GPU TECHNOLOGY CONFERENCE
MARCH 24-27, 2014 | SAN JOSE, CA

CISAS G. COLOMBO

# Accelerating Particle-Mesh Interaction for Particle-in-Cell Simulation


A. Madonna

Graduate Student, University of Padova

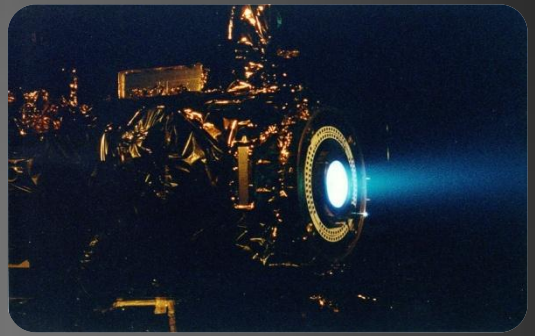
Supervisors:

D. Pavarin
Associate Professor
University of Padova

M. Manente
Senior R&D Engineer
Hit09 S.r.l.

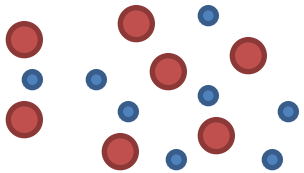
# Electric Spacecraft Propulsion

- Low absolute thrust (< 0.1N)
- High Specific Impulse ($I_{sp}$ > 1000s)
- Long duration
- Used in deep space missions or in Attitude Control Systems



# Plasma Physics 101

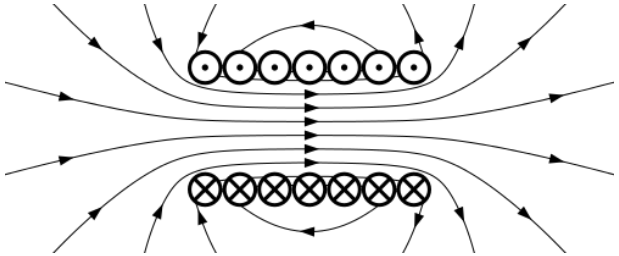
Charged Particles



Equation of Motion + Lorentz Eq.

$$\begin{cases} \dot{\mathbf{x}}_i &= \mathbf{v}_i \\ \dot{\mathbf{p}}_i &= q_i \left( \mathbf{E}(\mathbf{x}_i) + \dfrac{\mathbf{v}_i \times \mathbf{B}(\mathbf{x}_i)}{c} \right) \end{cases}$$

EM Field

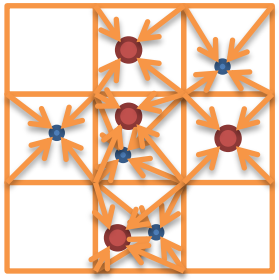


Maxwell's Equations

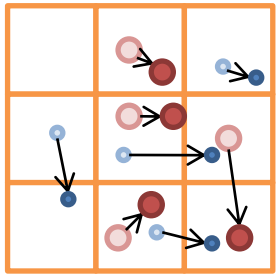
$$\begin{cases} \nabla \cdot \mathbf{B} &= 0 \\ \nabla \cdot \mathbf{E} &= \dfrac{\rho}{\varepsilon_0} \\ \nabla \times \mathbf{E} &= -\dfrac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{B} &= \mu_0 \left( \mathbf{J} + \varepsilon_0 \dfrac{\partial \mathbf{E}}{\partial t} \right) \end{cases}$$

