General Plasma Science Through Petascale Particle Simulations

Homa Karimabadi, UCSD
Kai Germaschewski, UNH

Contributors:
V. Roytershteyn, SciberQuest
Y. Omelchenko, H.X. Vu, UCSD
W. Daughton, LANL
M. Tatineni and A. Majumdar, SDSC
B. Loring, Prabhat, S. Byna, O. Ruebel, Lawrence Berkeley National Lab.

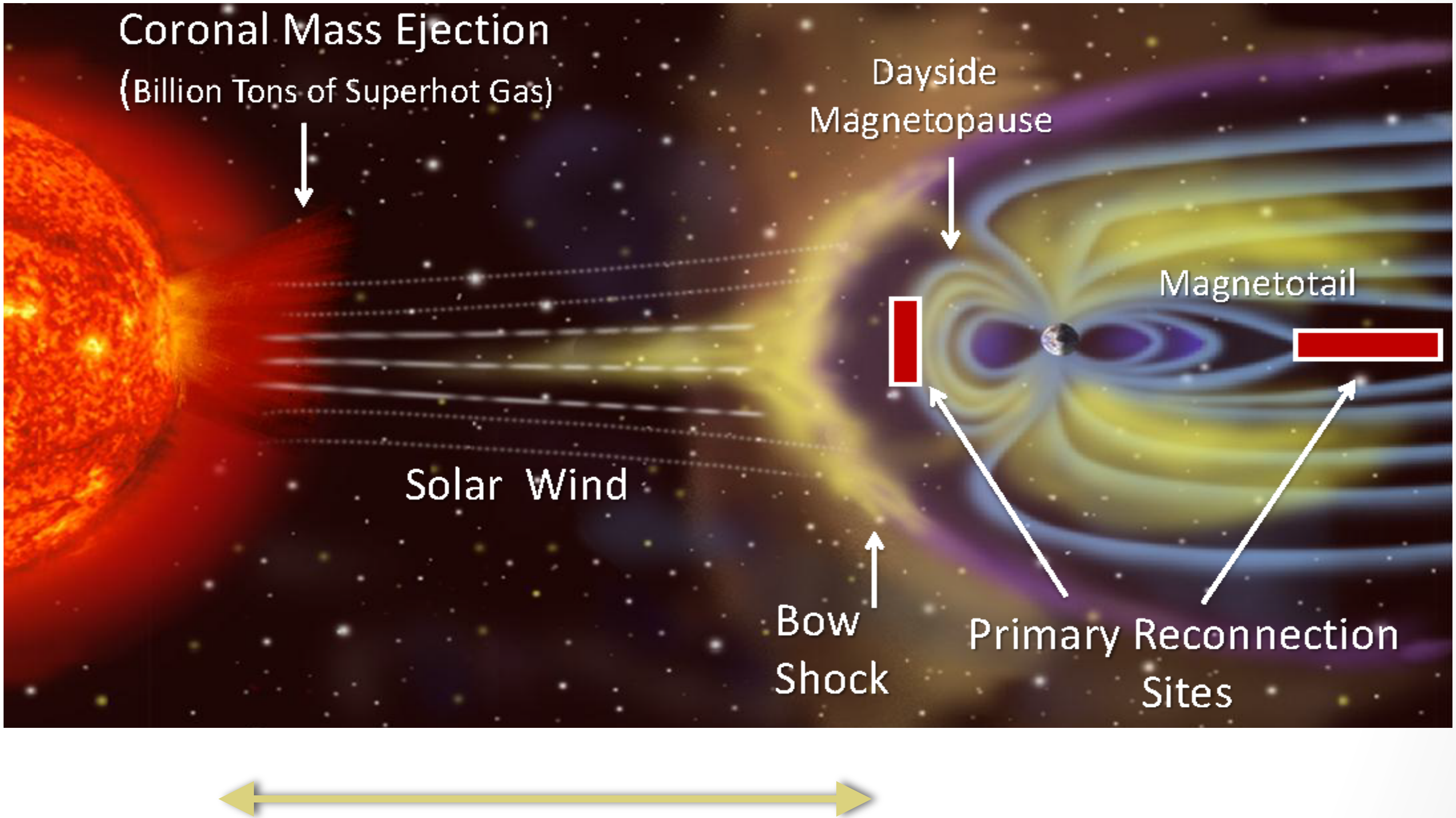
Large Scale Production Computing and Storage Requirements for Fusion Energy Science Research
March 19-20, 2013, Rockville, MD

