Progress Report:
New Particle-In-Cell Code For
Numerical Simulations of
Coherent Synchrotron Radiation

Balša Terzić and Rui Li
CASA, Jefferson Lab

CASA Group Meeting
May 26, 2010
Outline of the Talk

• Motivation and Background of Coherent Synchrotron Radiation:
  • Physical problem
  • Computational challenges
  • Two approaches: point-to-point (P2P) and particle-in-cell (PIC)
    • Why develop a new PIC code from an existing P2P code designed by Rui Li?

• New Particle-In-Cell CSR Code
  • Outline of the new algorithm
  • First results: benchmarking against analytical results

• Still to do...

• Summary
Coherent Synchrotron Radiation: Motivation and Background

- When a charged particle beam travels along a curved trajectory (bending magnet), beam emits synchrotron radiation.

- If the wavelength $\lambda$ of synchrotron radiation is longer than the bunch length $\sigma_s$, the resulting radiation is coherent (CSR).

  - ISR radiated power: $P = \frac{e^2 c N E^4}{6\pi\varepsilon_0 (m_0 c^2)^4 R^2}$
  - CSR radiated power: $P = \frac{2^{4/3}}{\pi} \left[ \Gamma \left( \frac{2}{3} \right) \right]^2 \frac{N^2 e^2}{R^{2/3} \sigma_s^{4/3}}$

  - ISR radiated power: $\frac{P_e}{P_p} = \left( \frac{m_p}{m_e} \right)^4 = 1.13 \times 10^{13}$

  - Synchrotron radiation mostly negligible in protons.
Coherent Synchrotron Radiation: Motivation and Background

- CSR is the low frequency part of the synchrotron radiation power spectrum

- $N$ particles in the bunch act in phase and enhance intensity by a factor $N$ (typically $N=10^9-10^{11}$)

- Therefore for shorter bunch ($\sigma_s$ small), CSR is more pronounced

\[ P \propto \frac{N^2}{R^{2/3} \sigma_s^{4/3}} \]
**Coherent Synchrotron Radiation: Motivation and Background**

- Short bunch lengths are desirable in many different contexts:
  - FEL, ERL, B-factories, linear colliders such as ILC...
  - The demand for short bunches is expected to increase in the future
- This presents a problem:
  - Short beam bunch $\Rightarrow$ CSR is dominant
  - $\Rightarrow$ Beam is a subject to adverse CSR effects
- Adverse CSR effects, which can seriously impair beam quality:
  - Energy spread
  - Longitudinal instability (microbunching)
  - Emittance degradation
- Having a trustworthy code to simulate CSR effects is of great importance
Coherent Synchrotron Radiation: Computational Challenges

- Numerical simulations of the CSR effects have proven to be extremely challenging because of:
  - Memory requirement associated with storing history of the beam bunch
  - Difficulty to accurately account for retardation
  - Large cancellation between $E$ and $B$ fields in the Lorentz force
  - Sensitivity to numerical noise
  - The manner in which self-interactions scale in numerical algorithms

- We present the new code which simultaneously deals with all of these issues
Coherent Synchrotron Radiation: Computational Challenges

• Dynamics of an electron bunch is governed by

\[
\frac{d}{dt}(\gamma m_e \vec{v}) = e(\vec{E} + \vec{\beta} \times \vec{B})
\]

\[
\vec{\beta} = \vec{v}/c
\]

\[
\vec{E} = \vec{E}^{\text{ext}} + \vec{E}^{\text{self}}
\]

\[
\vec{B} = \vec{B}^{\text{ext}} + \vec{B}^{\text{self}}
\]

