number of iterations needed for convergence with zero initial approximation. In both cases, the number of iterations is appreciably larger when preconditioning is turned off.

### C. Implementing boundary conditions

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,  $U(z \rightarrow \infty) \rightarrow 0$ . We choose the computational grid to have transverse dimensions several (generally 4–6) 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. Accordingly, the computation of BCs reduces to solving the following system of equations:

$$\rho^{lm}(z) = \frac{4}{AB} \int_0^A \int_0^B \rho(x, y, z) \sin(\alpha_l x) \sin(\beta_m y) dy dx, \quad (12)$$

$$\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\epsilon_0}, \quad (13)$$

$$\phi(x, y, z) = \sum_{l=1}^{M_x} \sum_{m=1}^{M_y} \phi^{lm}(z) \sin(\alpha_l x) \sin(\beta_m y), \quad (14)$$

where  $\rho$  is the charge distribution,  $\phi$  is the potential,  $\alpha_l = l\pi/A$ ,  $\beta_m = m\pi/B$ ,  $\gamma_{lm}^2 = \alpha_l^2 + \beta_m^2$ , and the geometry of the pipe is given by  $0 \leq x \leq A$  and  $0 \leq y \leq B$  [29]. Equation (14) is evaluated only on the surface of the computational grid, and for the predefined number of expansion coefficients  $M_x$  and  $M_y$ , thus yielding  $U_{\text{bnd}}$  from Eq. (3). This is only one of the ways to compute the potential on the surface of the grid.

An alternative to grounded-rectangular-pipe BCs is a grounded cylindrical pipe (which is not implemented here). In the case of a cylindrical pipe, the computation of BCs reduces to solving

$$\rho^{lm}(z) = \frac{1}{\pi R_{\text{wall}}^2} \int_0^{R_{\text{wall}}} J_l(\xi_{lm} R) \int_0^{2\pi} \rho(R, \theta, z) d\theta dR, \quad (15)$$

$$\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \xi_{lm}^2 \phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\epsilon_0}, \quad (16)$$

$$\phi(R, z) = \sum_{l=1}^{M_x} \sum_{m=1}^{M_y} \phi^{lm}(z) J_l(\xi_{lm} R), \quad (17)$$

where  $\xi_{lm} = j_l^m/R_{\text{wall}}$  and  $j_l^m$  is the  $m$ th root of the  $l$ th Bessel function of the first order  $J_l(x)$  (finite at  $x = 0$ ).

The inhomogeneous Dirichlet boundary-value problem in Eq. (3) has been made equivalent to the homogeneous one by transferring the inhomogeneous boundary-value terms to the source:

$$\begin{aligned} \tilde{F} = & F - h_x^{-2}[G(h_x, 0, 0) + G(N_1 h_x, 0, 0)] \\ & - h_y^{-2}[G(0, h_y, 0) + G(0, N_2 h_y, 0)] \\ & - h_z^{-2}[G(0, 0, h_z) + G(0, 0, N_3 h_z)], \end{aligned} \quad (18)$$

where  $N_1$ ,  $N_2$ , and  $N_3$  are grid resolutions in  $x$ -,  $y$ -, and  $z$ -directions, respectively [39]. After this adjustment, Eq. (7), which assumes  $U = 0$  outside the computational grid, can be used to iteratively solve the problem for  $U$ .

### III. NOISE IN PIC SIMULATIONS

The sources of numerical noise in PIC simulations are (i) *sampling noise*: the number of simulation particles is orders of magnitude smaller than the number of particles in the physical system  $N_{\text{physical}}$  ( $\approx 10^{10}$ – $10^{11}$ ); and (ii) *discrete computational domain*: all physical quantities are defined on a discrete, finite-resolution grid instead of the space-time continuum.

Thresholding the wavelet coefficients can remove the smallest-scale fluctuations usually associated with numerical noise. However, 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 very real processes. Although we do not explore this subject in the present study, 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.

The aims of this section are (i) to investigate the origin of noise in generic PIC simulations; (ii) to devise and implement a wavelet-based noise-removing scheme in the context of beam dynamics simulations; and (iii) to use ‘‘toy’’ models to illustrate the effectiveness of such denoising.

#### A. Origin and generic properties of noise

In PIC simulations,  $N$  particles sampling a charge distribution are deposited on a Cartesian computational grid with resolution  $N_i$ , and grid spacing  $h_i$ ,  $i = 1, \dots, D$ , in each coordinate of the  $D$ -dimensional system, for the total of  $N_{\text{grid}} \equiv \prod_{i=1}^D N_i$  grid points. The average number of particles per grid point in a simulation is defined as  $N_{\text{ppg}} \equiv N/N_{\text{grid}}$  [since the number of cells is  $N_{\text{cells}} \equiv \prod_{i=1}^D (N_i - 1)$  and the average number of particles per cell is defined as  $N_{\text{ppc}} \equiv N/N_{\text{cell}}$ ,  $N_{\text{ppg}}$  and  $N_{\text{ppc}}$  are close]. In a given realization with the total number of particles being  $N$ , there are  $n_j$  particles inside a  $V$ -neighborhood of the  $j$ th grid point, where  $V = [(-h_1/2, h_1/2), (-h_2/2, h_2/2), \dots, (-h_D/2, h_D/2)]$ . Each particle has the same charge  $q_0 \equiv Q_{\text{tot}}/N$ , where  $Q_{\text{tot}} = \sum_{j=1}^{N_{\text{grid}}} q_j$  is the total bunch charge and  $q_j = q_0 n_j$  is the charge in the  $V$ -neighborhood of the  $j$ th grid point.

There are two important particle-deposition schemes in PIC simulations.

(i) Nearest grid point deposition scheme (NGP DS), whereby a particle only contributes to the nearest grid point, with the particle-deposition function centered at each particle (see Fig. 6):

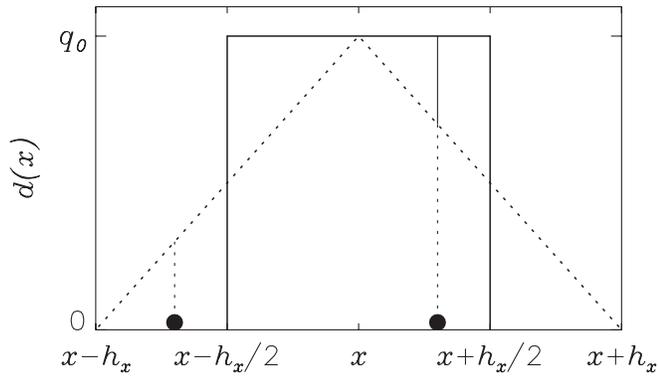


FIG. 6. Deposition functions  $d_{\text{NGP}}(x)$  (solid line) and  $d_{\text{CIC}}(x)$  (dashed line) in 1D.  $x$  denotes the location of a particle whose charge is to be deposited. The NGP DS affects only the nearest grid point to which it deposits all of its charge. The CIC DS affects the two closest grid points in each coordinate, as discussed in the text. Filled circles represent grid points.

$$d_{\text{NGP}}(\mathbf{x}) = \begin{cases} q_0 & \text{if } |x_i| \leq h_i/2 \text{ for all } i \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

where  $x_i$ ,  $i = 1, 2, \dots, D$  are coordinates.

(ii) Cloud-in-cell deposition scheme (CIC DS), where a particle linearly contributes to each of the vertices (grid points) of the cell it occupies (2 vertices in 1D, 4 in 2D, 8 in 3D), for which the particle-deposition function, centered at each particle, is given by (see Fig. 6)

$$d_{\text{CIC}}(\mathbf{x}) = q_0 \prod_{i=1}^D \left(1 - \frac{|x_i|}{h_i}\right). \quad (20)$$

In the NGP DS, the probability of a particle being deposited in the  $V$ -neighborhood of the  $j$ th grid point is given by the binomial distribution

$$P(n_j = n | p_j) = \frac{N!}{n!(N-n)!} p_j^n (1-p_j)^{N-n}, \quad (21)$$

where  $p_j \equiv \bar{n}_j/N$  and  $\bar{n}_j$  is the expectation of the number of particles in the  $V$ -neighborhood of the  $j$ th grid point. In the limit of large  $N$ , as pertains to  $N$ -body simulations, the binomial distribution converges to the Poisson distribution with mean  $\bar{n}_j$ :

$$P(n_j = n | p_j) = \frac{\bar{n}_j^n e^{-\bar{n}_j}}{n!}, \quad (22)$$

where  $n$  is an integer.

In what follows, we make use of a ‘‘global’’ measure of the error associated with the sampling-and-deposition noise, defined as the algebraic average of variances:

$$\sigma^2 \equiv \frac{1}{N_{\text{grid}}} \sum_{i=1}^{N_{\text{grid}}} \text{Var}(q_i). \quad (23)$$

For the two particle-deposition schemes considered in this paper, their values are (cf. Appendix A)

$$\sigma_{\text{NGP}}^2 \approx \frac{q_0^2 N}{N_{\text{grid}}} = \frac{Q_{\text{tot}}^2}{NN_{\text{grid}}}, \quad (24)$$

for the NGP DS, and

$$\sigma_{\text{CIC}}^2 \approx \frac{a^2 q_0^2 N}{N_{\text{grid}}} = \frac{a^2 Q_{\text{tot}}^2}{NN_{\text{grid}}}, \quad (25)$$

for the CIC DS, where

$$a = \left(\frac{2}{3}\right)^{D/2}. \quad (26)$$

The noise distribution for the CIC DS is therefore a *contracted* Poissonian, given by Eq. (22) with  $n_j \rightarrow an_j$ .

To summarize, the algebraic average of variances of noise in a PIC simulation depends sensitively on the parameters of the simulation and the particle-deposition scheme and only very weakly on the particle distribution (cf. Appendix A). For the types of simulations arising in beam dynamics, this weak dependence appears to be negligible.

We now demonstrate these findings on analytically known particle distributions, randomly sampled by  $N$  particles. To reiterate: the validity and applicability of our findings, however, depend only weakly on our knowledge of the exact distribution. This means that this discussion on noise in PIC simulations, as well as denoising via wavelet thresholding to be presented later in this section, are generic and should apply to realistic simulations of beams.

For our demonstration, we choose three different analytic particle distributions.

(i) Superimposed Gaussians, modeling a cathode with ‘‘hot spots’’:<sup>1</sup>

$$F(x, y, z) = \sum_{i=1}^{N_{\text{Gauss}}} A_i e^{-m_i^2}, \quad (27)$$

$$m_i^2 = \left(\frac{x - x_i^0}{a_i}\right)^2 + \left(\frac{y - y_i^0}{b_i}\right)^2 + \left(\frac{z - z_i^0}{c_i}\right)^2,$$

where  $x \in [-2, 2]$ ,  $y \in [-2, 2]$ ,  $z \in [-0.7, 0.7]$ ,  $N_{\text{Gauss}} = 3$ ,  $A_i = [1.0, 0.5, 0.4]$ ,  $a_i = [1.0, 0.2, 0.1]$ ,  $b_i = [1.0, 0.2, 0.1]$ ,  $c_i = [0.4, 0.07, 0.05]$ ,  $x_i^0 = [0.0, 0.5, -0.5]$ ,  $y_i^0 = [0.0, 0.5, -0.5]$ ,  $z_i^0 = [0.0, 0.0, 0.0]$ .

<sup>1</sup>We have also explored superimposed Gaussian models with a substantially larger number of ‘‘hot spots,’’ but have not observed any appreciable qualitative difference from the model used herein. Also, we looked into models with ‘‘spots’’ that scale as  $\exp(-|m|)$ , but, again, no notable difference was observed.

