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)

\[ F = -\nabla\Phi \]

- **system at** $t=t_0$
  - $N$ macro-particles
  - Newton's equations advance particles by $\Delta t$
  - system at $t=t_0+\Delta t$

- **interpolate to find** $F$ for each particle

- **bin macro-particles to obtain** particle distribution $\rho$
  - on a finite $N_x \times N_y \times N_z$ grid
  - solve the Poisson equation
    \[ \Delta \Phi = \rho \]
    - on a finite $N_x \times N_y \times N_z$ grid
Algorithms for \textit{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)

\begin{itemize}
  \item \textbf{Direct summation}: CPU cost scales as $N^2$
  \item \textbf{Tree}: direct summation nearby and statistical treatment farther away
  \item \textbf{Particle-In-Cell (PIC)}: particles binned in cells (grid)
\end{itemize}

\begin{align*}
  \text{system at } t = t_0 & \\
  N \text{ macro} & \\
  \text{macroparticles} & \\
  \text{Newton's equations} & \\
  \text{advance particles by } \Delta t & \\
  \text{system at } t = t_0 + \Delta t & \\
  \text{interpolate to find} & \\
  F \text{ for each particle} & \\
  F = - \nabla \Phi & \\
  \text{bin macro} & \\
  \text{macroparticles to obtain} & \\
  \text{particle distribution } \rho & \\
  \text{on a finite } N_x \times N_y \times N_z \text{ grid} & \\
  \text{solve the Poisson equation} & \\
  \Delta \Phi = \rho & \\
  \text{on a finite } N_x \times N_y \times N_z \text{ grid} & \\
  \text{use wavelets} & \\
\end{itemize}
Algorithms for $N$-body Simulations

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

Newton's equations advance particles by $\Delta t$

system at $t = t_0$

$N$ macroparticles

differentiate to get $F$ for each particle

system at $t = t_0 + \Delta t$

$F = -\nabla \Phi$

expand particle distribution $\rho$ in a finite orthogonal basis

$\rho(x,y,z) = \sum_{ijk} \rho_{ijk} \psi_{ijk}(x,y,z)$

solve the Poisson equation $\Delta \Phi = \rho$ in the same basis

$\Phi(x,y,z) = \sum_{ijk} \Phi_{ijk} \psi_{ijk}(x,y,z)$
Algorithms for N-body Simulations

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

\[
\rho(x,y,z) = \sum_{ijk} \rho_{ijk} \psi_{ijk}(x,y,z)
\]

\[
\Phi(x,y,z) = \sum_{ijk} \Phi_{ijk} \psi_{ijk}(x,y,z)
\]

Use scaled Gauss-Hermite basis

System at \( t = t_0 \)

\( N \) macroparticles

Newton's equations

Advance particles by \( \Delta t \)

\[
F = -\nabla \Phi
\]

Differentiate to get \( F \) for each particle

System at \( t = t_0 + \Delta t \)
Wavelets

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

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

$$f(x) = s_0^0 \phi_0^0(x) + \sum_i \sum_k d^k_i \psi^k_i(x)$$

- Discrete Wavelet Transform (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: \( N_{\text{simulation}} \ll N_{\text{physical}} \)
  - discreteness of the computational domain: \( \rho \) and \( \Phi \) 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

![Diagram](image)

- \( x \) – macroparticle location
- \( \bullet \) – gridpoint location
Numerical Noise in PIC Simulations

- For NGP, at each gridpoint, particle dist. is Poissonian:
  \[ P = (n!)^{-1} n_j^n e^{-n} \]
  \( n_j \) is the expected number in \( j^{\text{th}} \) cell; \( n \) integer

- For CIC, at each gridpoint, particle dist. is contracted Poissonian:
  \[ P = (n!)^{-1} (an_j)^n e^{-an} \]
  \( a = (2/3)^{(D/2)} \sim 0.54 \) (3D), 0.67 (2D), 0.82 (1D)

- Measure of error (noise) in depositing macroparticles onto a grid:
  \[ \sigma^2 = (N_{\text{grid}})^{-1} \sum_{i=1}^{N_{\text{grid}}} \text{Var}(q_i) \]
  \[ \sigma_{\text{NGP}}^2 = \frac{Q_{\text{total}}^2}{N N_{\text{grid}}} \]
  \[ \sigma_{\text{CIC}}^2 = \frac{a^2 Q_{\text{total}}^2}{N N_{\text{grid}}} \]
  where \( q_i = (Q_{\text{total}}/N)n_i \), \( Q_{\text{total}} \) total charge; \( N_{\text{grid}} \) number of gridpoints

