Orthogonal Basis Function Approximation of Particle Distributions In Numerical Simulations of Beams

Balša Terzić
Beam Physics and Astrophysics Group

Northern Illinois University

Jefferson Lab Beam Seminar
April 3, 2008

Motivation

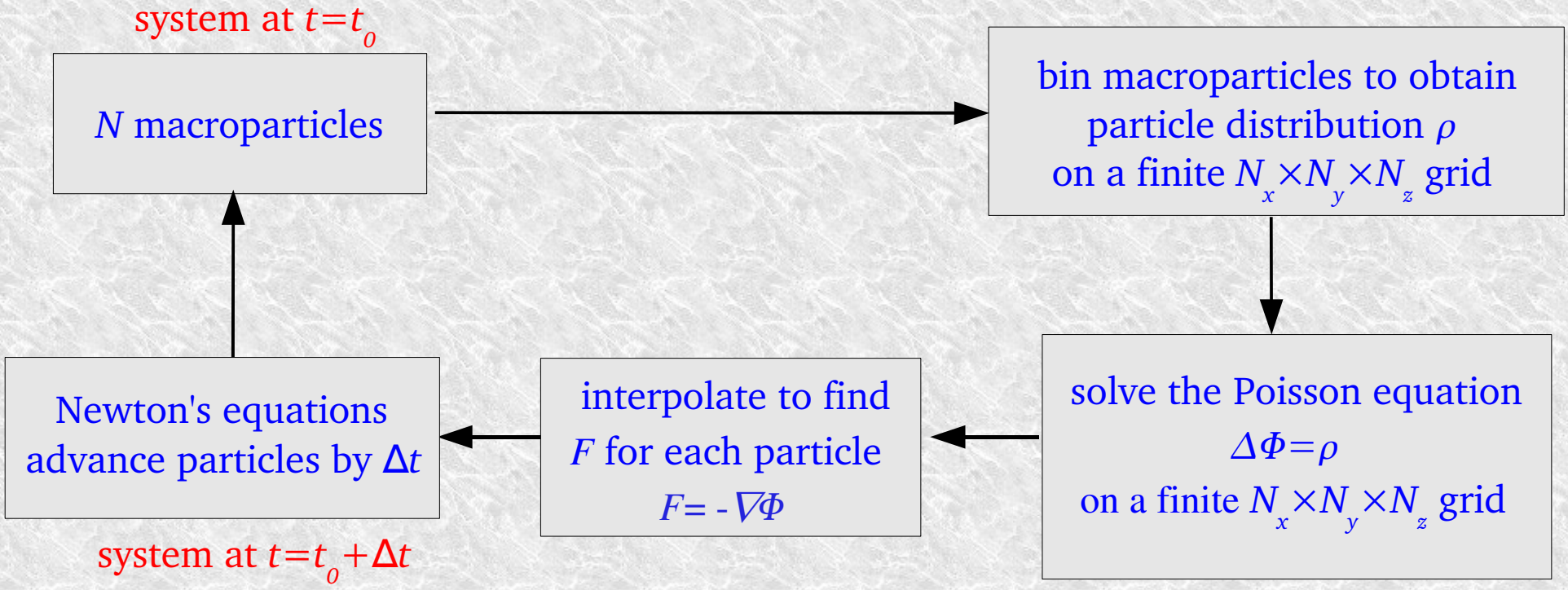
- Studying dynamics of multi-particle systems (charged particle beams, plasma, galaxies...) heavily relies on *N*-body simulations
- It is important for *N*-body codes to:
  - be as *efficient* as possible, without compromising accuracy
  - *minimize numerical noise* due to $N_{\text{simulation}} << N_{\text{physical}}$
  - account for *multiscale dynamics*
  - for some applications: have a compact representation of history
- We present two orthonormal bases which, as a part of an *N*-body code, address these requirements
  - wavelet basis
  - scaled Gauss-Hermite basis

Outline of the Talk

- Algorithms for *N*-body simulations
- Wavelet basis
  - brief overview of wavelets
  - wavelet-based Poisson equation solver
  - advantages
  - applications
- Scaled Gauss-Hermite basis
  - mathematical formalism
  - Poisson equation solver
  - applications
- Discussion of further work

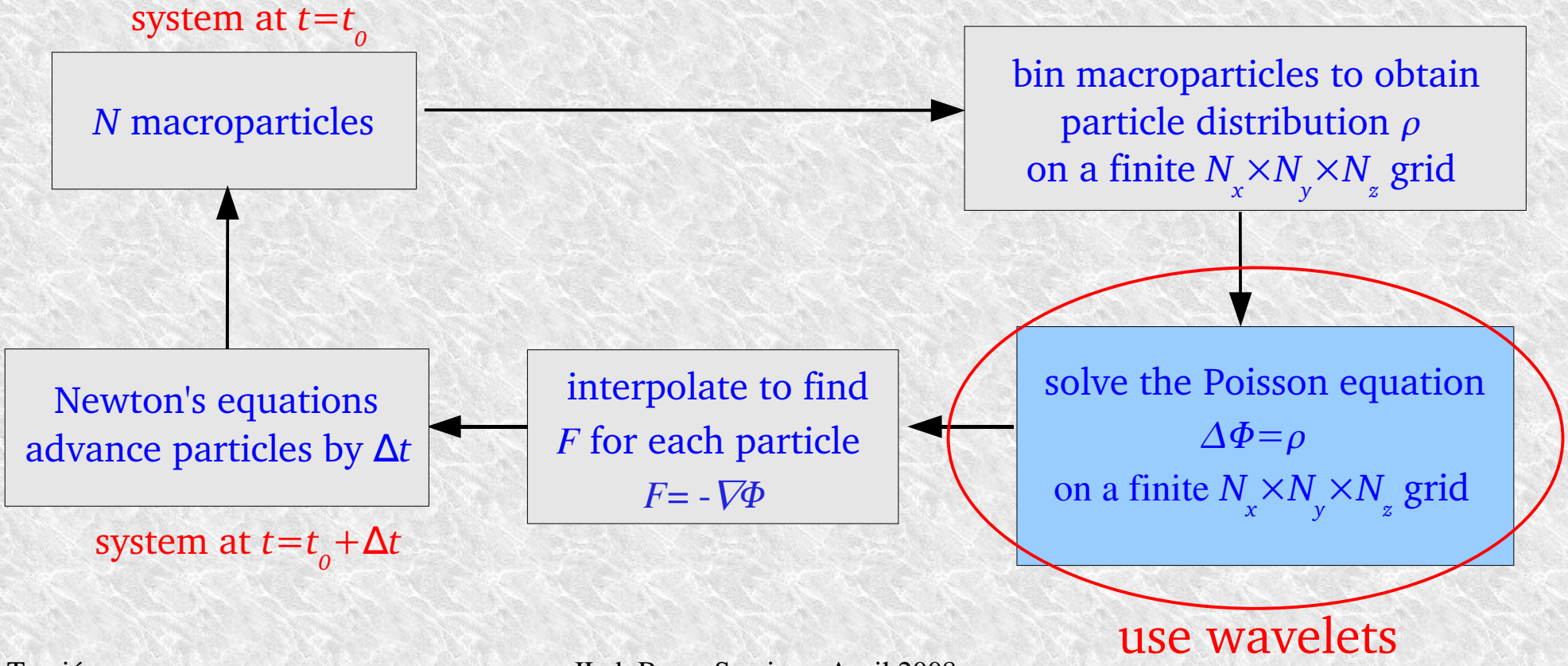
Algorithms for *N*-body Simulations

- Direct summation: CPU cost scales as N^2
- Tree: direct summation nearby and statistical treatment farther away
- Particle-In-Cell (PIC): particles binned in cells (grid)



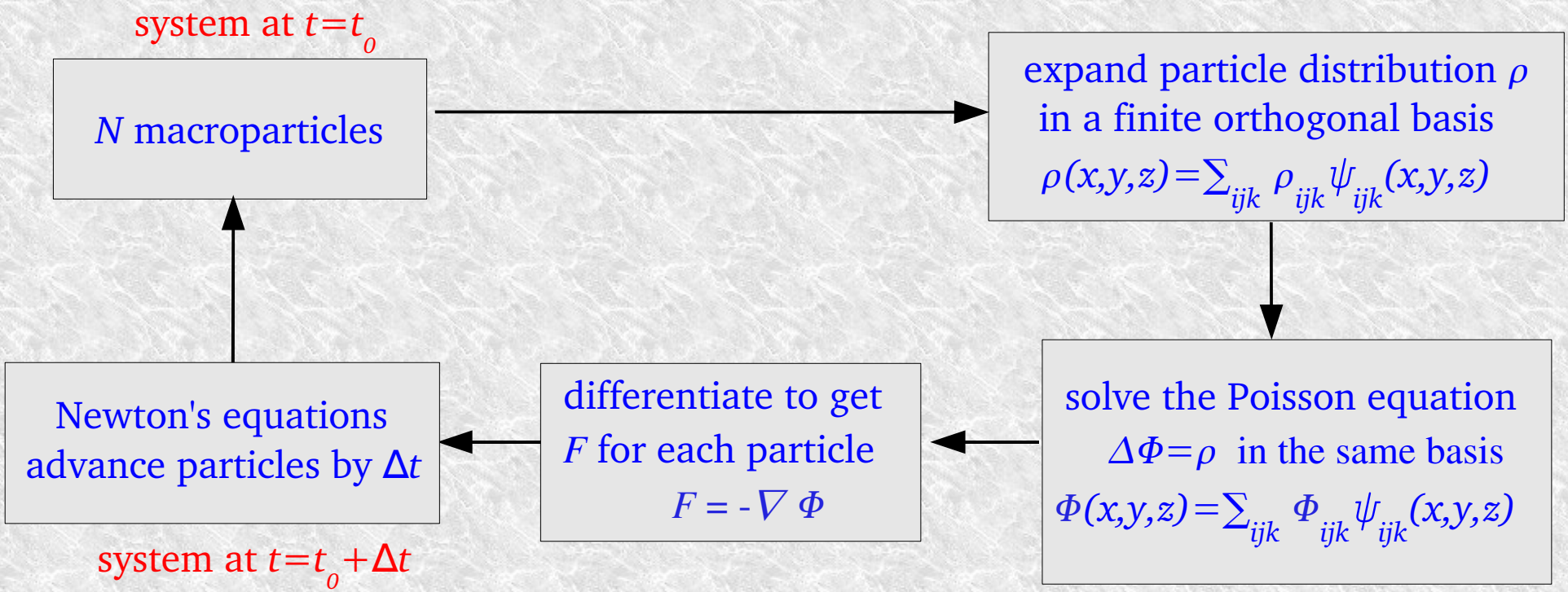
Algorithms for *N*-body Simulations

- Direct summation: CPU cost scales as N^2
- Tree: direct summation nearby and statistical treatment farther away
- Particle-In-Cell (PIC): particles binned in cells (grid)



