



OLD DOMINION
UNIVERSITY

I D E A F U S I O N

High-Performance CSR Simulations on GPUs

Balša Terzić

Department of Physics, Old Dominion University
Center for Accelerator Science, Old Dominion University

6th Microbunching Instability Workshop, Trieste, 6-8 October 2014

Collaborators

Center for Accelerator Science (CAS) at Old Dominion University (ODU):

Professors:

Physics: Alexander Godunov

Computer Science: Mohammad Zubair, Desh Ranjan

PhD student:

Computer Science: Kamesh Arumugam

Early advances on this project benefited from my collaboration with Rui Li (Jefferson Lab)

Outline

- Coherent Synchrotron Radiation (CSR)
 - Physical problem
 - Computational challenges
- New 2D Particle-In-Cell CSR Code
 - Outline of the new algorithm
 - Parallel implementation on Graphical Processing Units (GPUs)
 - Using wavelets to increase computational accuracy and efficiency
 - Benchmarking against analytical results
- Still to Come
- Summary

CSR: Physical Problem

- Beam's self-interaction due to CSR can lead to a host of adverse effects
 - Energy spread
 - Emittance degradation
 - Longitudinal instability (micro-bunching)
- Being able to quantitatively simulate CSR is the first step toward mitigating its adverse effects
- It is vitally important to have a trustworthy 2D CSR code

CSR: Computational Challenges

- Dynamics governed by the Lorenz force:

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

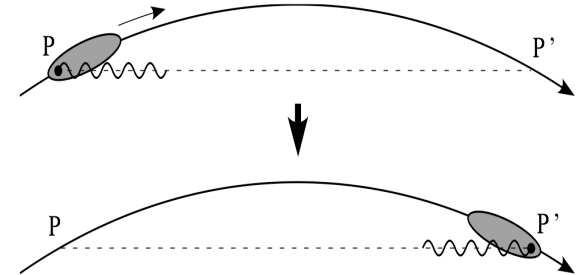
LARGE CANCELLATION

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

$$\vec{\beta} = \frac{\vec{v}}{c}$$

$$\vec{E} = \vec{E}^{ext} + \vec{E}^{self}$$

$$\vec{B} = \vec{B}^{ext} + \vec{B}^{self}$$



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

$$\vec{B}^{self} = \vec{\nabla} \times \vec{A}$$

**NUMERICAL NOISE
DUE TO GRADIENTS**

$$\begin{bmatrix} \phi(\vec{r}, t) \\ \vec{A}(\vec{r}, t) \end{bmatrix} = \int \begin{bmatrix} \rho(\vec{r}', t') \\ \vec{J}(\vec{r}', t') \end{bmatrix} \frac{d\vec{r}'}{|\vec{r} - \vec{r}'|} \left. \vphantom{\int} \right\} \text{retarded potentials}$$

$$t' = t - \frac{|\vec{r} - \vec{r}'|}{c} \quad \text{retarded time}$$

ACCURATE 2D INTEGRATION

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): $f(\vec{r}, \vec{v}, t)$

**Need to track the entire
history of the bunch**

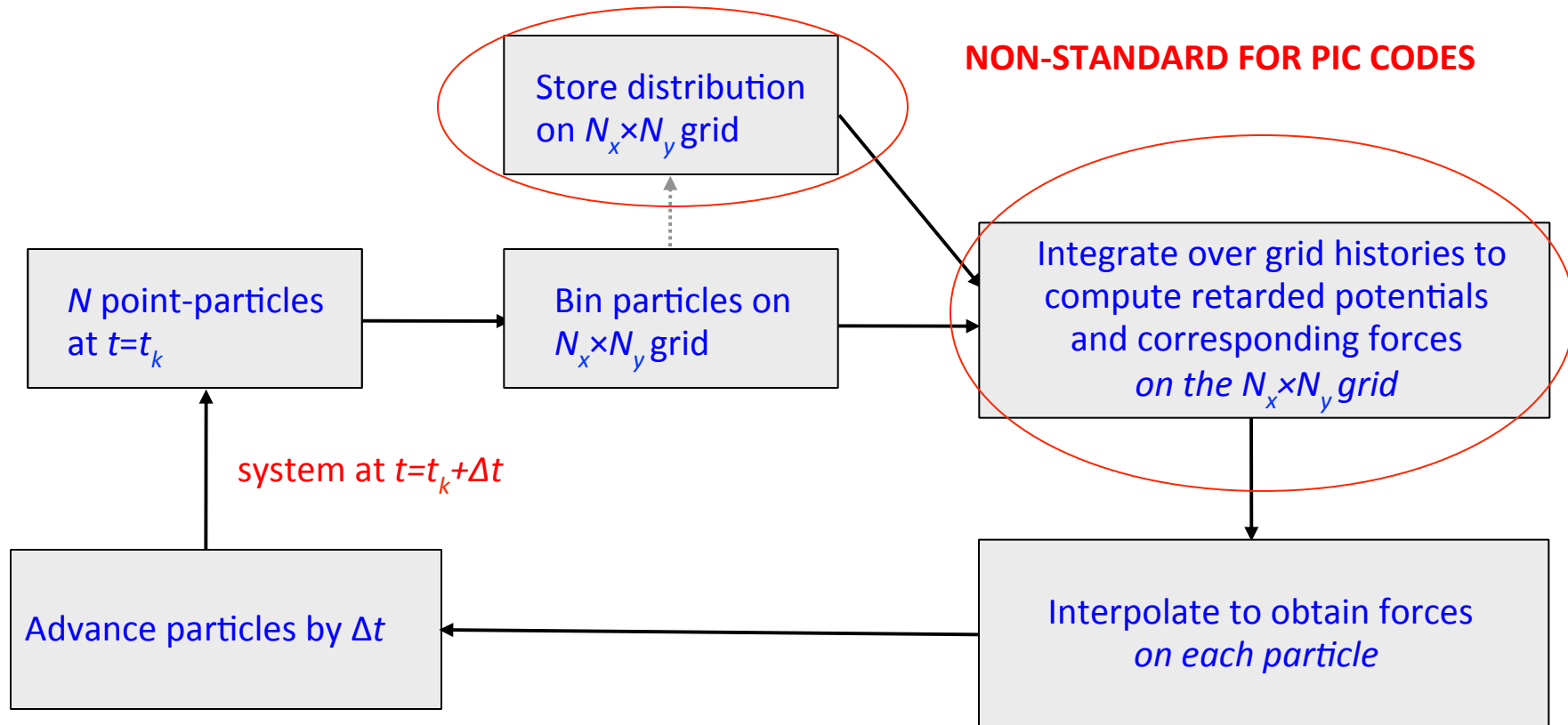
**ENORMOUS COMPUTATIONAL
AND MEMORY LOAD**

