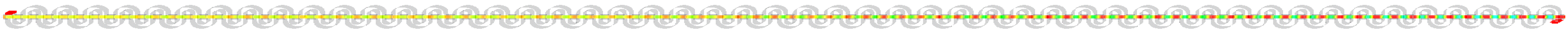


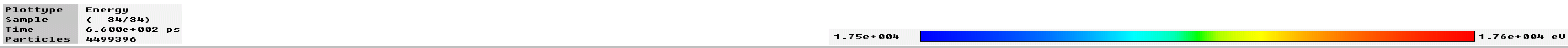
Particle-in-Cell solver on Multi-GPU platform

Martina Sofranac[1], Frank Hamme[1], Thomas Weiland[2]

[1]CST AG, Bad Nauheimer Str. 19, 64289 Darmstadt, Germany

[2] TEMF, TU Darmstadt, Schlossgartenstrasse 8, 64289 Darmstadt, Germany

CST



Motivation

Designing of particle accelerator structures is a complex and costly process. Thus, good simulation tools are necessary for reducing development costs. Particle-in-Cell (PIC) simulations tend to be quite time consuming and therefore the motivation is to take advantage of modern architectures in order to reduce simulation time. CST PARTICLE STUDIO® already supports Particle-in-Cell solver on a Single-GPU architechture. The Multi-GPU version would provide finer meshes and increased amount of particles.

Introduction

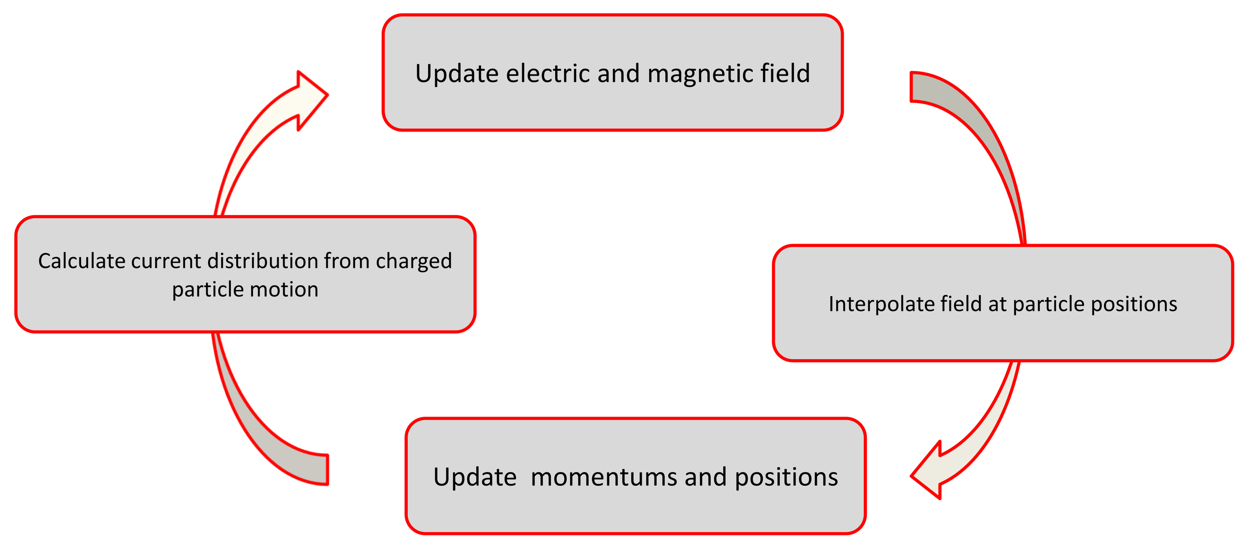
PIC is a common technique for simulation of charged particle dynamics in accelerator structures. In the implementation, the Finite Integration Technique (FIT) is applied to Maxwell equations in their integral form. The discretized matrix presentation of Maxwell equations is:

$$\mathbf{C}\widehat{\mathbf{e}} = -\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\widehat{\mathbf{b}}},$$
$$\widetilde{\mathbf{C}}\widehat{\mathbf{h}} = \widehat{\widehat{\mathbf{j}}} + \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\widehat{\mathbf{d}}},$$
$$\mathbf{S}\widehat{\widehat{\mathbf{b}}} = 0, \quad \widetilde{\mathbf{S}}\widehat{\widehat{\mathbf{d}}} = q$$

In the solver, particles are coupled with electromagnetic fields. Lorentz forces are calculated and particles are pushed by integrating the equations of motion:

