Particle-in-Cell Beam Dynamics Simulations With A Wavelet-Based Poisson Solver

Balsa Terzic$^1$, Ilya V. Pogorelov$^2$, Courtlandt L. Bohn$^1$

$^1$Beam Physics and Astrophysics Group, Northern Illinois University, DeKalb, IL 60115
$^2$Accelerator and Fusion Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

(Dated: September 1, 2006)

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 simulations. The solver is based on the operator formulation of the conjugate gradient algorithm, for which effectively diagonal preconditioners are available in wavelet bases. Due to 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.

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 [4-6], plasma physics [2, 7], heavy-ion-beam-driven inertial fusion [8], hydrodynamics of compressible fluid flows [9], and semiconductor device design [10]. 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 timestep as the initial approximation used in solving the Poisson equation one timestep 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 Poisson operator remains sparse, while, unlike in the original basis, there also exist in wavelet bases effectively diagonal preconditioners for the Poisson operator [11, 12]. 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 [13]. 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. 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 $O(N)$ 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. 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 wavelet 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
There are four main reasons that a wavelet-based Poisson solver is of interest:

1. Solving the problem in wavelet space enables retaining information about the dynamics over the hierarchy of scales spanned by the wavelet expansion.

2. Wavelet formulation also allows for natural removal of numerical noise (denoising) by thresholding of the wavelet coefficients.

3. By the same token, relevant data sets (the particle distribution and potential) can be represented compactly using only a fraction of the wavelet coefficients.

4. 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 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 timestep, 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 timestep 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.

II. WAVELET-BASED POISSON SOLVER

Almost from their inception, wavelets have been used to solve partial differential equations (PDEs), elliptic in particular [11, 12, 17–19]. A general discourse on solving PDEs including Poisson’s equation using wavelet formalism is provided in [17]. Wavelet-based methods were used to solve the Poisson equation with homogeneous boundary conditions (BCs) in 1D [11] and periodic BCs in 1D, 2D and 3D [12, 19]. 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

\[ \Delta u = f, \quad u_{\text{bnd}} = g \]  

where \( \Delta \) is the continuous Laplacian operator, and \( u_{\text{bnd}} \) is defined only on the boundary. In order to solve it computationally, one has to discretize the problem, \( i.e., \) restrict the domain of the functions and operators to a finite and discrete domain.

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

\[ LU = F, \quad U_{\text{bnd}} = G \]  

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.

The discretized Laplacian operator \( L \) is given in terms of the finite difference scheme involving a 3-point stencil:

\[ L \equiv L_x + L_y + L_z \equiv h_x^{-2} (\delta_x^- - 2 + \delta_x^+) + h_y^{-2} (\delta_y^- - 2 + \delta_y^+) + h_z^{-2} (\delta_z^- - 2 + \delta_z^+), \]
where $\delta^-$ and $\delta^+$ are backward and forward shift operators, respectively, in the coordinate noted in the subscript. $h_x$, $h_y$, and $h_z$ denotes 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:

$$D_{xx} = h_x^{-2}\left(\delta^--2+\delta^+\right) + O(h_x^2).$$

(5)

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

$$S_x(\omega) = e^{-i\omega} - 2 + e^{i\omega} = -(2 - 2\cos\omega).$$

(6)

In a similar manner, one can compute the transfer functions for other finite-difference approximations [such as 5-point stencil $h^{-2}(-\delta^-+16\delta^-+30+16\delta^+\delta^-)/12$], where $\delta^-$ and $\delta^+$ are double-backward and double-forward shifts, respectively. Figure 1 shows the transfer function due to the continuous operator $\Delta$ (solid line) and the ones corresponding to the discretized operator $L_x$ with 3-point (dotted line) and 5-point (dashed line) stencils. Whereas the discretized operator $L_x$ with a 5-point stencil provides a closer approximation to the continuous operator for a broader range of low frequencies, it does not suppress higher frequencies (often associated with numerical noise) as well as the one with a 3-point stencil. In our present study, we use a 3-point stencil. (Alternatively, the discretization of the physical problem can be done with finite-element methods, but that is beyond the scope of this paper. We will further explore this possibility in future versions of the algorithm.)

B. Preconditioned Conjugate Gradient Method: Preconditioning and Convergence Rate

Equation (3) represents a well-known problem in numerical analysis. It can be solved using a number of iterative methods, such as multigrid, successive over-relaxation, Gauss-Seidel, Jacobi, steepest descent, or CG. For the work presented here, we generalized to three dimensions the preconditioned conjugate gradient (PCG) method, because it best harnesses advantages afforded by operator preconditioning in wavelet space and a “smart” initial approximation [20, 21].

The PCG method updates the initial solution along the conjugate directions until the exit requirement

$$|LU - F|_2 \leq \varepsilon^2 |F|_2,$$  \hspace{1cm} (7)

is satisfied in 2-norm $|\cdot|_2$. The convergence rate of the method is dependent on the condition number $\kappa$ of the operator $L$, defined as the ratio of the largest and smallest eigenvalues:

$$|U - U^i|_2 \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^i |U|_2,$$  \hspace{1cm} (8)

where $U^i$ is the approximation to the exact solution $U$ after the $i$th iteration. The closer the condition number is to unity, the faster the approximation $U^i$ approaches the exact solution $U$. The condition number of the Laplacian operator $L$ on a grid is proportional to the square of the grid resolution (number of grid points in each coordinate) $N_i$, i.e., $\kappa(L) \propto O(N_i^2)$. More precisely, the condition numbers for the 3- and 5-point finite-difference stencils are given by

$$\kappa_3 = \frac{4}{2 - 2\cos(\pi/N_i)},$$  \hspace{1cm} (9)

$$\kappa_5 = \frac{32}{15 - 16\cos(\pi/N_i) + \cos(2\pi/N_i)}.$$  \hspace{1cm} (10)

