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:
  \[ \frac{d}{dt}(\gamma m_e \vec{v}) = e \left( \vec{E} + \vec{\beta} \times \vec{B} \right) \]
  \[ \vec{\beta} = \frac{\vec{v}}{c} \]
  \[ \vec{E} = \vec{E}^{\text{ext}} + \vec{E}^{\text{self}} \]
  \[ \vec{B} = \vec{B}^{\text{ext}} + \vec{B}^{\text{self}} \]

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

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

- \( \phi(\vec{r},t) \), \( \vec{A}(\vec{r},t) \) retarded potentials

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

- 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

LARGE CANCELLATION
NUMERICAL NOISE DUE TO GRADIENTS
ACCURATE 2D INTEGRATION

retarded time

\[ t' = t - \frac{|\vec{r} - \vec{r}'|}{c} \]
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

- **$N$ point-particles at $t=t_k$**
  - Advance particles by $\Delta t$
  - System at $t=t_k+\Delta t$

- **Bin particles on $N_x \times N_y$ grid**

- **Store distribution on $N_x \times N_y$ grid**

- **Integrate over grid histories to compute retarded potentials and corresponding forces on the $N_x \times N_y$ grid**

- **Interpolate to obtain forces on each particle**

**NON-STANDARD FOR PIC CODES**
## New Code: Computation of CSR Effects

### Computation retarding potentials:

**Major computational bottleneck**

### 3 coordinate frames for easier computation

<table>
<thead>
<tr>
<th>Map</th>
<th>DepositParticles</th>
<th>ComputePotential</th>
<th>ComputeForces</th>
<th>InterpForces</th>
<th>sim0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRENET FRAME ($t, x, s$)</td>
<td>Beam bunch ($s_n, x_n, p_n^x, p_n^z$)</td>
<td>Compute $\alpha$ offset $X_o, Y_o$</td>
<td>Compute $E_z, E_x, B_z$ on grid</td>
<td>Compute $E_z, E_x, B_z$ on grid</td>
<td>Advance particles in time with leap-frog scheme</td>
</tr>
<tr>
<td>LAB FRAME ($t, X, Y$)</td>
<td>Beam bunch ($X_n, Y_n, p_n^x, p_n^z$)</td>
<td>Vantage point: $(t_h, X_i, Y_j)$</td>
<td>Compute integration range</td>
<td>Compute $\partial x, \partial y, \partial z$ $A_{x,y}$ on grid</td>
<td></td>
</tr>
<tr>
<td>GRID FRAME ($t, \tilde{X}, \tilde{Y}$)</td>
<td>Beam bunch ($\tilde{X}_n, \tilde{Y}_n, p_n^x, p_n^z$) Bin particles $\rho, J$ on grid</td>
<td>Vantage point: $(t_h, \tilde{X}_i, \tilde{Y}_j)$</td>
<td>Integrate to get $\phi, A_{x}, A_{y}$ on grid</td>
<td>Take partial derivatives: $\partial x, \partial y, \partial z$ $A_{x,y}$ on grid</td>
<td></td>
</tr>
</tbody>
</table>

---

6 October 2014

HP CSR Simulations on GPUs

8
New Code: Computing Retarded Potentials

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

- Determine limits of integration in lab frame:

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

  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

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

![Graph showing speedup vs number of GPUs for different grid resolutions.](image)
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 macropraticles grows

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

- Signal-to-noise ratio in PIC simulations scales as $N_{ppc}^{-1/2}$
  
  [Terzić, Pogorelov & Bohn 2007, PR STAB 10, 034021]

  - Then the numerical noise scales as $N_{ppc}^{-1/2}$ ($N_{ppc}$: avg. # of particles per cell)

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)*:

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

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

\[\begin{align*}
SNR &= \sqrt{\frac{N_{grid}}{\sum_{i=1}^{N_{grid}} q_i^2}} \\
&\quad \div \sqrt{\sum_{i=1}^{N_{grid}} (q_i - \bar{q}_i)^2}
\end{align*}\]

\(\bar{q}_i\): exact

\(q_i\): grid

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 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]

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

\[ N = 512000 \]
\[ N_x = N_y = 64 \]
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: Particle-In-Cell

- Grid resolution is specified \textit{a priori} (fixed grid)
  - $N_X$: # of gridpoints in $X$
  - $N_Y$: # of gridpoints in $Y$
  - $N_{\text{grid}} = N_X \times N_Y$ total gridpts
  - Grid: $[X_{ij}, Y_{ij}]_{i=1,N_x, j=1,N_y}$
    - Inclination angle $\alpha$
    - Point-particles deposited on the grid via deposition scheme

- Grid is determined so as to tightly envelope all particles
  Minimizing number of empty cells $\Rightarrow$ optimizing spatial resolution
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** \((X~, Y~)\)
  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:** \( F_s^{\text{eff}} = \partial_s \phi - \beta_s \partial_s A_s \)

\[ N \approx 128000 \quad N_x = N_y = 32 \]

- 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:

  Model bunch compressor (chicane)
  
  \[ E = 70 \text{ MeV} \]
  \[ \sigma_{z0} = 0.5 \text{ mm} \]
  \[ u = -10.56 \text{ m}^{-1} \text{ energy chirp} \]

1D & 2D disagree in:
- Magnitude of CSR force
- Location of maximum force

$\Rightarrow$ 1D CSR model is inadequate

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

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

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

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

$$f(x) = s^0_0 \phi^0_0(x) + \sum_k \sum_i d^k_i \psi^k_i(x),$$

- 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 (DFT) is an $O(MN)$ operation, $M$ size of the wavelet filter, $N$ signal size
Advantages of Wavelet Formulation

- Wavelet basis functions have compact support ⇒ signal localized in space
- Wavelet basis functions have increasing resolution levels
  ⇒ signal localized in frequency
  ⇒ *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)
  ⇒ *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 ⇒ *Solving some PDEs is faster and more accurate*

- Provide a natural setting for numerical noise removal ⇒ *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):

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

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

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

- 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

6 October 2014
HP CSR Simulations on GPUs
CSR: P2P Vs. PIC

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

• Computational cost for PIC: \textbf{Total cost }\sim\textbf{ O}(N_{\text{grid}}^2)
  • Particle deposition (yields gridded charge & current densities): 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 each of }N\text{ particles): O(N) operation
  • Overall \sim\textbf{ O}(N_{\text{grid}}^2)+O(N) operations
    • But in realistic simulations: \textit{N}_{\text{grid}}^2>> N, so the total cost is \sim\textbf{ O}(N_{\text{grid}}^2)
    • Favorable scaling allows for larger }N\text{, and reasonable grid resolution
      \Rightarrow \textit{Improved spatial resolution}

• Fair comparison: P2P with }N\textit{ macroparticles and PIC with }N_{\text{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: \( N_X = N_Y = 32 \)

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

EXACT

P2P \( N=32^2 \) SNR=2.53

PIC \( N=50\times32^2 \) SNR=13.89

• PIC approach provides superior spatial resolution to P2P approach
• This motivates us to use a PIC code
Integrate over particle histories to compute retarded potentials and corresponding forces on each macroparticle.

Advance particles by $\Delta t$.

$N$ macroparticles at $t < t_k$.

$N$ macroparticles at $t = t_k$.

System at $t = t_k + \Delta t$. 

Outline of the P2P Algorithm.