- \( \vec{E}^{\text{ext}}, \vec{B}^{\text{ext}} \): external EM fields
- \( \vec{E}^{\text{self}}, \vec{B}^{\text{self}} \): self-interaction (CSR)

\[
\vec{E}^{\text{self}} = -\nabla \phi - \frac{1}{c} \frac{\partial \vec{A}}{\partial t}
\]

\[
\vec{B}^{\text{self}} = \nabla \times \vec{A}
\]

where

\[
\phi (\vec{r}, t) = \int \frac{d \vec{r}'}{|\vec{r} - \vec{r}'|} \rho(\vec{r}', t')
\]

\[
\vec{A} = \frac{1}{c} \int \frac{d \vec{r}'}{|\vec{r} - \vec{r}'|} \vec{J}(\vec{r}', t')
\]

\[
\rho(\vec{r}, t) = \int f(\vec{r}, \vec{v}, t) d\vec{v}
\]

\[
\vec{J}(\vec{r}, t) = \int \vec{v} f(\vec{r}, \vec{v}, t) d\vec{v}
\]

beam distribution function (DF): \( f(\vec{r}, \vec{v}, t) \)

Need to track the entire history of the bunch
Coherent Synchrotron Radiation: Computational Challenges

- Dynamics of an electron bunch is governed by
  \[ \frac{d}{dt}(\gamma m_e \vec{v}) = e(\vec{E} + \vec{B} \times \vec{B}) \]

  \[
  \frac{d}{dt}(\gamma m_e \vec{v}) = e(\vec{E} + \vec{B} \times \vec{B}) = e(\vec{E}_{\text{ext}} + \vec{E}_{\text{self}} + \vec{B}_{\text{ext}} + \vec{B}_{\text{self}})
  \]

LARGE CANCELLATION

- \( \vec{E}_{\text{ext}}, \vec{B}_{\text{ext}} \): external EM fields
- \( \vec{E}_{\text{self}}, \vec{B}_{\text{self}} \): self-interaction (CSR)

Frictional forces due to gradients

\[
\vec{E}_{\text{self}} = -\nabla \phi - \frac{1}{c} \frac{\partial \vec{A}}{\partial t}
\]

\[
\vec{B}_{\text{self}} = \nabla \times \vec{A}
\]

NUMERICAL NOISE DUE TO GRADIENTS

where

\[
\phi(\vec{r},t) = \int \frac{d \vec{r}'}{|\vec{r} - \vec{r}'|} \rho(\vec{r}',t')
\]

\[
\vec{A} = \frac{1}{c} \int \frac{d \vec{r}'}{|\vec{r} - \vec{r}'|} \vec{J}(\vec{r}',t')
\]

- Retarded potentials

Need to track the entire history of the bunch

MEMORY REQUIREMENT

- Charge density:
  \[
  \rho(\vec{r},t) = \int f(\vec{r},\vec{v},t) d\vec{v}
  \]

- Current density:
  \[
  \vec{J}(\vec{r},t) = \int \vec{v} f(\vec{r},\vec{v},t) d\vec{v}
  \]

- Beam distribution function (DF):
  \[
  f(\vec{r},\vec{v},t)
  \]
Coherent Synchrotron Radiation: Computational Challenges

- Storing and computing with a 4D (3 positions, 1 time) charge and current densities is prohibitively expensive
  ⇒ Need simplifications/approximations

- Possible simplifications to full dimensional CSR modeling:
  - 1D line approximation (IMPACT, ELEGANT): probably too simplistic
  - 2D approximation (vertically flat beam):
    - codes by Li 1998, Bassi et al. 2006

- Based on how the DF (and, consequently, charge and current densities) are represented, two approaches emerge:
  - **Point-to-point methods:**
    solve Maxwell's equations using retarded potentials with DF represented by macroparticles
  - **Particle-In-Cell (PIC) (mean field, grid, mesh) methods:**
    solve Maxwell's equations or retarded potentials on the grid, using finite difference, finite element, Green's function, retarded potentials...
Coherent Synchrotron Radiation: Point-to-Point Approach