Such large condition numbers lead to slow convergence. However, in wavelet space, there is an effectively diagonal preconditioner $P$ for the wavelet-transformed Laplacian operator $L_w$. In 1D, it is given by a $N_i \times N_i$ matrix

$$P_{k,l} = 2^l \delta_{k,l},$$  \hspace{1cm} (11)

where $n = \log_2 N_i$, $1 \leq j \leq n$, and $k$ and $l$ are such that $N_i/2^{n-1} \geq k,l \geq N_i/2^1 + 1$ and $P_{N_i,N_i} = N_i$ (see Fig. 2). This preconditioner was used to reduce the condition number of the periodized Laplacian operator to $O(1)$ [11, 12, 19]. Applying a preconditioner to data is equivalent to multiplying the wavelet-transformed data $w_{i'}$ ($i' = 1, \ldots, N_{data}$) by $P_{i',j'}$. Similarly, in 3D, applying a preconditioner to wavelet-transformed data is equivalent to multiplying each wavelet coefficient $w_{i',j',k'}$ by $P = \min\left(P_{i',i'},P_{j',j'},P_{k',k'}\right)$. The preconditioner effectively “bunches up” the eigenvalues of the system, thus

![FIG. 1: Transfer function $S(\omega)$ for the continuous operator $\Delta$ (solid line) and the ones corresponding to the discretized operator $L$ with a 3-point (dotted line) and a 5-point (dashed line) stencils. $\nu$ is the Nyquist frequency, the highest frequency attainable at a given resolution.](image-url)
FIG. 2: 1D diagonal preconditioner $P_{i,j}$ for $N_i = 32$ used for decreasing the condition number of the $N_i \times N_i$ wavelet-transformed Laplacian operator $L_w$.

reducing the ratio between the largest and the smallest one.

After transforming to wavelet space (denoted by the subscript $w$) and preconditioning, the linear algebraic problem in Eq. (3) then becomes

$$ (PL_w P)(P^{-1}U_w) = PF_w. \quad (12) $$

The preconditioner $P$ reduces the condition number of the Laplacian operator with inhomogeneous Dirichlet BCs in wavelet space to $\kappa(PL_w P) \propto O(N_i)$, thereby greatly improving the convergence rate. We also observe that, whereas after preconditioning the condition number becomes $\propto O(N_i)$, 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 3 shows the condition numbers, including the ratio between the second largest to the smallest eigenvalues after preconditioning in wavelet space of Daubechies family of order 2 (dashed line is $\propto \text{const}$).

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, $U(z \to \infty) \to 0$. 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:

$$ \rho^{lm}(z) = \frac{4}{AB} \int_0^A \int_0^B \rho(x,y,z) \sin(\alpha_l x) \sin(\beta_m y) dy dx, \quad (13) $$

$$ \frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\varepsilon_0}, \quad (14) $$

$$ \phi(x,y,z) = \sum_{l=1}^{M_x} \sum_{m=1}^{M_y} \phi^{lm}(z) \sin(\alpha_l x) \sin(\beta_m y), \quad (15) $$

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 $0 \leq y \leq B$. Equation (15) 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

$$\rho^{\text{m}}(z) = \frac{1}{\pi R_{\text{wall}}^2} \int_0^{R_{\text{wall}}} J_1(\xi_{\text{lm}} R) 2\pi \rho(R, \theta, z) d\theta dR,$$  \hspace{1cm} (16)

$$\frac{\partial^2 \phi^{\text{lm}}(z)}{\partial z^2} - \xi_{\text{lm}}^2 \phi^{\text{lm}}(z) = -\frac{\rho^{\text{m}}(z)}{\epsilon_0},$$  \hspace{1cm} (17)

$$\phi(R, z) = \sum_{l=1}^{M_x} \sum_{m=-1}^{M_y} \phi^{\text{lm}}(z) J_l(\xi_{\text{lm}} R),$$  \hspace{1cm} (18)

where $\xi_{\text{lm}} = j_l^m/R_{\text{wall}}$ 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 boundary-value terms to the source:

$$\hat{F} = F - h_x^{-2} (G(h_x, 0, 0) + G(N_1 h_x, 0, 0)) - h_y^{-2} (G(0, h_y, 0) + G(0, N_2 h_y, 0)) - h_z^{-2} (G(0, 0, h_z) + G(0, 0, N_3 h_z)),$$  \hspace{1cm} (19)

where $N_1$, $N_2$, and $N_3$ are grid resolutions in $x$-, $y$- and $z$-directions, respectively [22]. 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:

1. **Grainy distribution function**: 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
2. **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. Still, as we now discuss, 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:

1. To investigate the origin of noise in general PIC simulations;

2. To devise and implement a wavelet-based noise-removing scheme in the context of beam dynamics simulations; and

3. 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 computational grid with resolution $N_i$, and grid spacing $h_i$, $i = 1, ..., D$, in each coordinate of the $D$-dimensional system, for the total of $N_{\text{grid}} = \prod_{i=1}^{D} N_i$ 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 $N_{\text{cell}} = \prod_{i=1}^{D} (N_i - 1)$ 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 $V = [-(h_{1/2}, h_{1/2}, h_{1/2}), ..., (h_{1/2}, h_{1/2}, h_{1/2}), ..., -(h_{D/2}, h_{D/2}, h_{D/2})]$. Each particle has the same charge $q_0 \equiv Q_{\text{tot}}/N$, where $Q_{\text{tot}} = \sum_{j=1}^{N_{\text{tot}}} q_j$ 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:

1. **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. 4):

$$d_{\text{NGP}}(x) = \begin{cases} q_0 & \text{if } |x| \leq h_i/2 \text{ for all } i \\ 0 & \text{otherwise} \end{cases}, \hspace{1cm} (20)$$

where $x_i$, $i = 1, 2, ..., D$ are coordinates; and

2. **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. 4):

$$d_{\text{CIC}}(x) = q_0 \prod_{i=1}^{D} \left(1 - \frac{|x|}{h_i}\right), \hspace{1cm} (21)$$

In the NGP DS, the probability of a particle being deposited in the $V$-neighborhood of the $j$th grid point is
FIG. 4: Deposition functions $d_{NGP}(x)$ (solid line) and $d_{CIC}(x)$ (dashed line) in 1D. $x$ denotes the location of each particle. 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.

given by the binomial distribution

$$P(n_j = n|p_j) = \frac{N!}{n!(N - n)!} p_j^n (1 - p_j)^{N-n}, \quad (22)$$

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 $\bar{n}_j$:

$$P(n_j = n|p_j) = \frac{\bar{n}_j^n e^{-\bar{n}_j}}{n!}, \quad (23)$$

where $n$ is an integer.

We define an algebraic average of variances of noise by

$$\sigma^2 \equiv \frac{1}{N_{grid}} \sum_{i=1}^{N_{grid}} \text{Var}(q_i). \quad (24)$$

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

$$\sigma^2_{NGP} \approx \frac{\bar{n}_j^2 N}{N_{grid}} = \frac{Q_{tot}^2}{NN_{grid}}, \quad (25)$$

for the NGP DS, and

$$\sigma^2_{CIC} \approx \frac{a^2 \bar{n}_j^2 N}{N_{grid}} = \frac{a^2Q_{tot}^2}{NN_{grid}}, \quad (26)$$

for the CIC DS, where

$$a = \left( \frac{2}{3} \right)^{D/2}. \quad (27)$$

The noise distribution for the CIC DS is therefore a contracted Poissonian, given by Eq. (23) 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 B). 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 very 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 fully apply to realistic simulations of beams.

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

1. superimposed Gaussians, modeling a cathode with
tributions contracted because, from Fig. 6, the relations given in Eqs. (25)-(26) are con-
only allowed variations, thus agreement between the two is excellent, within the statistically allowed variations.

d, 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 = [0.0, 0.5, -0.5], c_i = [0.4, 0.07, 0.05], x_i = [0.0, 0.0, 0.0]; 2. a smooth polynomial distribution:

\[
F_{1D}(x) = (x_2 - x)(x_2 - x_1) \\
F_{2D}(x, y) = (x_2 - x)(x_2 - x_1)f_y \\
F_{3D}(x, y, z) = (x_2 - x)(x_2 - x_1)f_yf_z \\
\]

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

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

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

where \( q = \{x, y, z\}, x \in [-2, 2], y \in [-2, 2], z \in [-0.7, 0.7], x_1 = -2, x_2 = 2, y_1 = 2, y_2 = -2, z_1 = -0.7, z_2 = 0.7; \) and

3. a constant distribution:

\[
F(x, y, z) = 1,
\]

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

Figure 5 shows numerically computed distributions of noise for the first two analytic distributions, versus the distribution predicted by Eq. (23), 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. (23), and for the CIC DS by Eq. (23) with \( n \to an \). 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. 6. The relations given in Eqs. (25)-(26) are confirmed because, from Fig. 6,

\[
\sigma_{\text{NGP}}N_{\text{grid}}/Q_{\text{tot}} \approx aN_{\text{ppg}}^{-1/2},
\]

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 properties span a volume \( 2^D \) times larger than the volume of the NGP DS.

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 [4, 5]

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

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

\[
SNR_{\text{NGP}} \approx \frac{r}{\sigma_{\text{NGP}}N_{\text{grid}}^{1/2}} = \frac{r}{Q_{\text{tot}}N_{\text{ppg}}^{1/2}} \quad (34)
\]

\[
SNR_{\text{CIC}} \approx \frac{r}{\sigma_{\text{CIC}}N_{\text{grid}}^{1/2}} = \frac{r}{aQ_{\text{tot}}N_{\text{ppg}}^{1/2}} \quad (35)
\]

where \( r \) is a constant dependent on the charge distribution. From Eqs. (34)-(35), we see that \( SNR \propto N_{\text{ppg}}^{1/2} \), which is a well-known result. Equations (34)-(35) also states that for the same particle distribution, the CIC DS will yield a less noisy result than NGP DS, quantified by \( a^{-1} \) times higher SNR \( (\approx 1.22, 1.5 \text{ and } 1.84, \text{ for } 1D, 2D, 3D, \text{ respectively}) \), 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 [5]:

\[
DF = \frac{SNR_{\text{denoised}}}{SNR_{\text{noisy}}} = \frac{\sigma_{\text{noisy}}}{\sigma_{\text{denoised}}}.
\]

Combining Eqs. (34)-(35) and (36), 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 non-denoised signal with \( (DF)^2N_{\text{ppg}} \) particles per grid point [5]. This is true regardless of the dimensionality of the simulation.
FIG. 6: 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 $\sigma_{\text{grid}}/Q_{\text{tot}}$ 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.

C. Adaptive Vs. Fixed Wavelet Basis

From the early 1990s, Coifman, Meyer, Wickerhauser and co-workers [23–25] advanced the approach of adaptively selecting the best basis out of a library of bases for maximum compression of a signal (minimizing appropriately chosen entropy); in particular, they developed libraries of rapidly constructible and rapidly searchable orthonormal bases.