(For more details see Teržić, 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:
  
  \[ \text{signal} \rightarrow \text{few large wavelet coefficients } c_{ij} \]
  
  \[ \text{noise} \rightarrow \text{many small wavelet coefficients } c_{ij} \]

- **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 \( T \)

\[ T = 2 \sqrt{\log N_{\text{grid}}} \sigma \]

(\( \sigma \) 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

\[ SNR \sim \sqrt{\frac{N}{ppc}} \]

\( N_{ppc} \): avg. # of particles per cell  
\( N_{ppc} = \frac{N}{N_{\text{cells}}} \)
Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the Signal-to-Noise Ratio (SNR) which measures its quality

\[ \text{SNR} \sim \sqrt{N_{\text{ppc}}} \]

\[ N_{\text{ppc}} : \text{avg. # of particles per cell} \quad N_{\text{ppc}} = N/N_{\text{cells}} \]

2D superimposed Gaussians on 256×256 grid
Wavelet Denoising and Compression

- Whenever the discrete signal is analytically known, one can compute the Signal-to-Noise Ratio (SNR) which measures its quality

\[ \text{SNR} \sim \sqrt{\frac{N}{\text{ppc}}} \]

\[ N_{\text{ppc}} : \text{avg. \# of particles per cell} \quad N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \]

2D superimposed Gaussians on 256×256 grid

ANALYTICAL

\[ N_{\text{ppc}} = 3 \quad \text{SNR} = 2.02 \]
Whenever the discrete signal is analytically known, one can compute the Signal-to-Noise Ratio (SNR) which measures its quality

\[ SNR \sim \sqrt{N_{ppc}} \]

\[ N_{ppc} : \text{avg. # of particles per cell} \quad N_{ppc} = \frac{N}{N_{cells}} \]

2D superimposed Gaussians on 256×256 grid

**ANALYTICAL**

\[ N_{ppc} = 3 \quad SNR = 2.02 \]

\[ N_{ppc} = 205 \quad SNR = 16.89 \]
Wavelet Denoising and Compression

• Whenever the discrete signal is analytically known, one can compute the Signal-to-Noise Ratio (SNR) which measures its quality

\[ SNR \sim \sqrt{N_{\text{ppc}}} \]

\[ N_{\text{ppc}} : \text{avg. \# of particles per cell} \quad N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \]

2D superimposed Gaussians on 256×256 grid

• denoising by wavelet thresholding: if \( |c_{ij}| < T \), set to 0

ANALYTICAL

\( N_{\text{ppc}} = 3 \quad \text{SNR}=2.02 \)

\( N_{\text{ppc}} = 205 \quad \text{SNR}=16.89 \)
Whenever the discrete signal is analytically known, one can compute the **Signal-to-Noise Ratio (SNR)** which measures its quality

\[ SNR \sim \sqrt{N_{\text{ppc}}} \]

\[ N_{\text{ppc}} : \text{avg. \# of particles per cell} \quad N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \]

**2D superimposed Gaussians on 256×256 grid**

- denoising by wavelet thresholding: \( |c_{ij}| < T \), 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

Poisson equation in physical space

\[ \Delta u = f \]

discretize Poisson equation on a \( N_x \times N_y \times N_z \) grid

\[ L U = F \]

transform discretized Poisson eq. to wavelet space

\[ \text{wavelet-threshold source } F_w, \text{ operator } L_w \]

Preconditioned Conjugate Gradient in wavelet space

\[ |AX - B|_2 \leq \varepsilon^2 |B|_2 \]

precondition Laplacian \( L_w \) with diagonal preconditioner \( P \)

\[ (PL_w P) P^{-1} U_w = PF_w \]

solution \( U \) on the \( N_x \times N_y \times N_z \) grid

BCs: using Green's functions

\[ u_{\text{bnd}} = g \]

\[ \Delta u = f \]

DWT

\[ L_w U_w = F_w \]

\[ k (L) \sim O(N_x^2) \]

\[ k (L_w) \sim O(N_x^2) \]

\[ k (PL_w P) \sim O(N_x) \]
Preconditioned Conjugate Gradient (PCG)