- **Point-to-point approach (2D):** Li 1998
  
  \[
  f(\vec{r},\vec{\nu},t) = q \sum_{i=1}^{N} n_m(\vec{r} - \vec{r}_0^{(i)}(t)) \delta(\vec{\nu} - \vec{\nu}_0^{(i)}(t))
  \]

  - Charge density

  \[
  \rho(\vec{r},t) = q \sum_{i=1}^{N} n_m(\vec{r} - \vec{r}_0^{(i)}(t))
  \]

  - Current density

  \[
  \vec{J}(\vec{r},t) = q \sum_{i=1}^{N} \vec{\beta}_0^{(i)}(t) n_m(\vec{r} - \vec{r}_0^{(i)}(t))
  \]

  \[
  n_m(\vec{r} - \vec{r}_0^{(i)}(t)) = \frac{1}{2\pi\sigma_m^2} e^{-\frac{1}{2\sigma_m^2} \left(\frac{(x-x_0(t))^2 + (y-y_0(t))^2}{2\sigma_m^2}\right)}
  \]

  - Gaussian macroparticle

  - Charge density is sampled with \( N \) Gaussian-shaped 2D macroparticles (2D distribution without vertical spread)

  - At each timestep, each macroparticle experiences fields generated by all other macroparticles throughout history

  - **Expensive:** computation of retarded potentials and self fields \( \sim O(N^2) \)
    
    \[
    \Rightarrow \text{small number } N \Rightarrow \text{poor spatial resolution}
    \]
    
    \[
    \Rightarrow \text{difficult to see small-scale structure}
    \]

  - While useful in obtaining low-order moments of the beam, *point-to-point approach is not optimal for studying CSR*
Coherent Synchrotron Radiation: Particle-In-Cell Approach

- Particle-In-Cell approach with retarded potentials (2D):

\[
f(\vec{x}, \vec{v}, t) = q \sum_{i=1}^{N} \delta(\vec{x} - \vec{x}_0^{(i)}(t)) \delta(\vec{v} - \vec{v}_0^{(i)}(t))
\]

\[
\rho(\vec{x}_k, t) = q \sum_{i=1}^{N} \int_{-h}^{h} \delta(\vec{x}_k - \vec{x}_0^{(i)}(t) + \vec{X}) p(\vec{X}) d\vec{X}
\]

\[
\vec{J}(\vec{x}_k, t) = q \sum_{i=1}^{N} \vec{\beta}_0^{(i)}(t) \int_{-h}^{h} \delta(\vec{x}_k - \vec{x}_0^{(i)}(t) + \vec{X}) p(\vec{X}) d\vec{X}
\]

- Charge and current densities are sampled with \( N \) point-charges (\( \delta \)-functions) and deposited on a finite grid \( \vec{x}_k \) using a deposition scheme \( p(\vec{X}) \)

- Two main deposition schemes:
  - Nearest Grid Point (NGP) (constant: deposits to \( 1^D \) points)
  - Cloud-In-Cell (CIC) (linear: deposits to \( 2^D \) points)

There exist higher-order schemes

- Particles do not directly interact with each other, but only through a mean-field of the gridded representation
Coherent Synchrotron Radiation: P2P Vs. PIC

• Computational cost for P2P: Total cost \( \sim O(N^2) \)
  • Integration over history (yields self-forces): \( O(N^2) \) operation

• Computational cost for PIC: Total cost \( \sim O(N_{\text{grid}}^2) \)
  • Particle deposition (yields charge & current densities on the grid): \( O(N) \) operation
  • Integration over history (yields retarded potentials): \( O(N_{\text{grid}}^2) \) operation
  • Finite difference (yields self-forces on the grid): \( O(N_{\text{grid}}) \) operation
  • Interpolation (yields self-forces acting on \( N \) individual particles): \( O(N) \) operation
  • Overall \( \sim O(N_{\text{grid}}^2) + O(N) \) operations (but in realistic simulations: \( N_{\text{grid}}^2 \gg N \))