Research Areas

- Plasma Turbulence
- Magnetic Reconnection
- Dynamo
- Exploratory Fusion Concepts
- Space Weather

Computer Performance	
Name	**FLOPS**
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}

Space Weather



90 million miles or ~ 100 Suns

Goal: Develop Accurate Forecasts of Space Weather

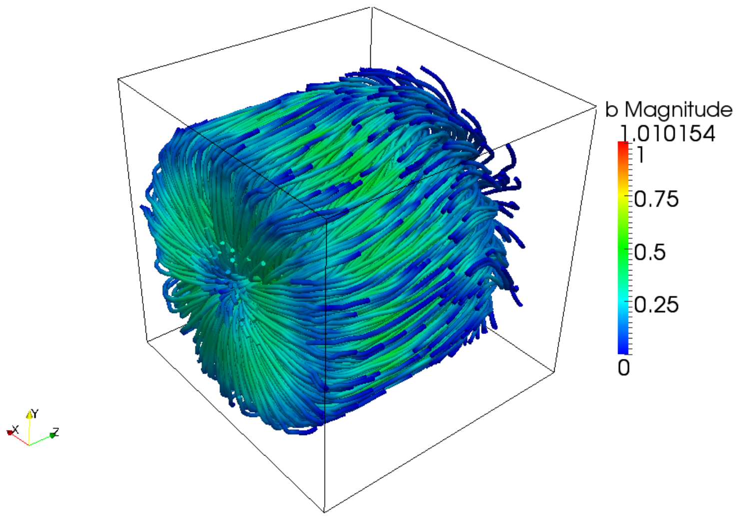
Space weather affects our technological systems:

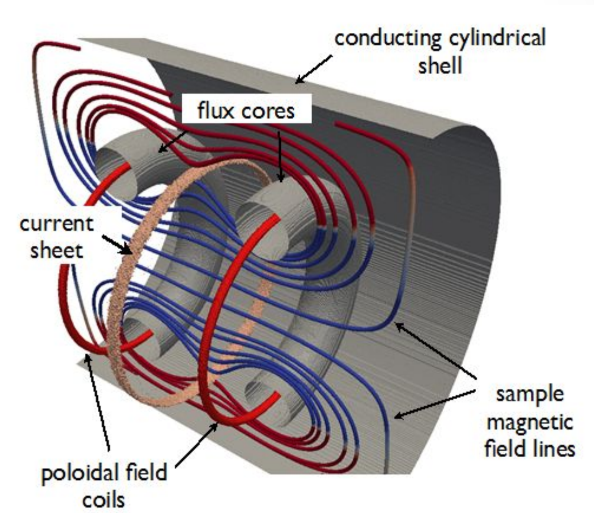
- Has caused over $4 billion in satellite losses
- A solar storm of the magnitude of the 1859 Solar Superstorm would cause over $2 trillion in damage today.
- Causes damage to sensitive electronics on orbiting spacecraft
- Causes colorful auroras, often seen in the higher latitudes
- Creates blackouts on Earth due to surges in power grids

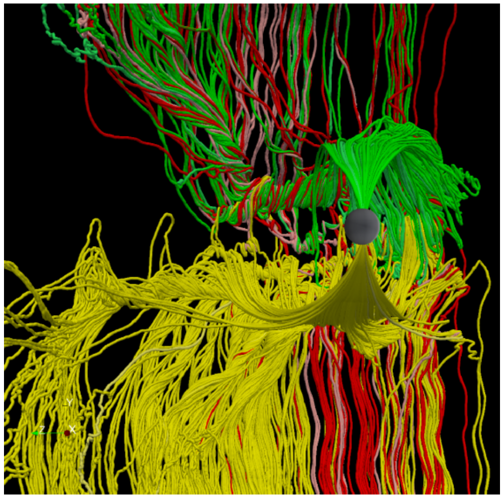
Funded by a new 5 year, multi-institutional NSF/NASA Collaborative Grant – PI A. Bhattacharjee (Princeton)