# Particle-in-Cell Method

1. Interpolation

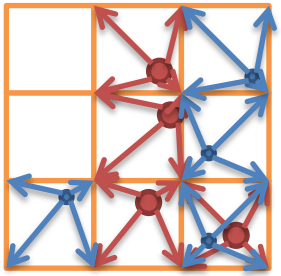


2. Particle Push

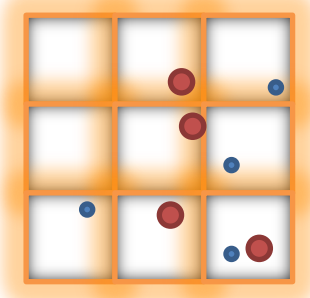


Δt

3. Deposition

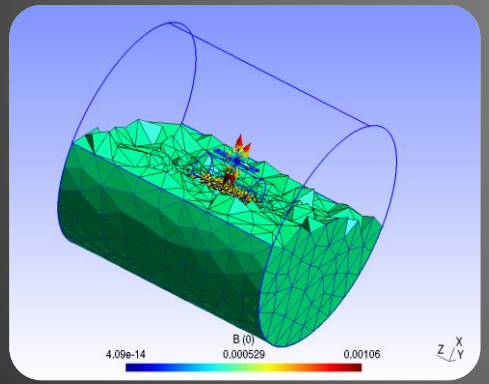


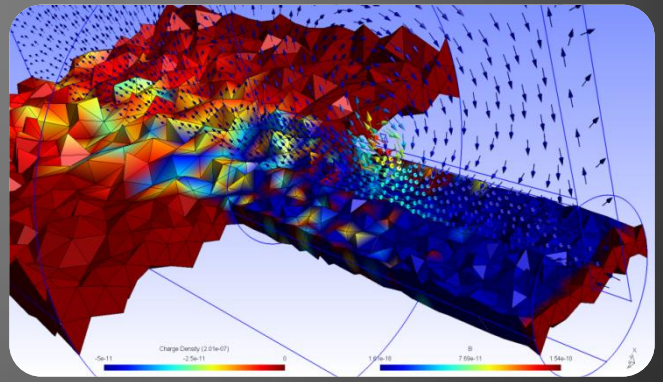
4. Solve Fields



# F3MPIC

- Particle-in-Cell simulation package
- Developed at CISAS to engineer Helicon thrusters
- Based on 3D Unstructured mesh



# PIC on GPU: Key elements



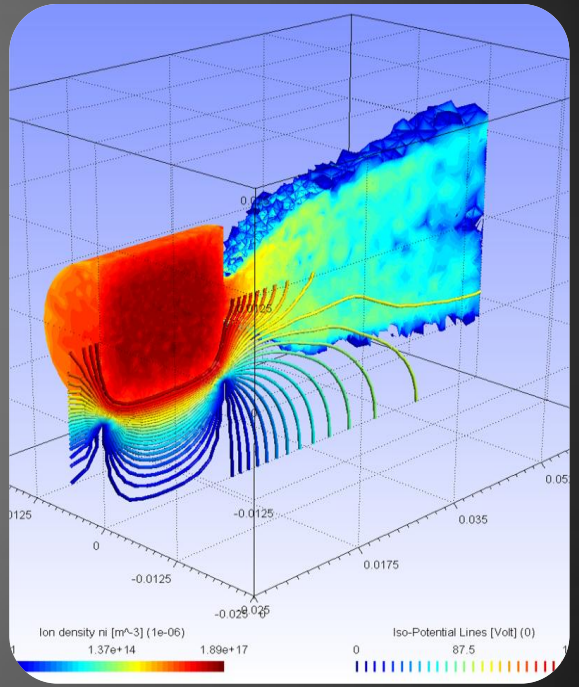
- Create cell clusters
- Sort particles based on clusters
- Atomic operations for charge accumulation

References:


- G. Stantchev, W. Dorland, N. Gumerov, *Fast parallel particle-to-grid interpolation for plasma PIC simulations on the GPU*, Journal of Parallel and Distributed Computing, 68, 2008
- X. Kong, M.C. Huang, C. Ren, V.K. Decyk, *Particle-in-Cell simulations with charge-conserving current deposition on graphic procesing units*, Journal of Computational Physics 230(4), 2011
- R. Joseph, G. Ravunnikutty, S. Ranka, E. D'Azevedo, S. Klasky, *Efficient GPU Implementation for Particle-in-Cell algorithm*, 2011 IEEE International Parallel & Distributed Processing Symposium, 2011
- J.E. Payne, *Implementation and Performance Evaluation of a GPU Particle-in-Cell Code*, Master Thesis, Massachusetts Institute of Technology, 2012

