Precise Simulations of Multi Bunches in High Intensity Cyclotrons

AMR in a particle accelerator framework

Matthias Frey

SNF project 200021_159936

08. March, 2018
Outline

1 Motivation
   - High Intensity Cyclotron
   - Neighbouring Bunch Effects
   - OPAL in a Nutshell (Slides by courtesy of A. Adelmann)
   - AMReX in a Nutshell

2 Challenges

3 Multigrid Poisson Solver

4 Outlook
Motivation
Cyclotron - High Intensity Cyclotron

High Intensity Cyclotron (PSI-Ring):
- particle: proton
- kin. energy: 72 - 590 MeV
- intensity: > 2 mA  
  (record: 2.4 mA)
- power: > 1.2 MW
- RF Cavity freq.: 50.65 MHz

PSI Ring Cyclotron  
(https://www.psi.ch/ltp/facilities)
Motivation

Cyclotron - Halo as Limiting Factor

- Process of halo creation and interactions need to be understood ...
- ... since halo leads to ...
  - particle losses
  - machine activation ($> 4$ MeV)
  - machine damage / shut down
  - machine intensity limitation
Motivation
Cyclotron - Beam Profile Measurement

- radial probes measure beam profiles
- turn separation decreases with energy
Motivation
Radial Profile Measurement (Thanks to J. Snuverink)
Motivation
Radial Profile Measurement (Thanks to J. Snuverink)
Motivation
Radial Profile Measurement (Thanks to J. Snuverink)
Motivation
Radial Profile Measurement (Thanks to J. Snuverink)
Motivation

- Neighbouring bunch effects occur in case of small turn separation for high intensity cyclotrons \(^1\)

Motivation

- Neighbouring bunch effects occur for small turn separation for high intensity cyclotrons.¹

Motivation

- **Requirements:**
  - Solving large-scale $N$-body problems of $O(10^9 \ldots 10^{10})$ particles coupled with Maxwell's equations
  - Particle-in-Cell with extremely fine mesh of $O(10^8 \ldots 10^9)$ grid points

- **Bottlenecks:**
  - Waste of memory and resolution in regions of void

- **Solution:**
  - Block-structured adaptive mesh-refinement
Tools

- **Software:**
  - OPAL (for physics)
  - AMReX (for grids, formerly: BoxLib) ([https://ccse.lbl.gov/AMReX](https://ccse.lbl.gov/AMReX))
  - Trilinos (Amesos2, Belos, Ifpack2, MueLu, Tpetra) ([https://trilinos.org/](https://trilinos.org/))

- **Hardware:**
  - Piz Daint at CSCS

Taken from [https://www.cscs.ch/computers/piz-daint/](https://www.cscs.ch/computers/piz-daint/)
## Tools - Top500 List November 2017

[https://www.top500.org/list/2017/11/](https://www.top500.org/list/2017/11/)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>System</th>
<th>Cores</th>
<th>Rmax (TFlop/s)</th>
<th>Rpeak (TFlop/s)</th>
<th>Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Supercomputing Center in Wuxi, China</td>
<td>Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRPC</td>
<td>10,649,600</td>
<td>93,014.6</td>
<td>125,435.9</td>
<td>15,371</td>
</tr>
<tr>
<td>2</td>
<td>National Super Computer Center in Guangzhou, China</td>
<td>Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT</td>
<td>3,120,000</td>
<td>33,862.7</td>
<td>54,902.4</td>
<td>17,808</td>
</tr>
<tr>
<td>3</td>
<td>Swiss National Supercomputing Centre (CSCS), Switzerland</td>
<td>Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect, NVIDIA Tesla P100 Cray Inc.</td>
<td>361,760</td>
<td>19,590.0</td>
<td>25,326.3</td>
<td>2,272</td>
</tr>
<tr>
<td>4</td>
<td>Japan Agency for Marine-Earth Science and Technology, Japan</td>
<td>Gyoukou - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz ExaScaler</td>
<td>19,860,000</td>
<td>19,135.8</td>
<td>28,192.0</td>
<td>1,350</td>
</tr>
</tbody>
</table>
OPAL - Object Oriented Parallel Accelerator Library

https://gitlab.psi.ch/OPAL/src
OPAL V.2.0 in a Nutshell I

Slides by courtesy of A. Adelmann

OPAL is an open-source tool for charged-particle optics in large accelerator structures and beam lines including 3D space charge, particle matter interaction, partial GPU support and multi-objective optimisation.

- OPAL is built from the ground up as a parallel application exemplifying the fact that HPC (High Performance Computing) is the third leg of science, complementing theory and the experiment
- OPAL runs on your laptop as well as on the largest HPC clusters
- OPAL uses the MAD language with extensions
- OPAL is written in C++, uses design patterns, easy to extend
- Webpage: https://gitlab.psi.ch/OPAL/src.wikis/home
- the OPAL Discussion Forum: https://lists.web.psi.ch/mailman/listinfo/opal
- \(\mathcal{O}(40)\) users
Vlasov-Poisson Equation
Addressing the multi scale challenge (Slides by courtesy of A. Adelmann)

When neglecting collisions, and taking advantage of the electrostatic approximation, the Vlasov-Poisson equation describes the (time) evolution of the phase space \( f(x, v; t) > 0 \) when considering electromagnetic interaction with charged particles.

\[
\frac{df}{dt} = \frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{q}{m} (E(x, t) + v \times B(x, t)) \cdot \nabla_v f = 0. \tag{1}
\]