Algorithms for *N*-body Simulations

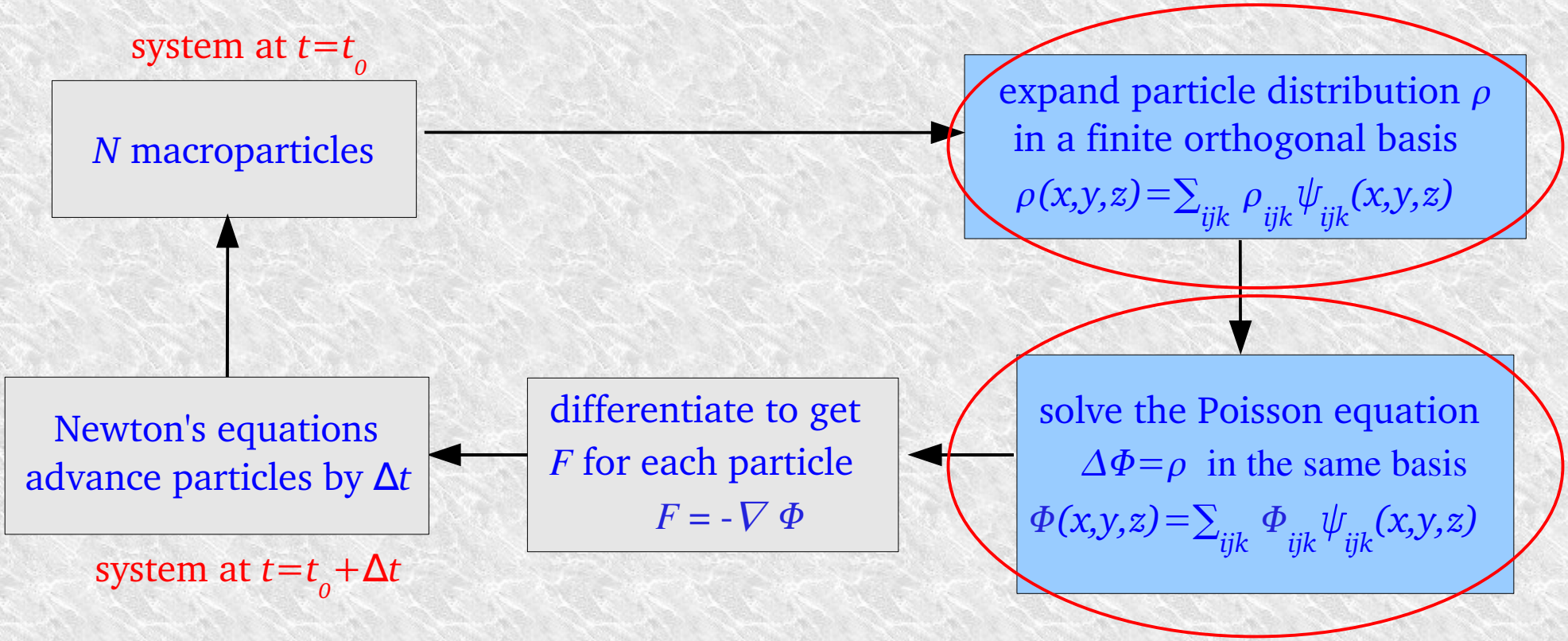
- Alternative *N*-body algorithm: analytical function approximation
  - analytical functions form a finite orthogonal basis
  - *N* macroparticles, but no grid



Algorithms for *N*-body Simulations

- Alternative *N*-body algorithm: analytical function approximation
  - analytical functions form a finite orthogonal basis
  - *N* macroparticles, but no grid

use scaled Gauss-Hermite basis

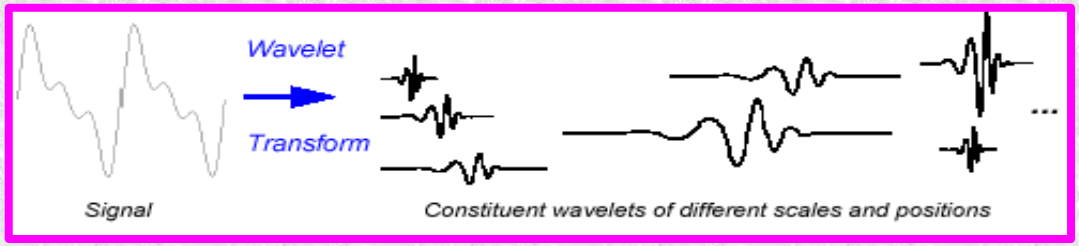


Wavelets

- **Wavelets:** orthogonal basis composed of scaled and translated versions of the same localized *mother wavelet* $\psi(x)$ and the *scaling function* $\phi(x)$:

$$\psi_i^k(x) = 2^{k/2}\psi(2^k x - i)$$

$$f(x) = s_0^0 \phi_0^0(x) + \Sigma_i \Sigma_k d_i^k \psi_i^k(x)$$



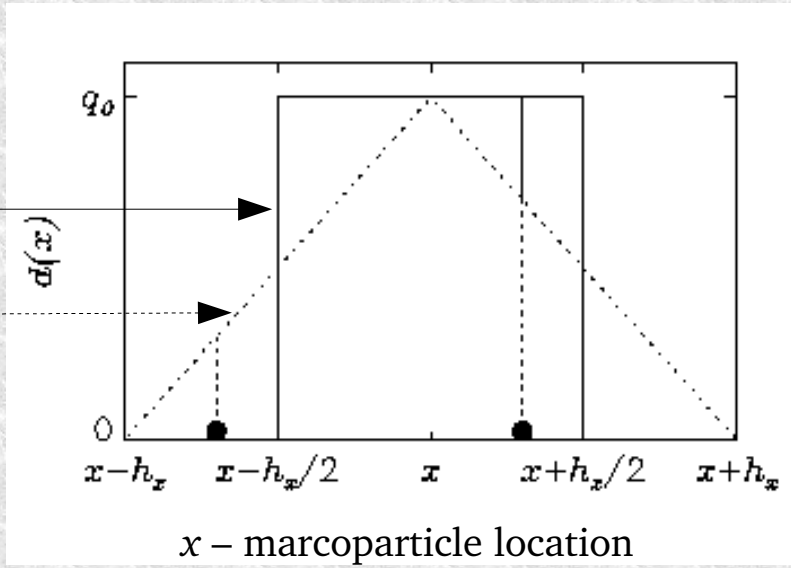
- Discrete Wavelet Transfrom (DFT) iteratively separates scales
  - $\sim O(MN)$ operation, M size of the wavelet filter, N size of the signal
- *Advantages:*
  - simultaneous localization in both space and frequency
  - compact representation of data, enabling *compression* (FBI fingerprints)
  - *signal denoising:* natural setting in which noise can be partially removed
    denoised simulation $\leftrightarrow$ simulation with more macroparticles

Numerical Noise in PIC Simulations

- Any *N*-body simulation will have numerical noise
- Sources of numerical noise in PIC simulations:
  - *graininess of the distribution function*: $N_{\text{simulation}} << N_{\text{physical}}$
  - *discreteness of the computational domain*: ρ and Φ specified on a finite grid
- Each macroparticle is deposited onto a finite grid by either:
  - Nearest Grid Point (NGP) dep. scheme
  - Cloud-In-Cell (CIC) dep. scheme

NGP

CIC



x – marcoparticle location

• – gridpoint location

Numerical Noise in PIC Simulations

- For NGP, at each gridpoint, particle dist. is Poissonian:

  $P=(n!)^{-1} n_j^n e^{-n_j}$ $\quad$ n_j is the expected number in j^{th} cell; $\quad$ n integer

- For CIC, at each gridpoint, particle dist. is *contracted* Poissonian:

  $P=(n!)^{-1}(an_j)^n e^{-an_j}$ $\quad$ $a=(2/3)^{(D/2)}\sim 0.54(3\mathrm{D}), 0.67(2\mathrm{D}), 0.82(1\mathrm{D})$

- Measure of error (noise) in depositing macroparticles onto a grid:

  $$\sigma^2=(N_{grid})^{-1}\sum_{i=1}^{N_{grid}} Var(q_i) \qquad \sigma^2_{NGP}=\frac{Q^2_{total}}{N N_{grid}} \qquad \sigma^2_{CIC}=\frac{a^2 Q^2_{total}}{N N_{grid}}$$

  where $q_i = (Q_{total}/N)n_i$, Q_{total} total charge; N_{grid} number of gridpoints

  (For more details see Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201)

- This error/noise estimate is crucial for optimal wavelet-denoising

- IDEA: Solve the Poisson equation in such a way so as to minimize numerical noise – *USE WAVELETS*

Numerical Noise in PIC Simulations

- In wavelet space:

  signal $\rightarrow$ few large wavelet coefficients c_{ij}

  noise $\rightarrow$ many small wavelet coefficients c_{ij}

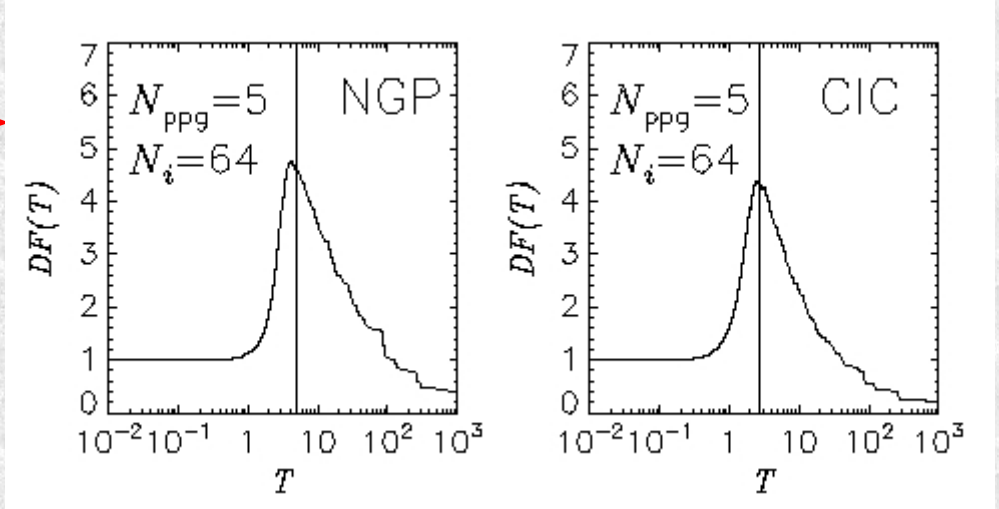
- Denoising by wavelet thresholding:

  if $|c_{ij}| < T$, set to $c_{ij} = 0$ (choose threshold T carefully!)