This approach is ideally suited to our setting in that it allows for simultaneous compression and denoising in situations when using subjective judgment as to what constitutes noise (as opposed to physically significant fine-scale structure) can no longer be justified. Qualitatively, when a sufficiently broad library of bases is available, the part of a signal that is not well compressed (compactly represented) in any basis is identified as noise. These ideas were put on a rigorous footing by Donoho and co-workers, who demonstrated how, under sufficiently general assumptions, one can carry out ideal denoising by thresholding in an adaptively chosen basis, resulting in a signal estimator that minimizes an appropriately chosen entropy and is within a logarithmic factor of the ideal risk [26, 27].

Our interest was in establishing whether in simulations without instabilities a full-scale application of the above techniques would be necessary or if a (small) number of bases consistently outperforms others when single-basis simulation/compression/denoising are carried out. We additionally looked at performance in terms of overall...
D. 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 [4, 5]. However, we remain unconvinced of the generality of that work’s central claim that simulations of dynamical evolution of nontrivial systems where fine scale structure gives rise to instabilities can, by virtue of denoising in a fixed wavelet basis, “become equivalent to simulations with two orders of magnitude more particles” [4].

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. The thresholding can be either:

1. hard, where the coefficients with magnitudes below certain threshold $T > 0$ are set to zero:

$$\tilde{w}_i = \begin{cases} w_i & \text{if } |w_i| > T \\ 0 & \text{if } |w_i| \leq T \end{cases},$$

or

2. soft, where the coefficients with magnitudes below certain threshold $T > 0$ are set to zero and the ones above it contracted by $T$:

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

The threshold $T$ has to be chosen based on physical arguments so as to remove as much numerical noise from the particle distribution as possible, without removing physical small-scale structure.

The most widely used noise threshold in the literature [4, 26–28] is given in terms of the standard deviation $\sigma$ of the noise as

$$T = \sqrt{2\log N_{\text{grid}}} \sigma.$$  

This is a universal threshold for signals with Gaussian white noise, which means that no better threshold can be obtained for all signals in all wavelet bases. It leads to noise removal which is within a small factor of ideal denoising [26, 27]. A number of variations on this threshold are shown to perform better in removing different features from a known noisy signal (cf. [28] and references therein).

Studies of wavelet denoising usually involve distributions contaminated with additive (distribution-independent) Gaussian noise [26–28]. However, in the previous Section, we showed that the noise in PIC simulations is Poisson-distributed and 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 [29] (see also [5, 30–33]).

$$X_G = 2\sqrt{X_P + \frac{3}{8}},$$  

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

$$m_G = \sqrt{m_P + \frac{3}{8} - \frac{1}{8m_P^{1/2}} + \frac{1}{64m_P^{1/2}}}.$$  

with $m_P$ being the mean of the Poissonian signal. Applying the transformation in Eq. (40) produces a bias in the data [4, 5, 29], 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 (40) 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 [33]. 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 [33].

After the Anscombe transformation is applied to data contaminated with Poisson noise as in Eq. (23), the resulting data is approximately normally distributed, with variance $\sigma_{NGP} = 1$ for the NGP DS. For the CIC DS, the data distribution is a contracted Poissonian, given by Eq. (23) with $n_j \rightarrow an_j$, which means that the resulting variance will be appropriately contracted by a factor $a$, i.e., $\sigma_{CIC} = a$. Combining these with Eq. (39) yields optimal noise thresholds for the Anscombe-transformed (variance-stabilized) data:

$$T_{NGP} = \sqrt{2\log N_{\text{grid}}},$$  

$$T_{CIC} = \sqrt{2\log N_{\text{grid}}} a.$$  

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 $SNR(T)$ is maximized is empirically found to be $\beta(N_{grid}) = 2\sqrt{\log N_{grid}}$. The resulting optimal thresholds for Poissonian data without Anscombe transformation for the two particle-deposition schemes are then given by:

$$T_{NGP} = 2\sqrt{\log N_{grid}} \frac{Q_{tot}}{(NN_{grid})^{1/2}}, \quad (44)$$

$$T_{CIC} = 2\sqrt{\log N_{grid}} \frac{aQ_{tot}}{(NN_{grid})^{1/2}}. \quad (45)$$

Thresholds for non-transformed data in Eqs. (44)-(45) play a role analogous to that of thresholds in Eqs. (42)-(43) for Anscombe-transformed data (cf. Fig. 7).

In Fig. 7, we show the efficiency of wavelet denoising, quantified by the DF given in Eq. (36), as a function of the thresholding parameter $T$ for the superimposed Gaussians model, for both Anscombe-transformed (dashed lines) and non-transformed (solid lines) data. The wavelet coefficients of a noisy realization are sorted by magnitudes, hard-thresholded with $T$ in the interval $[|\omega_i|_{\text{min}}, |\omega_i|_{\text{max}}]$, wavelet-transformed back to physical space, and their resulting $SNR_{\text{noisy}}$ divided by the $SNR_{\text{denoised}}$ of the noisy realization to yield $DF(T)$. The figure shows results for the realizations with $N_{ppg} = 1$ (first column), $N_{ppg} = 5$ (second column) on a $N_x = 32$ grid and $N_{ppg} = 5$ (fourth column) on a $N_x = 64$ grid. Top row represents the NGP DS and bottom the CIC DS. The thresholds $T_{NGP}$ (dashed) and $T_{NGP}$ (solid) are shown as vertical lines in the top row, and $T_{CIC}$ (dashed) and $T_{CIC}$ (solid) as vertical lines in the bottom row.

We observe that the thresholds for both transformed and non-transformed data, given in Eqs. (42)-(45), 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 an 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. (34)-(35) states 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 designed 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 sufficiently different model particle distributions, we can induce the following: i) as the number of particles in the simulation $N$ increases, the particle distribution 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 $N_{ppg}$, the effectiveness of denoising increases with resolution $N_x$; 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 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.