Solving with ES-PIC
- Hockney and Eastwood, \( h_x(t), h_y(t), h_z(t), M = M_x \times M_y \times M_z \)
- SAAMG-PCG solver with geometry
- change \( M \) during simulation (many different field solver instances)
- change \( \Delta t \) adaptively
- modern computational architectures
Software Architecture

MPI based + HW accelerators + Optimiser (Slides by courtesy of A. Adelmann)
AMReX in a Nutshell

https://amrex-codes.github.io/

AMReX

A software framework for massively parallel, block-structured adaptive mesh refinement (AMR) applications

Enable Your Science With AMReX
AMReX in a Nutshell

https://amrex-codes.github.io/

- Massively parallel block-structured adaptive mesh refinement library
- Support for cell-centered, face-centered, edge-centered and nodal data
- MPI and hybrid MPI / OpenMP parallelization
AMReX in a Nutshell

Tagging - Mark a Cell for Refinement

- Coordinate space discretized by grid
**AMReX in a Nutshell**

Tagging - Mark a Cell for Refinement

- Coordinate space discretized by grid
- Mark cell for refinement according to some criteria:
  - charge density per cell
  - potential gradient
  - potential magnitude
  - etc.
AMReX in a Nutshell
Tagging - Mark a Cell for Refinement

- Generate level 1
AMReX in a Nutshell

Tagging - Mark a Cell for Refinement

- Generate level 1
- Mark cell for refinement according to some criteria:
  - charge density per cell
  - potential gradient
  - potential magnitude
  - etc.
AMReX in a Nutshell

Tagging - Mark a Cell for Refinement

- Generate level 2
AMReX in a Nutshell
Tagging - Mark a Cell for Refinement

- Generate level 2
- Maximum level reached (user-defined)
Programming Challenges

- Computation is optimized for
  - grids (AMReX)
  - particles (OPAL)
- Different parallelization (i.e., MPI) strategy
- AMReX and OPAL developed independently
  - Implementation of a lightweight interface (AMReX as black box)
- Runtime selection of tracking model in OPAL: AMR vs. non-AMR
  - Additional particle layout manager in OPAL
Multigrid Poisson Solver

- Solve Poisson’s equation

\[ \Delta \phi(x, y, z) = -\frac{\rho}{\varepsilon_0} \]

\[ \phi \left( r = \sqrt{x^2 + y^2 + z^2} \right) = \mathcal{O} \left( \frac{1}{r} \right) \quad \text{for} \quad r \to \infty, \]

with charge density \( \rho \), vacuum permittivity \( \varepsilon_0 \) and potential \( \phi \).

- Implementation of an AMR multigrid solver based on Cartwright’s algorithm\(^2\) with Open BC.

---

Multigrid Poisson Solver

V-cycle (residual correction formulation) Algorithm

1: function \textsc{relax}(l)
2: \hspace{1em} if \( l = l^{max} \) then
3: \hspace{2em}  \( r^l \leftarrow \rho^l - \mathcal{L}\phi^l \)
4: \hspace{2em} (\( \mathcal{L} \): Laplace operator)
5: \hspace{1em} end if
6: \hspace{1em} if \( l > 0 \) then
7: \hspace{2em}  \( \phi^{l,\text{save}} \leftarrow \phi^l \)
8: \hspace{2em}  \( e^{l-1} \leftarrow 0 \)
9: \hspace{2em}  \textsc{smooth}(e^l, r^l)
10: \hspace{2em}  \phi^l \leftarrow \phi^l + e^l
11: \hspace{2em}  r^{l-1} \leftarrow \mathcal{R}r^l \triangleright \text{restrict residual}
12: \hspace{2em}  \textsc{relax}(l - 1)
13: \hspace{2em}  e^l \leftarrow \mathcal{P}e^l \triangleright \text{prolongate error}
14: \hspace{2em}  \delta e^l \leftarrow 0
15: \hspace{2em}  \textsc{smooth}(\delta e^l, r^l)
16: \hspace{2em}  e^l \leftarrow e^l + \delta e^l
17: \hspace{2em}  \phi^l \leftarrow \phi^{l,\text{save}} + e^l
18: \hspace{1em} \text{else}
19: \hspace{2em}  \text{solve } Ae^0 = r^0
20: \hspace{2em}  \phi^0 \leftarrow \phi^0 + e^0
21: \hspace{1em} \text{end if}
22: \text{end function}
Multigrid Poisson Solver
Third-Party Library

- **Matrix/vector:** MPI-\(X\) (\(X =\) OpenMP, GPU, etc.) support via back end *Kokkos*\(^3\)
- **Post- and Pre-smoothing:** Gauss-Seidel, Jacobi (*Ifpack2*\(^4\))
- **Bottom solver:** Smoothed Aggregation Multigrid (*MueLu*\(^5,6\))

---


Test Setting

- 3 Gaussian sources (each 16'777'216 particles)
- base grid: $288^3$
- max. grid: 24
  ($\rightarrow$ max. 1728 cores on base level)
- #level of refinement: 2
- 10 solves (move each particle randomly within $[-0.001, 0.001]$ after every solve)
Strong Scaling
72 to 1728 cores (no hyperthreading) on Piz Daint
Example Plot: 10 bunches
Outlook

**In progress ...**

- first PSI-Ring tests were performed
- test solver with OpenMP / CUDA enabled
- submitting proposal to CSCS (deadline: 11. May) (request for node hours for large scale simulations)

**Next steps ...**

- profile peak matching (measurement vs. simulation)
- uncertainty quantification (beam and solver)
- large scale simulations
Thanks for your attention!