(ii) A smooth polynomial distribution:

$$\begin{aligned}
 F_{1D}(x) &= (x_2 - x)(x - x_1), \\
 F_{2D}(x, y) &= (x_2 - x)(x - x_1)f_y + (y_2 - y)(y - y_1)f_x, \\
 F_{3D}(x, y, z) &= (x_2 - x)(x - x_1)f_y f_z \\
 &\quad + (y_2 - y)(y - y_1)f_x f_z \\
 &\quad + (z_2 - z)(z - z_1)f_x f_y,
 \end{aligned} \tag{28}$$

$$f_q = \frac{1}{12}[-q^4 + 2(q_1 + q_2)q^3 - 6q_1q_2q^2 + c_qq + d_q],$$

$$c_q = (q_1 + q_2)[6q_1q_2 - (q_1 + q_2)^2],$$

$$d_q = q_1q_2[-5q_1q_2 + (q_1 + q_2)^2],$$

where  $q \equiv \{x, y, z\}$ ,  $x \in [-2, 2]$ ,  $y \in [-2, 2]$ ,  $z \in [-0.7, 0.7]$ ,  $x_1 = -2$ ,  $x_2 = 2$ ,  $y_1 = 2$ ,  $y_2 = -2$ ,  $z_1 = -0.7$ ,  $z_2 = 0.7$ .

(iii) A constant distribution:

$$F(x, y, z) = 1, \tag{29}$$

where  $x \in [-2, 2]$ ,  $y \in [-2, 2]$ ,  $z \in [-0.7, 0.7]$ .

Figure 7 shows numerically computed distributions of noise for the first two analytic distributions, versus the distribution predicted by Eq. (22), for both the NGP DS and the CIC DS. Agreement between the two is excellent, within the statistically allowed variations, thus validating that the noise distribution for the NGP DS is well approximated by Eq. (22), and for the CIC DS by Eq. (22) with  $n \rightarrow an$ . Therefore, the noise in a discretized charge distribution is, to a good approximation, a superposition of  $N_{\text{grid}}$  Poisson distributions for the NGP DS, and a superposition of  $N_{\text{grid}}$  Poisson distributions *contracted* by a factor  $a$  for the CIC DS.

The (near)independence of the standard deviation of noise on types of particle distribution is demonstrated in Fig. 8. The relations given in Eqs. (24) and (25) are confirmed because, from Fig. 8,

$$\sigma_{\text{NGP}}N_{\text{grid}}/Q_{\text{tot}} \approx N_{\text{ppg}}^{-1/2}, \tag{30}$$

$$\sigma_{\text{CIC}}N_{\text{grid}}/Q_{\text{tot}} \approx aN_{\text{ppg}}^{-1/2}, \tag{31}$$

for all three particle distributions. The deviations from the  $N_{\text{ppg}}^{-1/2}$  law for the NGP DS with  $N_{\text{grid}} = 32$  for the superimposed Gaussians and polynomial distribution reflect the presence of elongated “tails” of the error distribution. They are due to a significant number of outlying grid points in the distribution having very little charge, on average less than the charge of a single particle, which induces large local sampling errors. This problem is less severe for the CIC DS because its intrinsic smoothing spans a volume  $2^D$  times larger than the volume of the NGP DS.

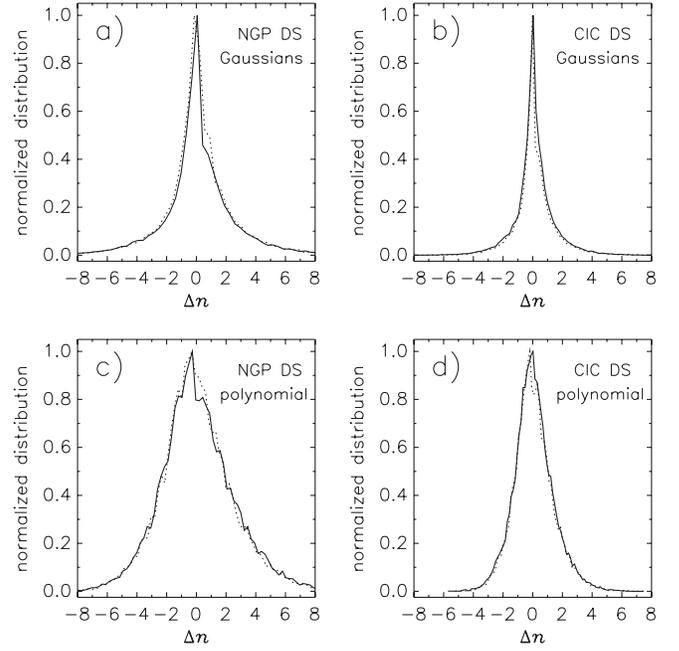


FIG. 7. For the 3D superimposed Gaussians model (top row) and the polynomial model (bottom row), numerically computed noise distribution over a single noisy realization (solid lines) versus distribution predicted by Eq. (22) (dashed lines) for both the NGP DS (left column) and the CIC DS (right column).  $N_{\text{ppg}} = 6$ , and  $N_i = 32$ . The abscissa represents error, defined as the difference in the number of particles in a  $V$ -neighborhood of a grid point between the exact distribution and a randomly sampled noisy distribution. Note that the graphs for CIC DS and NGP DS would nearly overlap if the abscissa of the NGP DS were contracted by a factor  $a$ .

## B. Quantifying noise level and denoising

Whenever the distribution is explicitly known, the quality of the noisy signal can be quantified via signal-to-noise ratio (SNR), which is defined as [8,9]

$$\text{SNR} \equiv \left[ \frac{\sum_{i=1}^{N_{\text{grid}}} \bar{q}_i^2}{\sum_{i=1}^{N_{\text{grid}}} (\bar{q}_i - q_i)^2} \right]^{1/2}. \tag{32}$$

The relationship between the SNR and the standard deviation of noise  $\sigma$  for the two particle-deposition schemes is found to be (cf. Appendix B)

$$\text{SNR}_{\text{NGP}} \approx \frac{r}{\sigma_{\text{NGP}}N_{\text{grid}}^{1/2}} = \frac{r}{Q_{\text{tot}}}N_{\text{ppg}}^{1/2}, \tag{33}$$

$$\text{SNR}_{\text{CIC}} \approx \frac{r}{\sigma_{\text{CIC}}N_{\text{grid}}^{1/2}} = \frac{r}{aQ_{\text{tot}}}N_{\text{ppg}}^{1/2}, \tag{34}$$

where  $r$  is a constant dependent on the charge distribution. From Eqs. (33) and (34), we see that  $\text{SNR} \propto N_{\text{ppg}}^{1/2}$ , which is a well-known result. Equations (33) and (34) also state that, for the same particle distribution, the CIC DS will yield a less noisy result than NGP DS, quantified by  $a^{-1}$  times

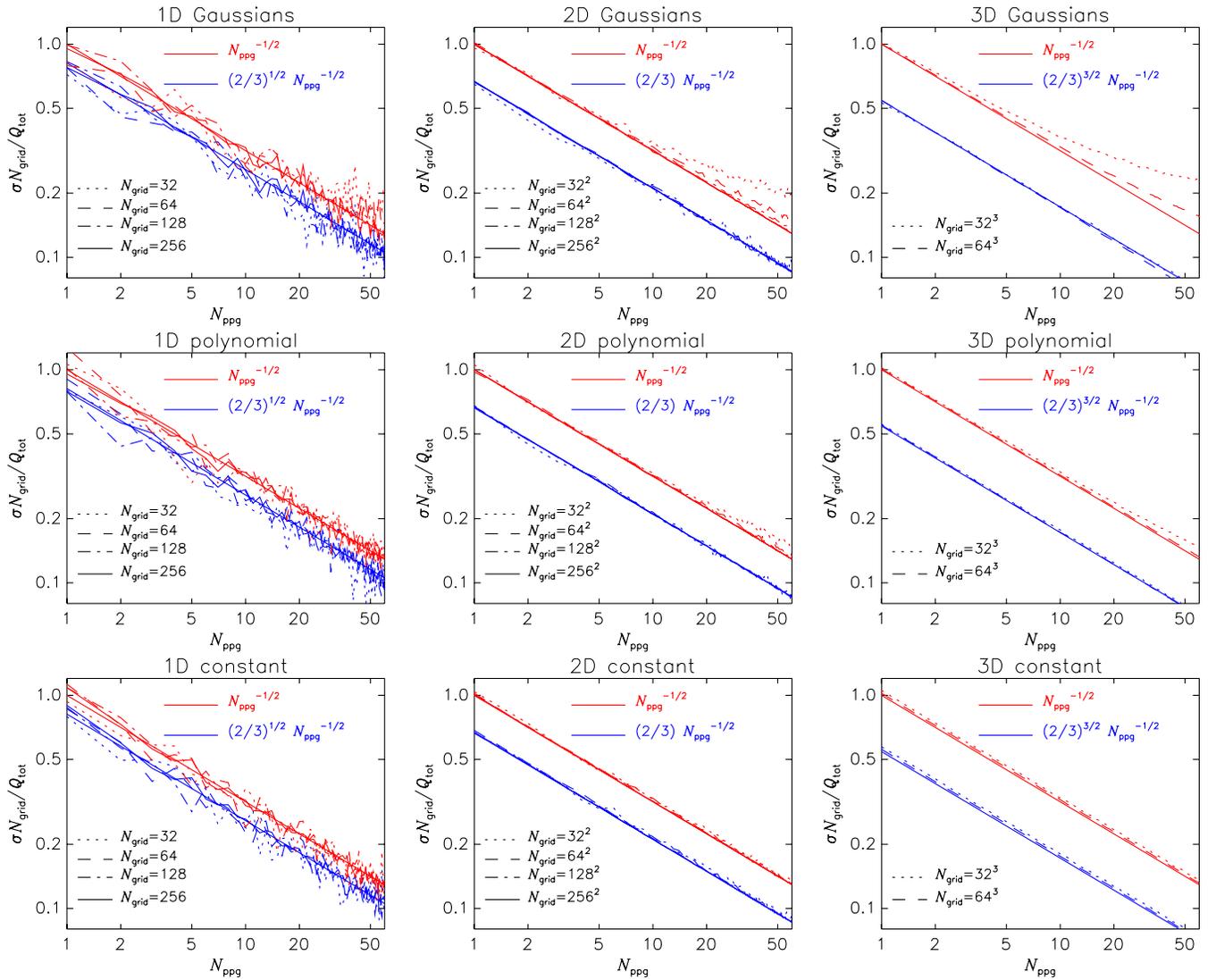


FIG. 8. (Color) For each of the three particle distributions [superimposed Gaussians (top row), polynomial (middle row), and constant (bottom row), in 1D (left column), 2D (middle column), and 3D (right column)], the normalized standard deviation of noise ( $\sigma N_{\text{grid}}/Q_{\text{tot}}$ ) for a single random noisy realization as a function of the average number of particles per grid point ( $N_{\text{ppg}}$ ). The red lines represent the NGP DS and blue the CIC DS.

higher SNR ( $\approx 1.22, 1.5,$  and  $1.84$ , for 1D, 2D, 3D, respectively), which shows the smoothing property of the CIC DS.

Whenever the SNR can be computed, one can also compute the denoising factor DF, which is defined to be the ratio of the SNR of the signal after the denoising is applied and the SNR of the original signal. It is also related to the ratio of the standard deviations of noise before and after denoising [9]:

$$\text{DF} \equiv \frac{\text{SNR}_{\text{denoised}}}{\text{SNR}_{\text{noisy}}} = \frac{\sigma_{\text{noisy}}}{\sigma_{\text{denoised}}}. \quad (35)$$

Combining Eqs. (33)–(35), one finds that the quality of a denoised signal, as measured by the SNR, represented with  $N_{\text{ppg}}$  particles per grid point is equivalent to a nondenoised

signal with  $(\text{DF})^2 N_{\text{ppg}}$  particles per grid point [9]. This is true regardless of the dimensionality of the simulation.

### C. Noise removal by wavelet thresholding in a fixed basis

Denoising in 2D PIC simulations using a fixed wavelet basis was first done by Romeo and collaborators [8,9]. (However, we remain unconvinced of the generality of that work’s central claim that simulations of dynamical evolution of nontrivial systems where the fine scale structure gives rise to instabilities can, by virtue of denoising in a fixed wavelet basis, “become equivalent to simulations with 2 orders of magnitude more particles” [8].)

