Electron Cooling for an Electron Ion Collider: Computational Methods and Code Development

Herman D. Schaumburg
Béla Erdélyi

DOE-NP Accelerator R&D PI Meeting
November 13, 2018
Main Goals

• Proof of principle demonstration of electron cooling with the first particle-level detailed simulations (Jones Report:: line 3: High, A)

• Develop a high-performance code with these capabilities, and other beam dynamics challenges beyond cooling (Jones Report:: line 4: High, A)

• Applications to electron-ion collider (JLEIC) modeling, design, and optimization (Jones Report:: line 39: High)
## Expenditures and Milestones

<table>
<thead>
<tr>
<th></th>
<th>FY10+FY11</th>
<th>FY12+FY13</th>
<th>FY14+FY15</th>
<th>FY16+FY17</th>
<th>TOTALS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Funds Allocated</strong></td>
<td>0+56</td>
<td>55+52=107</td>
<td>50+54=104</td>
<td>50+50=100</td>
<td>$367K</td>
</tr>
<tr>
<td><strong>Actual Costs to Date</strong></td>
<td>56</td>
<td>107</td>
<td>104</td>
<td>100</td>
<td>$367K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FY17</th>
<th>FY18</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quarter 1</strong></td>
<td>Shared memory parallelization of FMM data structures and integration with parallel FMM</td>
<td>Identification of speed and parallel efficiency bottlenecks in the current version of PHAD</td>
</tr>
<tr>
<td><strong>Quarter 2</strong></td>
<td>Variable order Picard integrator with automatic step size control parallelization</td>
<td>Amelioration of speed and efficiency bottlenecks in PHAD</td>
</tr>
<tr>
<td><strong>Quarter 3</strong></td>
<td>Binned time step implementation in parallel</td>
<td>DC electron cooling initial rate estimations for different ion species (charge states) at low energy</td>
</tr>
<tr>
<td><strong>Quarter 4</strong></td>
<td>Parallel PHAD integration, benchmarking and optimizations</td>
<td>Low energy electron cooling initial rate estimations for different initial distributions and external fields</td>
</tr>
</tbody>
</table>
PHAD Overview

• Accurate at expense of speed.
• No approximations (exact treatment).
• Algorithms employed are the most efficient available.
Main Outcomes

Stability & Performance

- Improved FMM.
- Improved local solver (Simo integrator).
- Memory improvements.
- Avoidance of digit cancelation.

User friendliness

- Parameter tracking.
- Additional timers.
- Error messages.
- Relaunch capability.
- New example.
Parallel Simo Integrator

N=50000

Superlinear speedup
Efficiency of weak scaling

![Efficiency plot](image-url)
Identifying Bottlenecks

- Implemented several different parallelization strategies.

- Strategies have competing benefits and costs.

Load balancing
Parallel computation

Memory usage
Communication
PHAD Parallelization Strategies

\[
\begin{align*}
N_1 & \quad \{ \text{Process 1} \\
& \quad \{ \text{Process 2} \\
& \quad \vdots \\
& \quad \{ \text{Process } p \\
& \quad \vdots \\
& \quad \{ \text{Process } p \\
N_2 & \quad \{ \text{Process 1} \\
& \quad \{ \text{Process 2} \\
& \quad \vdots \\
& \quad \{ \text{Process } p \\
& \quad \vdots \\
& \quad \{ \text{Process } p \\
N_k & \quad \{ \text{Process 1} \\
& \quad \{ \text{Process 2} \\
& \quad \vdots \\
& \quad \{ \text{Process } p \\
\end{align*}
\]
Balanced strategy

\[
\begin{bmatrix}
N_{\sigma(1)} & N_{\sigma(p+1)} & N_{\sigma(2p+1)} & \cdots & \text{Process 1} \\
N_{\sigma(2)} & N_{\sigma(p+2)} & N_{\sigma(2p+2)} & \cdots & \text{Process 2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
N_{\sigma(p)} & N_{\sigma(2p)} & N_{\sigma(3p)} & \cdots & \text{Process } p
\end{bmatrix}
\]

\[
\begin{bmatrix}
|N_1| & 1 \\
|N_2| & 2 \\
\vdots & \vdots \\
|N_k| & k
\end{bmatrix}
\xrightarrow{\text{indexed quicksort}}
\begin{bmatrix}
|N_{\sigma(1)}| & \sigma(1) \\
|N_{\sigma(2)}| & \sigma(2) \\
\vdots & \vdots \\
|N_{\sigma(k)}| & \sigma(k)
\end{bmatrix}
\]

\[
|N_{\sigma(1)}| \leq |N_{\sigma(2)}| \leq \cdots \leq |N_{\sigma(k)}|
\]
Parallel Simo Vs. Unbalanced

"Parallel Simo" scheme – 1 timestep

"Unbalanced" scheme – 5 timesteps

49.9K Particles, FMM order 6, q=60
Subprocedure Breakdown

"Parallel Simo" scheme – 1 timestep

"Unbalanced" scheme – 5 timesteps

49.9K Particles, FMM order 6, q=60
Balanced Vs. Unbalanced

"Balanced" scheme - 5 timesteps

"Unbalanced" scheme - 5 timesteps

49.9K Particles, FMM order 6, q=60
Subprocedure Breakdown

“Balanced” scheme – 5 timesteps

“Unbalanced” scheme – 5 timesteps

49.9K Particles, FMM order 6, q=60
Simo integrator ensures large timesteps leading to nonphysical results are not taken.

Relativistic gamma computed to avoid digit cancelation.

Modification allows smaller Simo order for small PHAD timesteps.

Several memory improvements.
Memory improvements

- FMM memory requirements do not scale with the number of processors.
- Memory for Taylor series in Simo integrator depends on q parameter instead of N.
- Memory improvement in indices storage.
Index Memory Improvements

Bonuses: Communication efficiency. Larger N with same RAM.
Parameter tracking - Simo integrator largest order and minimum timestep used.

Separate timers for read/write and some communications.

Descriptive error messages.

Relaunch capability.

Website improvement (in progress).

New example – beamlets.
• Generates all PHAD input files.
• Positions generated using 2D Gaussian and uniform beam size.
• Momentum generated from Maxwell-Boltzmann distribution.
• Detailed documentation.
Beamlet Setup
Beamlet Simulation
Snapshots
• PHAD is stable.
• Improvements to the FMM have improved its performance.
• Among the three tried strategies for parallelizing PHAD, the unbalanced strategy seems to be the best.
• Overall memory requirements have been significantly reduced.
• Several user friendly capabilities have been added (Relaunch, error messages, additional timers, parameter tracking)
• Further electron cooling simulations have begun.
• Use our code; we are here to help.

https://niu.edu/beamphysicscode/