CSR: Computational Challenges

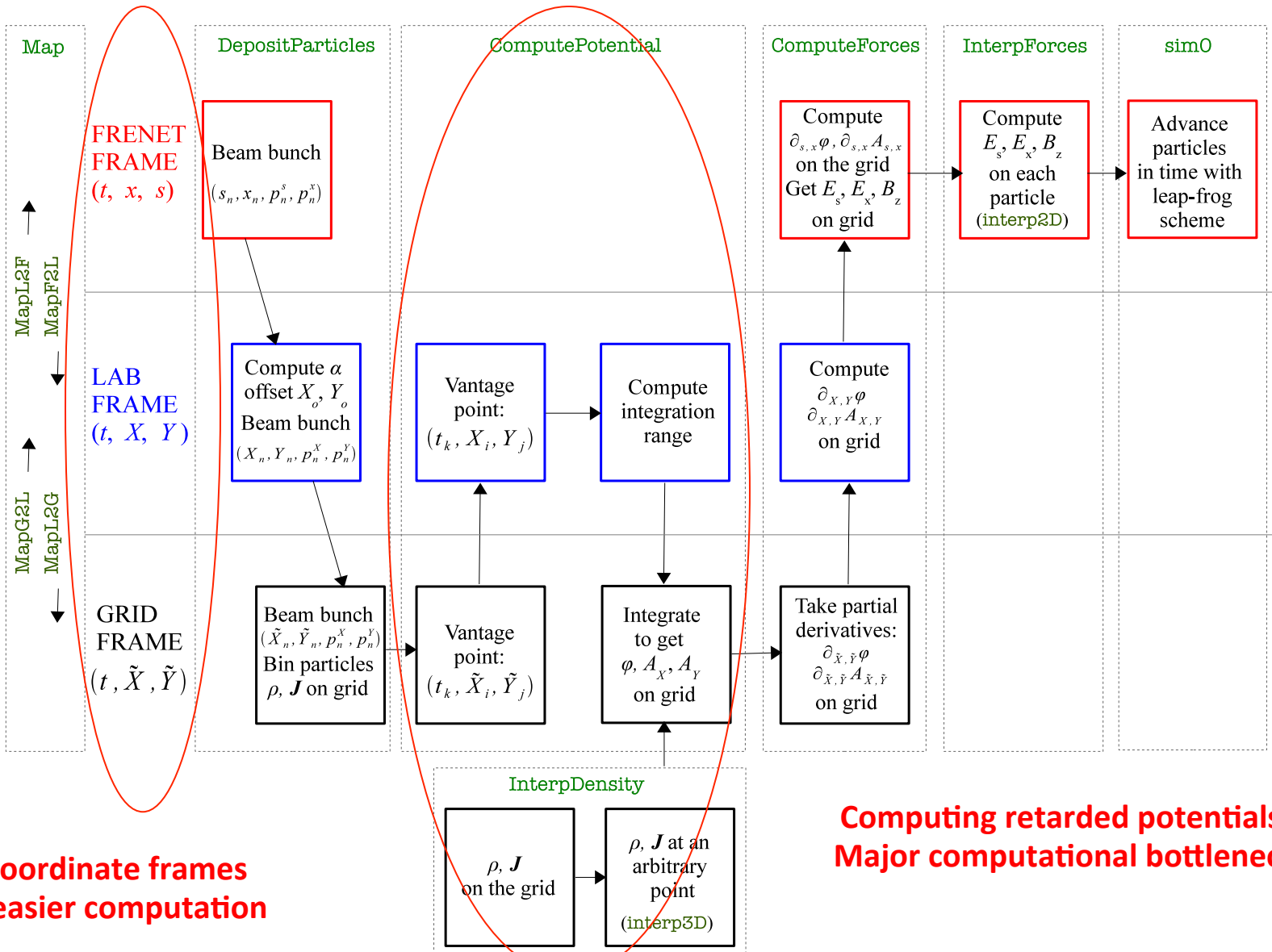
- Our new code solves the main computational challenges associated with the numerical simulation of CSR effects
 - Enormous computational and memory load (storing and integration over beam's history)
Massive parallelization on GPUs
 - Large cancellation in the Lorenz force
Developed high-accuracy, adaptive multidimensional integrators for GPUs
 - Scaling of the beam self-interaction
Particle-in-Cell (PIC) code
 - Self-interaction in PIC codes scales as grid resolution squared (Point-to-point codes: scales as number of macroparticles squared)
 - Numerical noise
Noise removal using wavelets

New Code: The Big Picture

NON-STANDARD FOR PIC CODES



New Code: Computation of CSR Effects

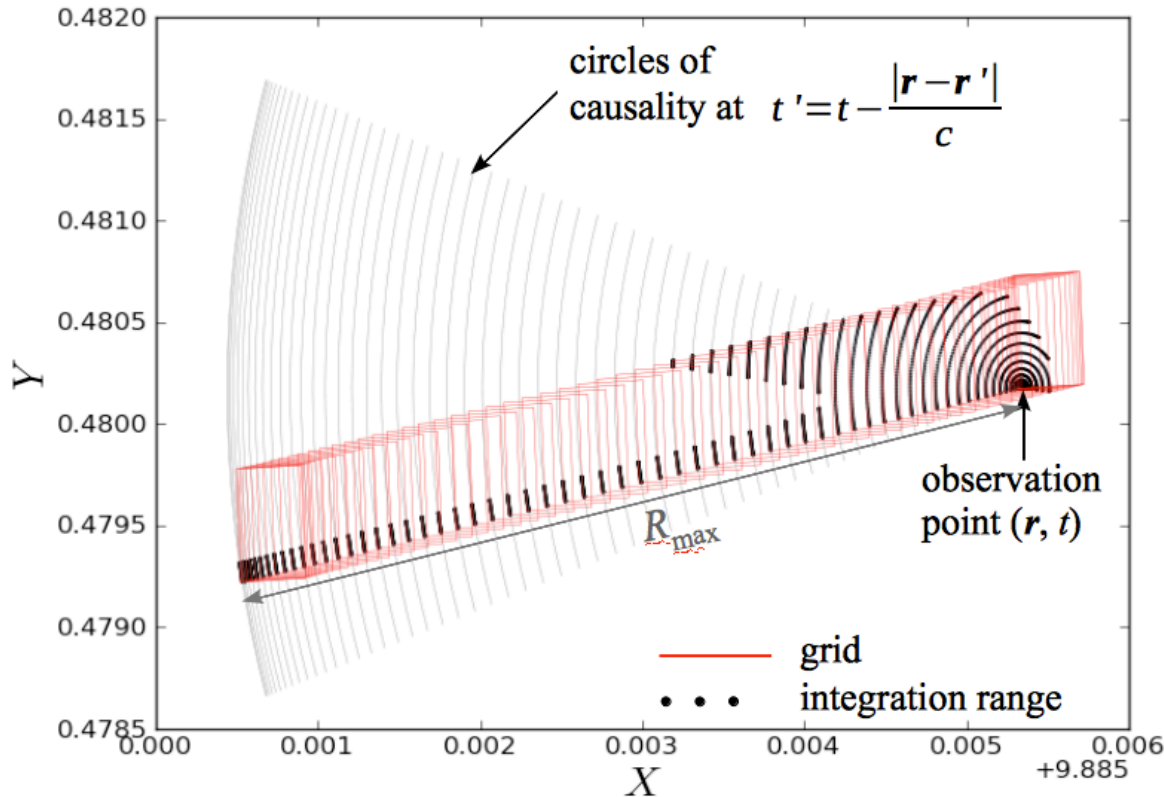


New Code: Computing Retarded Potentials

- Carry out integration over history:

$$\begin{bmatrix} \phi(\vec{r}, t) \\ \vec{A}(\vec{r}, t) \end{bmatrix} = \int \begin{bmatrix} \rho\left(\vec{r}', t - \frac{R'}{c}\right) \\ \vec{J}\left(\vec{r}', t - \frac{R'}{c}\right) \end{bmatrix} \frac{d\vec{r}'}{|\vec{r} - \vec{r}'|} = \sum_{i=1}^{M_{\text{int}}} \int_0^{R_{\text{max}}} \int_{\theta_{\text{min}}^i}^{\theta_{\text{max}}^i} \begin{bmatrix} \rho\left(\vec{r}', t - \frac{R'}{c}\right) \\ \vec{J}\left(\vec{r}', t - \frac{R'}{c}\right) \end{bmatrix} dR' d\theta'.$$