After transforming noisy data to wavelet space, the signal is generally represented by a smaller number of

large coefficients, while the noise is largely mapped to many small wavelet coefficients. Wavelet thresholding is a process whereby the contribution of the wavelet coefficients deemed to represent noise is eliminated.

Two commonly used thresholding procedures are as follows.

(i) *Hard thresholding*, where the coefficients with magnitudes below certain threshold  $T > 0$  are set to zero:

$$\bar{w}_i = \begin{cases} w_i & \text{if } |w_i| > T \\ 0 & \text{if } |w_i| \leq T. \end{cases} \quad (36)$$

(ii) *Soft thresholding*, where the coefficients with magnitudes below certain threshold  $T > 0$  are set to zero and the ones above it contracted by  $T$ :

$$\bar{w}_i = \begin{cases} \text{sign}(w_i)(|w_i| - T) & \text{if } |w_i| > T \\ 0 & \text{if } |w_i| \leq T. \end{cases} \quad (37)$$

The threshold can be chosen in an objective way by following a procedure detailed in [21–25], where an entropy-like objective (“cost,” “risk,” “penalty”) function is introduced to search for the best basis out the library of bases. The underlying idea is that the components of the signal (density) that correlate well with at least some basis functions in one or more bases will be represented compactly in that basis, with a small number of non-negligible coefficients; and the components of the signal (density) that do not correlate with any basis functions in any basis are identified with noise. The procedure eliminates the need for a subjective choice of the threshold (indeed, it yields a near-optimal value of  $T$ ), and allows for quantitative statements in regards to the fidelity of the estimation of the denoised/compressed signal from its noisy realization. Alternatively, one could rely on physical arguments to choose the threshold, as is often done in practice.

The most widely used noise threshold in the literature [8,23,24,40] is given in terms of the standard deviation  $\sigma$  of the noise as

$$T = \sqrt{2 \log N_{\text{grid}}} \sigma. \quad (38)$$

This is a *universal threshold* for signals with Gaussian white noise, which means that no better noise removal can be obtained for *all signals in all wavelet bases*. It leads to noise removal which is within a small factor of ideal denoising [23,24]. A number of variations on this threshold are shown to perform better in removing different features from a known noisy signal (cf. [40] and references therein).

Studies of wavelet denoising usually involve distributions contaminated with additive (distribution-independent) Gaussian noise [23,24,40]. However, in the previous section, we showed that the noise in PIC simulations is Poisson-distributed (and weakly distribution dependent). The basic assumption behind denoising techniques is that, regardless of the details of the noise, the small-scale fluctuations due to noise map to small-scale members of the wavelet family.

One way to assure that the wavelet-thresholding theory outlined in earlier work directly applies to PIC simulations is to transform Poisson-distributed density data into an approximately Gaussian distribution. This is achieved by using a variance-stabilizing transformation due to Anscombe [41] (see also [9,42–45]):

$$X_G = 2\sqrt{X_P + \frac{3}{8}}, \quad (39)$$

which transforms a Poisson-distributed signal  $X_P$  into an approximately Gaussian-distributed signal  $X_G$  with unit variance and mean

$$m_G = \sqrt{m_P + \frac{3}{8}} - \frac{1}{8m_P^{1/2}} + \frac{1}{64m_P^{3/2}}, \quad (40)$$

with  $m_P$  being the mean of the Poissonian signal. Applying the transformation in Eq. (39) produces a bias in the data [8,9,41], which can be removed by ensuring that the denoised and noisy data have the same mean (in simulations, this is equivalent to enforcing charge conservation).

When the number of particles per grid point in the PIC simulation is too low, the noise exhibits a departure from the Poissonian profile, and the transformation (39) is no longer applicable. The CIC DS is less sensitive to the low particle count because it essentially averages out particle counts (which is what NGP DS is) over  $2^D$  neighboring grid points. Similar averaging over several grid points (as a part of a radon transform) has been used to alleviate the problem of low particle counts in astronomical image representation [45]. The rationale is that the sum of the independent Poisson random variables is a Poisson random variable with intensity equal to the sum of the individual intensities [45].

After the Anscombe transformation is applied to data contaminated with Poisson noise as in Eq. (22), the resulting data is approximately normally distributed, with variance  $\sigma_{\text{NGP}} = 1$  for the NGP DS. For the CIC DS, the data distribution is a *contracted* Poissonian, given by Eq. (22) with  $n_j \rightarrow an_j$ , which means that the resulting variance will be appropriately contracted by a factor  $a$ , i.e.,  $\sigma_{\text{CIC}} = a$ . Combining these with Eq. (38) yields optimal noise thresholds for the Anscombe-transformed (variance-stabilized) data:

$$T_{\text{NGP}} = \sqrt{2 \log N_{\text{grid}}}, \quad (41)$$

$$T_{\text{CIC}} = \sqrt{2 \log N_{\text{grid}}} a. \quad (42)$$

We also looked for a threshold in the form of  $\tilde{T} = \beta(N_{\text{grid}})\sigma$  for data *without* the Anscombe transformation. The best approximation to the threshold at which the  $\text{SNR}(T)$  is maximized is empirically found to be  $\beta(N_{\text{grid}}) = 2\sqrt{\log N_{\text{grid}}}$ . The resulting optimal thresholds

for Poissonian data without Anscombe transformation for the two particle-deposition schemes are then given by

$$\tilde{T}_{\text{NGP}} = 2\sqrt{\log N_{\text{grid}}} \frac{Q_{\text{tot}}}{(NN_{\text{grid}})^{1/2}}, \quad (43)$$

$$\tilde{T}_{\text{CIC}} = 2\sqrt{\log N_{\text{grid}}} \frac{aQ_{\text{tot}}}{(NN_{\text{grid}})^{1/2}}. \quad (44)$$

Thresholds for nontransformed data in Eqs. (43) and (44) play a role analogous to that of thresholds in Eqs. (41) and (42) for Anscombe-transformed data (cf. Fig. 9).

In Fig. 9, we show the efficiency of wavelet denoising, quantified by the DF given in Eq. (35), as a function of the thresholding parameter  $T$  for the superimposed Gaussians model, for both Anscombe-transformed (dashed lines) and nontransformed (solid lines) data. The wavelet coefficients of a noisy realization are sorted by magnitudes, hard-thresholded with  $T$  in the interval  $[|w_i|_{\min}, |w_i|_{\max}]$ , wavelet-transformed back to physical space, and their resulting  $\text{SNR}_{\text{denoised}}$  divided by the  $\text{SNR}_{\text{noisy}}$  of the noisy realization to yield  $\text{DF}(T)$ . The figure shows results for the realizations with  $N_{\text{ppg}} = 1$  (first column),  $N_{\text{ppg}} = 5$  (second column) on a  $N_i = 32$  grid and  $N_{\text{ppg}} = 1$  (third column),  $N_{\text{ppg}} = 5$  (fourth column) on a  $N_i = 64$  grid. The top row represents the NGP DS and the bottom the CIC DS.

The thresholds  $T_{\text{NGP}}$  (dashed) and  $\tilde{T}_{\text{NGP}}$  (solid) are shown as vertical lines in the top row, and  $T_{\text{CIC}}$  (dashed) and  $\tilde{T}_{\text{CIC}}$  (solid) as vertical lines in the bottom row.

We observe that the thresholds for both transformed and nontransformed data, given in Eqs. (41)–(44), are extremely close to the ideal threshold at which the DF peaks. It is also apparent that denoising is at most only marginally more efficient when the Anscombe transformation is applied. The same qualitative behavior of the SNR as a function of the threshold  $T$ , as well as excellent agreement between the predicted noise threshold and the computed threshold at which the maximum in SNR occurs (also found for the other analytical models we studied), point to the generality of the findings. Recall Eqs. (33) and (34) state that for the same charge distribution CIC DS will have  $1/a \approx 1.84$  (in 3D) times higher SNR, which means that the relative comparison of SNR for the two particle-deposition schemes can be achieved by multiplying the  $y$ -values in the bottom row by 1.84.

In implementing the algorithm, we arranged that one of the run-time options in the simulation is whether the data is Anscombe transformed or not (see Table I).

We can generalize the findings of our study of analytical models to derive “reasonable expectations” on the efficiency of wavelet thresholding. From the work presented in this section, which we observe to hold for the three suffi-

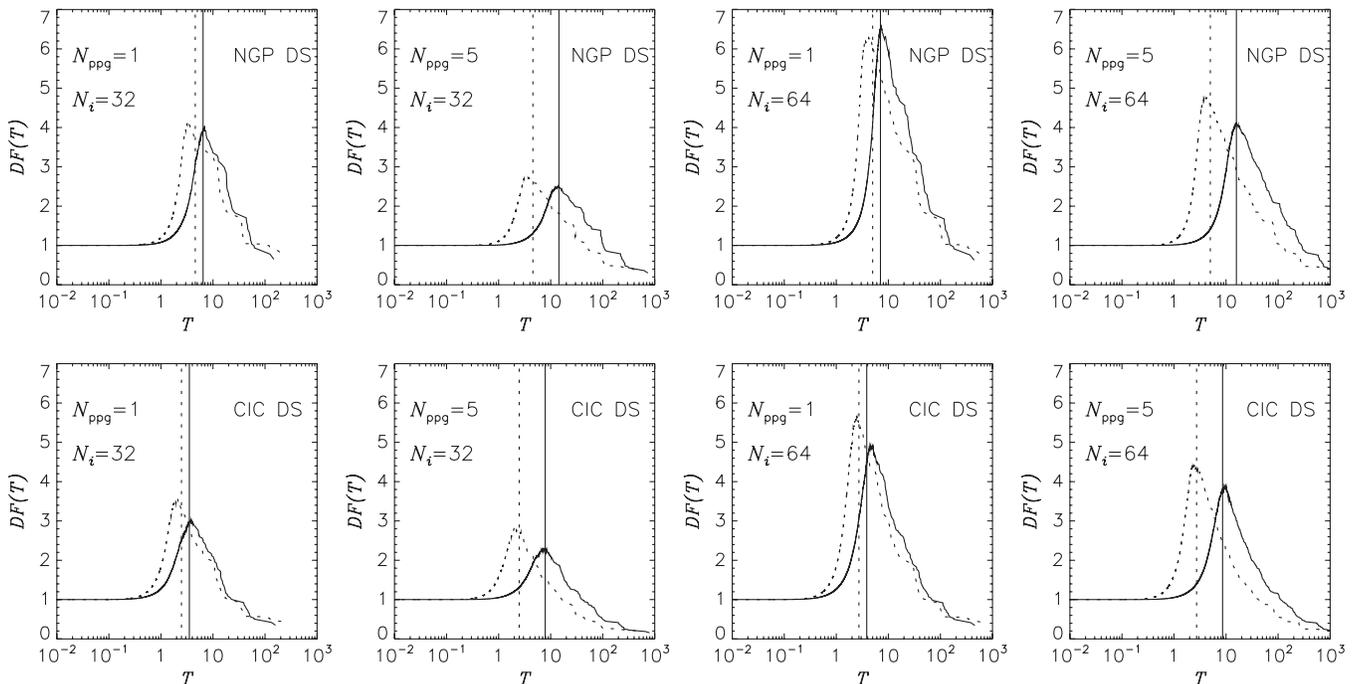


FIG. 9. Denoising factor (DF), defined in Eq. (35), as a function of the threshold  $T$  for the superimposed Gaussian model with:  $N_i = 32$  and  $N_{\text{ppg}} = 1$  (first column);  $N_i = 32$  and  $N_{\text{ppg}} = 5$  (second column);  $N_i = 64$  and  $N_{\text{ppg}} = 1$  (third column);  $N_i = 64$  and  $N_{\text{ppg}} = 5$  (fourth column). Daubechies wavelets of order 2 were used. The NGP DS is given in the top row, and the CIC DS in the bottom. Dashed lines represent thresholding with Anscombe transformation, and solid lines thresholding without Anscombe transformation. The vertical lines denote their corresponding thresholds predicted by Eqs. (41)–(44):  $T_{\text{NGP}}$  (dashed lines) and  $\tilde{T}_{\text{NGP}}$  (solid lines) in the top row, and  $T_{\text{CIC}}$  (dashed lines) and  $\tilde{T}_{\text{CIC}}$  (solid lines) in the bottom row.