• Favorable scaling allows for larger \( N \), and reasonable grid resolution \( \Rightarrow \) improved spatial resolution

• Fair comparison: P2P with \( N \) macroparticles and PIC with \( N_{\text{grid}} = N \)
Coherent Synchrotron Radiation: 

P2P Vs. PIC

- Difference in spatial resolution: An illustrative example
  - Analytical distribution sampled with
    - $N = N_X N_Y$ macroparticles (as in P2P)
    - On a $N_X \times N_Y$ grid (as in PIC)
  - 2D grid: $N_X = N_Y = 32$

Signal-to-Noise Ratio

$$SNR = \sqrt{\frac{\sum_i \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}}$$

- $\bar{q}_i = \text{exact}$
- $q_i = \text{approx.}$

- PIC approach provides superior spatial resolution to P2P approach
  \( \Rightarrow \) Modify Rui Li's P2P CSR code into a PIC
integrate over particle histories to compute retarded potentials and corresponding forces on each macroparticle

Outline of P2P Algorithm

- $N$ macroparticles at $t < t_k$
- $N$ macroparticles at $t = t_k$
- system at $t = t_k + \Delta t$
- advance particles by $\Delta t$

From P2P To PIC
Outline of PIC Algorithm

- \( N \) point-particles at \( t = t_k \)
- advance particles by \( \Delta t \)
- distribution on \( N \times N_y \) grid for \( t = t_k \)
- bin particles on \( N \times N_y \) grid
- integrate over grid histories to compute retarded potentials and corresponding forces on the \( N \times N_y \) grid
- interpolate to obtain forces on each particle
- system at \( t = t_k + \Delta t \)
Outline of PIC Algorithm

- **$N$ point-particles at $t=t_k$**
- **Advance particles by $\Delta t$**
- **$N \times N_y$ grid for $t=t_k$**
- **Bin particles on $N_x \times N_y$ grid**
- **Interpolate to obtain forces on each particle**
- **Integrate over grid histories to compute retarded potentials on the $N_x \times N_y$ grid**
To simplify calculations use 3 frames of reference:

- **Frenet frame** \((s, x)\)
  - \(s\) – along design orbit
  - \(x\) – deviation normal to direction of motion

- **Lab frame** \((X, Y)\)

- **Grid frame** \((\tilde{X}, \tilde{Y})\)
  - Scaled & rotated lab frame
  - particle deposition
  - grid interpolation
  - integration
  - always \([-0.5,0.5] \times [-0.5,0.5]\)
New PIC CSR Code: Outline

**Frenet Frame**

- `deposit_particles.f90`
  - Beam bunch
    - $(s, x, p_x, p_y, p_z)$

**Lab Frame**

- `compute_potential.f90`
  - Vantage point:
    - $(t, X, Y)$
  - Compute integration range

- `compute_forces.f90`
  - Transform to get
    - $A_x, A_y, A_z$ on grid
  - Get $E_x, E_y, B_z$ on each particle (interp2D)

**Grid Frame**

- `interp_forces.f90`
  - Transform
    - $\partial_x, A_x, \partial_y, \partial_z, \partial_r, \phi$
  - $E_x, E_y, B_z$ on each particle (interp3D)

- `interp_density.f90`
  - $\rho, J$ on grid
  - $\rho, J$ at an arbitrary pt. (interp3D)
New PIC CSR Code: Particle Deposition

- Grid resolution is specified *a priori* (fixed grid)
  - $N_X$: # of gridpoints in $X$
  - $N_Y$: # of gridpoints in $Y$
  - $N_{grid} = N_X \times N_Y$ total gridpts
  - Grid: $\mathbf{x}_k = [\tilde{X}_{ij}, \tilde{Y}_{ij}]$
    