- A great deal of study has been devoted to estimating optimal T

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

(σ was estimated earlier)



Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201

Wavelet Denoising and Compression

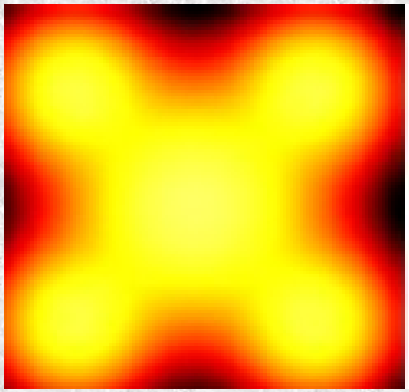
- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

2D superimposed Gaussians on 256×256 grid

ANALYTICAL

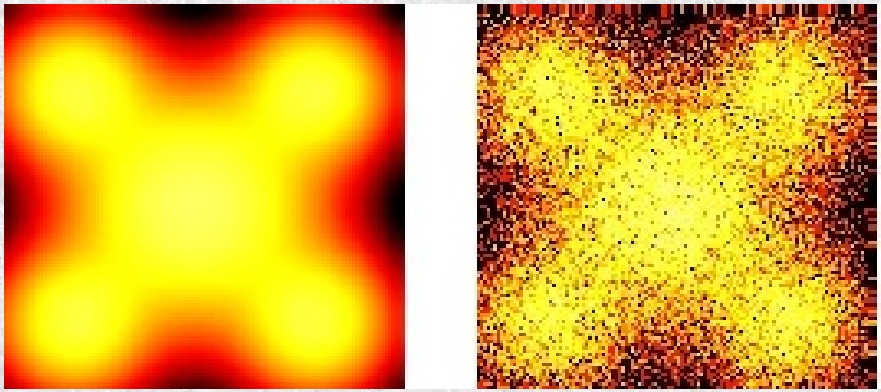


Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

2D superimposed Gaussians on 256×256 grid

ANALYTICAL $N_{ppc}=3$ $SNR=2.02$

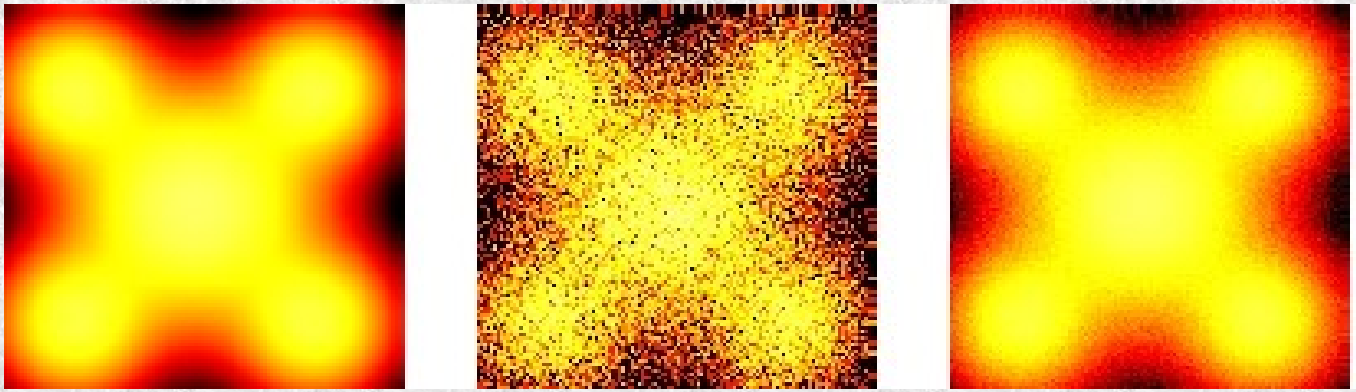


Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

2D superimposed Gaussians on 256×256 grid

ANALYTICAL $N_{ppc}=3$ $SNR=2.02$ $N_{ppc}=205$ $SNR=16.89$

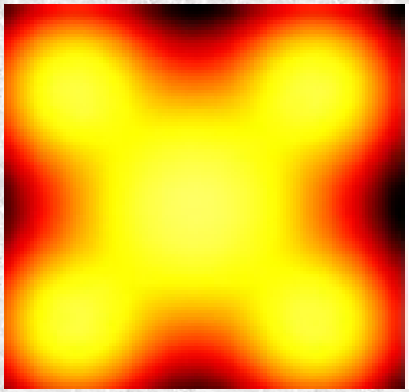


Wavelet Denoising and Compression

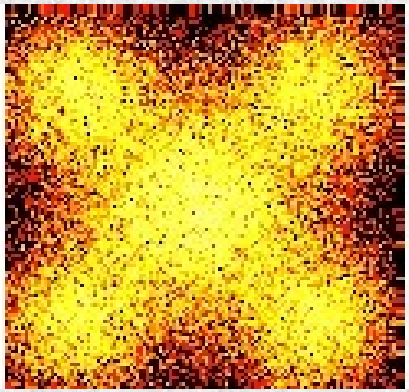
- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

2D superimposed Gaussians on 256×256 grid

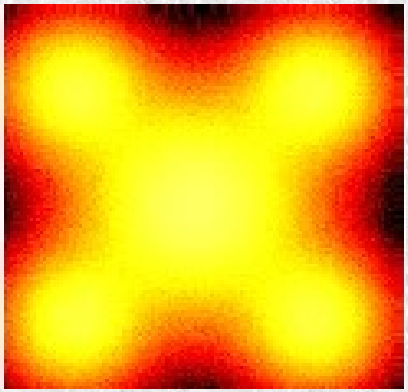
ANALYTICAL



N_{ppc}=3 *SNR*=2.02



N_{ppc}=205 *SNR*=16.89



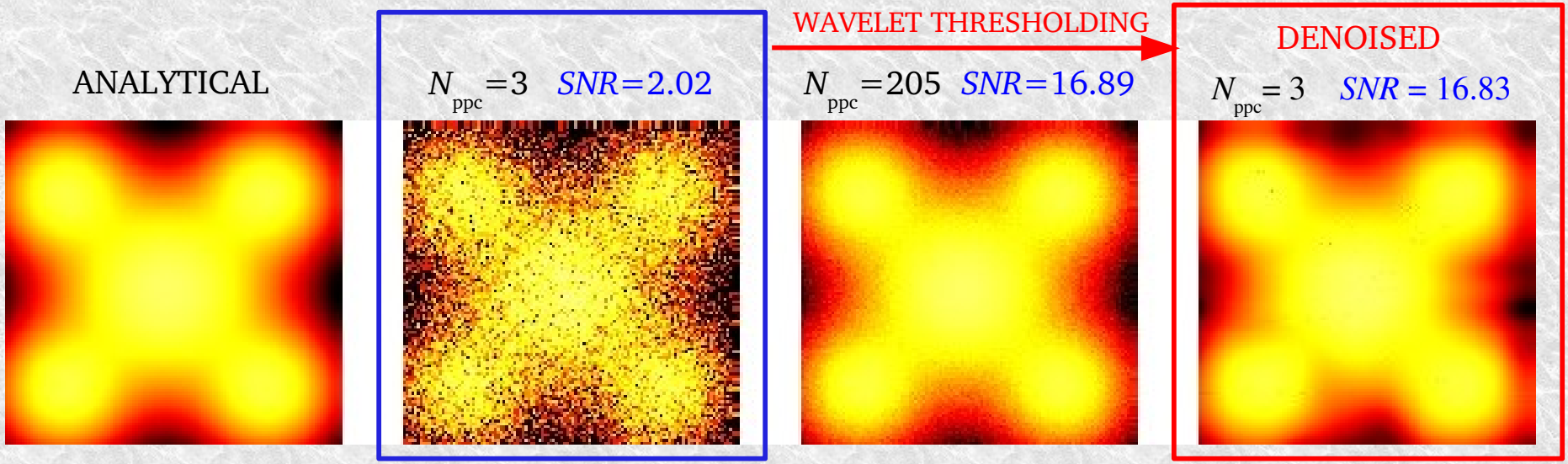
- denoising by wavelet thresholding: if $|c_{ij}| < T$, set to 0

Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the *Signal-to-Noise Ratio* (*SNR*) which measures its quality
- $SNR \sim \sqrt{N_{ppc}}$ N_{ppc}: avg. # of particles per cell $N_{ppc} = N/N_{cells}$

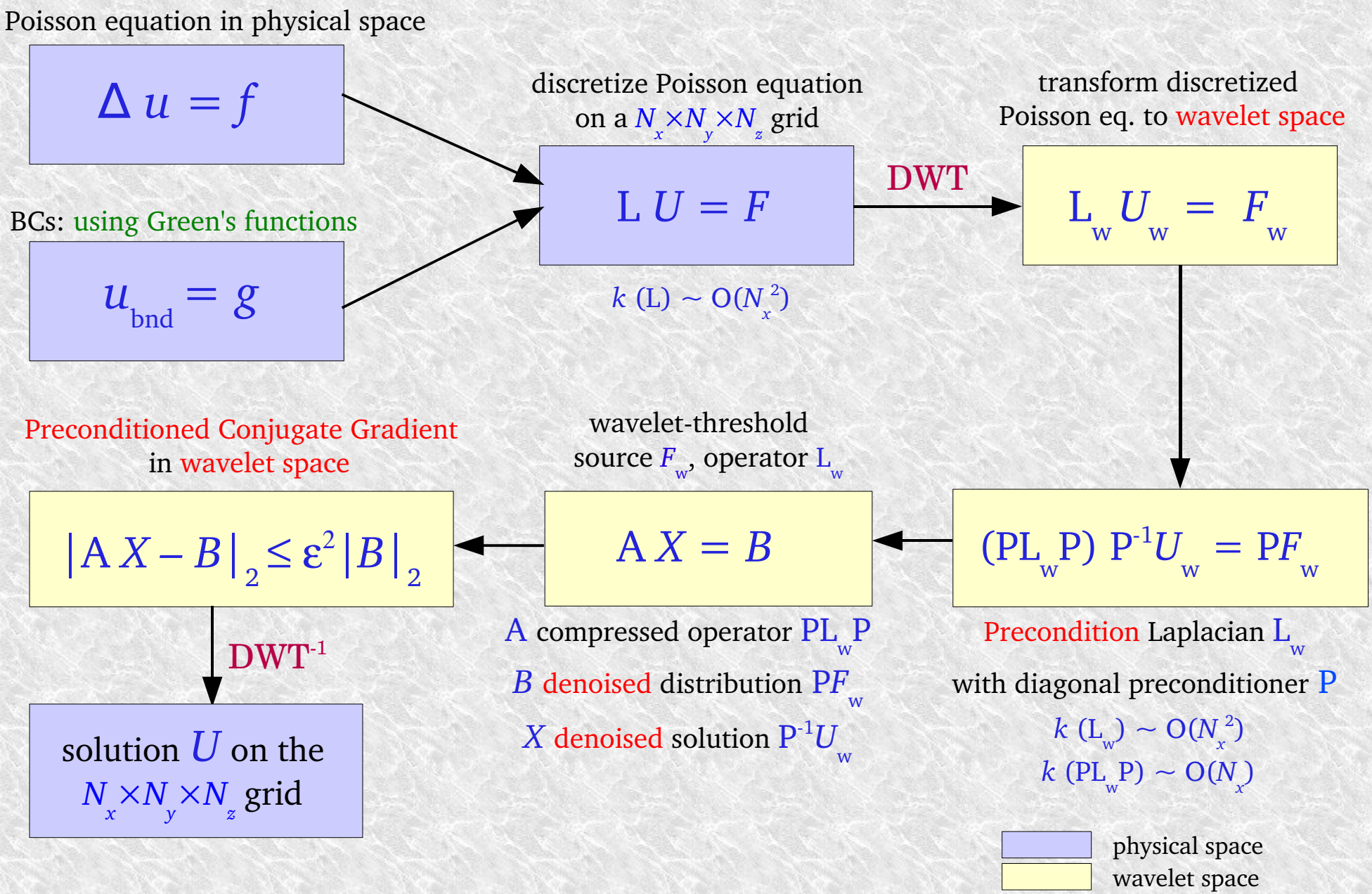
2D superimposed Gaussians on 256×256 grid