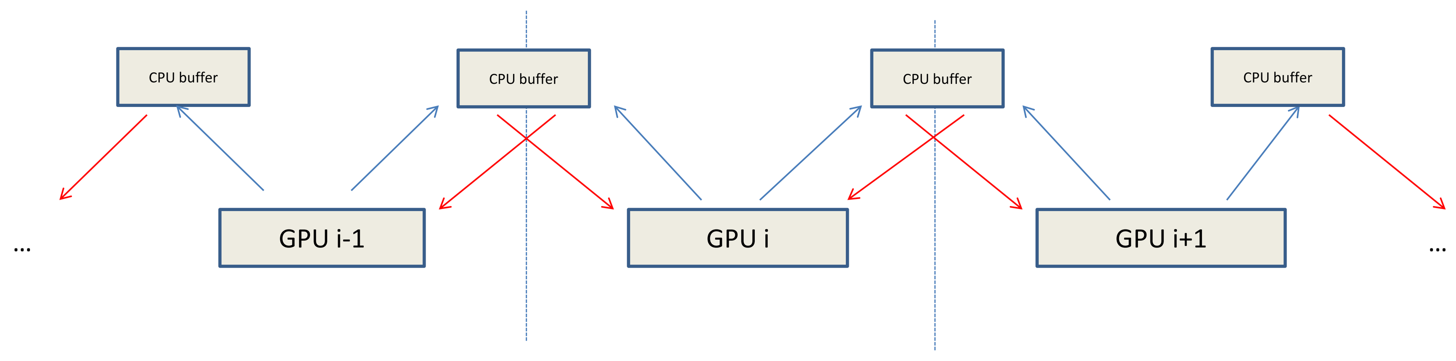
$$m\ddot{\vec{r}} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