TABLE I. Parameters required for the PCG Poisson equation solver within IMPACT-T.

Text	Code	Parameter	Recommended range or allowed values
$M_x$	Mx	Number of Green expansion functions in $x$ -coordinate	$\approx N_i$
$M_y$	My	Number of Green expansion functions in $y$ -coordinate	$\approx N_i$
	Niter	Maximum number of iterations of the PCG	200
$\epsilon$	eps	PCG exit requirement tolerance	$10^{-5}$ – $10^{-4}$
	thresh_L	Threshold for the wavelet-transformed Laplacian	$10^{-14}$ – $10^{-10}$
	wtype	Wavelet family	2, 3, 6, 10 (Daub); 54, 56, 58 (coif); 61, 62, 63 (sym)
	thr_charge	Threshold charge distribution	0 (no), 1 (yes)
	den_type	Denoising type	0 (with Anscombe transform), 1 (without)
	thr_type	Thresholding type	0 (hard), 1 (soft)
	rat_x	Ratio of sizes of pipe and computational box in $x$ -coordinate	4–6 (integer valued)
	rat_y	Ratio of sizes of pipe and computational box in $y$ -coordinate	4–6 (integer valued)
	ispars	Sparse matrix multiplication	0 (disabled), 1 (enabled)

ciently different model particle distributions, we can induce the following: (i) as the number of particles in the simulation  $N$  increases, the particle distribution, as expected, becomes less noisy, and denoising by thresholding becomes less effective, because there is less noise to remove; (ii) for the same average number of particles per grid point  $N_{\text{ppg}}$ , the effectiveness of denoising increases with resolution  $N_i$ ; (iii) CIC DS provides data smoothing, which removes some of the original noise and thus reduces the effectiveness of denoising by thresholding; (iv) the thresholds reported earlier in the section are excellent approximations to the ideal threshold that maximizes the denoising factor DF for charge distributions in a typical beam simulation. Based on these generalizations, we can conjecture that simulations using a Poisson solver with wavelet thresholding will inherently have less noise than those done with conventional solvers. However, it is not possible to quantify directly via the SNR the effectiveness of denoising in PIC simulations, where the “exact” signal is not known. One can conceivably run simulations with a varying number of particles and grid resolution, both with and without wavelet thresholding, to detect empirically the denoising effects. This study is currently underway, and we will report results in a separate publication. An example of such a comparison can be seen in our Figs. 15 and 16. Alternatively, one can implement the full library search and best basis selection approach as discussed in [21–24].

#### IV. APPLICATIONS

Our goal has been to develop a wavelet-based Poisson solver that can be easily merged into existing PIC codes for multiparticle dynamics simulations. As the first step toward that goal, we tested the PCG as a stand-alone Poisson solver.

##### A. Testing the PCG solver

We tested the PCG solver on two idealized particle distributions, one from stellar dynamics and the other

from beam dynamics. We used the PCG solver to compute the potential associated with the Plummer spherical stellar distribution [46] (Fig. 10). Both the density and potential are analytically known and are given by

$$F(r) = \frac{3}{(1+r^2)^{5/2}}, \quad U(r) = -\frac{1}{\sqrt{1+r^2}}, \quad (45)$$

where  $r = \sqrt{x^2 + y^2 + z^2}$ . Here we applied open BCs, which is the natural choice for self-gravitating systems. The potential on the surface of the rectangular computa-

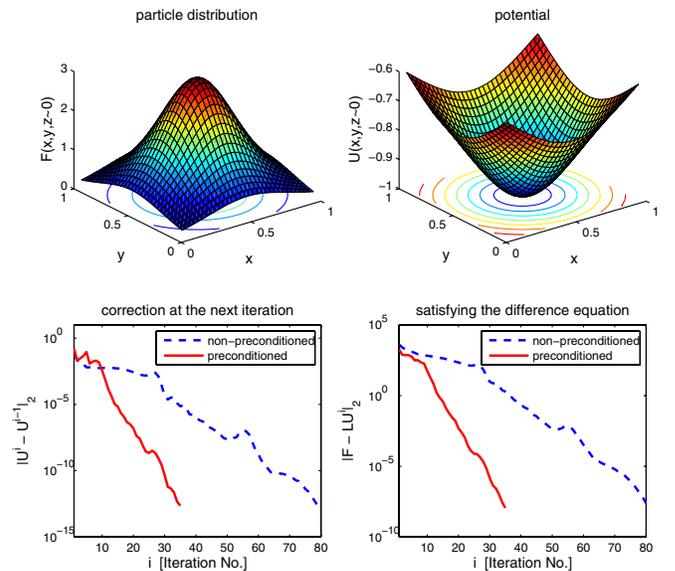


FIG. 10. (Color) Plummer spherical particle distribution (top left) and corresponding potential (top right) at the waist ( $z \approx 0$ ) obtained using the PCG solver. The lower panels show two convergence criteria—correction at the next iteration (bottom left) and the norm of the residual of the difference equation (bottom right)—with (solid line) and without (dashed line) the preconditioner. A poor initial approximation was chosen:  $U(\mathbf{x}, t = 0) = 0$ .

tional grid is specified analytically. The bottom panels of Fig. 10 demonstrate the substantial computational speedup gained by preconditioning.

We then applied the algorithm in a more realistic setting in which only the particle distribution is analytically known, and where the potential on the surface of the computational grid is computed using the analytically known Green function (Fig. 11). The density is an axially symmetric “fuzzy cigar”-shaped distribution of charged particles (mimicking a “beam bunch”) given by

$$F(x, y, z) = d_1(R)d_2(z), \quad (46)$$

$$d_1(R) = \begin{cases} 1 & 0 \leq R \leq R_1, \\ \frac{(R-R_2)^2[R-(3R_1-2R_2)]^2}{4(R_1-R_2)^4} & R_1 \leq R \leq R_2, \\ 0 & \text{otherwise,} \end{cases} \quad (47)$$

$$d_2(z) = \begin{cases} 1 & z_{12} \leq z \leq z_{21}, \\ \frac{(z-z_1)^2[z-(3z_{12}-2z_1)]^2}{4(z_{12}-z_2)^4} & z_1 \leq z \leq z_{12}, \\ \frac{(z-z_2)^2[z-(3z_{21}-2z_2)]^2}{4(z_{21}-z_2)^4} & z_{21} \leq z \leq z_2, \\ 0 & \text{otherwise,} \end{cases} \quad (48)$$

where the beam parameters  $R_1$ ,  $R_2$ ,  $z_1$ ,  $z_2$ ,  $z_{12}$ ,  $z_{21}$  are chosen so that  $0 \leq R_1 < R_2$  and  $z_1 < z_{12} \leq z_{21} < z_2$ . 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. As was true for the case of

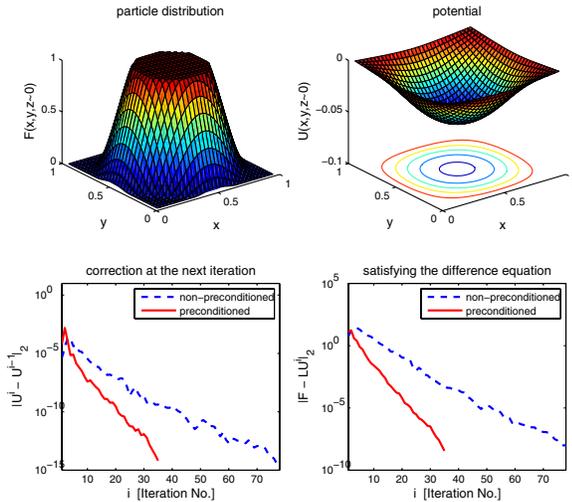


FIG. 11. (Color) Transverse particle distribution (top left) and corresponding potential (top right) at the waist ( $z \approx 0$ ) of the “fuzzy cigar” obtained using the PCG solver. The lower panels show two convergence criteria—correction at the next iteration (bottom left) and the norm of the residual of the difference equation (bottom right)—with (solid line) and without (dashed line) the preconditioner. A poor initial approximation was chosen:  $U(\mathbf{x}, t = 0) = 0$ .

the Plummer sphere, a high-accuracy solution is obtained in about 30 iterations with preconditioning, and 60 iterations without preconditioning. (In both cases,  $U^0 = 0$  was used as the initial approximation. In Sec. II B, we demonstrated that, when a Poisson solver is used as a part of a PIC simulation, a practical, computationally near-optimal way to reduce the number of iterations needed for convergence is to use the solution from the previous time step as an initial approximation at the current time step.)

## B. Integrating PCG into IMPACT-T code

Upon testing the PCG as a stand-alone Poisson solver, we replaced the standard FFT-based Poisson solver in the serial version of IMPACT-T [3,29] with the PCG solver. We chose IMPACT-T because it is a modular, state-of-the-art code for beam dynamics simulations, with ever-increasing popularity in the accelerator community. However, the PCG solver is by no means limited to IMPACT-T—it has been designed to be easily integrated into PIC codes in general.

As we have mentioned already, our approach involves the introduction of an auxiliary computational grid that envelops the beam bunch fairly tightly, and whose boundaries do not coincide with the boundaries of the physical system (i.e., the pipe walls) on which the BCs are prescribed. This means that the BCs on the surface of the computational grid must be calculated before the Poisson solver is invoked to compute the potential in the grid’s interior. In our solver, this is accomplished by using the Green function appropriate for the case of zero potential on the pipe walls and open BCs in the  $z$ -direction. The parameters  $M_x$  and  $M_y$  specify the number of Green-function expansion coefficients in the  $x$ - and  $y$ -directions, respectively.

We use routines for manipulating sparse matrices, which reduce computational load whenever the Laplacian operator is sparse.

We show all new parameters required by the PCG Poisson solver in Table I. Other parameters, such as the grid resolution ( $N_i$ ), grid size ( $h_x, h_y, h_z$ ), the number of particles ( $N$ ), are passed to the solver routine from the driver.

For the simulations presented here, we use hard thresholding,  $M_x = M_y = 30$  and  $\epsilon = 5 \times 10^{-5}$ .

## C. Code benchmarking: IMPACT-T with PCG vs FFT-based IMPACT-T

We tested the resulting wavelet-based code in a realistic setting by modeling the Fermilab/NICADD photoinjector [47] with a nonuniform initial particle distribution at the cathode, and comparing the simulation results to actual laboratory measurements. In addition, we performed extensive benchmarking of the PCG- against the FFT-based IMPACT-T, so as to verify that the two codes produce con-

sistent results. We also compare their performance and point out the advantages of the new wavelet-based Poisson solver.

To verify agreement between the space-charge computation of the two versions of the code, we tested them on two highly nonuniform transverse initial distributions: (i) a considerably nonuniform and asymmetric distribution generated from a real laboratory snapshot of the laser-illuminated photocathode in an actual experiment under suboptimal conditions (*Distribution 1*); and (ii) a 5-beamlet quincunx distribution that can be made by masking the photocathode (*Distribution 2*) [48]. We expect that the nonuniformity and asymmetry of the initial transverse beam distribution will strongly enhance space-charge effects *vis-à-vis* a uniform transverse distribution, thereby “stressing” the Poisson solvers.

We compare numerical results from simulations of these two distributions using IMPACT-T with PCG and the serial version of the FFT-based IMPACT-T code on several important points: (i) rms properties of the beam, (ii) phase-space detail, and (iii) computational speed.