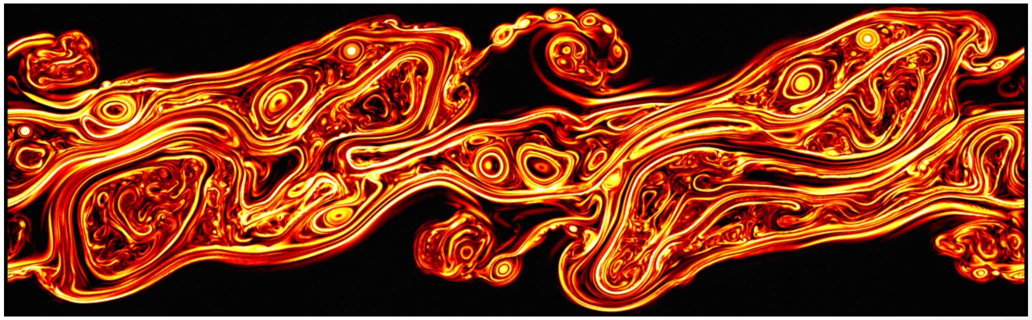
Approach – Develop next-generation global codes based on closure models of magnetic reconnection

Example Simulations



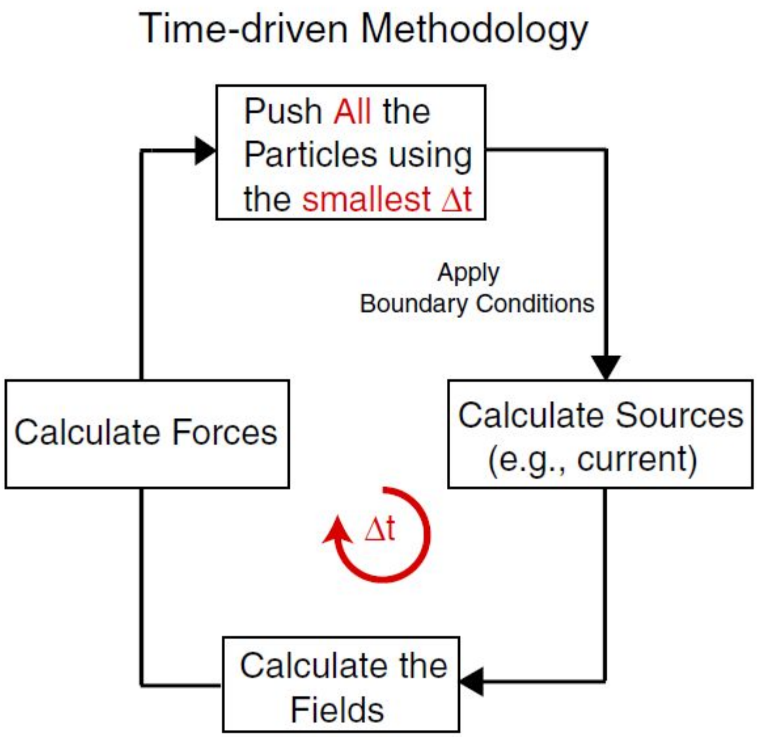


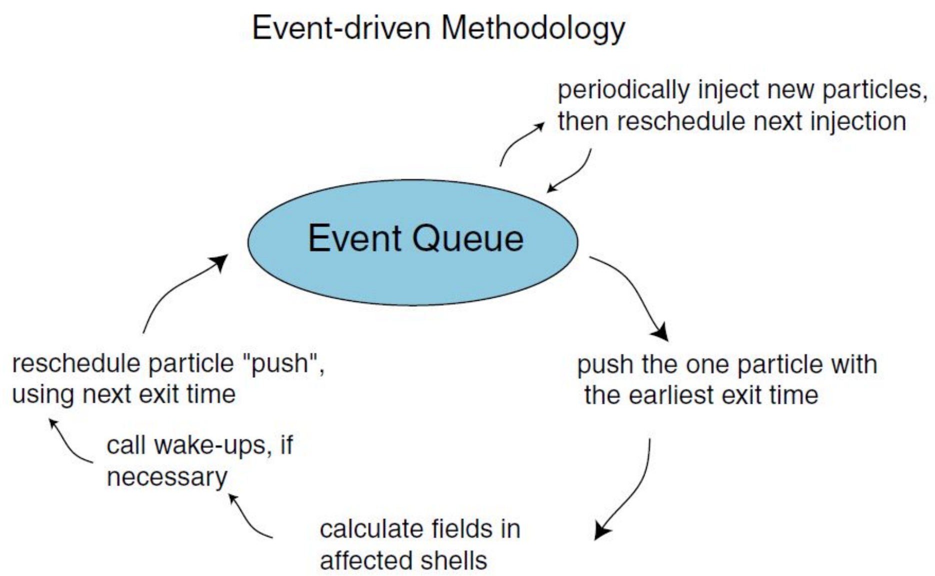




Particle-In-Cell Plasma Codes

- Fully kinetic (electrons and ions are treated as particles)
- Hybrid (electron fluid, particle ions)