# F3MPIC on the GPU: Problems

- Custom AoS data structures
- External 3rd Party FEM Field Solver
- 3D Tetrahedral mesh:
  - No mesh data locality
  - Undeterministic access pattern
  - Explicit particle tracking required

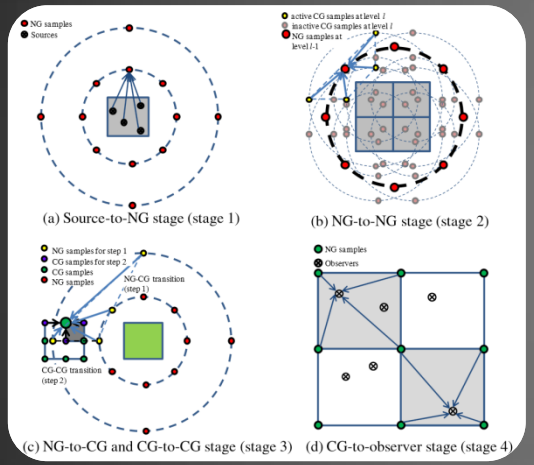


# F3MPIC GPU Modules: Baseline

- Convert data to SoA
- Keep field solve on CPU
- Thread Perspective:
    - Particles vs. Nodes
- One CUDA Stream per species

# FGM: First Attempts

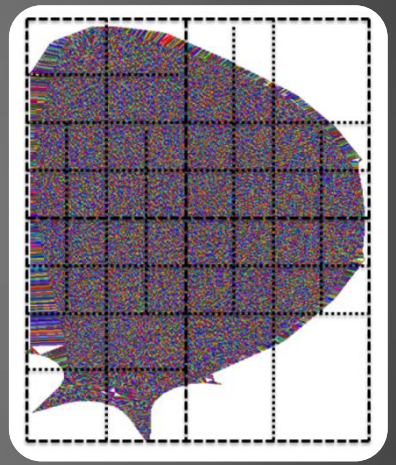
Superimpose octree to interpolate




From: S. Li, R. Chang, V. Lomakin, *Fast Electromagnetic Integral Equation Solvers on Graphics Processing Units*, GPU Computing Gems – Jade Edition, Morgan Kaufmann, 2012

Problem: Not good for charge density

Non-uniform partitioning to use shared memory

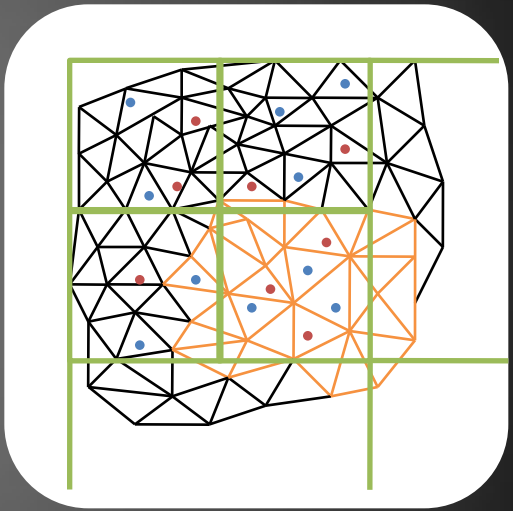


From: R. Joseph, G. Ravunnikutty, S. Ranka, E. D'Azevedo, S. Klasky, *Efficient GPU Implementation for Particle-in-Cell algorithm*, 2011 IEEE International Parallel & Distributed Processing Symposium

Good! We want to generalize this!