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

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

<table>
<thead>
<tr>
<th>no preconditioning</th>
<th>$U^i=0$ initial guess</th>
<th>75.2</th>
</tr>
</thead>
</table>

average over 30000-step run

**typical realistic beam simulation**

- $\#$ of PCG iterations
- simulation step #
Preconditioned Conjugate Gradient (PCG)

- convergence rate depends on condition number $k$
  $$|u - u'| \leq \left(\frac{\sqrt{k - 1}}{\sqrt{k + 1}}\right)^i |u|_2$$

- preconditioning (diagonal in wavelet space):
  $$k \sim O(N^2_x) \rightarrow k \sim O(N_x)$$

- good initial approximation: solution at previous time step

```plaintext
no preconditioning  $U' = 0$ initial guess  75.2
preconditioned  $U' = 0$ initial guess  60.7
```

average over 30000-step run

**typical realistic beam simulation**

**Diagram:**
- Graph showing the number of PCG iterations vs. simulation step number.
- Graph compares no preconditioning ($U' = 0$ initial guess) and preconditioned ($U' = 0$ initial guess) cases.

Terzić  JLab Beam Seminar, April 2008  21
Preconditioned Conjugate Gradient (PCG)

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

**Typical realistic beam simulation**

<table>
<thead>
<tr>
<th>Initial Guess</th>
<th>Preconditioning</th>
<th>Average over 30000-step run</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U^i = 0$</td>
<td>no</td>
<td>75.2</td>
</tr>
<tr>
<td>$U^i = 0$</td>
<td>preconditioned</td>
<td>60.7</td>
</tr>
<tr>
<td>$U^i = U^{i-1}$</td>
<td>no</td>
<td>4.8</td>
</tr>
</tbody>
</table>
Preconditioned Conjugate Gradient (PCG)

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

![Graph showing typical realistic beam simulation](image)

<table>
<thead>
<tr>
<th></th>
<th>no preconditioning</th>
<th>preconditioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U^i=0$ initial guess</td>
<td>75.2</td>
<td>60.7</td>
</tr>
<tr>
<td>$U^i=U^{i-1}$ initial guess</td>
<td>4.8</td>
<td>2.4</td>
</tr>
</tbody>
</table>

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: \textit{rms} Properties

Fermilab/NICADD photoinjector \(32 \times 32 \times 32\) grid 1 nC charge

\begin{align*}
\text{Good agreement} & \quad \text{to a few percent} \\
\text{rms beam radius} & \\
\text{rms normalized transverse emittance} & \\
\text{rms bunch length} & \\
\text{rms normalized longitudinal emittance} & 
\end{align*}
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
- $32 \times 32 \times 32$ grid, $N=200000$
- very good agreement in detail
- **Speed comparison**: IMPACT-T w/ PCG $\sim 10\%$ faster than the conventional serial IMPACT-T

Terzić
**Data Compression with PCG**

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

  **Fermilab/NICADD photoinjector:** Real Simulations

  **particle distribution**  **Laplacian operator**

  \[32 \times 32 \times 32 \text{ grid, } N=125 \ 000, \ N_{\text{ppc}} = 4.58 \sim 3.5\% \text{ coefficients retained on average}\]

  \[64 \times 64 \times 64 \text{ grid, } N=1 \ 000 \ 000, \ N_{\text{ppc}} = 4.58 \sim 1.75\% \text{ coefficients retained on average}\]

- 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: Basis functions:
  - oscillatory
  - exponentially decaying

Basis functions:
- \(0, 1, 2, 4, 5\)

- Infinite expansion (2D):
  \[ f(x, y) = \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} a_{lm} \psi_l(x) \psi_m(y) \]
  finite: \( \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} \rightarrow \sum_{l=0}^{L} \sum_{m=0}^{M} \)

- Scaled and translated version:
  \[ f\left(\frac{x}{\alpha_1} + \bar{x}, \frac{y}{\alpha_2} + \bar{y}\right) = \sum_{l=0}^{L} \sum_{m=0}^{M} a_{lm} \psi_l(x) \psi_m(y) \]
  \[ \begin{align*}
  \sigma_x^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\bar{x})^2 f(x, y) \, dx \, dy \\
  \sigma_y^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (y-\bar{y})^2 f(x, y) \, dx \, dy \\
  \bar{x} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) \, dx \, dy \\
  \bar{y} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x, y) \, dx \, dy \\
  \end{align*} \]