- Determine limits of integration in lab frame:



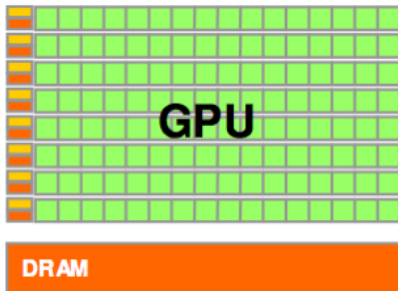
compute R_{max} and $(\vartheta_{\text{min}}^i, \vartheta_{\text{max}}^i)$

For each gridpoint, independently, do the same integration over beam's history

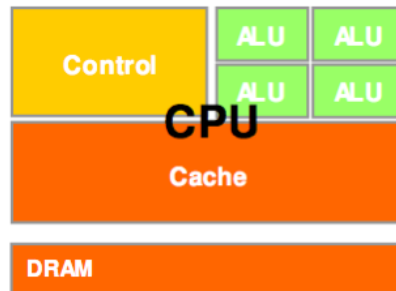
Obvious candidate for parallel computation

Parallel Computation on GPUs

- Parallel computation on GPUs
 - Ideally suited for algorithms with *high arithmetic operation/memory access ratio*
 - **Same Instruction Multiple Data (SIMD)**
 - *Several types of memories* with varying access times (global, shared, registers)
 - Uses extension to existing programming languages to handle new architecture
 - GPUs have many smaller cores (~400-500) designed for parallel execution
 - *Avoid branching and communication* between computational threads

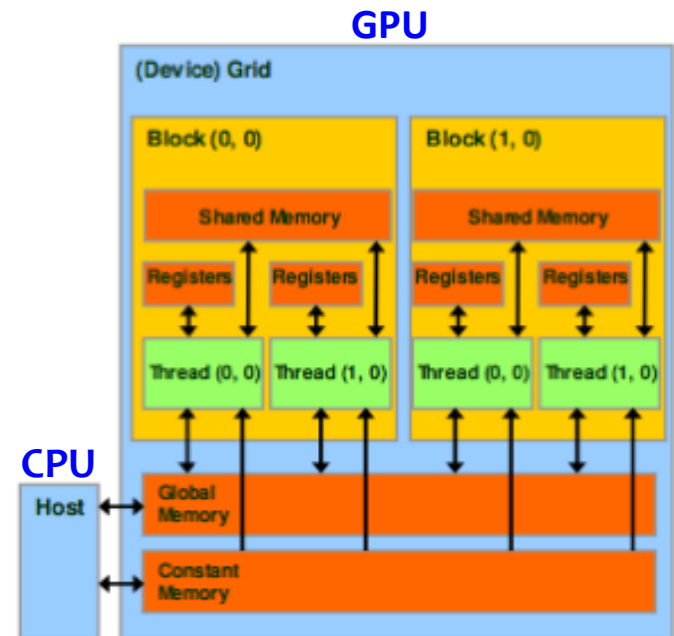


**More space for ALU,
less for cache
and flow control**



**GPU:
grid → blocks → threads**

Example: Tesla M2090 GPU has 512 cores

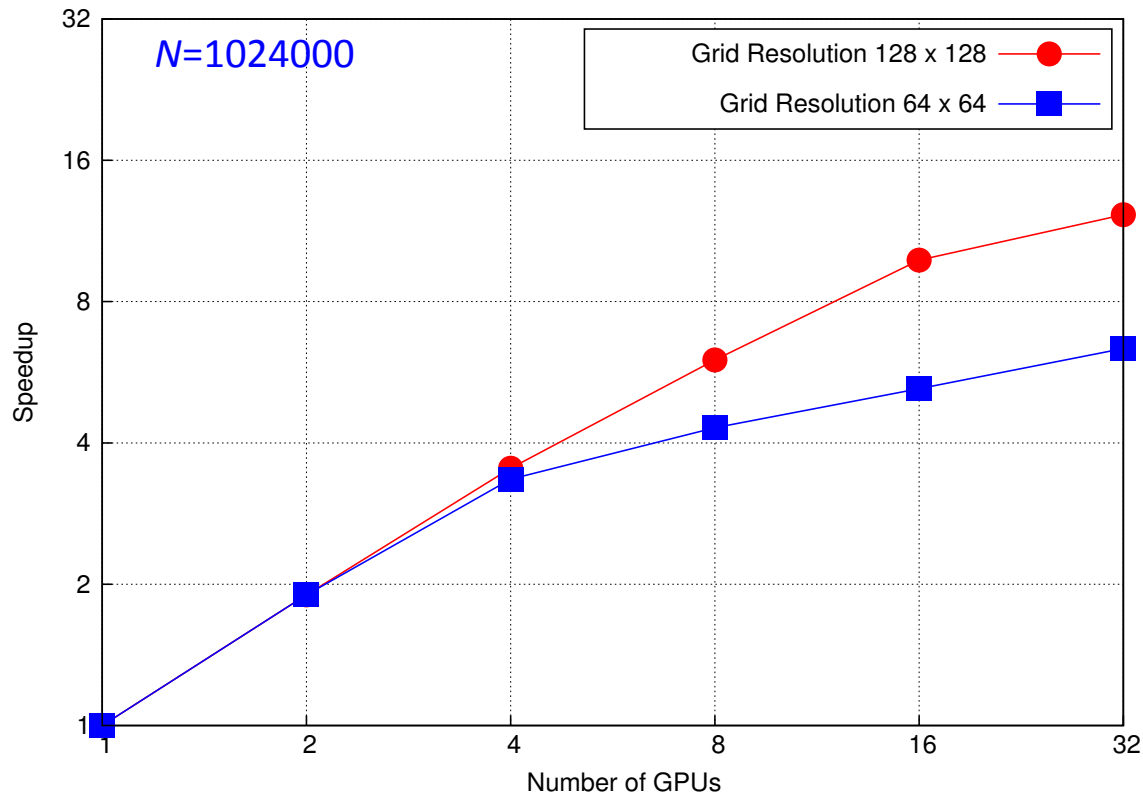


Parallel Computation on GPUs

- Computing the retarded potentials requires integrating over the entire bunch history – *very slow!* **Must parallelize.**
- Integration over a grid is ideally suited for GPUs
 - No need for communication between gridpoints
 - Same *kernel* executed for all (interpolation)
 - Can remove all branches from the algorithm
- We designed a new adaptive multidimensional integration algorithm optimized for GPUs
[\[Arumugam, Godunov, Ranjan, Terzić & Zubair 2013a,b\]](#)
 - NVIDIA's CUDA framework (extension to C++)
 - **About 2 orders of magnitude speedup over a serial implementation**
 - Useful beyond this project