# Octree domain partitioning

- Built adaptively on GPU (with Dynamic Parallelism)
- Establishes sorting parameter (Z-order node numbering + *Thrust* sort)
- Allows coalesced memory access
- Enables effective shared memory usage
- <u>Objective</u>: Leaf node == Thread Block



# Octree adaptivity

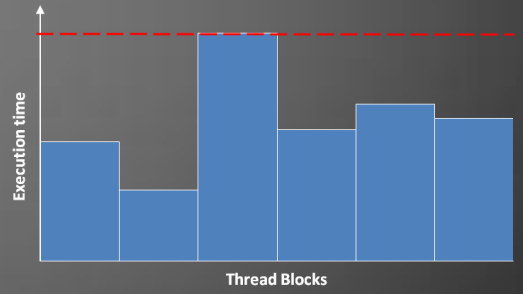
## Particle based

- ✓ Optimal workload distribution
- ✗ Requires rebuilding at every time step
- ✗ Inefficient and unpredictable use of per-Block resources
- ✗ Particles are millions!

  Octree could get too large!

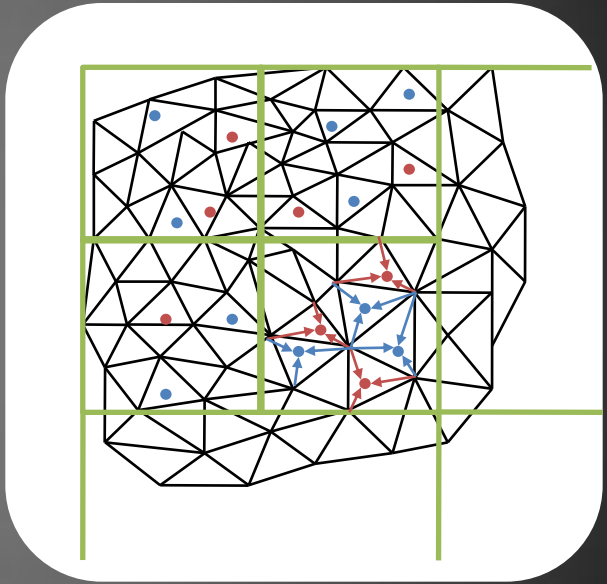
## Mesh based

- ✓ Built only at simulation start
- ✓ Good usage of shared resources
- ✗ We are bound by the work of the busiest node



# Interpolation & Push

- Load cells to shared memory
  - Octree helps caching!
- Coalesced load of particles
- Porting pusher code is straightforward



# Particle Tracking

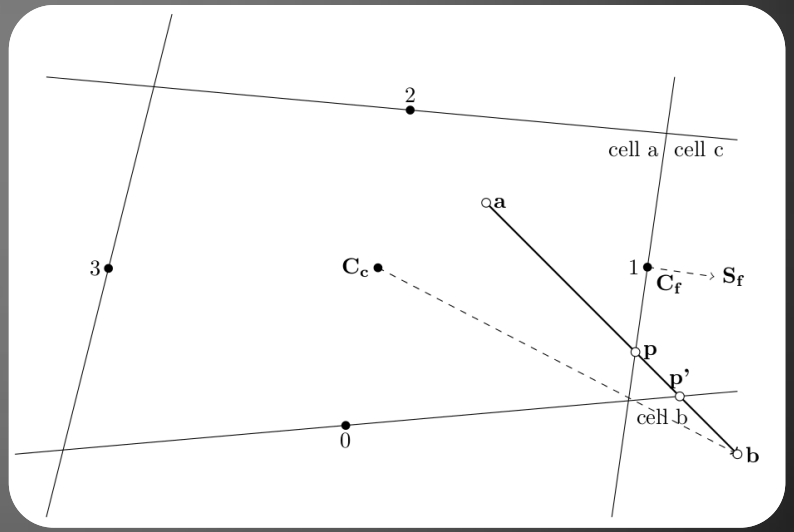
- Geometrically consistent algorithm

  G.B. Macpherson, N. Nordin, H.G. Weller, *Particle tracking in unstructured, arbitrary polyhedral meshes for use in CFD and molecular dynamics*, Communications in Numerical Methods in Engineering, 2008