  
    $i=1,..,N_X \quad j=1,..,N_Y$
  - Inclination angle $\alpha$

- Grid is determined so as to tightly envelope all particles
  
  Minimizing number of empty cells $\Rightarrow$ optimizing spatial resolution
New PIC CSR Code: Computing Retarded Potentials

1. Determine limits of integration in lab frame:

\[ \text{compute } R_{\text{max}} \text{ and } (\theta_{\text{min}}^i, \theta_{\text{max}}^i) \]

2. Carry out integration:

\[
\begin{bmatrix}
\phi(\vec{r}, t) \\
\vec{A}(\vec{r}, t)
\end{bmatrix} = \int \left[ \rho \left( \vec{r}', t - \frac{R'}{c} \right) \right] \frac{d\vec{r}'}{|\vec{r}' - \vec{r}|} = \sum_{i=1}^{M_{\text{int}}} \int_{0}^{R_{\text{max}}} \int_{\theta_{\text{min}}^i}^{\theta_{\text{max}}^i} \left[ \rho \left( \vec{r}', t - \frac{R'}{c} \right) \right] dR' d\theta'.
\]
New PIC CSR Code: Retarded Potentials

- All this yields scalar and vector potentials on the grid:

\[
N = 512000 \\
N_x = N_y = 64
\]
New PIC CSR Code: Computing Self-Forces

- From retarded potentials on the grid to forces on individual particles:
  - Compute derivatives on the grid: \( \partial_{\tilde{x}, \tilde{y}} A_{x,y}, \partial_t A_{x,y}, \partial_{\tilde{x}, \tilde{y}} \phi \)
  - Transform to lab frame: \( \partial_{x,y} A_{x,y}, \partial_t A_{x,y}, \partial_{x,y} \phi \)
  - Transform to Frenet frame: \( \partial_{s,x} A_{s,x}, \partial_t A_{s,x}, \partial_{s,x} \phi \)
  - Interpolate from grid to individual particle positions to obtain forces acting on them:
    \[
    \begin{align*}
    \vec{E}^{self} &= -\nabla \phi - \frac{1}{c} \frac{\partial \vec{A}}{\partial t} \\
    \vec{B}^{self} &= \nabla \times \vec{A}
    \end{align*}
    \]
    \[
    \frac{d}{dt} (\gamma m_e \vec{v}) = e (\vec{E} + \vec{\beta} \times \vec{B})
    \]
- Finally able to advance particles in time
New PIC CSR Code: Large Cancellation At Work

- Traditionally difficult to track large quantities which mostly cancel out:

- High accuracy of the implementation able to accurately track these cancellations over 5 orders of magnitude
New PIC CSR Code: Benchmarking Against Analytic Results

- Analytic steady state solution available for a rigid line Gaussian bunch (Derbenev & Shiltsev 1996)

- Excellent agreement between analytic solution and the computed provides a proof of concept for the new code

\[ \frac{N}{x} = 512000 \]

\[ \frac{N}{y} = 64 \]
New PIC CSR Code:  
Still To Do...

- Polish:
  - Optimize and parallelize

- Additional functionalities:
  - Wavelet denoising and compression:
    - Less numerical noise \( \Rightarrow \) cleaner results
    - Grid representation compactly stored (compression \( \sim 100 \) times)
      decrease memory requirement
      optimize computations

- Further benchmarking and testing

- Applications:
  - Simulating real machines: JLab FEL, JLAMP, LCSL, ...