Figure 12 shows the rms properties of the beam in the Fermilab/NICADD photoinjector for *Distribution 1* simulated by FFT-based IMPACT-T (black lines), and IMPACT-T with PCG: without thresholding (green line), thresholded

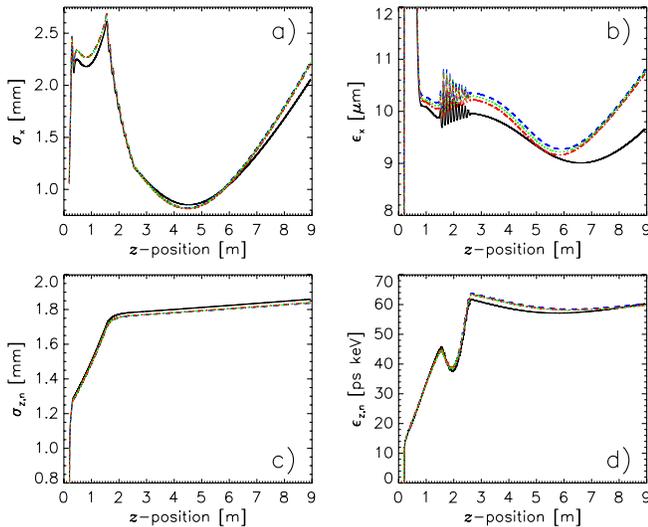


FIG. 12. (Color) *Distribution 1*: Simulation results for the Fermilab/NICADD photoinjector performed with  $N_i = 32$ ,  $N = 200\,000$ , and the standard version of IMPACT-T (black), IMPACT-T with PCG without denoising (green), IMPACT-T with PCG with thresholding and Anscombe transformation (blue), and IMPACT-T with PCG with thresholding and without Anscombe transformation (red): rms beam radius [top left, panel (a)], rms normalized transverse emittance [top right, panel (b); note that the ordinate is magnified], rms bunch length [bottom left, panel (c); note that the ordinate is magnified], rms normalized longitudinal emittance [bottom right, panel (d)]. For IMPACT-T with PCG, we use Daubechies wavelets of order 2 ( $M = 4$ ).

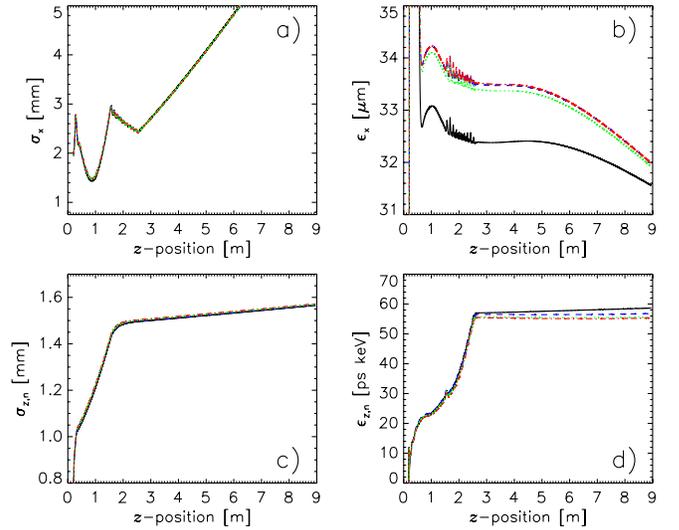


FIG. 13. (Color) *Distribution 2*: The same as in Fig. 12, only for *Distribution 2*.

after Anscombe transform (blue line), and thresholded without Anscombe transform (red line). Figure 13 shows the same for *Distribution 2*. The agreement in rms properties between the FFT-based IMPACT-T and IMPACT-T with PCG is excellent, to within a few percent.

For *Distribution 1*, the beam size in the experiment was measured at different positions of the beam line. Figure 14 compares experiment with numerical simulations using FFT-based IMPACT-T (red line) and IMPACT-T with PCG (blue line).

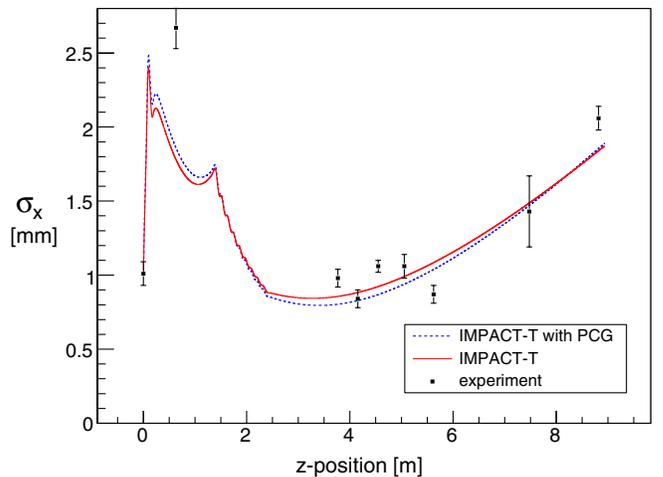


FIG. 14. (Color) *Distribution 1*: rms beam radius for the Fermilab/NICADD photoinjector: as measured in the lab (black dots) and as simulated with FFT-based IMPACT-T (red) and IMPACT-T with PCG (blue). Numerical simulations were done on a  $N_i = 32$  grid and with  $N = 200\,000$  particles, Daubechies wavelets of order 2 and no thresholding. The configuration of the lattice is different from the one used to generate Fig. 12.

These results clearly demonstrate that simulations using both the wavelet-based Poisson solver and the FFT-based solver are in excellent agreement with regard to the computation of beam moments. They also match the measured rms beam size reasonably well.

Figures 15 and 16 show, for the two distributions, integrated transverse cross sections of the beam at different positions down the beam line. Detailed agreement between

FFT-based IMPACT-T and IMPACT-T with PCG in the configuration space is clearly very good, even when the number of macroparticles in the latter is 5 times smaller (also resulting in simulation times being significantly shorter).

Keeping all parameters of the simulation and the number of macroparticles the same, the computational speed of IMPACT-T with PCG is comparable to that of FFT-based IMPACT-T. Since fast wavelet transforms scale as  $\propto$

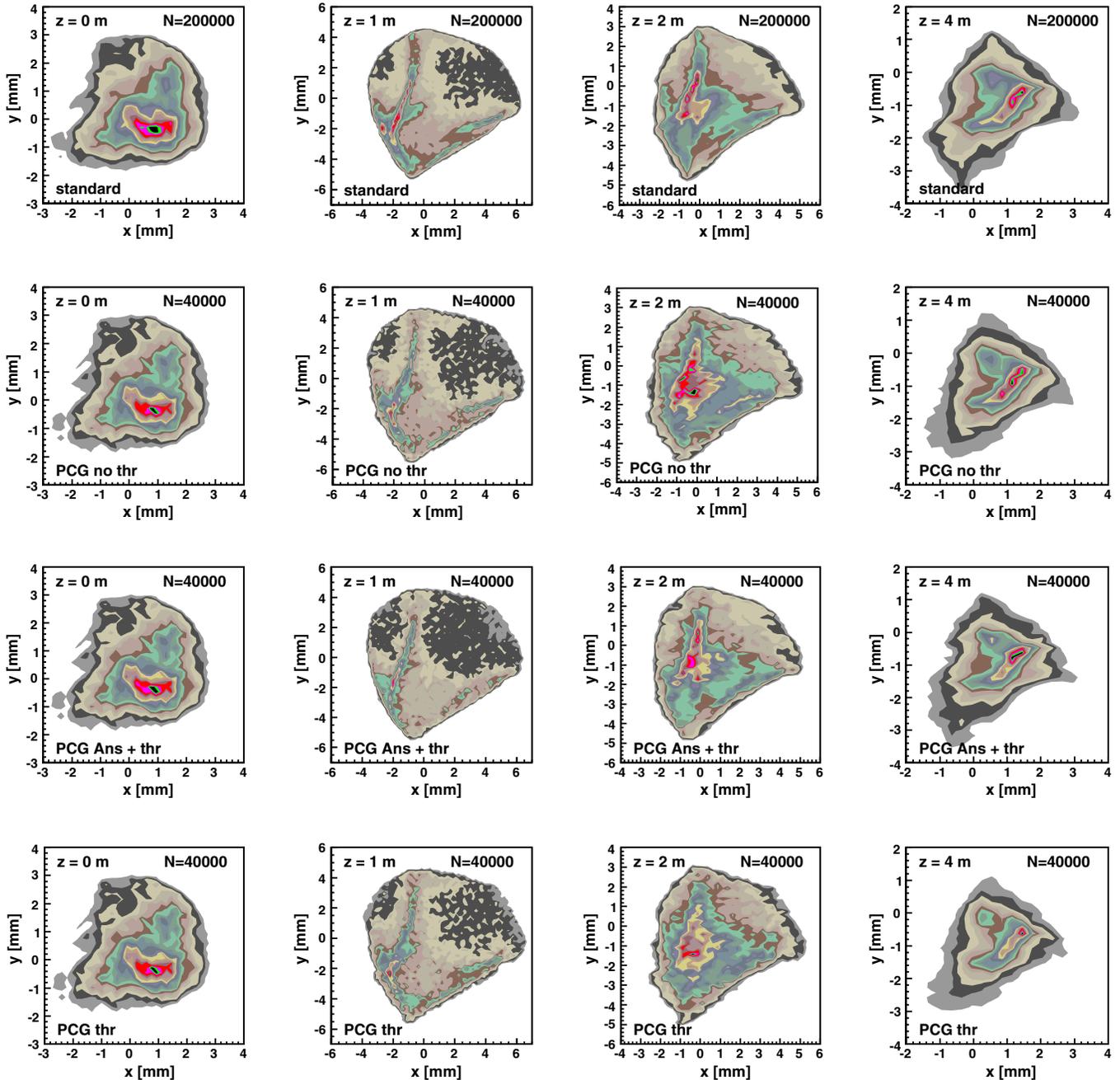


FIG. 15. (Color) *Distribution 1*: Integrated transverse cross section of the beam at different positions down the beam line for Fermilab/NICADD photoinjector simulations with FFT-based IMPACT-T with  $N = 200\,000$  (first row), and IMPACT-T with PCG with  $N = 40\,000$ : with no thresholding (second row), with Anscombe transforms and thresholding (third row), with thresholding (fourth row). The first column shows the transverse cross section of the beam leaving the cathode, second at  $z = 1$  m, third at  $z = 2$  m, and fourth at  $z = 4$  m. Grid resolution is  $N_i = 32$ . Daubechies wavelets of order 2 are used.

