Present and Future Computational Requirements
General Plasma Physics
Center for Integrated Computation and Analysis of Reconnection and Turbulence (CICART)

Kai Germaschewski, Homa Karimabadi

Amitava Bhattacharjee, Fatima Ebrahimi, Will Fox, Liwei Lin

CICART
Space Science Center / Dept. of Physics
University of New Hampshire

March 18, 2013

Outline

Project Information

Center for Integrated Computation and Analysis of Reconnection and Turbulence

Director: Amitava Bhattacharjee, PPPL / Princeton University
Co-Director: Ben Chandran, University of New Hampshire

CICART has a dual mission in research: it seeks fundamental advances in physical understanding, and works to achieve these advances by means of innovations in computer simulation methods and theoretical models, and validation by comparison with laboratory experiments and space observations. Our research program has two elements: niche areas in the physics of magnetic reconnection and turbulence which build on past accomplishments of the CICART group and to which the group is well-positioned to contribute, and high-performance computing tools needed to address these topics.

Objectives

Magnetic Reconnection

- Reconnection and in laser-generated plasma bubbles
- Reconnection and secondary instabilities in large, high-Lundquist-number plasmas
- Particle acceleration in the presence of multiple magnetic islands
- Gyrokinetic reconnection: comparison with fluid and particle-in-cell models

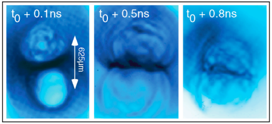
Turbulence

- Imbalanced turbulence
- Ion heating
- Turbulence in laboratory (including fusion-relevant) experiments

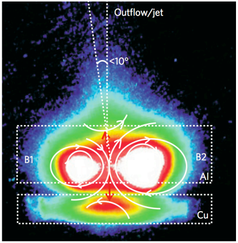
Bubble reconnection

Reconnection observed in laser-driven plasma experiments

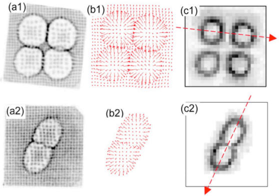
Rutherford [Nilson, *et al* PRL 2006, PoP 2008, Willingale *et al* PoP 2010]



Shenguang [Zhong *et al* Nature Phys 2010]



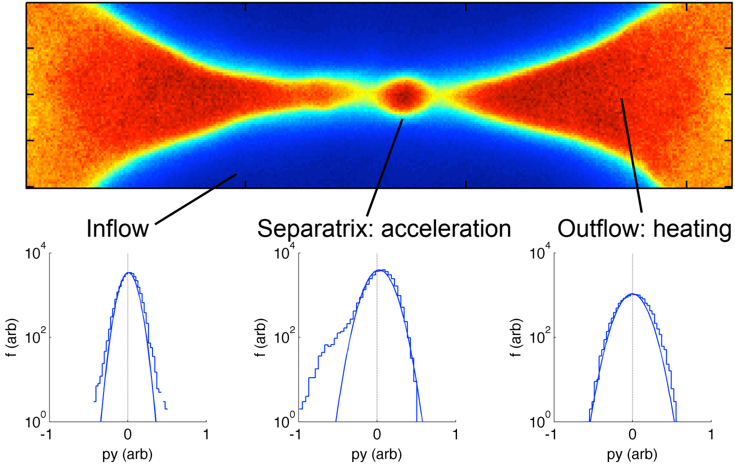
Omega: [C.K. Li, et al PRL 2007]



WFox APS 2012

Bubble reconnection

Particle energization is under study



WFox APS 2012

Bubble reconnection

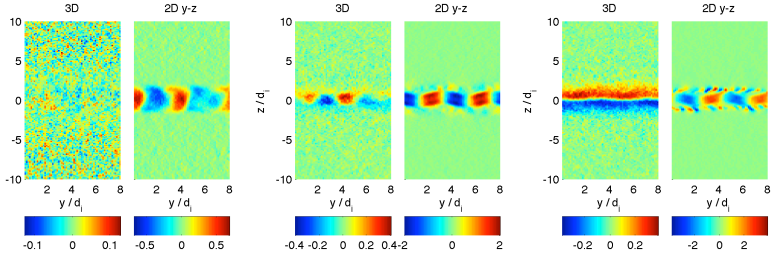
2D vs 3D: J_z evolution

Time →

Early Weibel growth for 2d

Weibel begins for 3-d (note scales)

Late time (peak reconnection)



Quiz: what is bipolar Jz in 3-D at late time?

Answer: the Hall current system near the x-line

WFox APS 2012

Current computational methods

Kinetic: Particle Simulation Code (PSC)

- solves Vlasov-Maxwell equations, in 1D/2D/3D
- 1st and 2nd order shape functions
- binary collisions
- explicit timestepping, parallelized by domain decomposition
- modular design
- special features: dynamic load balancing, AMR (wip)

Fluid: Magnetic Reconnection Code (MRC)

- solves extended MHD: Generalized Ohm's Law
- finite-volume, div B = 0, arbitrary curvilinear grids
- explicit, implicit time integration through PETSc
- automatic code generator generates r.h.s., Jacobian
- parallelized by domain decomposition / MPI

Current HPC usage

We used to run most simulations on a local Beowulf cluster, but in recent years, we have run almost all simulations at NERSC and other supercomputing centers.