- Long term plan:
  - 3D
  - Image charges
Summary

• Motivated the need for accurate CSR codes
• Demonstrated that the PIC approach is better because of:
  – Better spatial resolution (a “must” for small-scale instabilities)
  – Better scaling with the number of particles $N$
• Presented the new PIC code:
  – Some details of the inner workings of the code
  – How the new code resolves traditional computational difficulties
  – Proof of concept: excellent agreement with analytical results
• Outlined the path ahead toward simulating real machines
• Closing in on our immediate goal: having an accurate, efficient and trustworthy code which faithfully simulates CSR effects
• Near-term goal: being able to quantitatively simulate CSR in real machines, as the first step toward controlling its adverse effects
Backup Slides
Numerical Noise in the PIC Simulations

• There are the two major sources of numerical noise in MF simulations:
  • *graininess of the distribution function*: $N_{\text{simulation}} \ll N_{\text{physical}}$
  • *discreteness of the computational domain*: quantities defined on a finite grid
• One must first understand the profile of the numerical noise associated with the discreteness of the computational in order to be able to remove it
• Systematic removal of numerical noise from the PIC simulations leads to physically more reliable results, equivalent to simulations with many more particles
Numerical Noise in the Mean Field Simulations

- If many random realizations of a given particle distribution have are deposited onto a grid, the number of particles in each gridpoint is Poisson-distributed (variance = mean) \( \Rightarrow \) noise is signal-dependent

- Wavelet denoising is at its most powerful (and mathematically strongest ground) when the noise is Gaussian-distributed (signal-independent, white)

- Signal contaminated with Poissonian noise can be transformed to signal with Gaussian noise by a variance-stabilizing Anscombe transform (1948):

\[
Y_G = 2 \sqrt{Y_P + \frac{3}{8}}
\]

- After the transformation the noise in each gridpoint is (nearly) Gaussian-distributed with variance \( \sigma = 1 \)

- Essentially, we have pre-processed the signal before denoising it

- This error/noise estimate \( \sigma \) is crucial for optimal wavelet noise removal

[For more details see Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201]
Removing Numerical Noise from PIC Simulations

- It is desirable to remove noise from the PIC simulations
  - less numerical noise $\Leftrightarrow$ running simulations with more particles
  - $\Rightarrow$ increased sensitivity to physical small-scale structure

- Noise removal from the PIC simulations can be done in several ways:
  - Particle deposition schemes:
    - Higher order deposition schemes serve as smoothing filters
  - Filtering:
    - Savitzky-Golay smoothing filter (local polynomial regression)
  - In Fourier space:
    - Truncating the highest Fourier frequencies
  - In wavelet space:
    - Wavelet coefficient thresholding

- Wavelets provide a natural setting for judicious noise removal (other methods indiscriminantly smooth over/truncate small scale structures)
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 $4N \log_2 N$
• Advantages of wavelet formulation:

- Wavelet basis functions have compact support \(\Rightarrow\) signal localized in space
  Wavelet basis functions have increasing resolution levels
  \(\Rightarrow\) signal localized in frequency

\(\Rightarrow\) simultaneous localization in space and frequency (FFT only frequency)

- Wavelet basis functions correlate well with various signal types
  (including signals with singularities, cusps and other irregularities)

\(\Rightarrow\) compact and accurate representation of data (compression)

- Wavelet transform preserves hierarchy of scales

- In wavelet space, discretized operators (Laplacian) are also sparse and have an efficient preconditioner \(\Rightarrow\) solving some PDEs is faster and more accurate

- Wavelets provide a natural setting for noise removal \(\Rightarrow\) wavelet denoising
Brief Overview of Wavelets

- **Wavelets**: orthogonal basis composed of scaled and translated versions of the same localized wavelet $\psi(x)$:
  \[ \psi^k_i(x) = 2^{k/2} \psi(2^k x - i) \quad k, i \in \mathbb{Z} \]
  \[ f(x) \approx \sum_k \sum_i d^k_i \psi^k_i(x) \]
  Each new resolution level $k$ is orthogonal to the previous levels

- Wavelets are derived from the scaling function $\phi(x)$ which satisfies
  \[ \phi(x) = \sqrt{2} \sum_j h_j \phi(2x - j) \]
  \[ \psi(x) = \sqrt{2} \sum_j g_j \phi(2x - j) \]
  (only finite number of filter coefficients $h_j$ and $g_j$ are non-zero)