\( \alpha_1 = \frac{1}{\sqrt{2 \sigma_x}}, \quad \alpha_2 = \frac{1}{\sqrt{2 \sigma_y}} \)
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: 
  \[ f(\tilde{\gamma}_j, \tilde{\beta}_k) = \sum_{l=0}^L \sum_{m=0}^M a_{lm} \psi_l(\gamma_j) \psi_m(\beta_k) \]

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

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

- 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:

\[
b_{nm} = \begin{cases} 
  \frac{\kappa}{2 \alpha_2^2 \sqrt{(m+2)(m+1)}} a_{l+2m}, & m=0,1 \land l \geq 2, \\
  \frac{\kappa}{2 \alpha_1^2 \sqrt{(l+2)(l+1)}} a_{lm+2}, & l=0,1 \land m \geq 2,
\end{cases}
\]

- No need to invert the difference equation: compute “boundary” first, and then work inside \( \rightarrow \) 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 $\bar{x}, \bar{y}, \alpha_1, \alpha_2$:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

$$\alpha_1 = \left[ \frac{2}{N} \sum_{i=1}^{N} (x_i - x)^2 \right]^{-1/2}$$

$$\gamma_j = \frac{\gamma_j}{\alpha_1} + \bar{x}$$

$$\alpha_2 = \left[ \frac{2}{N} \sum_{i=1}^{N} (y_i - y)^2 \right]^{-1/2}$$

$$\tilde{\beta}_k = \frac{\beta_k}{\alpha_2} + \bar{y}$$

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 $\bar{x}, \bar{y}, \alpha_1, \alpha_2$:
  \[ \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \]
  \[ \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \]
  \[ \alpha_1 = \left[ \frac{2}{N} \sum_{i=1}^{N} (x_i-x)^2 \right]^{-1/2} \]
  \[ \alpha_2 = \left[ \frac{2}{N} \sum_{i=1}^{N} (y_i-y)^2 \right]^{-1/2} \]
  \[ \tilde{\gamma}_j = \frac{\gamma_j}{\alpha_1} + \bar{x} \]
  \[ \tilde{\beta}_k = \frac{\beta_k}{\alpha_2} + \bar{y} \]

  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)

\[
f(\tilde{\gamma}_l, \tilde{\beta}_m) = \frac{\int \int f(\tilde{\gamma}_l + \tilde{x}, \tilde{\beta}_m + \tilde{y}) d\tilde{y} d\tilde{x}}{\int \int d\tilde{y} d\tilde{x}}
\]

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 “wigglers”* 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

$$\begin{align*}
N_x &= N_y = 4 \\
N_x &= N_y = 10 \\
N_x &= N_y = 16
\end{align*}$$

128x128 grid
Applying Scaled Gauss-Hermite Expansion

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

- ratio \( t_{\cos} / t_{\text{sGH}} \sim O(L) + O(N_{\text{part}}/L^2) \) (assume \( L=M \))
Applying Scaled Gauss-Hermite Expansion

- Scaled Gauss-Hermite expansion is computationally appreciably faster:
  - cosine expansion:  \( t_{\cos} \sim O(LMN_{\text{part}}) \)
  - scaled Gauss-Hermite:  \( t_{s\text{GH}} \sim O((L+M)N_{\text{part}}) + O(L^2M^2) \)

- ratio  \( \frac{t_{\cos}}{t_{s\text{GH}}} \sim O(L) + O(N_{\text{part}}/L^2) \) (assume \( L=M \))

\[ \text{\~20 times faster} \]
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
Auxiliary Slides
Computing BCs: Current Implementation

- Green's function corresponding to a grounded rectangular pipe

\[
\rho^{ln}(z) = \frac{4}{ab} \int_{0}^{a} \int_{0}^{b} \rho(x, y, z) \sin(\alpha_l x) \sin(\beta_m y) \, dx \, dy
\]

\[
\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = -\rho^{lm}(z)
\]

\[
\phi(x, y, z) = \sum_{l=1}^{N_x} \sum_{m=1}^{N_y} \phi^{lm}(z) \sin(\alpha_l x) \sin(\beta_m y)
\]

EIGENFUNCTIONS
\[\sin(\alpha_l x)\]
\[\sin(\beta_m y)\]

EIGENVALUES
\[\alpha_l = l \pi / a\]
\[\beta_m = m \pi / b\]
\[\gamma_{lm}^2 = \alpha_l^2 + \beta_m^2\]
Computing BCs: Adaptive Grid