COMPACT: only 0.12% of coeffs



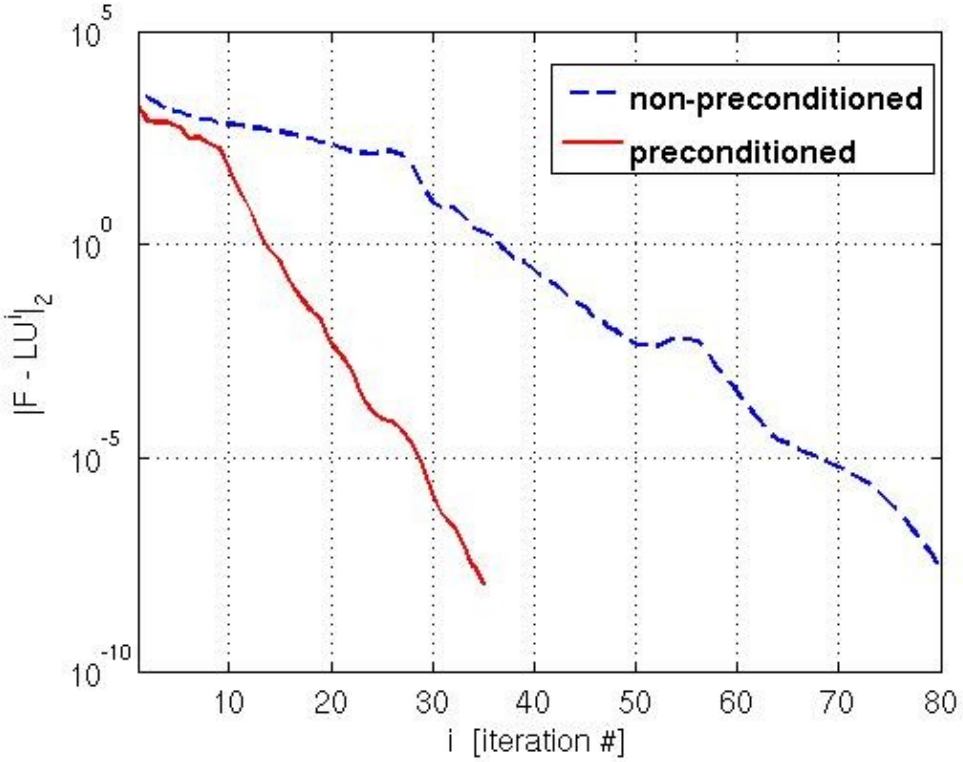
- denoising by wavelet thresholding: if $|c_{ij}| < T$, set to 0
- *Advantages:* - increase in *SNR* by $c \leftrightarrow c^2$ more macroparticles (here c=8.3, c^2=69)
  - compact storage in wavelet space (in this example 79/65536: 0.12%)

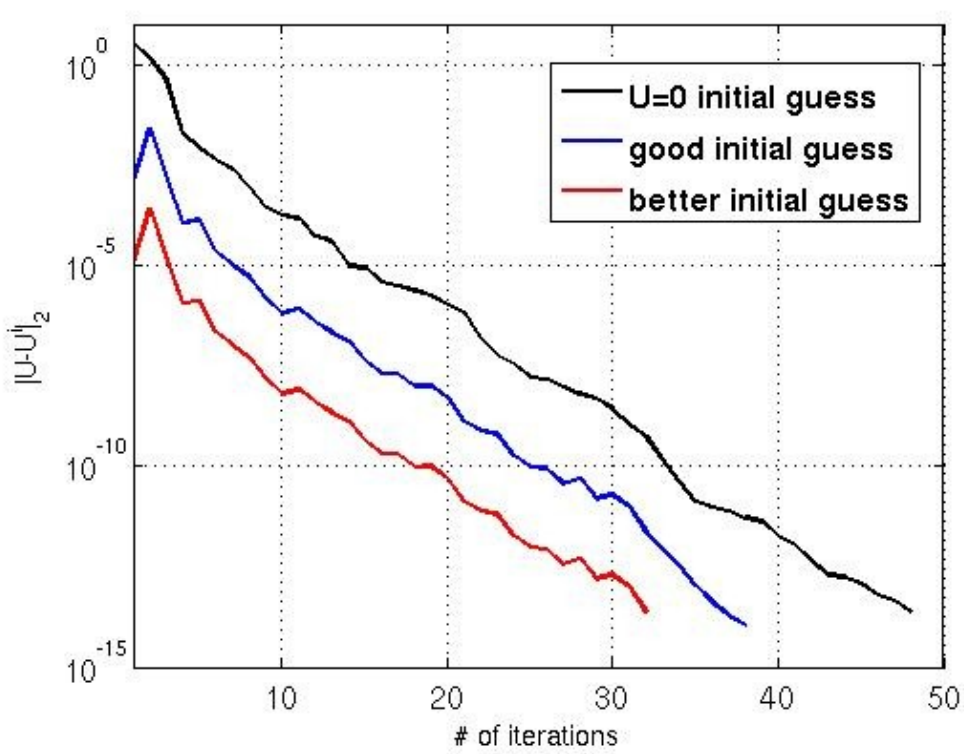
Wavelet-Based Poisson Equation Solver



Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number k $\quad |u-u^i|_2 \le \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |u|_2$
- preconditioning (diagonal in wavelet space): $\quad k \sim O(N_x^2) \;\rightarrow\; k \sim O(N_x)$
- good initial approximation: solution at previous time step

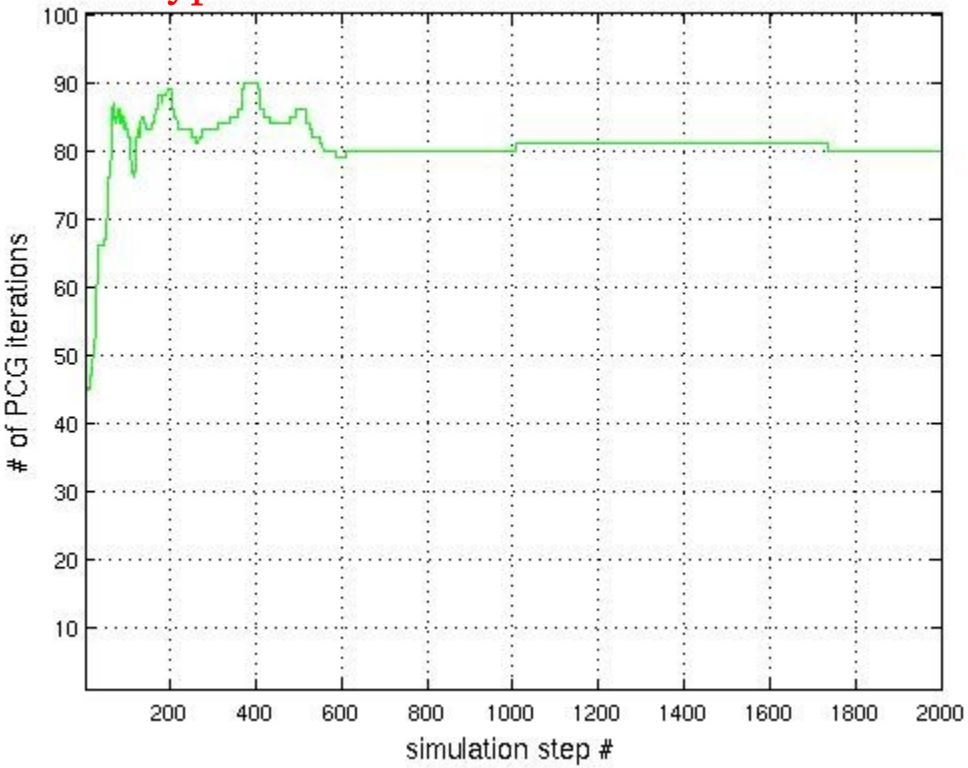




Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number k $\quad |u-u^i|_2 \le \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |u|_2$
- preconditioning (diagonal in wavelet space): $\quad k \sim O(N_x^2) \;\rightarrow\; k \sim O(N_x)$
- good initial approximation: solution at previous time step

typical realistic beam simulation

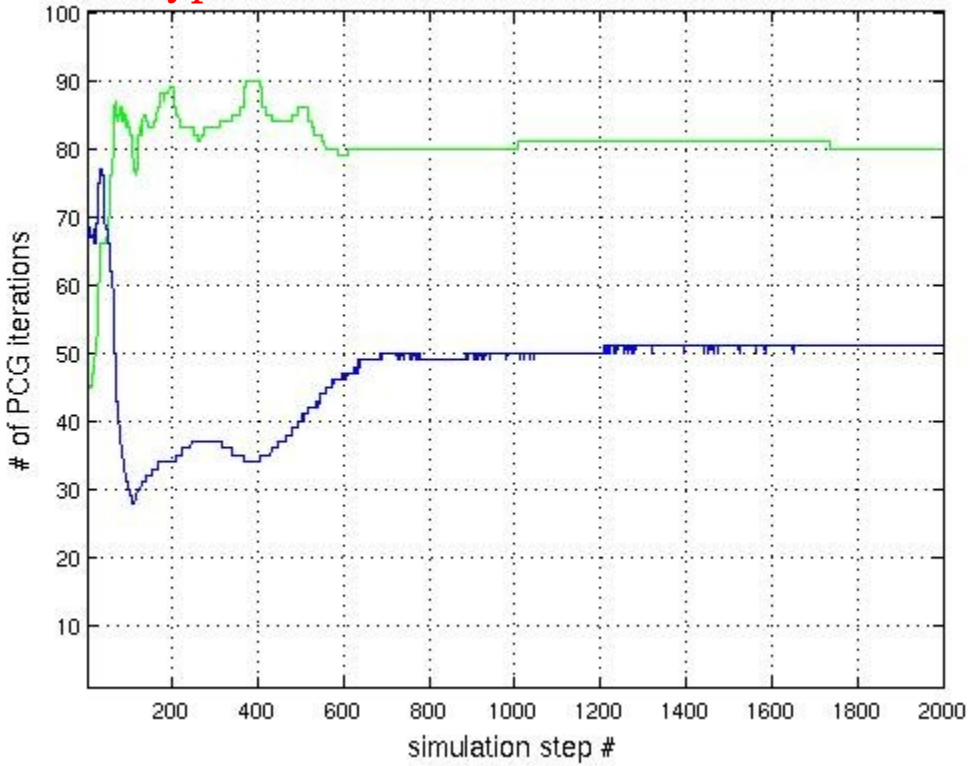


	average over 30000-step run
no preconditioning U^i=0 initial guess	75.2

Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number k $\quad |u-u^i|_2 \le \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |u|_2$
- preconditioning (diagonal in wavelet space): $\quad k \sim O(N_x^2) \rightarrow k \sim O(N_x)$
- good initial approximation: solution at previous time step