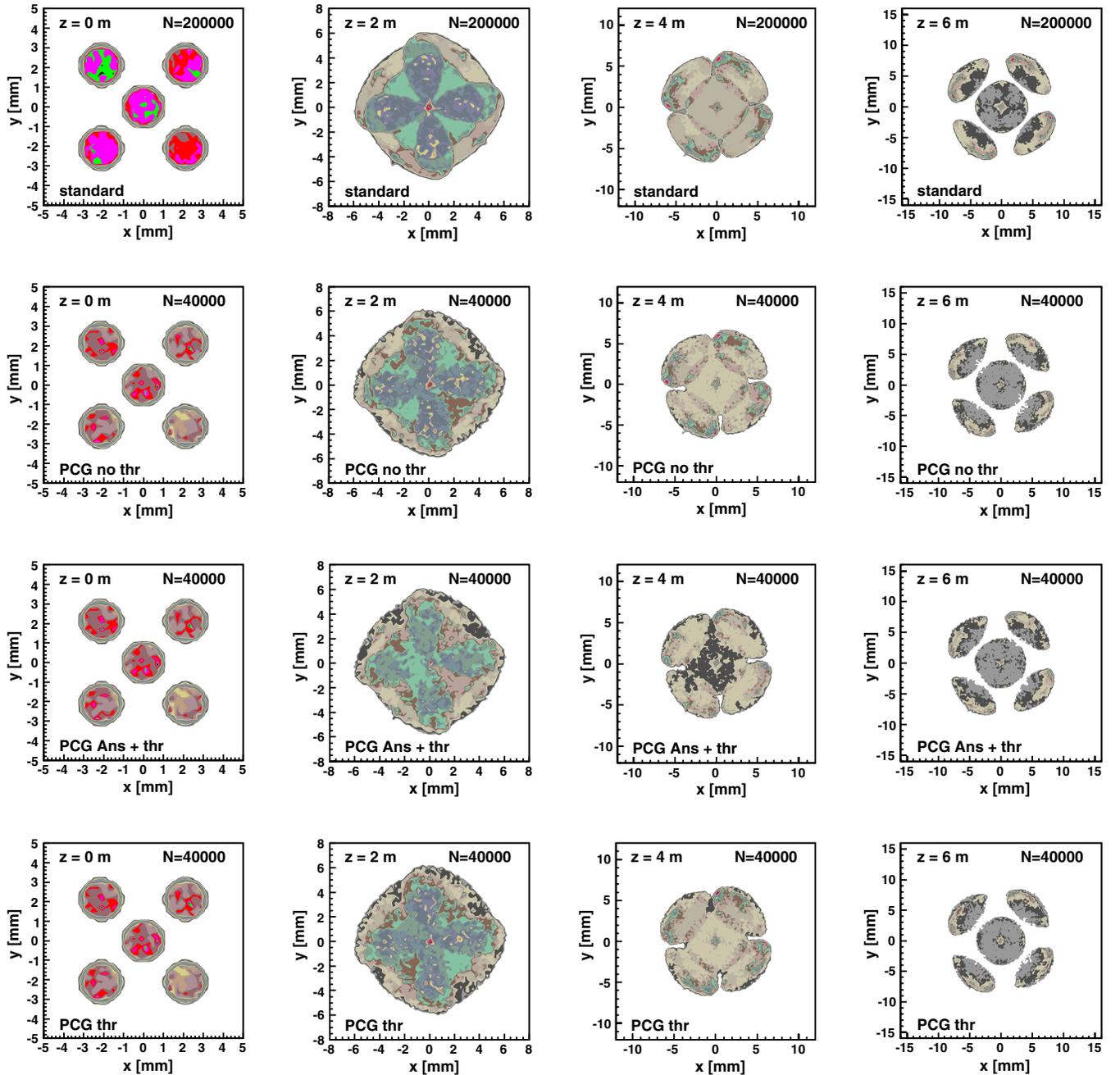


FIG. 16. (Color) *Distribution 2*: The same as in Fig. 15, only for *Distribution 2* and  $z = 2$  m (second column),  $z = 4$  m (third column), and  $z = 6$  m (fourth column).

$\mathcal{O}(MN_{\text{grid}})$ , transforms with wavelet families having larger-size support  $M$  take longer to perform, yield denser operators, and consequently adversely affect the computational speed. Figure 17 shows the relative execution times of different “variants” of IMPACT-T with PCG (no thresholding, thresholding with Anscombe transform, and thresholding without Anscombe transform) relative to the speed of the serial FFT-based IMPACT-T for a 30 000-step simulation with *Distribution 1* initial conditions in the Fermilab/

NICADD photoinjector. We observe that the fastest simulations are achieved with wavelets having smallest compact support, as expected. However, simulations with Daubechies family of order 6 ( $M = 12$ ) are faster than some simulations with wavelet families having smaller support because the convergence of the PCG algorithm at each step of the simulation requires fewer iterations (see Fig. 18).

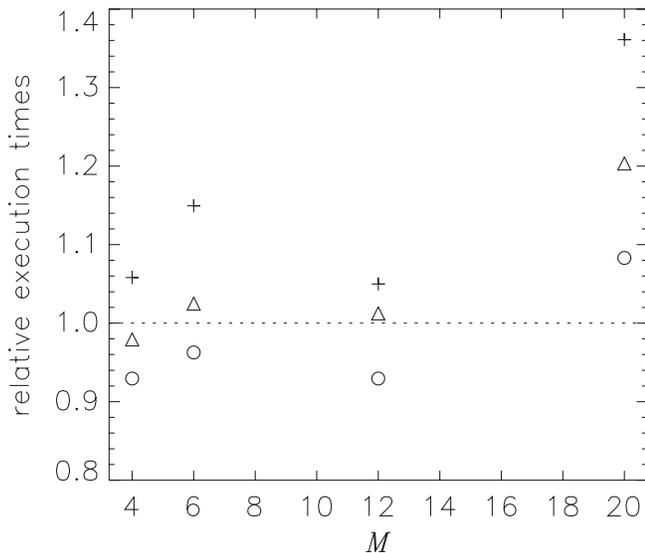


FIG. 17. Comparison of relative execution times of IMPACT-T with PCG vs standard FFT-based IMPACT-T for the full simulation for *Distribution 1* with 30 000 steps,  $N = 125\,000$  particles and resolution  $N_i = 32$ . Speed of the standard IMPACT-T is represented by unity. Circles represent PCG with no thresholding, crosses PCG with Anscombe transformation and thresholding, and triangles PCG with thresholding and no Anscombe transformation.  $M = 4$  represents Daubechies wavelet family of order 2,  $M = 6$  Daubechies family of order 3,  $M = 12$  Daubechies family of order 6,  $M = 20$  Daubechies family of order 10. This figure illustrates only the numerical speed of the new method when compared to the FFT-based method with all parameters of the simulation and the number of macroparticles being equal. The substantial computational savings come from the IMPACT-T simulation with a wavelet-based Poisson solver requiring fewer macroparticles than the FFT-based IMPACT-T for comparable quality results (see text and Figs. 15 and 16).

#### D. Operator and data compression

Formulating the Poisson equation in wavelet space renders operators and data sets sparse. We exploit this sparsity by implementing routines for manipulation of sparse matrices where applicable, which correspondingly reduces the computational load of the algorithm.

A good choice of a wavelet family should provide a compact representation of the signal. The work of Donoho and Johnstone [23,24] demonstrated that for a given application there exist the “ideal” wavelet basis in which the entropy-like “risk” (“cost”) function is minimized and optimal compression is achieved. The closer the actual basis is to the ideal basis, the better the compression that results. In Fig. 19, we compare fractions of wavelet coefficients of the charge retained after thresholding in a typical realistic simulation on a  $N_i = 32$  grid with  $N = 125\,000$  particles for 10 different wavelet families.

Figure 20 shows the average sparsity of the particle distribution in a typical simulation of the Fermilab/NICADD photoinjector done with IMPACT-T with PCG on

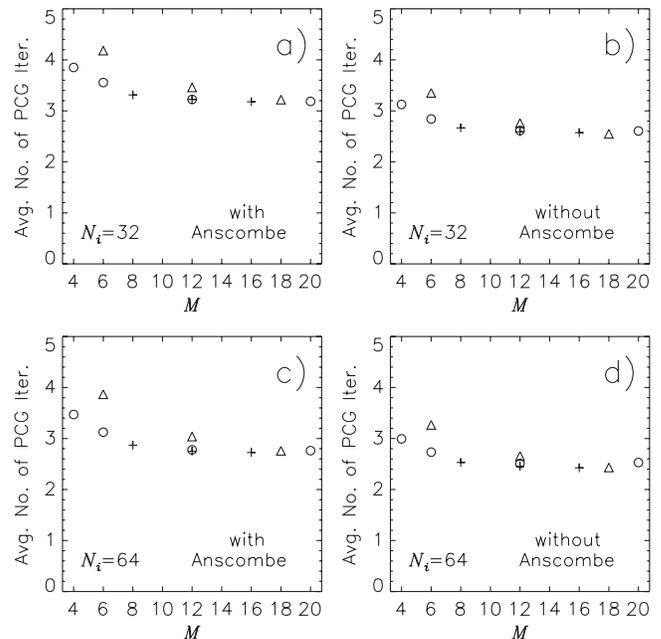


FIG. 18. Average number of iterations of IMPACT-T with PCG for a 30 000-step full simulation of the Fermilab/NICADD photoinjector with *Distribution 1*, and  $N = 125\,000$  and  $N_i = 32$ . The size of the compact support of the wavelet family ( $M$ ) is given on the abscissa for Daubechies wavelets of order 2, 3, 6, and 10 (empty circles), symlets of order 4, 6, 8 (crosses), coiflets of order 1, 2, 3 (triangles).

$N_i = 32$  and  $N_i = 64$  grids (top two rows; left and right, respectively), the same  $N_{\text{ppg}} = 4.58$  and with 10 different wavelet families. The sparsity of the Laplacian operator in wavelet space is shown in the bottom row.

We find that the average fraction of wavelet coefficients retained after thresholding is somewhat smaller for Anscombe-transformed data. Also, the average fraction of coefficients retained after thresholding halves as the resolution is doubled, while the number of particles per grid point remains fixed, for both Anscombe-transformed and nontransformed data.

For these simulations, optimal compression requires, on average, only about 3.7% (2.0%) coefficients from the full expansion, i.e., about 1212 out of 32 768 (about 5243 out of 262 144) for thresholding after applying the Anscombe transform and about 6.1% (3.5%), i.e., about 1999 of 32 768 (about 9175 out of 262 144) for thresholding without the Anscombe transform on a  $N_i = 32$  ( $N_i = 64$ ) grid. When the number of particles per grid point  $N_{\text{ppg}}$  is reduced, the fraction of coefficients retained after thresholding decreases, because the particle distribution becomes more noisy, and thresholding is more efficient in both compression and noise removal (cf. Sec. III).

Therefore, one might expect an even more compact representation of density for simulations with lower  $N_{\text{ppg}}$ . These results are generally valid as long as  $N_{\text{ppg}}$  is large

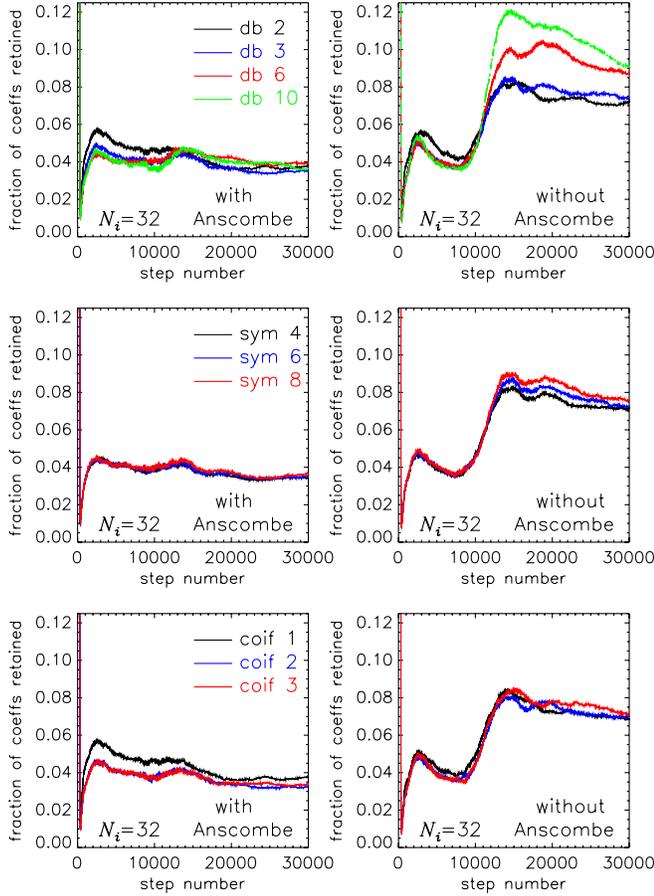


FIG. 19. (Color) Fraction of the wavelet coefficients of the particle distribution retained after thresholding in a fully 3D IMPACT-T with PCG simulations of the NICADD/Fermilab photoinjector on a  $N_i = 32$  grid, with  $N = 125\,000$  particles ( $N_{\text{ppg}} = 4.58$ ). The left column represents the fraction of coefficients retained when the data is Anscombe-transformed, and the right column when the data is not Anscombe-transformed for the 10 different orthogonal wavelet families: Daubechies of order 2, 3, 6, and 10 (top row); symlets of order 4, 6, and 8 (middle row); coiflets of order 1, 2, and 3 (bottom row). Recall, full expansion for  $N_i = 32$  resolution requires  $32^3 = 32\,768$  coefficients, and for  $N_i = 64$  resolution  $64^3 = 262\,144$  coefficients.

enough to resolve the relevant fine-scale structure in the physical distribution.