Parallel Computation on GPUs

- The higher the resolution, the larger the fraction of time spent on computing integrals
 - The code scales better on multiple GPUs
 - We expect the scaling at larger resolutions to be nearly linear



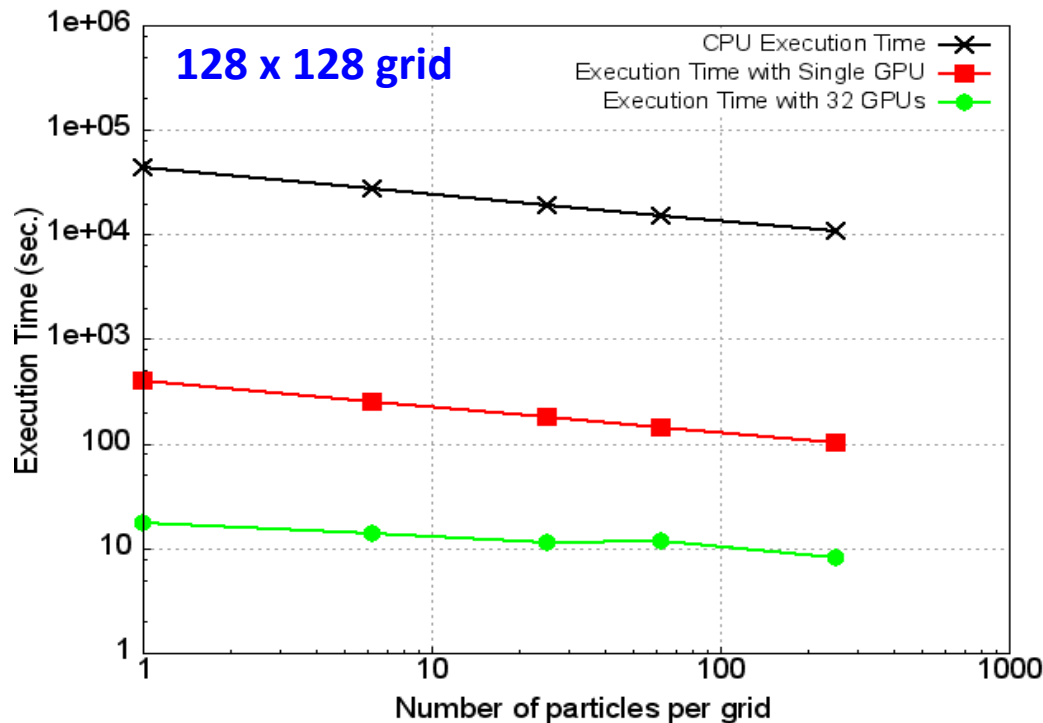
Parallel Computation on GPUs

- The largest resolution tested so far is 128x128
- 1 step of the simulation on a 128x128 grid and 32 GPUs: ~ 10 s
- Execution time *reduces* as the number of macroparticles grows

Number of Particles (N)	Grid Resolution	Sequential Time(sec.)	Single GPU		32 GPUs	
			Time (sec.)	Speedup	Time (sec.)	Speedup
102400	32 × 32	145.52	1.48	98	1.29	113
	64 × 64	1736.24	16.78	104	1.13	1537
	128 × 128	27049.30	256.85	105	13.88	1950
1024000	32 × 32	121.41	1.30	93	1.23	99
	64 × 64	1140.15	11.12	103	1.75	652
	128 × 128	15153.60	144.03	105	11.78	1287
4096000	32 × 32	119.73	1.29	93	1.23	97
	64 × 64	939.96	9.19	102	1.74	540
	128 × 128	10654.00	101.37	105	9.33	1142

Importance of Numerical Noise

- Signal-to-noise ratio in PIC simulations scales as $N_{\text{ppc}}^{1/2}$
[Terzić, Pogorelov & Bohn 2007, PR STAB 10, 034021]
 - Then the numerical noise scales as $N_{\text{ppc}}^{-1/2}$ (N_{ppc} : avg. # of particles per cell)



Execution time for integral evaluation also scales as $N_{\text{ppc}}^{-1/2}$

Less numerical noise = more accurate and faster simulations

[Terzić, Pogorelov & Bohn 2007, PR STAB 10, 034021; Terzić & Bassi 2011, PR STAB 14, 070701]

Wavelet Denoising and Compression

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

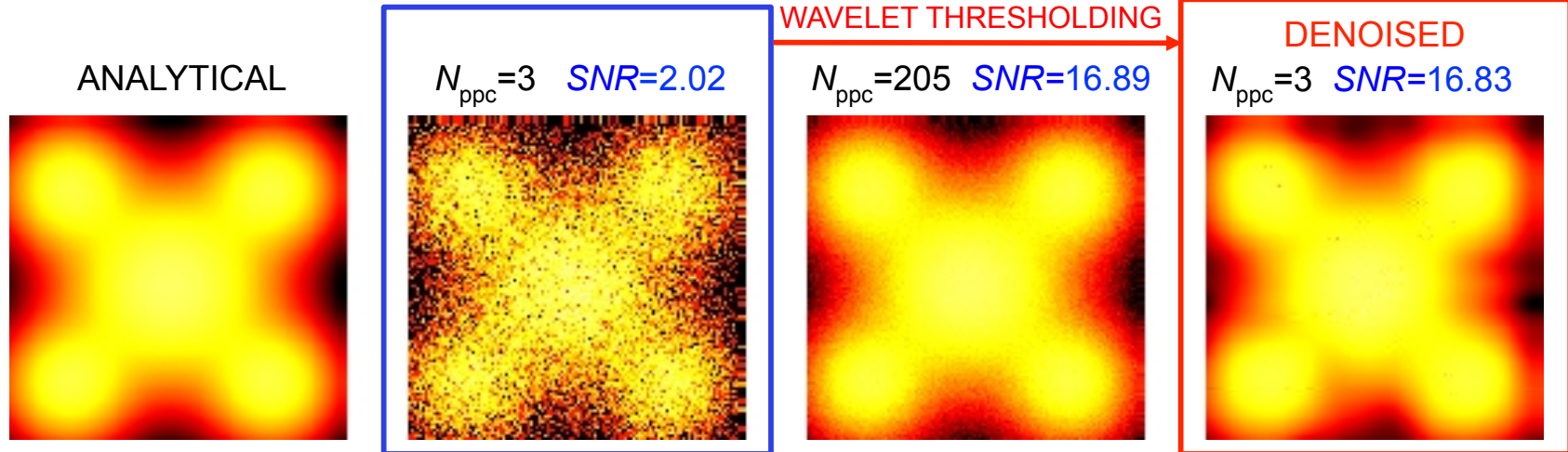
$$SNR = \sqrt{N_{ppc}}$$

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

$$SNR = \sqrt{\frac{\sum_{i=1}^{N_{grid}} \bar{q}_i^2}{\sum_{i=1}^{N_{grid}} (q_i - \bar{q}_i)^2}} \quad \begin{array}{l} \bar{q}_i \text{ exact} \\ q_i \text{ grid} \end{array}$$