- In order to attain orthogonality of different scales, their shapes are strange
  - Makes them suitable to represent irregularly shaped functions

- For discrete signals (gridded quantities), fast Discrete Wavelet Transform (DFT) is an $O(MN)$ operation, $M$ size of the wavelet filter, $N$ signal size
Brief Overview of Wavelets

- Wavelet transform separates scales
Wavelet Denoising

- 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 $c_{ij} = 0$

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

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

($\sigma = 1$ after Anscombe transform)

Denoising factor (DF):

$$DF = \frac{\text{Error}_{\text{original}}}{\text{Error}_{\text{denoised}}}$$

[Terzić, Pogorelov & Bohn 2007, PR STAB, 10, 034201]
When the signal is known, one can compute Signal-to-Noise Ratio (SNR):

\[ \text{SNR} = \sqrt{\frac{\sum_i \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}} \]

\[ \bar{q}_i = \text{exact} \quad q_i = \text{approx.} \]

- \( \text{SNR} \sim \sqrt{N_{\text{ppc}}} \)
  - \( N_{\text{ppc}} \): avg. # of particles per cell
  - \( N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \)
When the signal is known, one can compute Signal-to-Noise Ratio (SNR):

$$SNR \approx \sqrt{\frac{\sum \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}}$$

where $\bar{q}_i = $ exact $q_i = $ approx.

$SNR \sim \sqrt{N_{ppc}}$  $N_{ppc}$ : avg. # of particles per cell  $N_{ppc} = N/N_{cells}$

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

- When the signal is known, one can compute Signal-to-Noise Ratio (SNR):
  \[ SNR = \sqrt{\frac{\sum_i \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}} \]
  \( \bar{q}_i = \text{exact} \)
  \( q_i = \text{approx.} \)

\( SNR \sim \sqrt{N_{\text{ppc}}} \)

\( N_{\text{ppc}} \) : avg. # of particles per cell

\( N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \)

- 2D superimposed Gaussians on 256×256 grid

ANALYTICAL

\( N_{\text{ppc}} = 3 \quad SNR = 2.02 \)
When the signal is known, one can compute *Signal-to-Noise Ratio* (SNR):

\[ SNR = \sqrt{\frac{\sum_i \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}} \]

\( q_i = \text{approx.} \)

\( \bar{q}_i = \text{exact} \)

\( N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \)

**ANALYTICAL**

- \( N_{\text{ppc}} = 3 \) \( SNR = 2.02 \)
- \( N_{\text{ppc}} = 205 \) \( SNR = 16.89 \)

2D superimposed Gaussians on 256×256 grid
When the signal is known, one can compute Signal-to-Noise Ratio (SNR):

\[ SNR = \sqrt{\frac{\sum q_i^2}{\sum (q_i - \bar{q}_i)^2}} \]

\( q_i = \text{exact} \)
\( \bar{q}_i = \text{approx.} \)

- SNR \( \sim \sqrt{N_{\text{ppc}}} \)
  
  \( N_{\text{ppc}} \): avg. # of particles per cell

\( N_{\text{ppc}} = \frac{N}{N_{\text{cells}}} \)

- 2D superimposed Gaussians on 256\( \times \)256 grid

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

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

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

- When the signal is known, one can compute Signal-to-Noise Ratio (SNR):

\[
SNR = \sqrt{\frac{\sum_i \bar{q}_i^2}{\sum_i (q_i - \bar{q}_i)^2}}
\]

- \( SNR \sim \sqrt{N_{ppc}} \)  
  \( N_{ppc} \): avg. # of particles per cell  
  \( N_{ppc} = \frac{N}{N_{cells}} \)

2D superimposed Gaussians on 256×256 grid

- Wavelet denoising yields a representation which is:
  - Appreciably more accurate than non-denoised representation
  - Sparse (if clever, we can translate this sparsity in computational efficiency)