typical realistic beam simulation

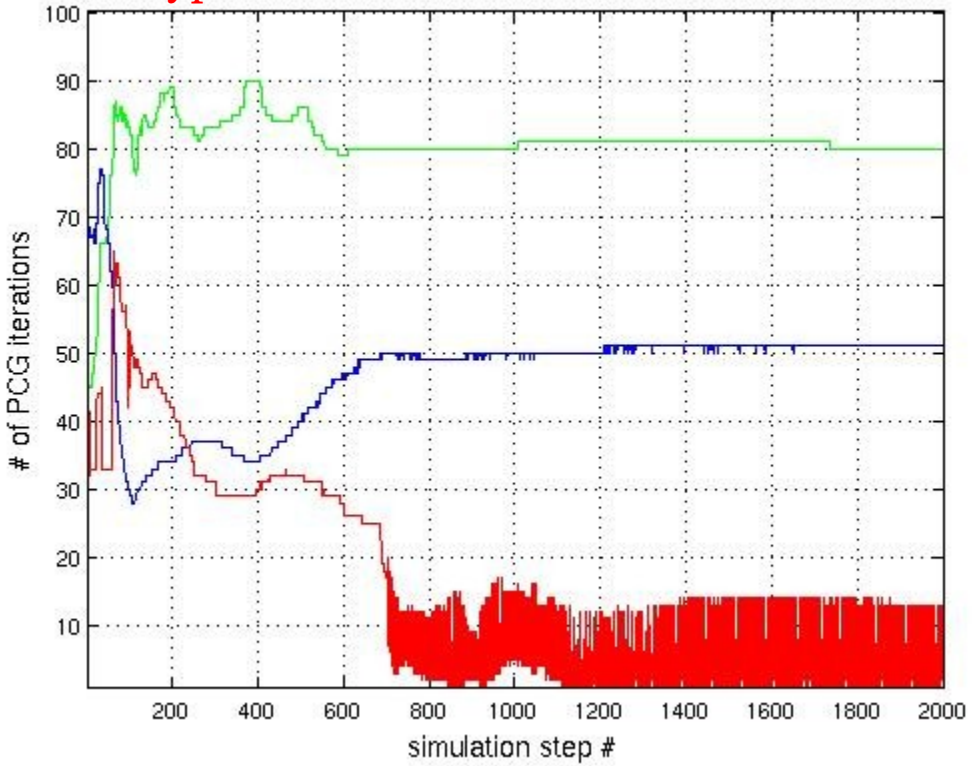


	average over 30000-step run
no preconditioning U^i=0 initial guess	75.2
preconditioned U^i=0 initial guess	60.7

Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number k $\quad |u-u^i|_2 \le \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |u|_2$
- preconditioning (diagonal in wavelet space): $\quad k \sim O(N_x^2) \rightarrow k \sim O(N_x)$
- good initial approximation: solution at previous time step

typical realistic beam simulation

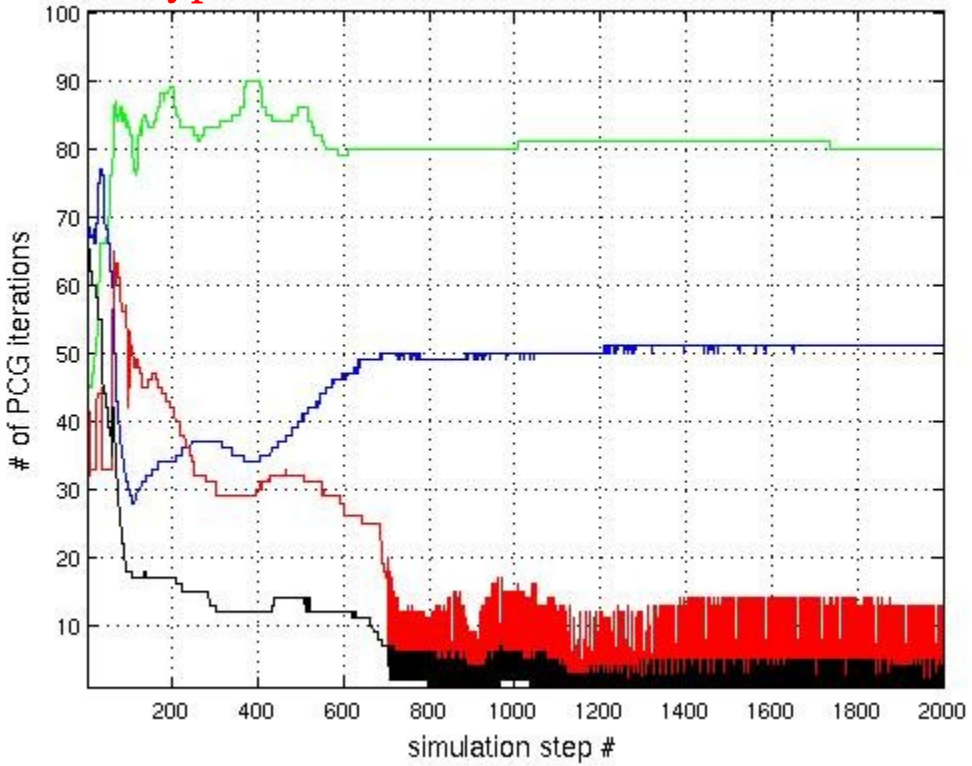


	average over 30000-step run
no preconditioning U^i=0 initial guess	75.2
preconditioned U^i=0 initial guess	60.7
no preconditioning $U^i=U^{i-1}$ initial guess	4.8

Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number k $\quad |u-u^i|_2 \le \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i |u|_2$
- preconditioning (diagonal in wavelet space): $\quad k \sim O(N_x^2) \rightarrow k \sim O(N_x)$
- good initial approximation: solution at previous time step

typical realistic beam simulation



	average over 30000-step run
no preconditioning $U^i=0$ initial guess	75.2
preconditioned $U^i=0$ initial guess	60.7
no preconditioning $U^i=U^{i-1}$ initial guess	4.8
preconditioned $U^i=U^{i-1}$ initial guess	2.4

considerable computational speedup

(Terzić, Pogorelov & Bohn 2007)

Using PCG in Numerical Simulations

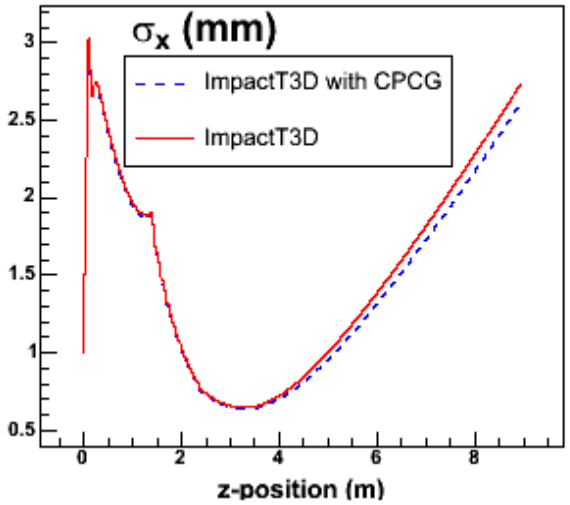
- **Our goal:** develop wavelet-based Poisson solver which can easily be integrated into existing PIC codes
- **First:** test the PCG as a stand-alone solver on examples from:
  - *beam dynamics*
  - *galactic dynamics*
- **Second:** insert the PCG Poisson solver into an existing PIC code (IMPACT-T) and run realistic charged particle beam simulations
  (Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201)
  - compare (conventional FFT-based) IMPACT-T Vs. IMPACT-T with PCG:
    - *rms* properties
    - level of detail
    - computational speed

Conventional IMPACT-T vs. IMPACT-T with PCG
Code Comparison: *rms* Properties

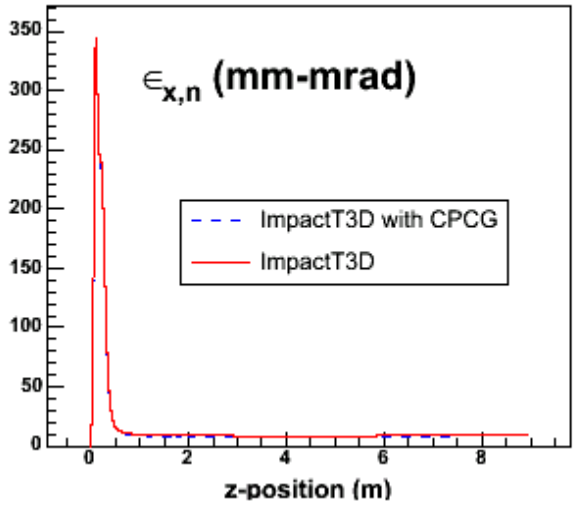
Fermilab/NICADD photoinjector **32×32×32 grid** **1 nC charge**