2D superimposed Gaussians on 256x256 grid

COMPACT: only 0.12% of coeffs

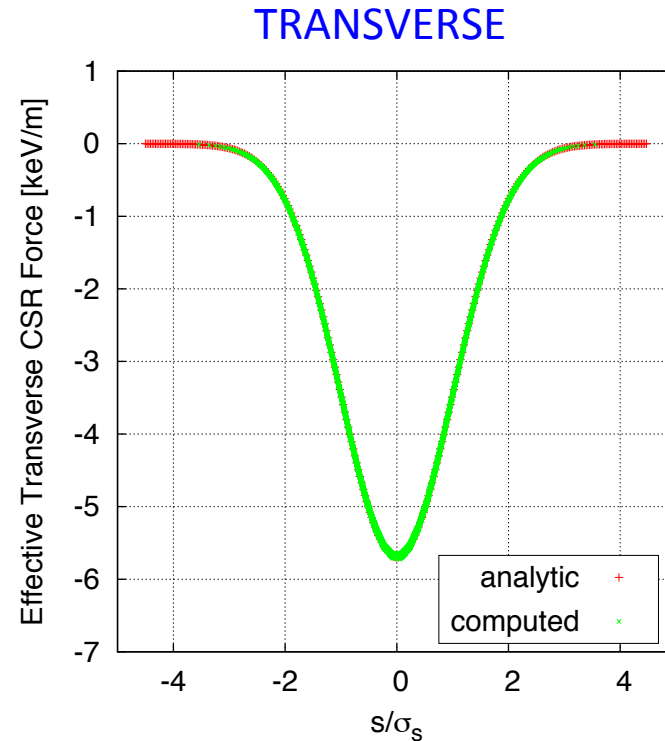
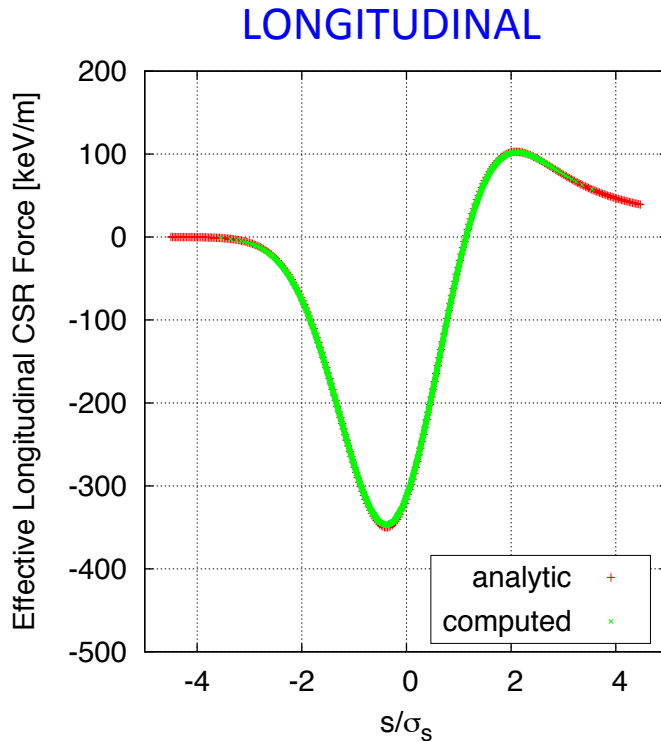


Wavelet denoising yields a representation which is:

- Appreciably more accurate than non-denoised representation
- Sparse (if clever, we can translate this sparsity into computational efficiency)

Benchmarking Against Analytic 1D Results

- Analytic steady state solution available for a rigid line Gaussian bunch [Derbenev & Shiltsev 1996, SLAC-Pub 7181]



$N=512000$
 $N_x=N_y=64$

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

New PIC CSR Code: Efforts Currently Underway

- Compare to 2D semi-analytical results (chirped bunch)
- Compare to other 2D codes (for instance Bassi *et al.* 2009)
- Simulate a test chicane
- Further Afield:
 - Various boundary conditions
 - Shielding
 - Using wavelets to increase efficiency and accuracy
 - Explore the need and feasibility of generalizing the code from 2D to 3D

Summary

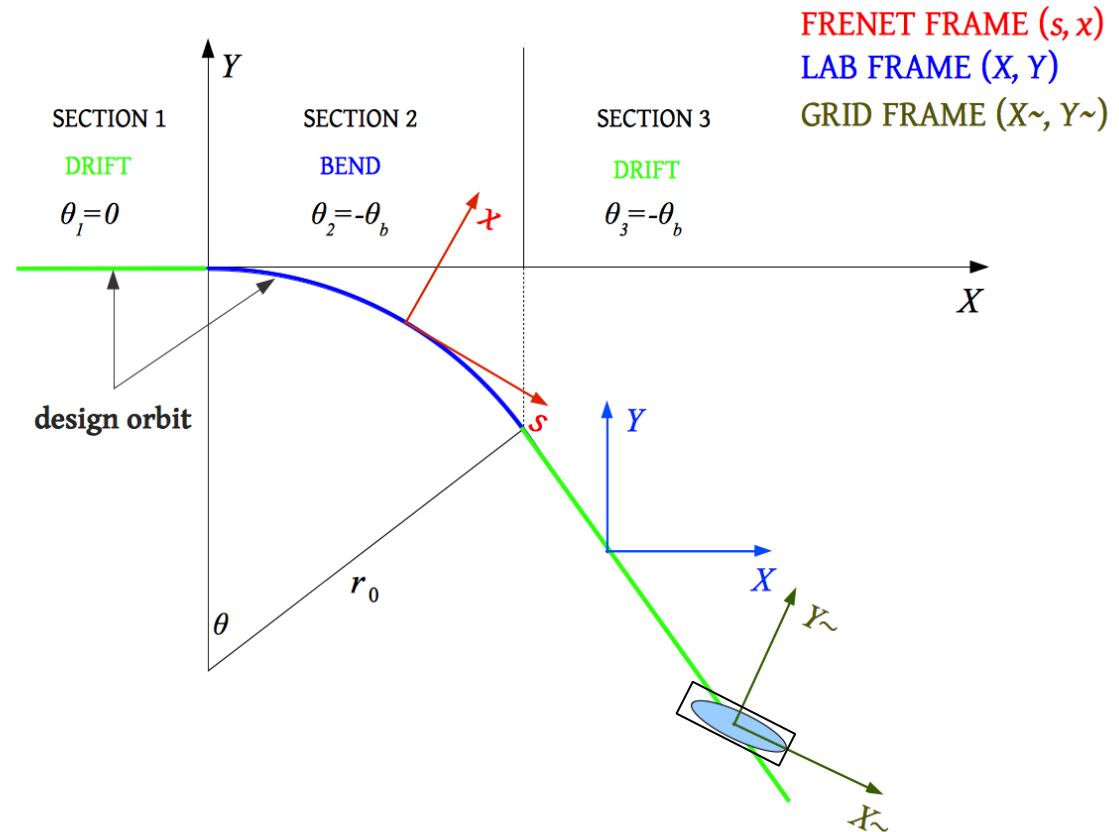
- Presented the new 2D PIC code:
 - Resolves traditional computational difficulties using new computational and mathematical methodologies – GPUs and wavelets
 - Proof of concept: excellent agreement with analytical 1D results
- Outlined outstanding issues that will soon be implemented
- Closing in on our goal
 - Accurate and efficient code which faithfully simulates CSR effects