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 [34] (Fig. 8). Both the density and potential are analytically known and are given by

$$F(r) = \frac{3}{(1 + r^2)^{5/2}}, \quad U(r) = -\frac{1}{\sqrt{1 + r^2}}. \quad (46)$$

where $r = \sqrt{x^2 + y^2 + z^2}$. The potential on the surface of the rectangular computational grid is specified analytically. The bottom panels of Fig. 8 demonstrate the substantial computational speedup gained by preconditioning. Here we applied open BCs, which is the natural choice for self-gravitating systems.

We then applied the alogorithm 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. 9). It is an axially symmetric “fuzzy
FIG. 7: Denoising factor (DF), defined in Eq. (36), as a function of the threshold $T$ for the superimposed Gaussians model with: $N_i = 32$ and $N_{ppg} = 1$ (first column); $N_i = 32$ and $N_{ppg} = 5$ (second column); $N_i = 64$ and $N_{ppg} = 1$ (third column); $N_i = 64$ and $N_{ppg} = 5$ (fourth column). Daubechies wavelets of order 2 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. (42)-(45): $T_{NGP}$ (dashed lines) and $T_{NGP}$ (solid lines) in the top row, and $T_{CIC}$ (dashed lines) and $T_{CIC}$ (solid lines) in the bottom row.

cigar”-shaped distribution of charged particles (mimicking a “beam bunch”) given by

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

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

$$d_2(z) = \begin{cases} 
1 & z_{12} \leq z \leq z_{21}, \\
\frac{(z-z_{21})^2[(z-(3z_{12}-2z_{21}))^2]}{4(z_{12}-z_{21})^2} & z_{12} \leq z \leq z_{21}, \\
\frac{(z-z_{21})^2[(z-(3z_{21}-2z_{12}))^2]}{4(z_{21}-z_{12})^2} & z_{21} \leq z \leq z_{22}, \\
0 & \text{otherwise},
\end{cases}$$

where the beam parameters $R_1$, $R_2$, $z_{12}$, $z_{21}$, $z_{22}$ are chosen so that $0 \leq R_1 < R_2$ and $z_{12} \leq z_{21} < z_{22}$. 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 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 Section IV C, we illustrate a practical way to reduce the number of iterations needed for convergence by using a different, near-optimal initial approximation.

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, 16] 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 $\epsilon = 0.00005$.

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 [40] 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 new code against the FFT-based IMPACT-T, so as to verify that the two
codes produce consistent 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 codes, 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) [35]. We expect that the non-uniformity 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,
Table I: Parameters required for the PCG Poisson equation solver within IMPACT-T.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Code</th>
<th>Recommended Range or Allowed Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_x$</td>
<td>Rx</td>
<td>number of Green expansion functions in $x$-coord.</td>
</tr>
<tr>
<td>$M_y$</td>
<td>Ry</td>
<td>number of Green expansion functions in $y$-coord.</td>
</tr>
<tr>
<td>$N_{\text{iter}}$</td>
<td></td>
<td>maximum number of iterations of the PCG</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>eps</td>
<td>PCG exit requirement tolerance</td>
</tr>
<tr>
<td>threshold $L$</td>
<td>thrL</td>
<td>threshold for the wavelet-transformed Laplacian</td>
</tr>
<tr>
<td>wavelet family</td>
<td>wtype</td>
<td>2, 3, 6, 10 (Daub); 54, 56, 58 (coif); 61, 62, 63 (sym)</td>
</tr>
<tr>
<td>threshold charge distribution</td>
<td>thr_charge</td>
<td>0 (no), 1 (yes)</td>
</tr>
<tr>
<td>denoising type</td>
<td>den_type</td>
<td>0 (with Anscombe transform), 1 (without)</td>
</tr>
<tr>
<td>thresholding type</td>
<td>thr_type</td>
<td>0 (hard), 1 (soft)</td>
</tr>
<tr>
<td>ratio of sizes of pipe and computational box in $x$-coord.</td>
<td>rat_x</td>
<td>4-6 (integer valued)</td>
</tr>
<tr>
<td>ratio of sizes of pipe and computational box in $y$-coord.</td>
<td>rat_y</td>
<td>4-6 (integer valued)</td>
</tr>
<tr>
<td>sparse matrix multiplication</td>
<td>impar</td>
<td>0 (disabled), 1 (enabled)</td>
</tr>
</tbody>
</table>

ii) phase-space detail, and iii) computational speed.

Figure 10 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 after Anscombe transform (blue line), and thresholded without Anscombe transform (red line). Figure 11 shows the same for the 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 12 compares experiment with numerical simulations using FFT-based IMPACT-T (red line) and IMPACT-T with PCG (blue line).

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 13-14 show, for the two distributions, integrated transverse cross-sections of the beam at different positions down the beamline. Detailed agreement between FFT-based IMPACT-T and IMPACT-T with PCG in the configuration space is clearly very good.

The computational speed of IMPACT-T with PCG is comparable to that of FFT-based IMPACT-T. Since fast wavelet transforms scale as $O(MN_{\text{grid}})$, transforms with wavelet families having larger-size support $M$ take longer to perform, yield denser operators, and consequently adversely affect the computational speed. Figure 15 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 30000-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. 16).

The computational speedup due to operator preconditioning and a good initial approximation is illustrated in Fig. 17. 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 17 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 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.

**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 [26, 27] demonstrated that for a given application there exist the “ideal” wavelet basis in which the “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. 18, we compare fractions of wavelet coefficients of the charge retained after thresholding in
FIG. 10: Distribution 1: Simulation results for the Fermilab/NICADD photoinjector performed with \( N_i = 32, N = 200000 \), 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 (\( M = 4 \)).

a typical realistic simulation on a \( N_i = 32 \) grid with \( N = 125000 \) particles for 10 different wavelet families.