- Already proven on GPU

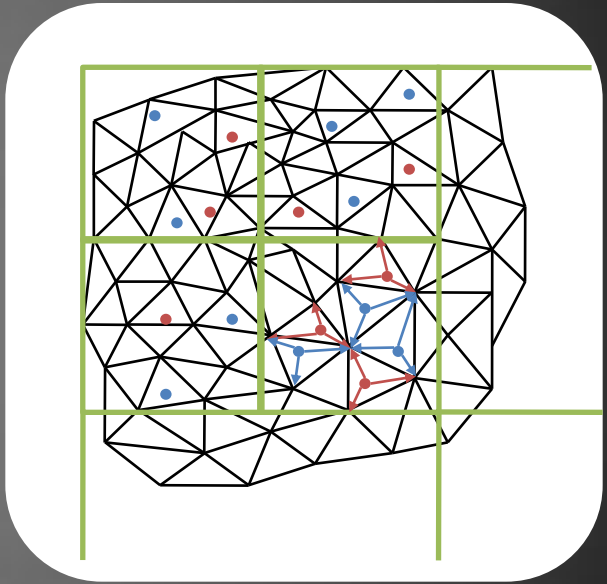
  N. Brünggel, *Lagrangian Particle Tracking on a GPU*, Lucerne University of Applied Sciences and Arts, 2011

- Adapted for F3MPIC custom mesh format



# Deposition

- Load cells to shared memory
  - Octree helps caching!
- Coalesced load of particles
- Different outcomes possible
  - Node-based Thread Blocks reduce warp divergence!
- Charge accumulation with atomic operations



# Octree benefits

**DATA LOCALITY (Classic):**

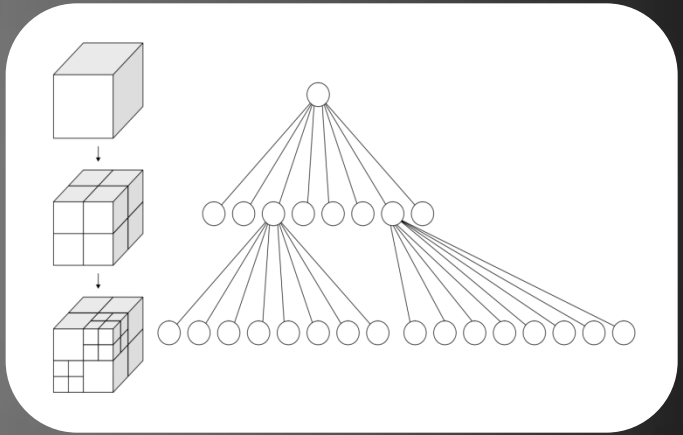
- Allows sorting → Coalesced Access
- Enables shared memory usage