Backup Slides

New Code: Frames of Reference

- Choosing a correct coordinate system is of crucial importance
- To simplify calculations use 3 frames of reference:

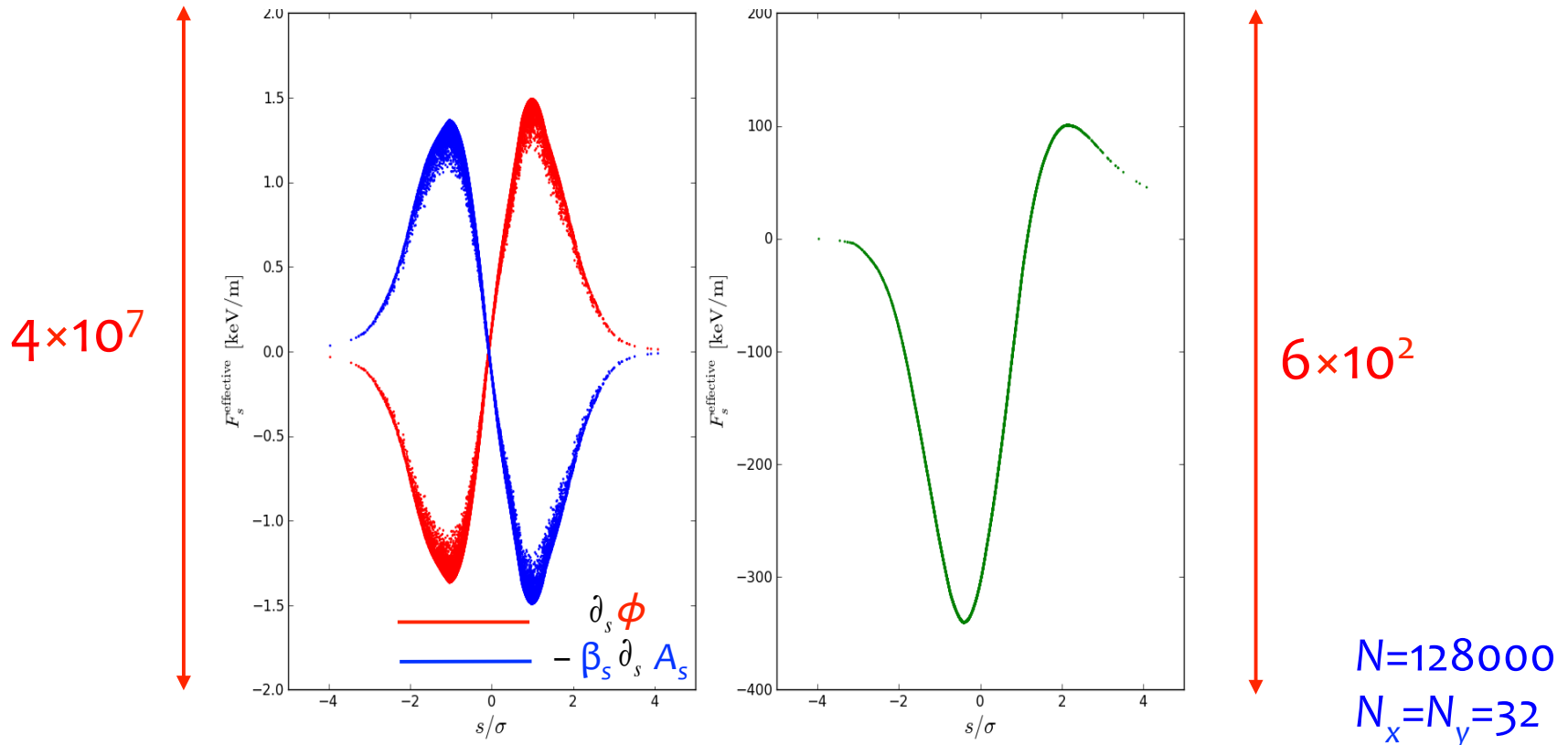
- **Frenet frame (s, x)**
 - s – along design orbit
 - x – deviation normal to direction of motion
 - Particle push
- **Lab frame (X, Y)**
 - Integration range
 - Integration of retarded potentials
- **Grid frame ($X\sim, Y\sim$)**
 - Scaled & rotated lab frame
 - always $[-0.5, 0.5] \times [-0.5, 0.5]$
 - Particle deposition
 - Grid interpolation
 - History of the beam



Large Cancellation in the Lorenz Force

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

Effective Longitudinal Force: $F_s^{\text{eff}} = \partial_s \phi - \beta_s \partial_s A_s$



- High accuracy of the implementation able to track accurately these cancellations over 5 orders of magnitude

Semi-Analytic 2D Results: 1D Model Breaks Down

- Analytic steady state solution is justified for $\kappa = \frac{\sigma_x}{(R\sigma_z^2)^{1/3}} \ll 1$ [Derbenev & Shiltsev 1996]
- Li, Legg, Terzić, Bisognano & Bosch 2011:

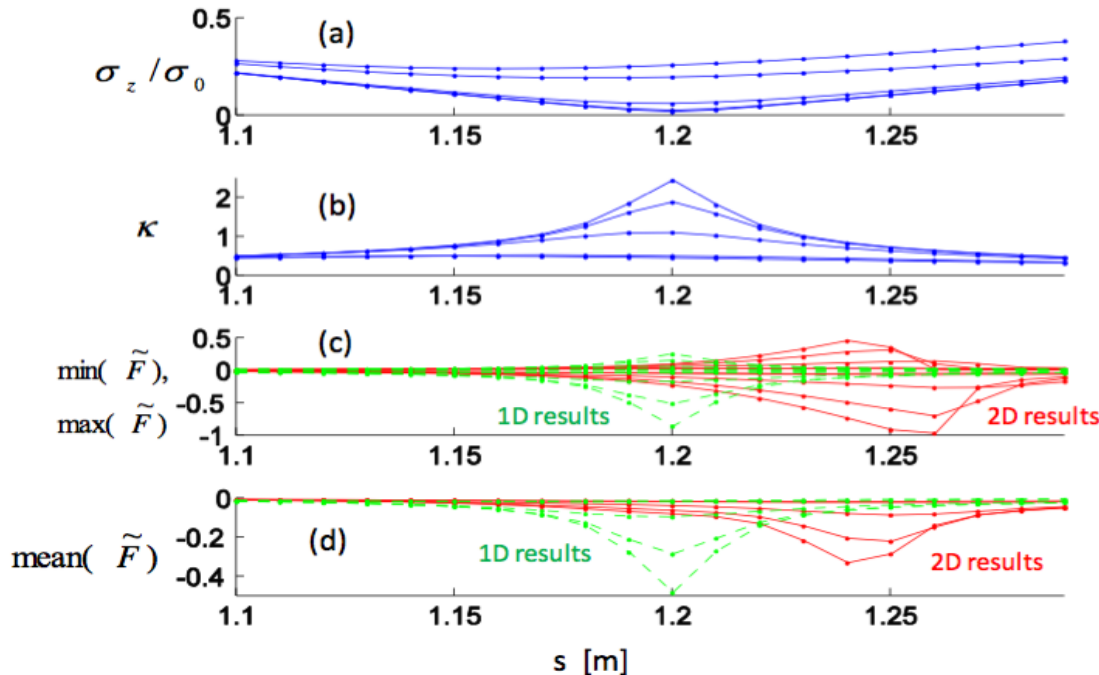
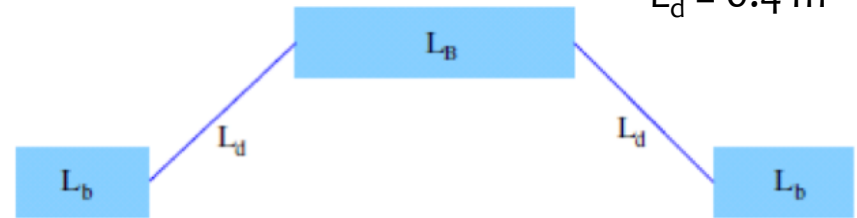
$L_b = 0.3 \text{ m}$
 $L_B = 0.6 \text{ m}$
 $L_d = 0.4 \text{ m}$