V=0

computational grid

V=0
Computing BCs: Adaptive Grid

V=0

computational grid

V=0
Computing BCs: Adaptive Grid

V=0

computational grid
Preconditioner in Wavelet Space
Laplacian in Standard Form in Wavelet Space
Laplacian in Non-Standard Form in Wavelet Space
Wavelet Decomposition

The continuous wavelet transform of a function \( f(t) \) is

\[
\gamma(s, \tau) = \int_{-\infty}^{\infty} f(t) \psi_{s,\tau}(t) \, dt
\]

\[
\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi \left( \frac{t - \tau}{s} \right)
\]

\( \psi(t) \) mother wavelet with scale and translation dimensions \( s \) and \( \tau \) respectively.
How Do Wavelets Work?

Wavelet analysis (wavelet transform):

- **Approximation** – apply low-pass filter to Signal and down-sample
- **Detail** – apply high-pass filter to Signal and down-sample
- **Wavelet synthesis** (inverse wavelet transform): up-sampling & filtering
- **Complexity**: \(4MN\), \(M\) the size of the wavelet, \(N\) number of cells
  - Recall: FFT complexity \(4N \log_2 N\)

S - signal
A - approximation
D - detail
Wavelets

- Wavelet transform separates scales
Numerical Noise in PIC Simulations

- In wavelet space:
  - signal $\rightarrow$ few large wavelet coefficients $c_{ij}$
  - noise $\rightarrow$ many small wavelet coefficients $c_{ij}$

- Poissonian noise $\rightarrow$ Gaussian noise

- 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 $T$

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

($\sigma$ was estimated earlier)

---

Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201
3D Plummer sphere on a 32x32x32 grid

BCs: open in all directions (analytically specified)
3D 'fuzzy cigar' on a 32x32x32 grid

BCs: grounded rectangular pipe ($V=0$ on the sides), open in $z$-direction (Green's function solution)
2D EXAMPLE: “MICROBUNCHED” DENSITY DISTRIBUTION (with $U=0$ starting guess)

BCs: open in all directions (analytically specified)
IMPACT-T Vs. IMPACT-T with PCG: 

**rms properties**

Fermilab/NICADD photoinjector \((32 \times 32 \times 32 \text{ grid})\) 1 nC charge

bunch compressor at \(z=5.65\text{m}\) from cathode

- **rms beam radius**
- **rms normalized transverse emittance**

- **rms bunch length**
- **rms normalized longitudinal emittance**
IMPACT-T Vs. IMPACT-T with PCG: level of detail

- transverse charge distribution for the Fermilab/NICADD photoinjector simulation
- 5-beamlets (masked photocathode)
- $32 \times 32 \times 32$ grid, $N=200000$
- very good agreement in detail
IMPACT-T Vs. IMPACT-T with PCG: level of detail

- transverse charge distribution for the AES/JLab low-charge photoinjector simulation
- very non-axisymmetric beam
- $32 \times 32 \times 32$ grid, $N=200000$
- very good agreement in detail
IMPACT-T Vs. IMPACT-T with PCG: level of detail

- transverse charge distribution for the AES/JLab low-charge photoinjector simulation
- very non-axisymmetric beam
- $32 \times 32 \times 32$ grid, $N = 200000$
- very good agreement in detail
Scaled Gauss-Hermite Basis

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

- Hermite polynomials $H_n(x)$ relations:

$$ H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x) $$
$$ H_n'(x) = 2xH_{n-1}(x) $$

- Orthonormal: $\int_{-\infty}^{\infty} \psi_l(x)\psi_m(x)e^{x^2}dx = \delta_{lm}$

$$ \int_{-\infty}^{\infty} \psi_m(x)dx = \delta_m $$
$$ \delta_{lm}, \delta_m \text{ Kronecker delta} $$

Basis functions:
- oscillatory
- exponentially decaying

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

- Infinite expansion (2D): $f(x, y) = \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} a_{lm} \psi_l(x)\psi_m(y) $$

finite: $\sum_{l=0}^{\infty} \sum_{m=0}^{\infty} \rightarrow \sum_{l=0}^{L} \sum_{m=0}^{M}$

- Scaled and translated version:

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

$$ \left\{ \sigma_{x}^2, \sigma_{y}^2 \right\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( (x-\bar{x})^2 \right) f(x, y) dx \, dy $$

$$ \left\{ \bar{x}, \bar{y} \right\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ x, y \right\} f(x, y) dx \, dy $$

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

0, 1, 2, 4, 5