**WORKLOAD MANAGEMENT:**

- Balances workloads
- Reduces warp divergence
- Helps caching behavior

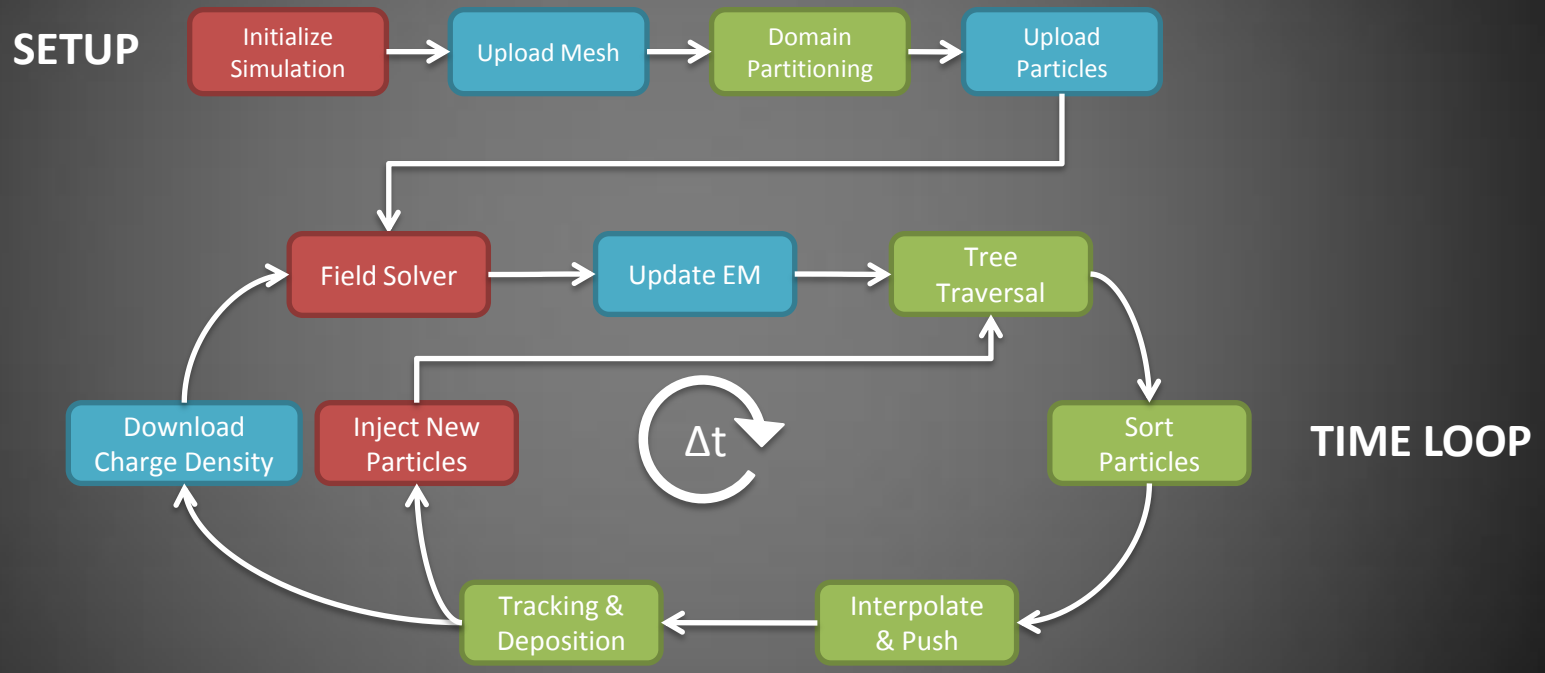
**FLEXIBILITY:**

- Opens to new functionalities



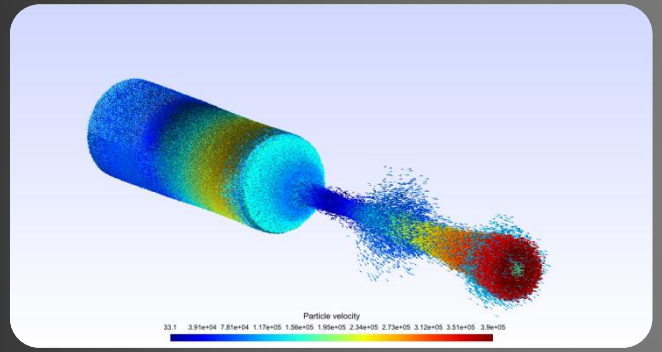
**<u>The Octree is the algorithm backbone!</u>**