Model bunch compressor (chicane)

$E = 70 \text{ MeV}$

$\sigma_{z0} = 0.5 \text{ mm}$

$u = -10.56 \text{ m}^{-1}$ energy chirp



1D & 2D disagree in:

Magnitude of CSR force

Location of maximum force

⇒ 1D CSR model is inadequate

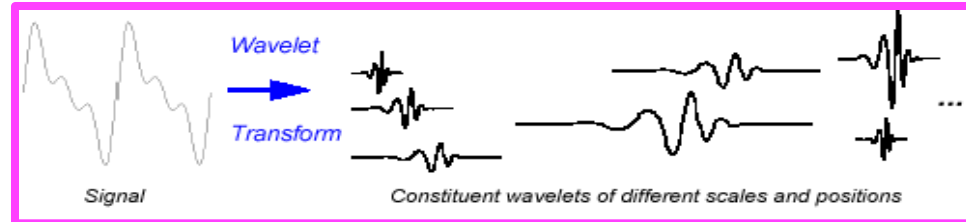
Preliminary simulations show good agreement between 2D semi-analytic results and results obtained with our code

Wavelets

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

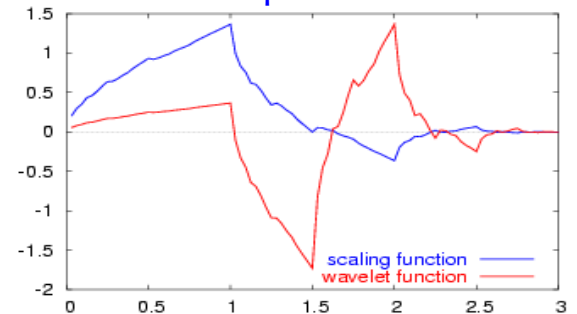
$$\psi_i^k(x) = 2^{k/2} \psi(2^k x - i), \quad k, i \in \mathbb{Z}$$

$$f(x) = s_0^0 \phi_0^0(x) + \sum_k \sum_i d_i^k \psi_i^k(x),$$



- Each new resolution level k is orthogonal to the previous levels
- Compact support*: finite domain over which nonzero
- In order to attain orthogonality of different scales, their shapes are strange
 - Suitable to represent irregularly shaped functions
- For discrete signals (gridded quantities), fast Discrete Wavelet Transform (DWT) is an $O(MN)$ operation, M size of the wavelet filter, N signal size

Daubachies 4th order wavelet



Advantages of Wavelet Formulation

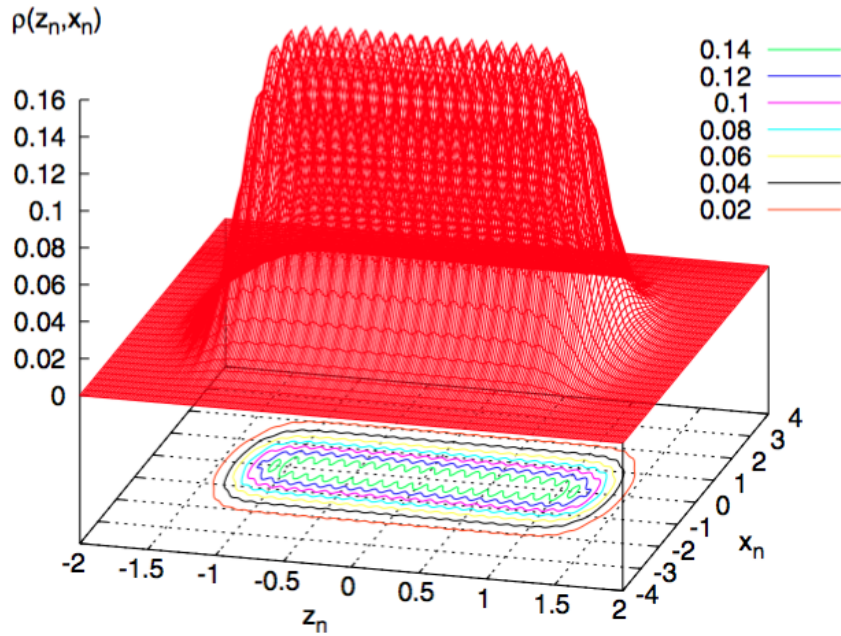
- Wavelet basis functions have compact support \Rightarrow signal localized in space
Wavelet basis functions have increasing resolution levels
 \Rightarrow signal localized in frequency
 \Rightarrow *Simultaneous localization in space and frequency* (FFT only frequency)
- Wavelet basis functions correlate 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*
- Provide a natural setting for numerical noise removal \Rightarrow *Wavelet denoising*
Wavelet thresholding: If $|w_{ij}| < T$, set $w_{ij} = 0$.

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

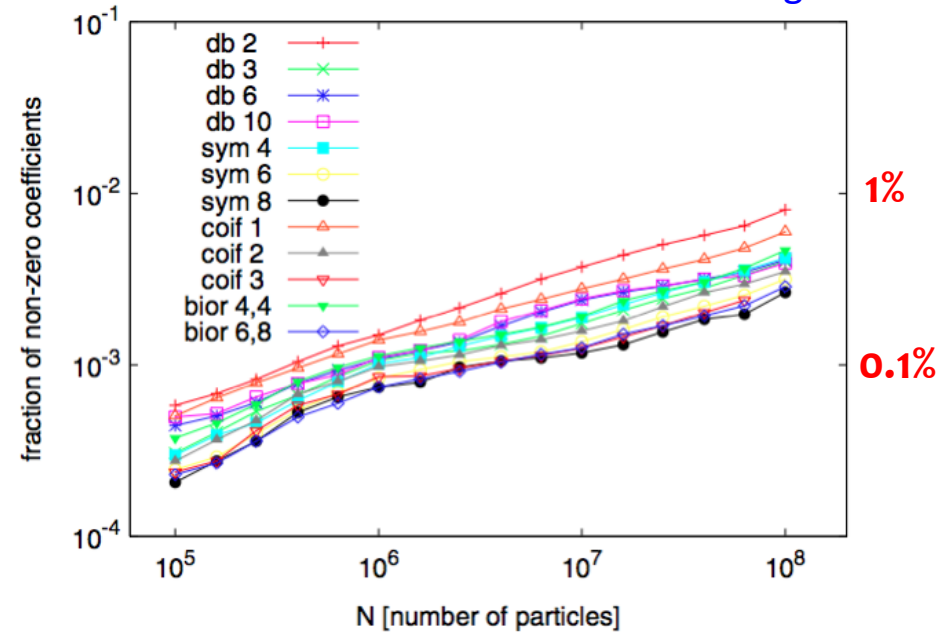
[Terzić & Bassi 2011, PR STAB 14, 070701]