## V. DISCUSSION AND CONCLUSION

We developed a 3D wavelet-based solver for the Poisson equation with Dirichlet BCs and optimized it for use in PIC simulations. The work represents a natural extension of the earlier work done on wavelet-based preconditioned conjugate gradient solvers for the Poisson equation with periodized or homogeneous Dirichlet BCs [15,16,28]. Whereas some of the methodology and treatment presented here was first reported elsewhere, our formulation of the discretized problem, treatment of BCs and the Laplacian operator is

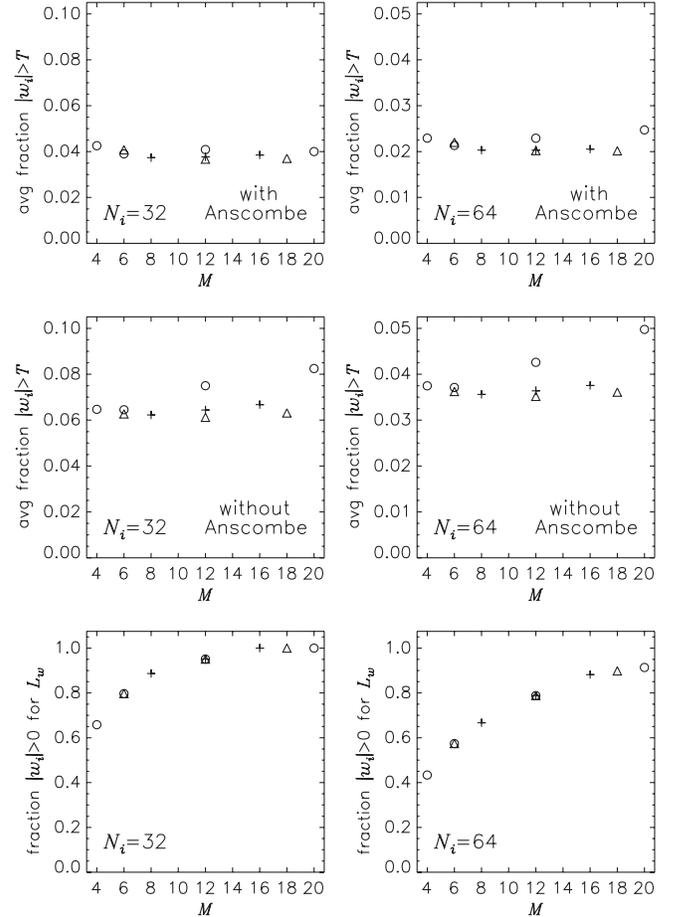


FIG. 20. Average fraction of wavelet coefficients retained to store a particle distribution in a simulation of the Fermilab/NICADD photoinjector done with IMPACT-T with PCG with  $N_{\text{ppg}} = 4.58$ : with the Anscombe transformation (top row), without the Anscombe transformation (middle row). The bottom row shows the number of nonzero wavelet coefficients in the Laplacian operator in wavelet space. The left column shows  $N_i = 32$  ( $N = 125\,000$  particles) and right  $N_i = 64$  ( $N = 1\,000\,000$  particles). The size of the compact support of the wavelet family ( $M$ ) is given on the abscissa for Daubechies wavelets of order 2, 3, 6, and 10 (empty circles), symlets of order 4, 6, 8 (crosses), coiflets of order 1, 2, 3 (triangles).

appreciably different from the periodized or homogeneous Dirichlet problem. To our knowledge, the work reported here constitutes the first application of the wavelet-based multiscale methodology to 3D computer simulations in beam dynamics.

We employ wavelet thresholding to remove effects of numerical noise from simulations. We expect that, in simulations where errors associated with graininess of the distribution function dominate, this denoising procedure will translate into greatly improved overall simulation fidelity.

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 photoinjector simulations performed using IMPACT-T with

the “native” Poisson solver (based on Green 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 and transverse beam spots). This result enables us to move from the proof-of-concept stage to the advanced optimization and application-specific algorithm design.

Our results confirm the expectation that one can achieve significant compression of the charge density data in realistic simulations. As seen in Fig. 19, for each of the 10 wavelet bases tested for this paper, of the order of only 5% or less of the total number of wavelet coefficients remained nonzero after thresholding carried out according to prescription in [23]. Consistent with the fact that dynamical evolution in these simulations did not involve development of instabilities, none of the 10 bases is clearly preferable to others at any point throughout the simulation. (In terms of overall computational speed, Daubechies families of order 2 and 6 enjoy a moderate advantage, as seen in Fig. 17.) The above comparison can be thought of as a (intentionally) simplified version of the full basis-library search approach of Coifman, Meyer, and Wickerhauser [21,22]; clearly, one would like to use the same basis throughout the simulation and avoid, if possible, a full library search at every time step. However, it is our expectation that an entropy-based basis selection process will be indispensable in simulations where instabilities such as microbunching are prominently present. While subjective choice of a basis carries with it an obvious danger of “smoothing away” the physically important fine-scale structure that serves as a seed for the instability growth, the search for best basis based on a clearly defined objective function allows for simultaneous compression and denoising, along with a quantifiable degree of certainty that what has been discarded is actually noise. We plan to apply and further explore these ideas in simulations where longitudinal space charge- and CSR-driven microbunching instability is important.

Our current efforts are focused on several areas that encompass both algorithm optimization and applications. On the optimization side, we continue work on implementing procedures for storage and manipulation of sparse operators and data sets, which will directly translate into increased computational efficiency. We are also exploring ways to compute more efficiently the potential on the boundary of the computational grid (as distinct from the physical boundaries of the system), so as to reduce the computational overhead at each step of the simulation. Finally, we have yet to address 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 applying the multiscale wavelet formulation to the problem of high-precision 3D modeling of CSR and its effects on the dynamics of beams in a variety of accelerator systems.

Our solver can also be integrated into existing PIC codes for modeling self-gravitating systems, such as star clusters, galaxies, or clusters of galaxies. We therefore plan to cross into astrophysics, which will provide an important field of application of our solver.

## ACKNOWLEDGMENTS

We are thankful to Daniel Mihalcea for generating Fig. 14, initial conditions and lattice configuration files for numerical simulations, and his many valuable suggestions. Benjamin Sprague has been instrumental in optimizing IMPACT-T with PCG. Ji Qiang provided valuable advice in integrating the solver into the IMPACT-T suite. This work was supported by Air Force Contract No. FA9471-040C-0199 to NIU and by Department of Energy Contract No. DE-AC02-05CH11231 to LBNL and Department of Energy Grant No. DE-FG02-04ER41323 to NIU.

## APPENDIX A: VARIANCE OF NOISE FOR THE NGP AND CIC DEPOSITION SCHEMES

In this Appendix we consider the process of sampling a continuous charge density distribution by  $N$  particles, with subsequent deposition of the charge onto a grid. We limit discussion to the NGP and CIC particle-deposition schemes. Our goal is to calculate the expectation and variance of  $Q^{(i)}$ , the aggregate charge assigned to the  $i$ th node of the lattice, in the two schemes. It is assumed that each particle carries the same charge  $q_0 \equiv Q_{\text{tot}}/N$ .

The aggregate charge  $Q^{(i)}$  deposited onto the node  $i$  can be viewed as the sum of  $N$  independent and identically distributed random variables  $\{Q_1, \dots, Q_N\}$ :

$$Q^{(i)} = \sum_{k=1}^N Q_k; \quad (\text{A1})$$

it may be convenient to visualize the sampling process as particles being added one at a time. All  $\{Q_k\}$  are then distributed as the “prototype” random variable  $Q$ , the charge assigned to the node  $i$  when a new particle is added to the sample. For notational convenience, we also introduce an auxiliary variable  $I$ , which assumes value 1 if the sampled position of a  $Q_k$  is within the support  $\Omega^{(i)}$  of the charge particle-deposition function of the CIC method, centered on the node  $i$ , and zero value otherwise. (For both NGP and CIC,  $\Omega^{(i)}$  is defined to be a  $D$ -dimensional cube of edge length  $2h$ , centered on the node  $i$ .)

In what follows, we assume that the mesh is sufficiently fine for the probability density function to be approximated by a linear function on  $\Omega^{(i)}$ . With this assumption, one readily finds

$$E[Q|I=1] = q_0 2^{-D}, \quad E[Q^2|I=1] = q_0^2 2^{-D} \quad (\text{A2})$$

and

$$\text{Var}(Q|I=1) = \left(\frac{1}{2^D} - \frac{1}{4^D}\right)q_0^2 \quad (\text{A3})$$

for the NGP DS, and similarly

$$E[Q|I=1] = q_0 2^{-D}, \quad E[Q^2|I=1] = q_0^2 3^{-D} \quad (\text{A4})$$

and

$$\text{Var}(Q|I=1) = \left(\frac{1}{3^D} - \frac{1}{4^D}\right)q_0^2 \quad (\text{A5})$$

for the CIC DS. Clearly,  $E[Q|I=0] = 0$  and  $\text{Var}(Q|I=0) = 0$  for both NGP DS and CIC DS.

We now introduce a new random variable  $K$ , the number of particles for which  $I=1$ ; for a given realization,  $K$  will assume a value  $k$  between 0 and  $N$  ( $0 \leq k \leq N$ ). For a given  $k$ ,

$$E[Q^{(i)}|K=k] = kE[Q|I=1] \quad (\text{A6})$$

and

$$\text{Var}(Q^{(i)}|K=k) = k \text{Var}(Q|I=1), \quad (\text{A7})$$

since the  $\{Q_1, \dots, Q_N\}$  are independent and identically distributed (as  $Q$ ). Since  $K$  itself is a random variable, the expectation and variance of  $Q^{(i)}$  has to be calculated using the double expectation theorem:

$$E[Q^{(i)}] = E[E[Q^{(i)}|K]] = E[K]E[Q|I=1], \quad (\text{A8})$$

$$\begin{aligned} \text{Var}(Q^{(i)}) &= E[\text{Var}(Q^{(i)}|K)] + \text{Var}(E[Q^{(i)}|K]) \\ &= E[K]\text{Var}(Q|I=1) + (E[Q|I=1])^2 \text{Var}(K). \end{aligned} \quad (\text{A9})$$

$E[K]$  and  $\text{Var}(K)$  that enter the expressions above can be calculated by noting that  $K$  is the number of ‘‘successes’’ ( $I=1$ ) in a series of  $N$  trials, so that it is distributed according to a binomial distribution with ‘‘probability of success’’  $P_i$  equal to the continuous probability density function integrated over the  $\Omega^{(i)}$ :

$$E[K] = NP_i, \quad (\text{A10})$$

$$\text{Var}(K) = NP_i(1 - P_i). \quad (\text{A11})$$

Analogously to  $P_i$ , one can define  $p_i$  as the value of the integral of the continuous probability density over  $\omega^{(i)}$ , a  $D$ -dimensional cube of edge length  $h$  centered on the node  $i$ . With this notation,

$$P_i \approx 2^D p_i, \quad (\text{A12})$$

and, combining Eqs. (A2), (A4), (A8), (A10), and (A12), we finally obtain

$$E[Q^{(i)}] = q_0 N p_i \quad (\text{A13})$$

for both NGP DS and CIC DS, and, combining Eqs. (A2)–

(A5) and (A9)–(A12),

$$\text{Var}(Q^{(i)}) = q_0^2 N P_i \left(\frac{1}{2^D} - \frac{P_i}{4^D}\right) \approx q_0^2 N p_i (1 - p_i) \quad (\text{A14})$$

for the NGP DS,

$$\text{Var}(Q^{(i)}) = q_0^2 N P_i \left(\frac{1}{3^D} - \frac{P_i}{4^D}\right) \approx q_0^2 N p_i \left[\left(\frac{2}{3}\right)^D - p_i\right] \quad (\text{A15})$$

for the CIC DS.