# FGM: Algorithm



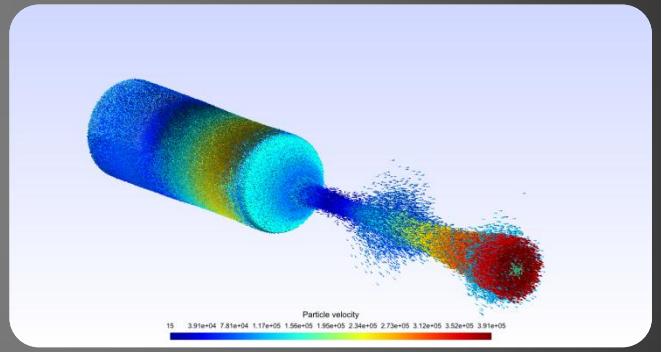
# FGM: Accuracy

## F3MPIC




```
Charge conservation check
Global: sum of charges 1.231251e-10 C, integral of charge density 1.232762e-10,
error 0.122737%
Species 0: sum of charges -1.383870e-11 C, integral of charge density -1.385641e
-11, error -0.127917%
Species 1: sum of charges 1.369638e-10 C, integral of charge density 1.371326e-1
0, error 0.123260%
```

## F3MPIC + FGM

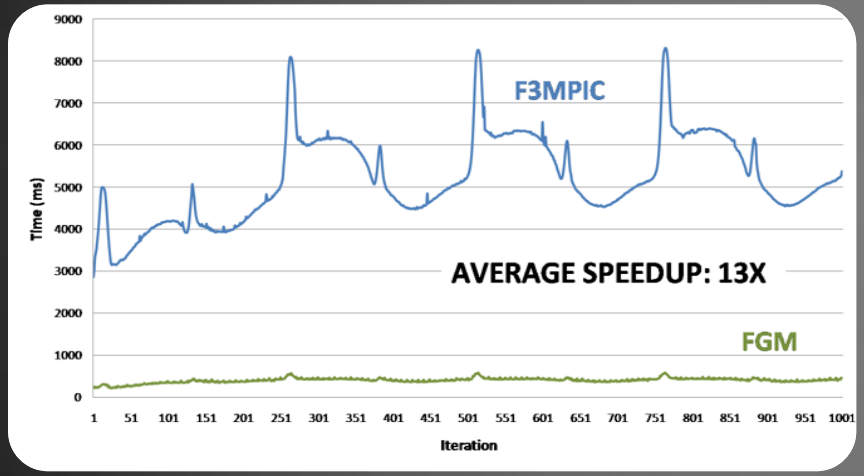



```
Charge conservation check
Global: sum of charges 1.230899e-10 C, integral of charge density 1.230899e-10,
error 0.000000%
Species 0: sum of charges -1.364573e-11 C, integral of charge density -1.364572e
-11, error -0.000078%
Species 1: sum of charges 1.367356e-10 C, integral of charge density 1.367356e-1
0, error 0.000008%
```