Wavelet Compression

Modulated flat-top particle distribution



Fraction of non-zero coefficients retained after wavelet thresholding



[From Terzić & Bassi 2011, PR STAB 14, 070701]

CSR: Point-to-Point Approach

- **Point-to-Point approach (2D): [Li 1998]**

$$f(\vec{r}, \vec{v}, t) = q \sum_{i=1}^N n_m (\vec{r} - \vec{r}_0^{(i)}(t)) \delta(\vec{v} - \vec{v}_0^{(i)}(t)) \quad \text{DF}$$

$$\rho(\vec{r}, t) = q \sum_{i=1}^N n_m (\vec{r} - \vec{r}_0^{(i)}(t)) \quad \text{Charge density}$$

$$\vec{J}(\vec{r}, t) = q \sum_{i=1}^N \vec{\beta}_0^{(i)}(t) n_m (\vec{r} - \vec{r}_0^{(i)}(t)) \quad \text{Current density}$$

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

- Charge density is sampled with N Gaussian-shaped 2D macroparticles (2D distribution without vertical spread)
- Each macroparticle interacts with each macroparticle throughout history
- **Expensive:** computation of retarded potentials and self fields $\sim O(N^2)$
 - \Rightarrow small number $N \Rightarrow$ *poor spatial resolution*
 - \Rightarrow 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

CSR: Particle-In-Cell Approach

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

$$f(\vec{r}, \vec{v}, t) = q \sum_{i=1}^N \delta(\vec{r} - \vec{r}_0^{(i)}(t)) \delta(\vec{v} - \vec{v}_0^{(i)}(t)) \quad \text{DF (Klimontovich)}$$

$$\rho(\vec{x}_{\vec{k}}, t) = q \sum_{i=1}^N \int_{-h}^h \delta(\vec{x}_{\vec{k}} - \vec{x}_0^{(i)}(t) + \vec{X}) p(\vec{X}) d\vec{X} \quad \text{Charge density}$$

$$\vec{J}(\vec{x}_{\vec{k}}, t) = q \sum_{i=1}^N \vec{\beta}_0^{(i)}(t) \int_{-h}^h \delta(\vec{x}_{\vec{k}} - \vec{x}_0^{(i)}(t) + \vec{X}) p(\vec{X}) d\vec{X} \quad \text{Current density}$$

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

- Two main deposition schemes

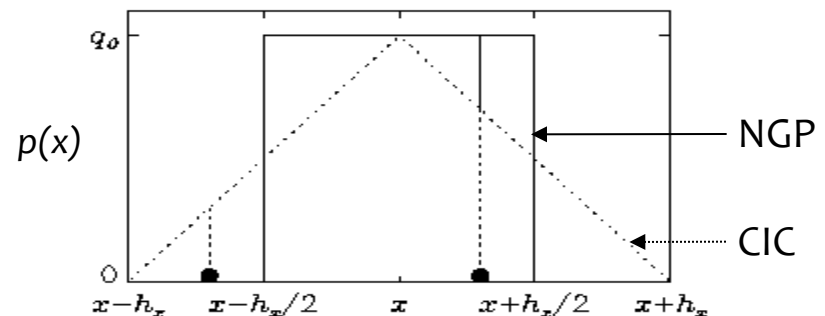
- Nearest Grid Point (NGP)

(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

x – macroparticle location

• – gridpoint location

CSR: 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 $\sim O(N_{grid}^2)$**
 - Particle deposition (yields gridded charge & current densities): $O(N)$ operation
 - Integration over history (yields retarded potentials): $O(N_{grid}^2)$ operation
 - Finite difference (yields self-forces on the grid): $O(N_{grid})$ operation
 - Interpolation (yields self-forces acting on each of N particles): $O(N)$ operation
 - Overall $\sim O(N_{grid}^2) + O(N)$ operations
 - But in realistic simulations: $N_{grid}^2 \gg N$, so the total cost is $\sim O(N_{grid}^2)$
 - Favorable scaling allows for larger N , and reasonable grid resolution
 \Rightarrow Improved spatial resolution
- Fair comparison: P2P with N macroparticles and PIC with $N_{grid} = N$

CSR: P2P Vs. PIC

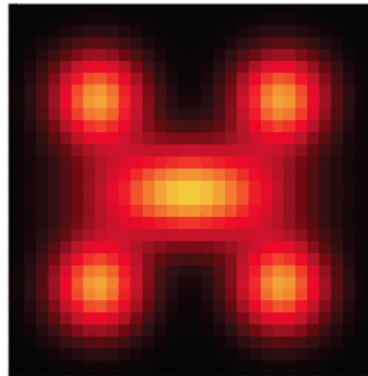
- Difference in spatial resolution: An illustrative example
 - Analytical distribution sampled with
 - $N = N_x N_y$ macroparticles (as in P2P)
 - On a $N_x \times N_y$ grid (as in PIC)
 - 2D grid: $N_x = N_y = 32$

Signal-to-Noise Ratio

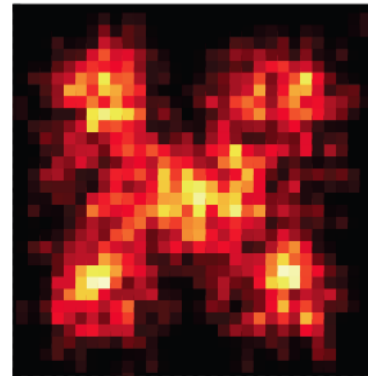
$$SNR = \sqrt{\frac{\sum_{i=1}^{N_{grid}} \bar{q}_i^2}{\sum_{i=1}^{N_{grid}} (q_i - \bar{q}_i)^2}}$$

\bar{q}_i exact
 q_i grid

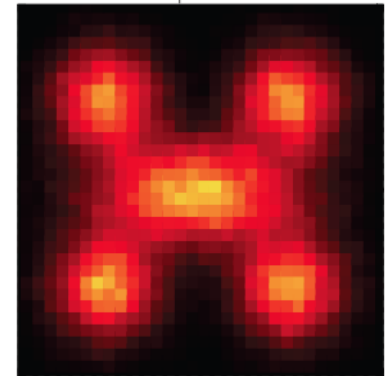
EXACT



P2P $N=32^2$ SNR=2.53



PIC $N=50 \times 32^2$ SNR=13.89



- PIC approach provides superior spatial resolution to P2P approach
- This motivates us to use a PIC code

Outline of the P2P Algorithm