Usage 2009

- NERSC: 500,000 hrs
- local cluster: 1,400,000 hrs

Usage 2012

- NERSC: 3,500,000 hrs
- Jaguar: 15,200,000 hrs
- XSEDE: 2,000,000 hrs
- BlueWaters: 100,000 hrs
- local cluster: 100,000 hrs

Typical jobs

PSC: Particle-in-Cell

- 20,000 - 60,000 cores, 4 - 24 hours
- Data written: 5 TB (data read: 3 TB on restart)
- Memory used: 3 TB, 1 GB / core

MRC/HMHD/OpenGGCM: MHD / Hall-MHD

- 4,000 - 20,000 cores, 12 - 48 hours
- Data written: 2 TB
- Memory used: 10 GB, < 100 MB / core

Necessary software, infrastructure

- HDF5, PETSc, (Parallel Python?)
- visualization cluster (Paraview, python, matlab) with large memory and fast disk access

HPC requirements for 2017

PIC runs are by far most expensive, so they dominate the requirements.

Requirements

- Hours: 100,000,000
- nr cores: 1,000,000 1-2 GB / core
- wallclock: 72 hrs
- I/O: 50 TB + 1 PB checkpoint (!?!)
- online file storage: 200 TB / 1.2 PB (checkpoint)
- offline file storage: 1 PB
- data analysis becomes even more challenging ;-(

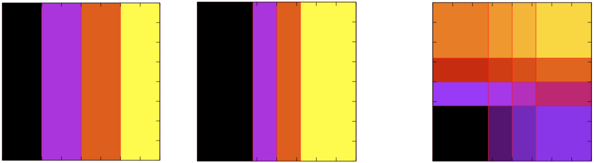
Enables: End-to-end plasma bubble reconnection, extended MHD space weather modeling, predictive 3-d simulations of lab / fusion devices

Dynamic Load Balancing

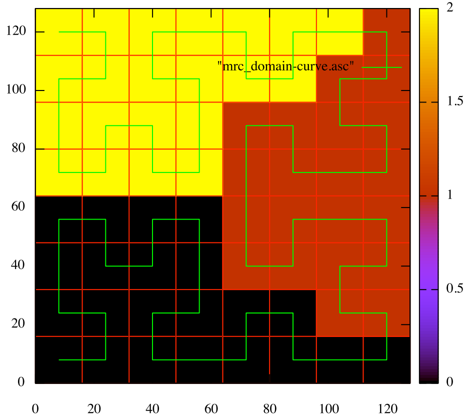
Problem with PIC simulations: *Those particles just keep on moving!*

PIC codes parallelized via domain decomposition often become unbalanced over time – even if balanced nicely at the start of the simulation.

Rebalance by moving processor domain boundaries:



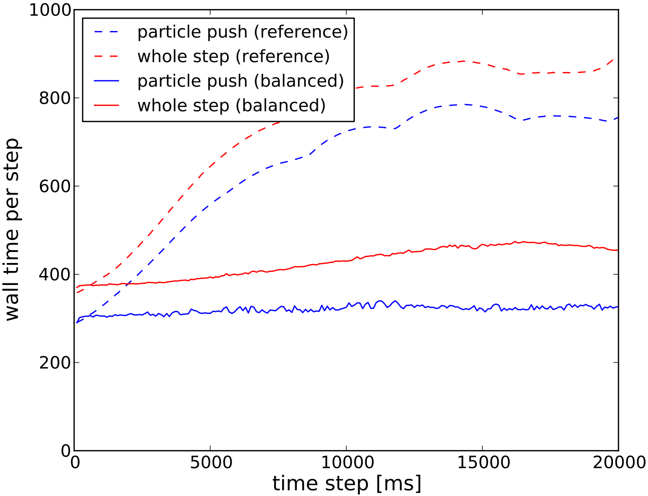
Dynamic Load Balancing



Color indicates the processor responsible for the corresponding part of the domain.

Dynamic Load Balancing

with patch-based load balancing



nvidia GPU, Intel MIC

Performance on TitanDev / BlueWaters:
16-core AMD 6274 CPU, Nvidia Tesla M2090 / Tesla K20X

Kernel	Performance [particles/sec]
2D push & V-B current, CPU (AMD)	130×10^6
2D push & V-B current, GPU (M2090)	565×10^6
2D push & V-B current, GPU (K20X)	710×10^6

For best performance, need to use GPU and CPU simultaneously.
Patch-based load balancing enables us to do that: On each node, we have 1 MPI-process that has $\approx$ 30 patches that are processed on the GPU, and 15 MPI-processes that have 1 patch each that are processed on the remaining CPU cores.

Suggestions

- Max. queue limits of > 24 hrs is highly desired
- Wait time of less than 2 weeks
- Sufficient scratch space for data analysis & purging no more frequent than 3 months
- Efficient I/O
- Establish a POC for large users
- Support from visualization experts is critical
- Maintain good interactive access
- Fault tolerance solutions
- PGAS / new programming models / load balancing