Figure 19 shows the average sparsity of the particle distribution in a typical simulation of the Fermilab/NICADD photoinjector done with IMPACT-T with PCG on \( N_i = 32 \) and \( N_i = 64 \) grids (top two rows; left and right, respectively), the same \( N_{ppg} = 4.58 \) 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 non-transformed 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 32768 (about 5243 out of 262144) for thresholding after applying the Anscombe transform and about 6.1% (3.5%), i.e. about
1999 of 32768 (about 9175 out of 262144) for thresholding without the Anscombe transform on a $N_i = 32$ ($N_i = 64$) grid. When the number of particles per grid point $N_{ppg}$ 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. Section III).

Therefore, one might expect an even more compact representation for simulations with lower $N_{ppg}$. These results are generally valid for $N_{ppg} \geq 1$.

V. DISCUSSION AND CONCLUSION

We formulated and implemented a 3D wavelet-based solver for the Poisson equation with Dirichlet BCs, optimized for use in PIC simulations. The solver is based on the preconditioned conjugate gradient. We built on previous implementations of wavelet-based solvers for the Poisson equation with homogeneous ($U = 0$) Dirichlet BCs in 1D [11] and periodic BCs in 1D, 2D and 3D [12, 19], to design and implement a solver for the 3D Poisson equation with general inhomogeneous Dirichlet BCs. Our formulation of the discretized problem, which includes the treatment of the BCs and the Laplacian operator, differs significantly from the periodized 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 graining 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. 18, 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 non-zero after thresholding carried out according to prescription in [26]. 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. 15). The above comparison can be thought of as a (intentionally) simplified version of the full basis-library search approach of Coifman, Meyer, and Wickerhauser [24]: clearly, one would like to use the same basis throughout the simulation and avoid, if possible, a full library search at every timestep. 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.

Another example where compact representation enables study of a previously intractable issue is simulating beam halo with good statistics. The idea is to insert a large population of test particles at/near the cathode, i.e., at \( t = 0 \). These particles interact with the bulk potential but do not contribute to it. Integrating test-particle orbits in the compactly stored time-dependent potential is a fast operation; one can track several million such particles and thereby provide excellent statistics. What will be of most interest is the halo extent versus time (i.e., position along the beamline), and this is determined at each timestep from the test particle that lies furthest away from the accelerator’s centerline. Using a large population of test particles will lend confidence to the predicted halo extent. With such a code one can also explore the consequences of different initial conditions for the test particles inasmuch as these conditions can never be known precisely in practice.

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 clus-
FIG. 13: Distribution 1: Integrated transverse cross-section of the beam at different positions down the beamline for Fermilab/NICADD photoinjector simulations with $N = 200000$: FFT-based IMPACT-T (first row), IMPACT-T with PCG and no thresholding (second row), IMPACT-T with PCG and Anscombe transform and thresholding (third row), IMPACT-T with PCG and thresholding (fourth row). 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.

Acknowledgments

We are thankful to Daniel Mihalcea for generating Fig. 12, initial conditions and lattice configuration files for numerical simulations, and his many valuable suggestions. Benjamin Sprague has been instrumental in opti-
mizing 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 FA9471-040C-0199 to NIU and by Department of Energy contract DE-AC02-05CH11231 to LBNL and Department of Energy grant DE-FG02-04ER41323 to NIU.


FIG. 15: Comparison of relative execution times of IMPACT-T with PCG vs. standard FFT-based IMPACT-T for the full simulation for Distribution 1 with 30000 steps, \( N = 125000 \) 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.

FIG. 16: Average number of iterations of IMPACT-T with PCG for a 30000-step full simulation of the Fermilab/NICADD photoinjector with Distribution 1, and \( N = 125000 \) 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).

FIG. 17: Number of iterations of the PCG algorithm for the first 2000 steps of IMPACT-T with PCG with: 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 of Distribution 1 on the Fermilab/NICADD photoinjector, with \( N = 125000 \) 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 30000-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.


FIG. 18: 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 \( N = 125000 \) particles (\( N_{ppg} = 4.58 \)). Left column represents 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 \( 32^3 = 32768 \) coefficients, and for \( N_i = 64 \) resolution \( 64^3 = 262144 \) coefficients.

[39] 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.

[40] See http://www.nicadd.niu.edu/fnpl/
FIG. 19: 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 \( N_{\text{pps}} = 4.58 \): with the Anscombe transformation (top row), without the Anscombe transformation (middle row). Bottom row shows the number of non-zero wavelet coefficients in the Laplacian operator in wavelet space. Left column shows \( N_i = 32 \) (\( N = 125000 \) particles) and right \( N_i = 64 \) (\( N = 1000000 \) 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).

APPENDIX A: WAVELETS AND WAVELET TRANSFORMS

In this Appendix, we provide some essential background on wavelets and wavelet transforms. This material is available in other sources [4, 17, 36], but is offered here for the convenience of the reader.

Wavelets and wavelet transforms are a relatively new concept, introduced in the 1980’s [17, 25, 36, 37]. The discrete wavelet transform (DWT), like the discrete Fourier transform (DFT), is a fast, linear operation on data sets with size of integer power of two 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 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.