Good agreement to a few percent

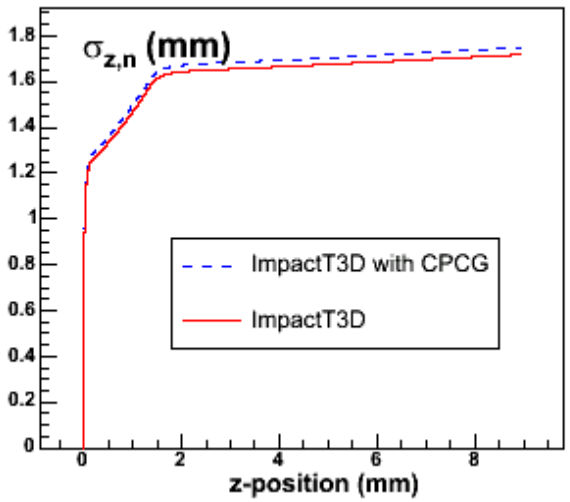
rms beam radius



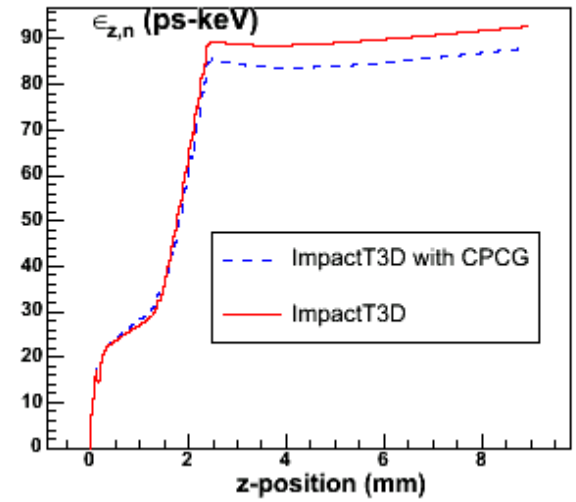
rms normalized transverse emittance



rms bunch length

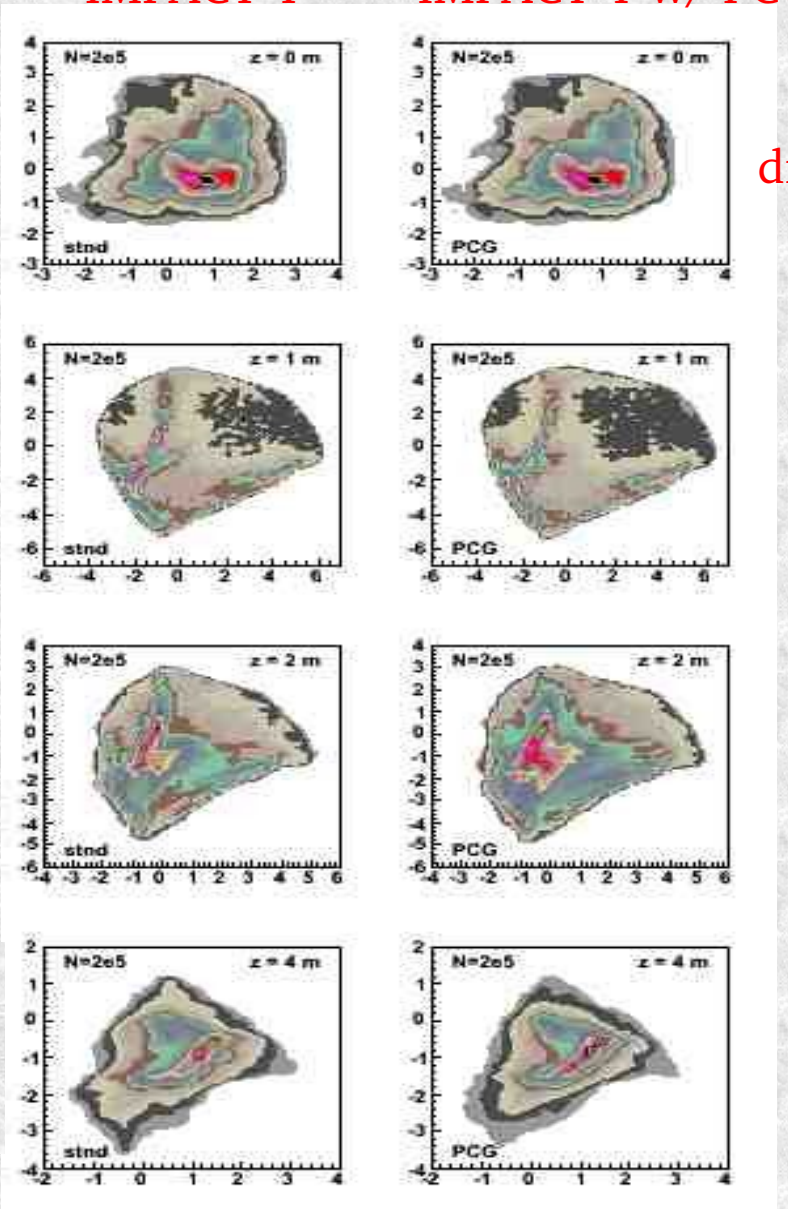


rms normalized longitudinal emittance



Conventional IMPACT-T vs. IMPACT-T with PCG
Code Comparison: Level of Detail & Speed

- transverse charge distribution for the Fermilab/NICADD photoinjector simulation
- very non-axisymmetric beam
- 32×32×32 grid, N=200000
- very good agreement in detail
- Speed comparison: IMPACT-T w/ PCG ~ 10% faster than the conventional serial IMPACT-T

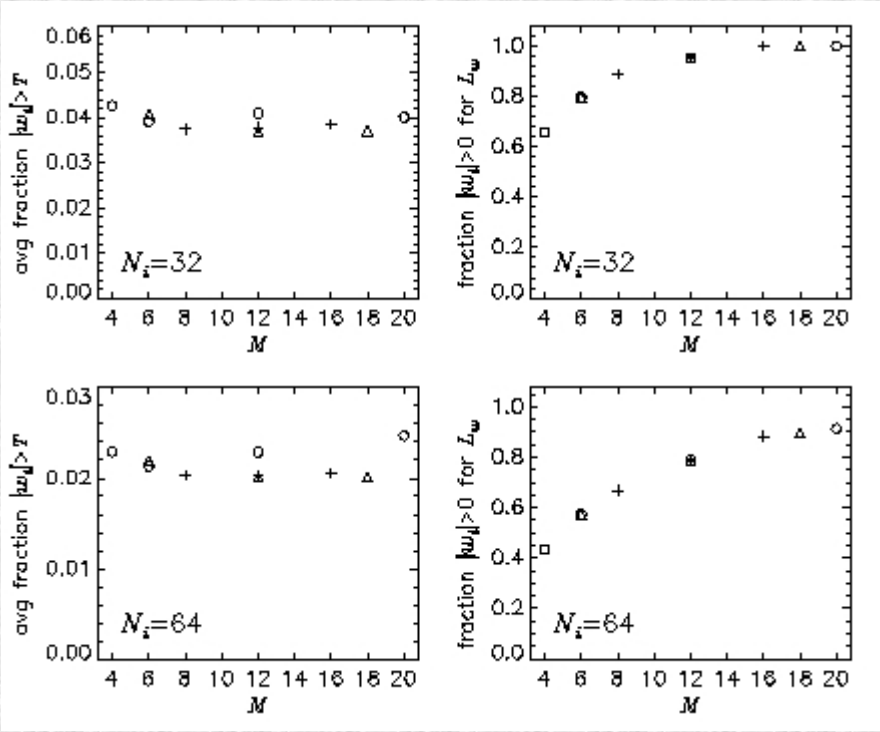


Data Compression with PCG

- PCG provides excellent compression of data and operators in wavelet space

Fermilab/NICADD photoinjector: Real Simulations

particle distribution **Laplacian operator**



$32\times32\times32$ grid, $N=125\,000$, $N_{ppc}=4.58$

~ 3.5% coefficients retained on average

$64\times64\times64$ grid, $N=1\,000\,000$, $N_{ppc}=4.58$

~ 1.75% coefficients retained on average

- compact storage of beam's distribution history needed for CSR simulations
- compact storage of beam's potential needed for modeling halo formation

Ongoing Project: Improving the PCG Solver

- Currently, we (graduate student Ben Sprague and I) are working on a number of improvements to the wavelet-based Poisson equation solver:
  - change from fixed to adaptive grid
    - simplify BC computation (currently a computational bottleneck)
    - further exploit sparsity of operators and data sets
  - use a non-standard operator form to better separate scales
  - use more sophisticated wavelet families (biorthogonal, lifted)
  - explore other preconditioners
  - parallelize and optimize
- Possible future applications of PCG solver:
  - CSR simulations: computation of retarded potentials requires integration over history of the system – compactly represented in wavelet space
  - develop a new PIC code to simulate self-gravitating systems

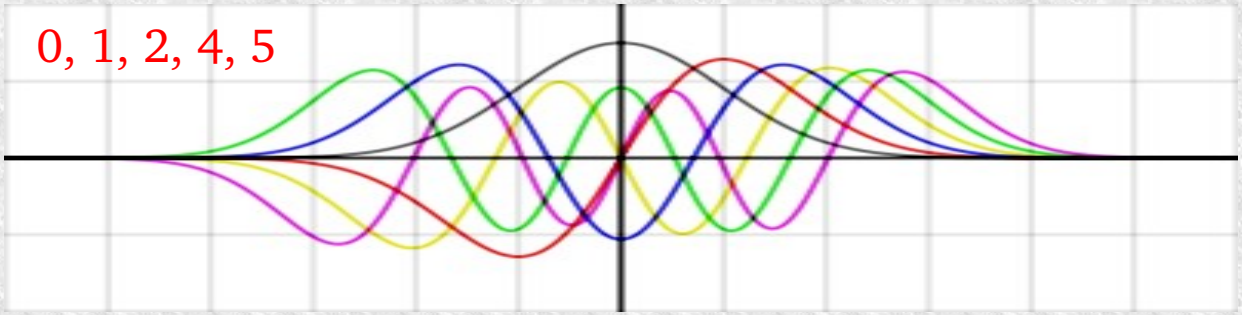
Scaled Gauss-Hermite Basis

- Gauss-Hermite orthonormal basis:
  (solution to quantum harmonic oscillator)

$$\psi_n(x)=\frac{1}{\sqrt{2^n n!\sqrt{\pi}}}H_n(x)e^{-x^2}$$

- Orthonormal: $\int_{-\infty}^{\infty}\psi_l(x)\psi_m(x)e^{x^2}dx=\delta_{lm}$ $\int_{-\infty}^{\infty}\psi_m(x)dx=\delta_m$ δ_{lm},δ_m Kronecker delta

Basis functions:
- oscillatory
- exponentially decaying



- Infinite expansion (2D): $f(x,y)=\sum_{l=0}^{\infty}\sum_{m=0}^{\infty}a_{lm}\psi_l(x)\psi_m(y)$ finite: $\sum_{l=0}^{\infty}\sum_{m=0}^{\infty}\to\sum_{l=0}^{L}\sum_{m=0}^{M}$
- Scaled and translated version:

$$f\left(\frac{x}{\alpha_1}+\bar{x},\frac{y}{\alpha_2}+\bar{y}\right)=\sum_{l=0}^{L}\sum_{m=0}^{M}a_{lm}\psi_l(x)\psi_m(y)$$

$$\alpha_1=\frac{1}{\sqrt{2}\sigma_x},\quad \alpha_2=\frac{1}{\sqrt{2}\sigma_y}$$

$$\begin{Bmatrix}\sigma_x^2\\ \sigma_y^2\end{Bmatrix}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\begin{Bmatrix}(x-\bar{x})^2\\ (y-\bar{y})^2\end{Bmatrix}f(x,y)\,dx\,dy$$

$$\begin{Bmatrix}\bar{x}\\ \bar{y}\end{Bmatrix}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\begin{Bmatrix}x\\ y\end{Bmatrix}f(x,y)\,dx\,dy$$

Scaled Gauss-Hermite Basis

- Define collocation points: $\{\tilde{\gamma}_j\}_{j=0}^{N}$ roots of $H_{L+1}(x)$
  $\{\tilde{\beta}_k\}_{k=0}^{M}$ roots of $H_{M+1}(y)$
- At collocation points: $f(\tilde{\gamma}_j, \tilde{\beta}_k) = \sum_{l=0}^{L}\sum_{m=0}^{M} a_{lm}\psi_l(\gamma_j)\psi_m(\beta_k)$ $\qquad \tilde{\gamma}_j = \frac{\gamma_j}{\alpha_1} + \bar{x}$ $\qquad \tilde{\beta}_k = \frac{\beta_k}{\alpha_2} + \bar{y}$
- Take advantage of the relation for Hermite polynomials:

$$\sum_{k=0}^{n} \frac{H_k(x)H_k(y)}{2^k k!} = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{2^{n+1} n!(x-y)}$$

to obtain

$$a_{lm} = \sum_{j=0}^{L}\sum_{k=0}^{M} \frac{1}{C_{jk}} f(\tilde{\gamma}_j, \tilde{\beta}_k)\psi_l(\gamma_j)\psi_m(\beta_k) \qquad 0 \le l \le L,\ 0 \le m \le M$$

$$C_{jk} = \sum_{l=0}^{L}[\psi_n(\gamma_j)]^2 \sum_{m=0}^{M}[\psi_m(\beta_k)]^2 \qquad 0 \le j \le L,\ 0 \le k \le M$$

