Hybrid Particle-in-cell (Hybrid PIC)

Team Members

Radhika Achikanath Chirakkara, PhD student, Australian National University

Mentors

Fred Fung,
National Computational Infrastructure

Australian National University

NCI AUSTRALIA

Hybrid Particle-In-Cell Algorithm

E : electric field
B : magnetic field
J : total current

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \cdot \mathbf{E} = \frac{\rho_{\text{total}}}{\epsilon_0} \; (\rho_{\text{total}} = \rho_I + \rho_e \approx 0)$$

Lorentz force

Turbulent driving

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \qquad \frac{d\mathbf{v}}{dt} = \frac{q}{m}(\mathbf{E}^* + \mathbf{v} \times \mathbf{B}) + \mathbf{f}$$

Particle evolution

Density (ρ) and ion currents ($\mathbf{J}_I$)

Calculated on particles interpolated to grid

Constrained-transport algorithm

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

Compute magnetic field

Faraday's Law

Particle-in cell / MHD codes

Thermoelectric effect

Resistivity

Ampere's law

$$\mathbf{J} = \frac{\nabla \times \mathbf{B}}{\mu_0}$$

$$\mathbf{E} = \frac{1}{\rho}\left[(\mathbf{J} - \mathbf{J}_I) \times \mathbf{B}\right] - \frac{\nabla p_{\text{electron}}}{\rho} + \eta \mathbf{J} - \eta_{\text{hyper}} \nabla^2 \mathbf{J}$$

Compute electric field

Hyper-resistivity

Problem with $\nabla \cdot \mathbf{B} = 0$

- This condition cannot be satisfied for any type of discretization
- This is a problem for magnetohydrodynamic and particle-in-cell codes

Discretization of physical equations

- Cell Centered Methods
  - Projection method
  - 8-wave method
  - Divergence cleaning

- Constrained Transport (Staggered method)
  - Magnetic fields (B) have a staggered representation where different components live on different faces – also requires a staggered representation of electric fields

Constrained Transport (Staggered method)

Magnetic fields (B or H in below figure) have a staggered representation where different components live on different faces – also requires a staggered representation of electric fields (E)

Pros :

Very good at conserving $\nabla \cdot \mathbf{B} = 0$ on computational domain

Cons :

Face-centered and edge-centered variables to be introduced and evolved

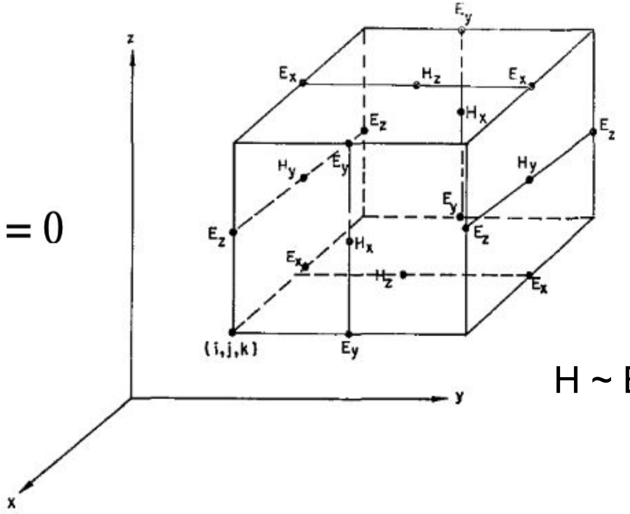

Fig. 1. Positions of various field components. The E-components are in the middle of the edges and the H-components are in the center of the faces.

H ~ B

Yee (1966)

Implementing Constrained Transport (CT) for a hybrid particle-in–cell code

- Problem trying to solve - trying to implement an algorithm with better properties for magnetic field evolution in a hybrid PIC code which traditionally uses a cell-centered algorithm
- Scientific driver for the chosen algorithm - to preserve the div. B = 0 (B : magnetic fields), so that the discretized equations do not create magnetic monopoles as the physical solutions evolve in numerical simulations
- What’s the algorithmic motif? - the constrained transport method (uses geometric discretisation of Stoke’s theorem)
- What parts are you focusing on? - different parts of the CT implementation

Evolution and Strategy

- What was your goal coming here?
- To understand the geometry of the constrained transport (CT) method
- Write a finite difference discretisation scheme for the physical equations (Ampere’s law and Faraday’s Law)

- What was your initial strategy?

- Write the sub-routine which can evolve magnetic fields using CT

- How did this strategy change?

- The estimation of electric fields is complicated for CT - hybrid PIC

Results and Final Profile

- What were you able to accomplish?
  - Wrote three Fortran90 subroutines for the evolution of electric fields, current and magnetic fields
  - Integrated these subroutines into FLASH hybrid PIC
  - Compiled the new source code on Gadi without errors

- What did you learn?
  - I learnt to implement the constraint transport method
  - Achieved new scientific goals? - Yes, this improved source code preserves div.B = 0 to machine precision using geometry, which is very important for MHD and PIC codes

What problems have you encountered?

- Understanding the geometry of the problem
- Calculating the currents and electric fields appropriately
- Understanding the FLASH data structures : FACE_VARS, SCRATCH_VARS
- Compilation errors

Was it worth it?

- Was this worth it? Yes
- Will you continue development?
  - Optimising the algorithm
  - Writing IO compatible with the new source code
  - Testing the new source code with test problems
- What sustained resources/support will be critical for your work after the event?
  - NCI computational resources for testing the code

Summarize your team’s achievements during this Hackathon

- Defining face centered variables
- Defining the total current from face centered variables
- Defining the electric field variables
- Finite difference scheme to evolve magnetic fields using CT