A continuous wavelet transform of a function \( f(x) \) is given by

\[
\mathcal{F}(a, b) = \int_{-\infty}^{\infty} f(x) \Psi_b^a(x) dx, \tag{A1}
\]

\[
\Psi_b^a(x) = \frac{1}{\sqrt{a}} \Psi \left( \frac{x - b}{a} \right), \tag{A2}
\]

where \( \Psi(x) \) is a mother wavelet, from which the wavelet basis functions are computed by scaling (by way of the parameter \( a \)) and/or translation (by way of the parameter \( b \)) [4, 17, 36]. DWT can be viewed as a discrete analog of the continuous wavelet transform defined for a discrete signal of length \( N_{\text{data}} = 2^K+1 \) (where \( K \) is an integer):

\[
f(x) = s_0^0 \phi_0^0(x) + \sum_{k=0}^{2^K-1} \sum_{l=0}^{K-1} d_{l}^k \psi_{l}^k(x), \tag{A3}
\]

\[
\phi_i^k(x) = 2^{k/2} \phi \left( 2^k x - i \right), \tag{A4}
\]

\[
\psi_i^k(x) = 2^{k/2} \psi \left( 2^k x - i \right), \tag{A5}
\]

where \( \psi(x) \) is the mother wavelet, and \( \phi(x) \) is the scaling function, with the scaling and translation parameters being \( k \) and \( i \), respectively. Extending wavelet transforms to higher dimensions is achieved by forming product functions [36]. In 2D, the space of scaling functions at a resolution \( k \) is given by

\[
\phi[s]_i^k(x, y) = \phi_i^k(x) \phi_i^k(y), \tag{A6}
\]

and the space of wavelets is comprised of

\[
\psi[s]_i^k(x, y) = \phi_i^k(x) \psi_i^k(y), \tag{A7}
\]

\[
\psi[d]_i^k(x, y) = \psi_i^k(x) \phi_i^k(y), \tag{A8}
\]

\[
\psi[d]_i^k(x, y) = \psi_i^k(x) \psi_i^k(y). \tag{A9}
\]

Similarly, the wavelet transforms in 3D [17] require one family of scaling functions of \([sss]\) type, and 7 different classes of wavelets: \([ssd]\), \([sds]\), \([sdd]\), \([dss]\), \([dds]\), \([ddd]\).

The coefficients \( s_{i}^{k-1} \) and \( d_{i}^{k-1} \) in the above expression can be computed recursively by a forward wavelet transform defined by

\[
s_{i}^{k-1} = \sum_{j=-m}^{m} h_{j} s_{2i+j}^{k}, \tag{A8}
\]

\[
d_{i}^{k-1} = \sum_{j=-m}^{m} g_{j} s_{2i+j}^{k}, \tag{A9}
\]
A backward wavelet transform is defined as

\[ s_{2i}^{k+1} = \sum_{j=-m/2}^{m/2} (h_{2j} s_{i-j}^k + g_{2j} d_{i-j}^k), \quad (A10) \]

\[ s_{2i+1}^{k+1} = \sum_{j=-m/2}^{m/2} (h_{2j+1} s_{i-j}^k + g_{2j+1} d_{i-j}^k); \quad (A11) \]

(recall \( m \) is even) [17].

An orthogonal wavelet family of degree \( m \) is defined by 2 filters \( h_j \) and \( g_j \) whose non-zero values are generally in the interval \( j = -m, ..., m \), where \( m \) is even. If this is not the case, it is acceptable to “pad” the filter with zeros until they fit this format. The low-pass filter \( h_j \) and the high-pass filter \( g_j \) satisfy [17]:

1. orthogonality relations

\[ \sum_l h_{l-2i} h_{l-2j} = \delta_{i,j}, \quad (A12) \]

\[ \sum_l g_{l-2i} g_{l-2j} = \delta_{i,j}, \quad (A13) \]

\[ \sum_l h_{l-2i} g_{l-2j} = 0, \quad (A14) \]

where \( \delta_{i,j} \) is a Kronecker delta function defined as \( \delta_{i,j} = 1 \) if \( i = j \), and \( \delta_{i,j} = 0 \) otherwise;

2. symmetry relation

\[ g_{i+1} = (-1)^{i+1} h_{-i}, \quad (A15) \]

3. refinement relations

\[ \phi(x) = \sqrt{2} \sum_{j=-m}^{m} h_j \phi(2x - j), \quad (A16) \]

\[ \psi(x) = \sqrt{2} \sum_{j=-m}^{m} g_j \phi(2x - j), \quad (A17) \]

and

\[ \phi_i^k(x) = \sum_{j=-m}^{m} h_j \phi_{2i+j}^{k+1}(x), \quad (A18) \]

\[ \psi_i^k(x) = \sum_{j=-m}^{m} g_j \phi_{2i+j}^{k+1}(x), \quad (A19) \]

Bi-orthogonal wavelets are given by four filters: \( h_j \) and \( g_j \) in primal space and their analogs in dual space \( \hat{h}_j \) and \( \hat{g}_j \). For an equivalent set of requirements for bi-orthogonal wavelets, see [17].

There exists a fast DWT algorithm for which the number of operations scales as \( O(mN_{\text{data}}) \) (see [17, 36]). The data set is first decomposed into two parts: approximation \( (s_i^k) \) and detail \( (d_i^k) \). The approximation is obtained by passing the data set through the low-pass filter \( h_j \) and rejecting every other data point (down-sampling). The detail is constructed by applying a high-pass filter \( g_j \) and down-sampling. This two-step process of applying low- and high-pass filters and down-sampling is then recursively applied to each approximation. The forward fast wavelet transform (wavelet decomposition), therefore, consists of data-filtering and down-sampling, while the inverse fast wavelet transform (wavelet synthesis) is composed of up-sampling (inserting a point between each two data points) and filtering [25, 36, 37].

Further borrowing nomenclature and illustrations from [17], we can illustrate the fast DWT via Table II. For filter coefficients of several important wavelet families, as well as more mathematically detailed treatment of multi-resolution analysis, see [17, 25, 36–38].