One possible global measure of the error associated with sampling-and-deposition noise is the norm-like quantity

$$\sigma^2 \equiv \frac{1}{N_{\text{grid}}} \sum_{i=1}^{N_{\text{grid}}} \text{Var}(Q^{(i)}). \quad (\text{A16})$$

In a typical beam dynamics PIC simulation, the charge density distribution hardly ever possesses highly prominent, strongly localized peaks; in addition, the grid is always adjusted so as to minimize (or, at least, greatly reduce) the number of grid points in the regions of zero density. This means that, although

$$\frac{1}{N_{\text{grid}}} \leq \sum_{i=1}^{N_{\text{grid}}} p_i^2 \leq 1 \quad (\text{A17})$$

(where the lower bound is attained for the uniform distribution, and the upper bound is reached when the whole distribution is assigned to a single node), in practice the sum in Eq. (A17) is  $\mathcal{O}(1/N_{\text{grid}})$ , and can be neglected in computing  $\sigma^2$  using Eq. (A16). With this simplification, for the two particle-deposition schemes considered here we obtain

$$\sigma_{\text{NGP}}^2 \approx \frac{q_0^2 N}{N_{\text{grid}}} = \frac{Q_{\text{tot}}^2}{NN_{\text{grid}}} \quad (\text{A18})$$

and

$$\sigma_{\text{CIC}}^2 \approx \left(\frac{2}{3}\right)^D \frac{q_0^2 N}{N_{\text{grid}}} = \left(\frac{2}{3}\right)^D \frac{Q_{\text{tot}}^2}{NN_{\text{grid}}}. \quad (\text{A19})$$

## APPENDIX B: RELATING SNR TO THE STANDARD DEVIATION OF NOISE $\sigma$

Combining Eq. (23) with Eq. (32), we obtain

$$\text{SNR} = \frac{[\sum_{i=1}^{N_{\text{grid}}} \bar{q}_i^2]^{1/2}}{\sigma N_{\text{grid}}} \equiv \frac{R(N_{\text{grid}})}{\sigma N_{\text{grid}}}. \quad (\text{B1})$$

In a PIC simulation, the total charge is conserved for each resolution level:

$$Q_{\text{tot}} \equiv \sum_{i=1}^{N_{\text{grid}}} \bar{q}_i = \sum_{i=1}^{N_{\text{grid}}} q_i = \text{const.} \quad (\text{B2})$$

This means that as the grid is refined from the resolution

level  $k - 1$  to  $k$ , where the number of grid points in each coordinate is  $N_i^k = 2^k$ , and the total number of grid points is  $N_{\text{grid}} = 2^{Dk}$ , the contribution of the points on the coarser grid will be about  $1/2^D$  of the new total. The new, refined grid can be viewed as a superposition of  $2^D$  coarse grids appropriately shifted. If one assumes that all such shifted grids have the same properties, which is quite reasonable considering that they all sample the same charge distribution,

$$Q_{\text{tot}} \equiv \sum_{i=1}^{2^{Dk}} \bar{q}_{k,i} \approx 2^D \sum_{i=1}^{2^{D(k-1)}} \frac{1}{2^D} \bar{q}_{k-1,i}, \quad (\text{B3})$$

then

$$\begin{aligned} R^2(2^{Dk}) &\equiv \sum_{i=1}^{2^{Dk}} \bar{q}_{k,i}^2 \approx 2^D \sum_{i=1}^{2^{D(k-1)}} \left( \frac{1}{2^D} \bar{q}_{k-1,i} \right)^2 \\ &= 2^{-D} \sum_{i=1}^{2^{D(k-1)}} \bar{q}_{k-1,i}^2 = 2^{-D} R^2(2^{D(k-1)}) \\ &\approx 2^{-kD} R^2(2^0) \equiv r^2 N_{\text{grid}}^{-1}, \end{aligned} \quad (\text{B4})$$

which means that

$$R(N_{\text{grid}}) \approx r N_{\text{grid}}^{-1/2}, \quad (\text{B5})$$

where  $r$  is essentially a constant which depends on the distribution  $q$ . Combining the relationship given in Eq. (B5) with Eq. (B1) and Eqs. (24) and (25) establishes the inverse relationship between the SNR and the standard deviation of noise for both particle-deposition schemes:

$$\text{SNR}_{\text{NGP}} \approx \frac{r}{\sigma_{\text{NGP}} N_{\text{grid}}^{1/2}} = \frac{r N^{1/2}}{Q_{\text{tot}} N_{\text{grid}}^{1/2}} = \frac{r}{Q_{\text{tot}}} N_{\text{ppg}}^{1/2}, \quad (\text{B6})$$

$$\text{SNR}_{\text{CIC}} \approx \frac{r}{\sigma_{\text{CIC}} N_{\text{grid}}^{1/2}} = \frac{r N^{1/2}}{a Q_{\text{tot}} N_{\text{grid}}^{1/2}} = \frac{r}{a Q_{\text{tot}}} N_{\text{ppg}}^{1/2}. \quad (\text{B7})$$

We have thus affirmed that  $\text{SNR} \propto N_{\text{ppg}}^{1/2}$ , which is well known.

---

[1] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (Institute of Physics Publishing, London, 1988).  
[2] J. Dawson, *Rev. Mod. Phys.* **55**, 403 (1983).  
[3] J. Qiang, R. Ryne, S. Habib, and V. Decyk, *J. Comput. Phys.* **163**, 434 (2000).  
[4] F. Combes, F. Debbasch, D. Friedli, and D. Pfenniger, *Astron. Astrophys.* **233**, 82 (1990).  
[5] H. Couchman, *Astrophys. J.* **368**, L23 (1991).  
[6] A. Melott, R. Splinter, S. Shandarin, and Y. Suto, *Astrophys. J.* **479**, L79 (1997).

[7] R. Splinter, A. Melott, S. Shandarin, and Y. Suto, *Astrophys. J.* **497**, 38 (1998).  
[8] A. Romeo, C. Horellou, and J. Bergh, *Mon. Not. R. Astron. Soc.* **342**, 337 (2003).  
[9] A. Romeo, C. Horellou, and J. Bergh, *Mon. Not. R. Astron. Soc.* **354**, 1208 (2004).  
[10] K. Heitmann, P. Ricker, M. Warren, and S. Habib, *Astrophys. J. Suppl. Ser.* **160**, 28 (2005).  
[11] C. Birdsall and A. Langdon, *Plasma Physics via Computer Simulations* (Institute of Physics Publishing, London, 1991).  
[12] A. Friedman, D. Grote, and I. Haber, *Phys. Fluids B* **4**, 2203 (1992).  
[13] F. H. Harlow, *Meth. Comput. Phys.* **3**, 319 (1964).  
[14] D. Hamilton, F. Lindholm, and A. Marshak, *Principles and Applications of Semiconductor Device Modeling* (Rinehart and Winston, New York, 1971).  
[15] G. Beylkin, in *Wavelets: Mathematics and Applications*, edited by J. Benedetto and M. Frazier (CRC Press LLC, Boca Raton, Florida, 1993).  
[16] A. Averbuch, G. Beylkin, R. Coifman, P. Fischer, and M. Israeli, in *Signal and Image Representation in Combined Spaces*, edited by Y. Zeevi and R. Coifman (Academic Press, San Diego, 1995).  
[17] Z. Huang, M. Borland, P. Emma, J. Wu, C. Limborg, G. Stupakov, and J. Welch, *Phys. Rev. ST Accel. Beams* **7**, 074401 (2004).  
[18] I. Pogorelov, J. Qiang, R. Ryne, M. Venturini, A. Zholents, and R. Warnock, *Proceedings of the 9th International Computational Accelerator Physics Conference, 2006*, p. 182 (WEP01).  
[19] M. Berger and P. Colella, *J. Comput. Phys.* **82**, 64 (1989).  
[20] J. Bella, M. Berger, J. Saltzman, and M. Welcome, *SIAM J. Sci. Comput.* **15**, 127 (1994).  
[21] R. Coifman and M. Wickerhauser, *IEEE Trans. Inf. Theory* **38**, 713 (1992).  
[22] R. Coifman, Y. Meyer, and M. Wickerhauser, in *Wavelets and Their Applications*, edited by M. B. Ruskai *et al.* (Jones and Bartlett, Boston, 1992).  
[23] D. Donoho and I. Johnstone, *Biometrika* **81**, 425 (1994).  
[24] D. Donoho and I. Johnstone, *Stanford University internal report, 1994*, <http://www-stat.stanford.edu/~donoho/Reports/1994/idealbasis.pdf>.  
[25] N. Saito, in *Wavelets in Geophysics*, edited by E. Foufoula-Georgiou and P. Kumar (Academic Press, San Diego, 1994).  
[26] C. Bohn, *AIP Conf. Proc.* **647**, 81 (2002).  
[27] G. Bassi, T. Agoh, M. Dolhus, L. Giannessi, R. Hajima, A. Kabel, T. Limberg, and M. Quattromini, *Nucl. Instrum. Methods Phys. Res., Sect. A* **557**, 189 (2006).  
[28] A. Averbuch, G. Beylkin, R. Coifman, P. Fischer, and M. Israeli, *Institut de Mathématiques de Bordeaux Internal Report*, 2003.  
[29] J. Qiang and R. Ryne, *Comput. Phys. Commun.* **138**, 18 (2001).  
[30] J. Qiang, S. Lidia, R. Ryne, and C. Limborg-Deprey, *Phys. Rev. ST Accel. Beams* **9**, 044204 (2006).  
[31] I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992).  
[32] M. Wickerhauser, *Adaptive Wavelet Analysis From Theory to Software* (A K Peters, Wellesley, Massachusetts, 1994).

- [33] M. Misiti, Y. Misiti, G. Oppenheim, and J. Poggi, *Wavelet Toolbox for Use with MATLAB* (The MathWorks, Inc., Natick, Massachusetts, 1997).
- [34] S. Goedecker, *Wavelets and Their Applications* (Presses Polytechniques et Universitaires Romandes, Lausanne, 1998).
- [35] S. Mallat, *A Wavelet Tour of Signal Processing* (Elsevier, San Diego, 1998).
- [36] S. Goedecker and O. Ivanov, *Comput. Phys.* **12**, 548 (1998).
- [37] G. Golub and C. Van Loan, *Matrix Computations* (Johns Hopkins, Baltimore, 1996).
- [38] M. R. Hestenes and E. Stiefel, *J. Res. Nat. Bur. Stand.* **49**, 409 (1952).
- [39] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing, 2nd edition* (University of Cambridge, Cambridge, England, 1992).
- [40] R. Coifman and D. Donoho, in *Wavelet and Statistics*, edited by A. Antoniadis and G. Oppenheim (Springer, New York, 1995).
- [41] F. Anscombe, *Biometrika* **35**, 246 (1948).
- [42] A. Stuart and J. Ord, *Advanced Theory of Statistics* (Oxford University Press, New York, 1991), Vol. 2.
- [43] D. Donoho, Stanford University Internal Report, 1993, <http://www-stat.stanford.edu/~donoho/Reports/1993/toulouse.ps.Z>.
- [44] F. Murtagh, J. Starck, and A. Bijaoui, *Astron. Astrophys. Suppl. Ser.* **112**, 179 (1995).
- [45] J. Starck, D. Donoho, and E. Candés, *Astron. Astrophys.* **398**, 785 (2003).
- [46] H. Plummer, *Mon. Not. R. Astron. Soc.* **71**, 460 (1911).
- [47] See <http://www.nicadd.niu.edu/fnpl/>
- [48] R. Tikhoplav, A. Melissinos, J. Li, and P. Piot, Particle Accelerator Conference, 2005.