General Agorithm Scheme of PIC

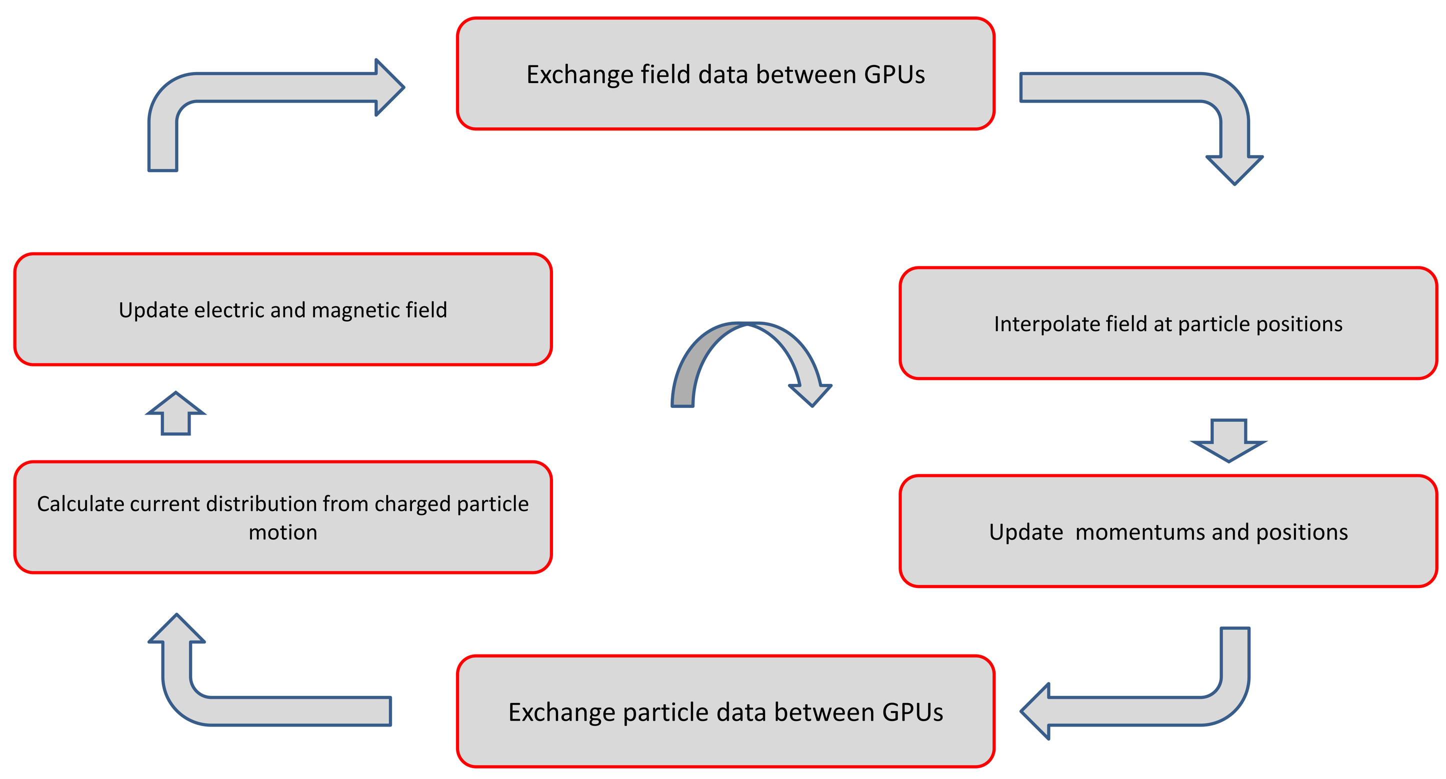


Memory Management

The GPU is a throughput-oriented device. In comparison with CPU the GPU cache is very small and latency is an issue. DRAM is slow access memory and data should be organized properly. If there is no memory continuity then DRAM bursting causes the loading of unnecessary data.
The computational domain is divided such that cutting planes are along direction in which CPU-GPU transferring of data is minimal. Each GPU handles one sub-domain and they exchange data via buffers on CPU.

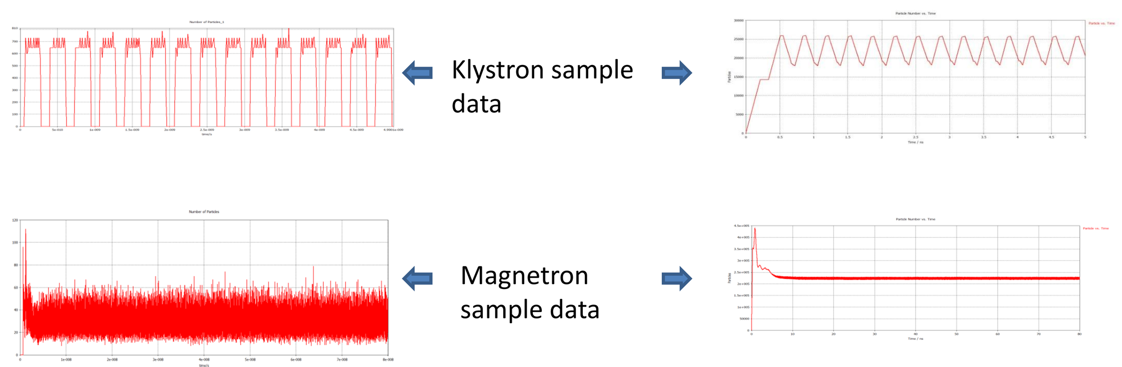


Domain Decomposition Algorithm Scheme for PIC on Multi-GPU



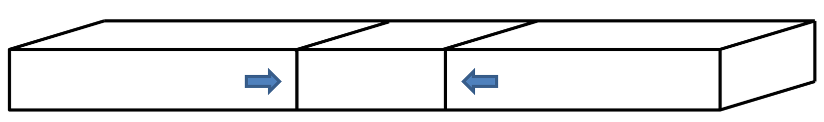
Pinned vs. Non-Pinned

The solution has to be general and provide simulations of very different accelerator structures. One of the challenges is the design and size of buffers.



Load Balancing Scheme

A weighting function is needed as a measure of load distribution. It will provide decomposition that results in an equally balanced load between GPUs. In the scheme cutting planes are not fixed anymore and the issue is to handle resizing of sub-domains.



Final goal of the project is to compare memory usage and performance of both schemes when used for simulation of different devices. The single node solution will work for up to 8 GPUs and will be the basis for implementation of message passing interface(MPI).

References:

[1] Burau, H. at al. „PIConGPU: A Fully Relativistic Particle-in-Cell Code for a GPU Cluster“, IEEE Transactions on Plasma Sciences, Volume:38, Issue: 10, 2010

[2] Felix Wolfheimer, Erion Gjonaj, Thomas Weiland, “A parallel 3D particle-in-cell code with dynamic load balancing”, ICAP 2004, p. 202 - 204

OPAC | EUROPEAN COMMISSION | MARIE CURIE ACTIONS | SEVENTH FRAMEWORK PROGRAMME | TEMF | TECHNISCHE UNIVERSITÄT DARMSTADT

Project is funded by the European Commission under contract PITN-GA-2011-289485