- This formalism is general and can easily be extended to higher dimensions

Poisson Equation in Scaled Gauss-Hermite Basis

- Poisson equation:

$$\Delta\Phi\left(\frac{x}{\alpha_1}+\bar{x},\frac{y}{\alpha_2}+\bar{y}\right) = \left[\partial_x^2+\partial_y^2\right]\Phi\left(\frac{x}{\alpha_1}+\bar{x},\frac{y}{\alpha_2}+\bar{y}\right) = \kappa f\left(\frac{x}{\alpha_1}+\bar{x},\frac{y}{\alpha_2}+\bar{y}\right)$$

$$\Phi\left(\frac{x}{\alpha_1}+\bar{x},\frac{y}{\alpha_2}+\bar{y}\right) = \sum_{l=0}^{\infty}\sum_{m=0}^{\infty} b_{lm}\psi_l(x)\psi_m(y)$$

where b_{lm} are given by the difference relation:

$$2\alpha_1^2\sqrt{l(l-1)}\,b_{n-2m} + 2\alpha_2^2\sqrt{m(m-1)}\,b_{lm-2} = \kappa a_{lm}$$

with "boundary" coefficients:

$$b_{nm}=\begin{cases}\dfrac{\kappa}{2\alpha_2^2\sqrt{(m+2)(m+1)}}\,a_{lm+2}, & l=0,1 \quad\wedge\quad m\geq 2,\\ \dfrac{\kappa}{2\alpha_1^2\sqrt{(l+2)(l+1)}}\,a_{l+2m}, & m=0,1 \quad\wedge\quad l\geq 2,\end{cases}$$

- No need to invert the difference equation: compute "boundary" first, and then work inside $\longrightarrow$ *computationally simple and efficient*

Simulating Multiparticle Systems with Scaled Gauss-Hermite Expansion

- *N*-body realization of the discrete particle distribution:

$$f(x,y) = \frac{1}{N}\sum_{i=1}^{N}\delta(x-x_i)\delta(y-y_i)$$

- Expanding *f(x,y)* in scaled Gauss-Hermite basis reduces to the following steps:

1. tabulate the unchanging quantities:

$$C_{jk} = \sum_{l=0}^{L}[\psi_l(\gamma_j)]^2\sum_{m=0}^{M}[\psi_m(\beta_k)]^2$$

$$p_{lmjk} = \frac{\psi_l(\gamma_j)\psi_m(\beta_k)}{C_{jk}}$$

2. compute $\bar{x}, \bar{y}, \alpha_1, \alpha_2$:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N}x_i \qquad \alpha_1 = \left[\frac{2}{N}\sum_{i=1}^{N}(x_i-x)^2\right]^{-1/2} \qquad \tilde{\gamma}_j = \frac{\gamma_j}{\alpha_1}+\bar{x}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N}y_i \qquad \alpha_2 = \left[\frac{2}{N}\sum_{i=1}^{N}(y_i-y)^2\right]^{-1/2} \qquad \tilde{\beta}_k = \frac{\beta_k}{\alpha_2}+\bar{y}$$

3. evaluate $f(\tilde{\gamma}_j, \tilde{\beta}_k)$ at the nodes

4. compute coefficients $a_{lm} = \sum_{j=0}^{L}\sum_{k=0}^{M}p_{lmjk}f(\tilde{\gamma}_j, \tilde{\beta}_k)$

Simulating Multiparticle Systems with Scaled Gauss-Hermite Expansion

- *N*-body realization of the discrete particle distribution:

$$f(x,y) = \frac{1}{N}\sum_{i=1}^{N}\delta(x-x_i)\delta(y-y_i)$$

- Expanding *f(x,y)* in scaled Gauss-Hermite basis reduces to the following steps:

1. tabulate the unchanging quantities:

$$C_{jk} = \sum_{l=0}^{L}[\psi_l(\gamma_j)]^2\sum_{m=0}^{M}[\psi_m(\beta_k)]^2$$

$$p_{lmjk} = \frac{\psi_l(\gamma_j)\psi_m(\beta_k)}{C_{jk}}$$

2. compute $\bar{x}, \bar{y}, \alpha_1, \alpha_2$:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N}x_i \qquad \alpha_1 = \left[\frac{2}{N}\sum_{i=1}^{N}(x_i-x)^2\right]^{-1/2} \qquad \tilde{\gamma}_j = \frac{\gamma_j}{\alpha_1}+\bar{x}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N}y_i \qquad \alpha_2 = \left[\frac{2}{N}\sum_{i=1}^{N}(y_i-y)^2\right]^{-1/2} \qquad \tilde{\beta}_k = \frac{\beta_k}{\alpha_2}+\bar{y}$$

3. evaluate $f(\tilde{\gamma}_j, \tilde{\beta}_k)$ at the nodes

4. compute coefficients $a_{lm} = \sum_{j=0}^{L}\sum_{k=0}^{M}p_{lmjk}f(\tilde{\gamma}_j, \tilde{\beta}_k)$

function estimation from a discrete sample

Simulating Multiparticle Systems with Scaled Gauss-Hermite Expansion

- Evaluate $f(\tilde{\gamma}_j, \tilde{\beta}_k)$ from a discrete sample (nonparametric density estimation)

$$f(\tilde{\gamma}_l, \tilde{\beta}_m) = \frac{\int_{-h_x}^{h_x}\int_{-h_y}^{h_y} f(\tilde{\gamma}_l+\tilde{x}, \tilde{\beta}_m+\tilde{y})\, d\tilde{y}\, d\tilde{x}}{\int_{-h_x}^{h_x}\int_{-h_y}^{h_y} d\tilde{y}\, d\tilde{x}}$$

shifted histogram estimator

with "window" $[-h_x,h_x]\times[-h_y,h_y]$

- Optimal size of the window: $h_{opt} = \left(\frac{9}{2}\right)^{1/5}\left[\int_{-h_x}^{h_x} f''(x)\, dx\right]^{1/5} N^{-1/5}$

- Integrated means square error (*IMSE*)

$$IMSE = \frac{5}{4} 2^{-4/5} 9^{-1/5}\left[\int_{-h_x}^{h_x} f''(x)\, dx\right]^{1/5} N^{-4/5} \sim N^{-4/5}$$

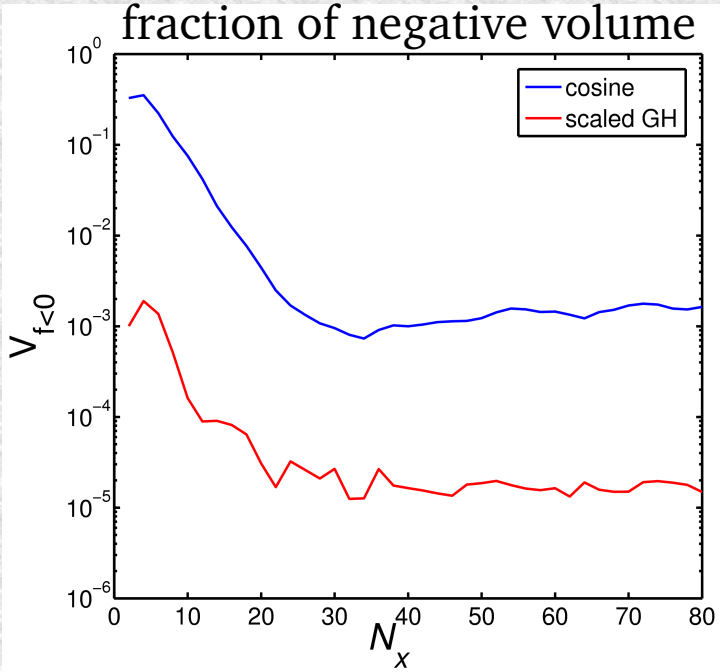
- Other, more sophisticated estimators are available:
  - kernel, adaptive estimators, and others...