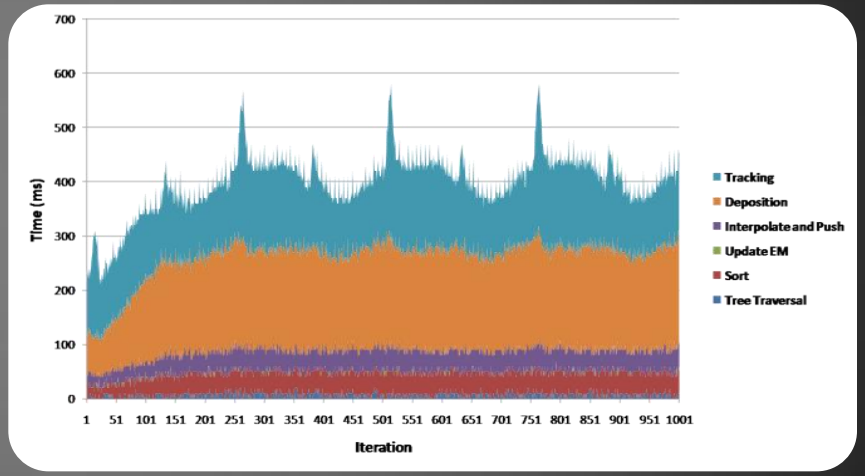
- New tracking algorithm
- FMAD on GPU

# FGM: Performance

## Total interaction times



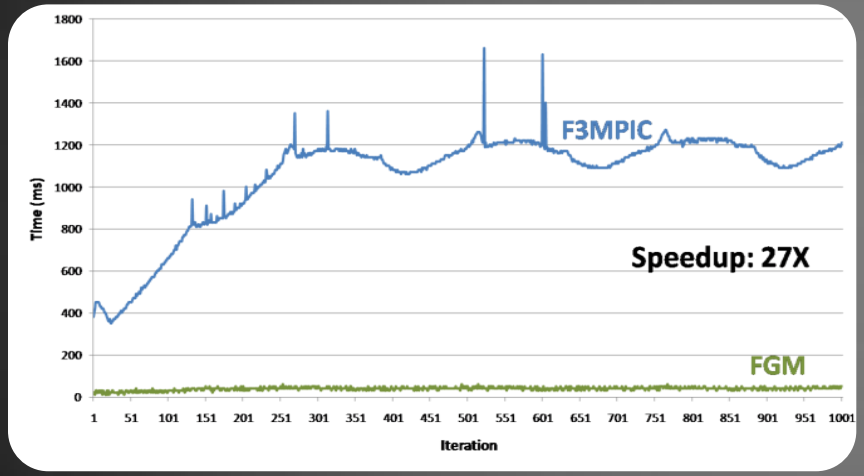

## FGM Breakdown



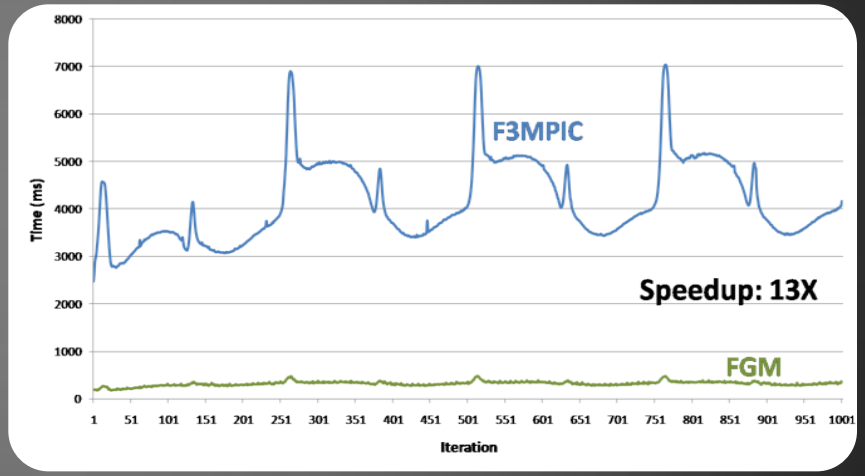

CPU: Intel Core i7-4770 @ 3.4GHz (Haswell microarchitecture)
GPU: NVIDIA GeForce GTX 780

# FGM: Performance

## Interpolation & Push




## Tracking & Deposition




CPU: Intel Core i7-4770 @ 3.4GHz (Haswell microarchitecture)
GPU: NVIDIA GeForce GTX 780

# FGM: Flexibility



- Monte-Carlo Collisions (MCC)
- Not just for plasma: direct particle interaction
  - Octree sorting → Broad phase collision detection
  - Narrow phase detection and contact algorithms: NNS, SPH, ...
- Agnostic to Field Solver
  - FEM
  - CFD
  - Discontinuous Galerkin (DG) – could be on GPU, too!
- Tracking works for polyhedral cells
- Scalable to Multi-GPU (Hilbert curve node numbering?)

# FGM: Exploitation

- Combustion with reagent atomization
- Multiphase flows (e.g. Droplets, etc.)
- Erosion phenomena
- Magneto-fluid Dynamics
- …

# QUESTIONS?

Contacts:


Alberto Madonna: madonna.alberto@gmail.com

Marco Manente: m.manente@hit09.com

Daniele Pavarin: daniele.pavarin@unipd.it