There exists an infinite number of wavelet families arising from various compromises between space and frequency localization. Each application imposes a different set of requirements on the properties of wavelets, which will in the end determine the choice of the wavelet family. (For a detailed account of wavelets, their derivation and implementation, see [17, 22, 25, 36]).

**APPENDIX B: 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 \( \{Q_1, ..., Q_N\} \):

\[ Q^{(i)} = \sum_{k=1}^{N} Q_k; \quad (B1) \]

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 assump-
for the NGP DS, and similarly

\[ E[Q^2|I = 1] = q_0 2^{-D}, \quad E[Q^2|I = 1] = q_0^2 2^{-D} \]  

(B2)

and

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

(B3)

for the NGP DS, and similarly

\[ E[Q|I = 1] = q_0 2^{-D}, \quad E[Q^2|I = 1] = q_0^2 3^{-D} \]  

(B4)

and

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

(B5)

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 between 0 and \( N \) (\( 0 \leq k \leq N \)). For a given \( k \),

\[ E[Q^{(i)}|K = k] = k E[Q|I = 1] \]  

(B6)

and

\[ \text{Var}(Q^{(i)}|K = k) = k \text{Var}(Q|I = 1) \]  

(B7)

since the \( \{Q_1, ..., Q_N\} \) 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:

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

(B8)

\[ \text{Var}(Q^{(i)}) = E[\text{Var}(Q^{(i)}|K)] + \text{Var}(E[Q^{(i)}|K]) \]  

(B9)

\[ = E[K] \text{Var}(Q|I = 1) + (E[Q|I = 1])^2 \text{Var}(K). \]

\( E[K] \) 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 \( \Omega^{(i)} \):

\[ E[K] = N P_i, \]  

(B10)

\[ \text{Var}(K) = N P_i (1 - P_i). \]  

(B11)

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,

\[ P_i \approx 2^D p_i, \]  

(B12)

and, combining Eqs. (B2), (B4), (B8), (B10) and (B12), we finally obtain

\[ E[Q^{(i)}] = q_0 N p_i \]  

(B13)

for both NGP DS and CIC DS, and, combining Eqs. (B2)-(B5), (B9)-(B12),

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

(B14)

for the NGP DS,

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

(B15)

for the CIC DS.

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

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

(B16)

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

\[ \frac{1}{N_{\text{grid}}} \leq \sum_{i=1}^{N_{\text{grid}}} p_i^2 \leq 1, \]  

(B17)

(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. (B17) is $O(1/N_{grid})$, and can be neglected in computing $\sigma^2$ using Eq. (B16). With this simplification, for the two particle-deposition schemes considered here we obtain

$$\sigma^2_{NGP} \approx \frac{q_0^2 N}{N_{grid}} = \frac{Q_{tot}^2}{NN_{grid}}$$  \hspace{1cm} (B18)$$

and

$$\sigma^2_{CIC} \approx \left(\frac{2}{3}\right)^D \frac{q_0^2 N}{N_{grid}} = \left(\frac{2}{3}\right)^D \frac{Q_{tot}^2}{NN_{grid}}.$$  \hspace{1cm} (B19)

**APPENDIX C: RELATING SNR TO THE STANDARD DEVIATION OF NOISE $\sigma$**

Combining Eq. (24) with Eq. (33), we obtain

$$SNR = \left[\sum_{i=1}^{N_{grid}} q_i^2\right]^{1/2}/\sigma N_{grid} = R(N_{grid})/\sigma N_{grid}.$$  \hspace{1cm} (C1)

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

$$Q_{tot} \equiv \sum_{i=1}^{N_{grid}} q_i = \sum_{i=1}^{N_{grid}} q_i = const.$$  \hspace{1cm} (C2)

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 $N_{grid} = 2^{Dk}$, 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,

$$Q_{tot} \equiv \sum_{i=1}^{2^{Dk}} \tilde{q}_{k,i} \approx 2^D \sum_{i=1}^{2^{D(k-1)}} \frac{1}{2^D} \tilde{q}_{k-1,i}$$  \hspace{1cm} (C3)$$

then

$$R^2(2^{Dk}) \equiv \sum_{i=1}^{2^{Dk}} \tilde{q}_{k,i}^2 \approx 2^D \sum_{i=1}^{2^{D(k-1)}} \left(\frac{1}{2^D} \tilde{q}_{k-1,i}\right)^2$$  \hspace{1cm} (C4)$$

$$\approx 2^{-D} \sum_{i=1}^{2^{D(k-1)}} \tilde{q}_{k-1,i}^2 = 2^{-D}R^2(2^{D(k-1)})$$

$$\approx 2^{-kD}R^2(2^0) = r^2 N_{grid}^{-1},$$

which means that

$$R(N_{grid}) \approx rN_{grid}^{-1/2},$$  \hspace{1cm} (C5)$$

where $r$ is essentially a constant which depends on the distribution $q$. Combining the relationship given in Eq. (C5) with Eq. (C1) and Eqs. (25)-(26) establishes the inverse relationship between the $SNR$ and the standard deviation of noise for both particle-deposition schemes:

$$SNR_{NGP} \approx \frac{r}{\sigma_{NGP} N_{grid}^{1/2}} = \frac{rN_{grid}^{1/2}}{Q_{tot} N_{grid}^{1/2}} = \frac{r}{Q_{tot}} N_{ppg}^{1/2},$$  \hspace{1cm} (C6)$$

$$SNR_{CIC} \approx \frac{r}{\sigma_{CIC} N_{grid}^{1/2}} = \frac{rN_{grid}^{1/2}}{aQ_{tot} N_{grid}^{1/2}} = \frac{r}{aQ_{tot}} N_{ppg}^{1/2}.\hspace{1cm} (C7)$$

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