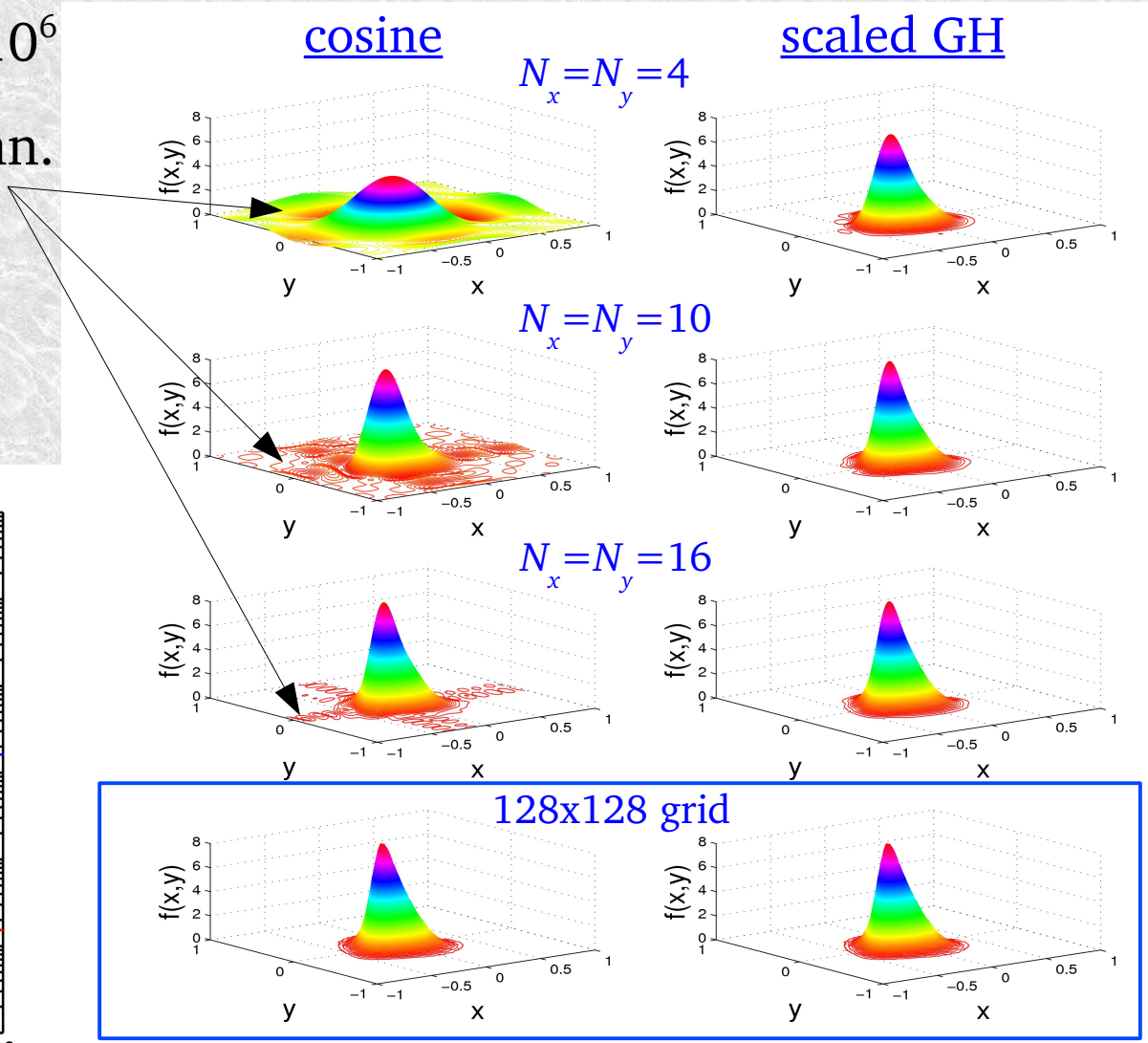
Applying Scaled Gauss-Hermite Expansion

- Future application of scaled Gauss-Hermite approximation:
  2D CSR code of Bassi, Ellison, Heinemann and Warnock:
  - particle distribution is sampled by *N* macroparticles
  - distribution is approximated at each timestep with a cosine expansion
  - beam self-forces are computed from the analytic expansion
  - Problems:
    - *unphysical "wiggles"* in the tails of the distribution
    - *computational speed*: each coefficient requires *N* cosine evaluations
  - Problems resolved (?) with scaled Gauss-Hermite:
    - no wiggles – basis functions are exponentially decaying
    - computing coefficients scales more favorably and does not involve evaluation of any expensive function (addition & multiplication)

Applying Scaled Gauss-Hermite Expansion

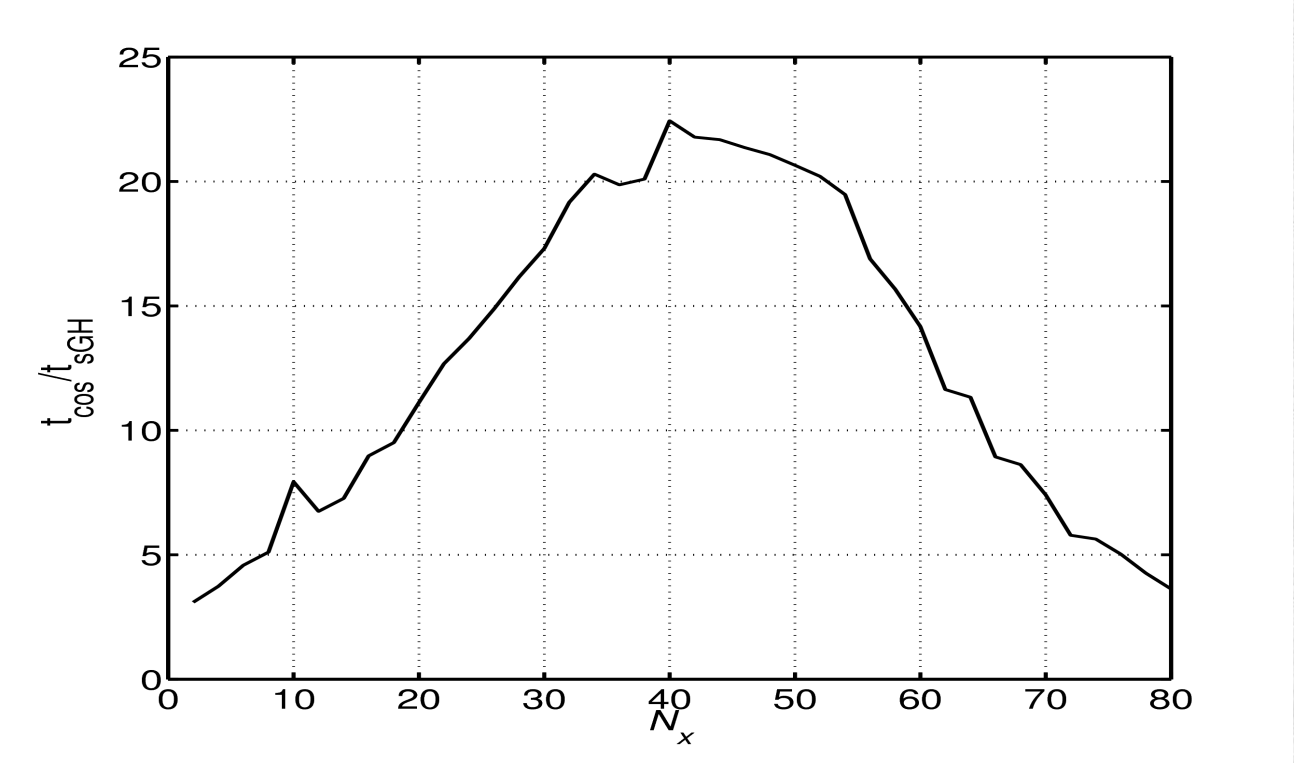
- Typical simulation: $N=10^6$
- "Wiggles" in cosine expan.
- Reduced by orders of mag. in scaled GH





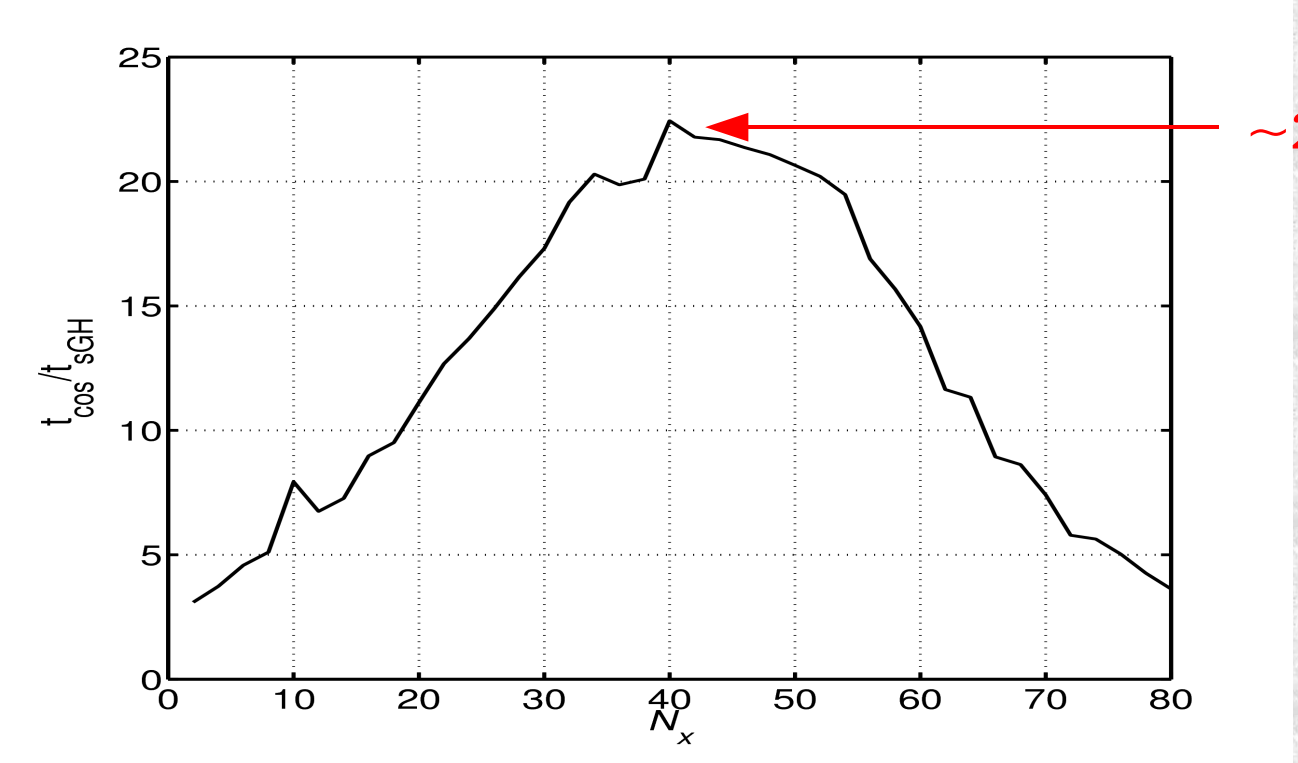
Applying Scaled Gauss-Hermite Expansion

- Scaled Gauss-Hermite expansion is computationally appreciably faster:
  - cosine expansion: $t_{\cos} \sim \mathrm{O}(LMN_{part})$
  - scaled Gauss-Hermite: $t_{\mathrm{sGH}} \sim \mathrm{O}((L+M)N_{part}) + \mathrm{O}(L^2M^2)$
    - ratio $t_{\cos}/t_{\mathrm{sGH}} \sim \mathrm{O}(L) + \mathrm{O}(N_{part}/L^2)$ (assume $L=M$)



Applying Scaled Gauss-Hermite Expansion

- Scaled Gauss-Hermite expansion is computationally appreciably faster:
  - cosine expansion: $t_{\cos} \sim \mathrm{O}(LMN_{part})$
  - scaled Gauss-Hermite: $t_{\mathrm{sGH}} \sim \mathrm{O}((L+M)N_{part}) + \mathrm{O}(L^2M^2)$
    - ratio $t_{\cos}/t_{\mathrm{sGH}} \sim \mathrm{O}(L) + \mathrm{O}(N_{part}/L^2)$ (assume $L=M$)



Scaled Gauss-Hermite Expansion: Loose Ends

- There are several issues with the scaled Gauss-Hermite expansion that we are still exploring/resolving:
  - convergence
    - different estimators for evaluating $f(\tilde{\gamma}_j, \tilde{\beta}_k)$
    - optimal number of basis functions (when is “more” less?)
  - what do we lose by using an analytic expansion?
    - avoid danger of smoothing over physical small-scale structures
  - adequate resolution: can this approach resolve physical small-scale structure?
- When these issues are properly addressed, we will have another tool with which to attack CSR and integration over beam's history

Summary

- Designed an iterative wavelet-based Poisson solver (PCG)
  - wavelet *compression* and *denoising* achieves computational speedup
  - *preconditioning* and *sparsity* of operators and data in wavelet space reduce CPU load
  - integrated PCG into a PIC code (IMPACT-T) for beam dynamics simulations
  - *current efforts*: adaptive grid, parallelization, optimization
  - *future uses*: - probe usefulness of wavelet methodology in CSR simulations
    - simulate self-gravitating systems
- Developed a scaled Gauss Hermite approximation (still a prototype):
  - efficient representation of particle distribution
  - Poisson equation solved directly at a marginal cost
  - *current efforts*: resolving issues of convergence, truncation of expansion
  - *future uses (?)*: - in Bassi *et al.*'s 2D CSR code
    - in Rui Li's 2D CSR code