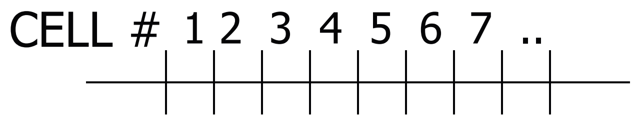




Karimabadi et al., JCP, 2005

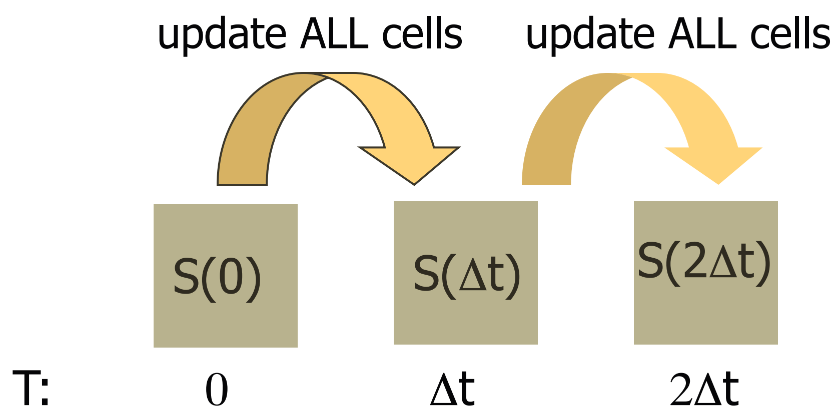
Discrete Event vs Time Stepping

S(T) =



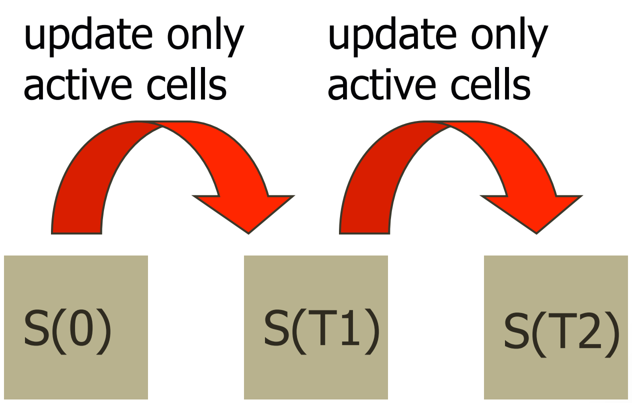
Time-driven simulation

updates the entire system

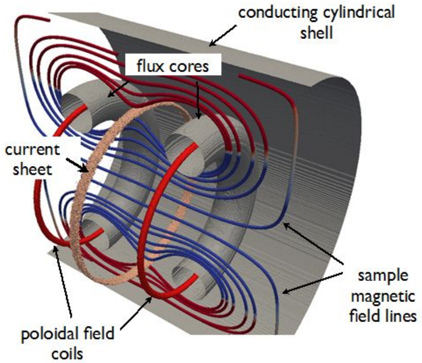


Event-driven simulation

updates active cells only

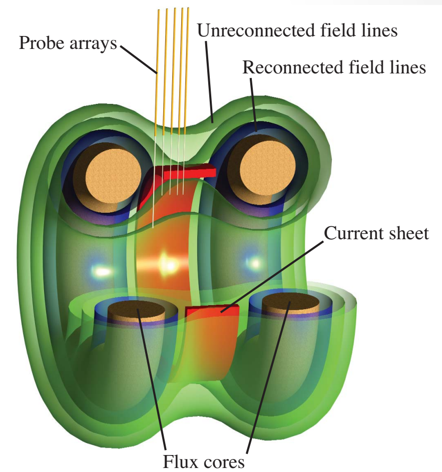


Code Validation

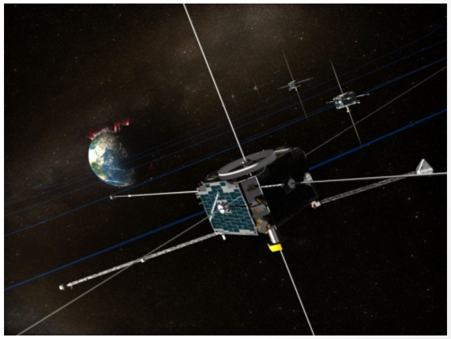


Simulations

Lab. Experiments

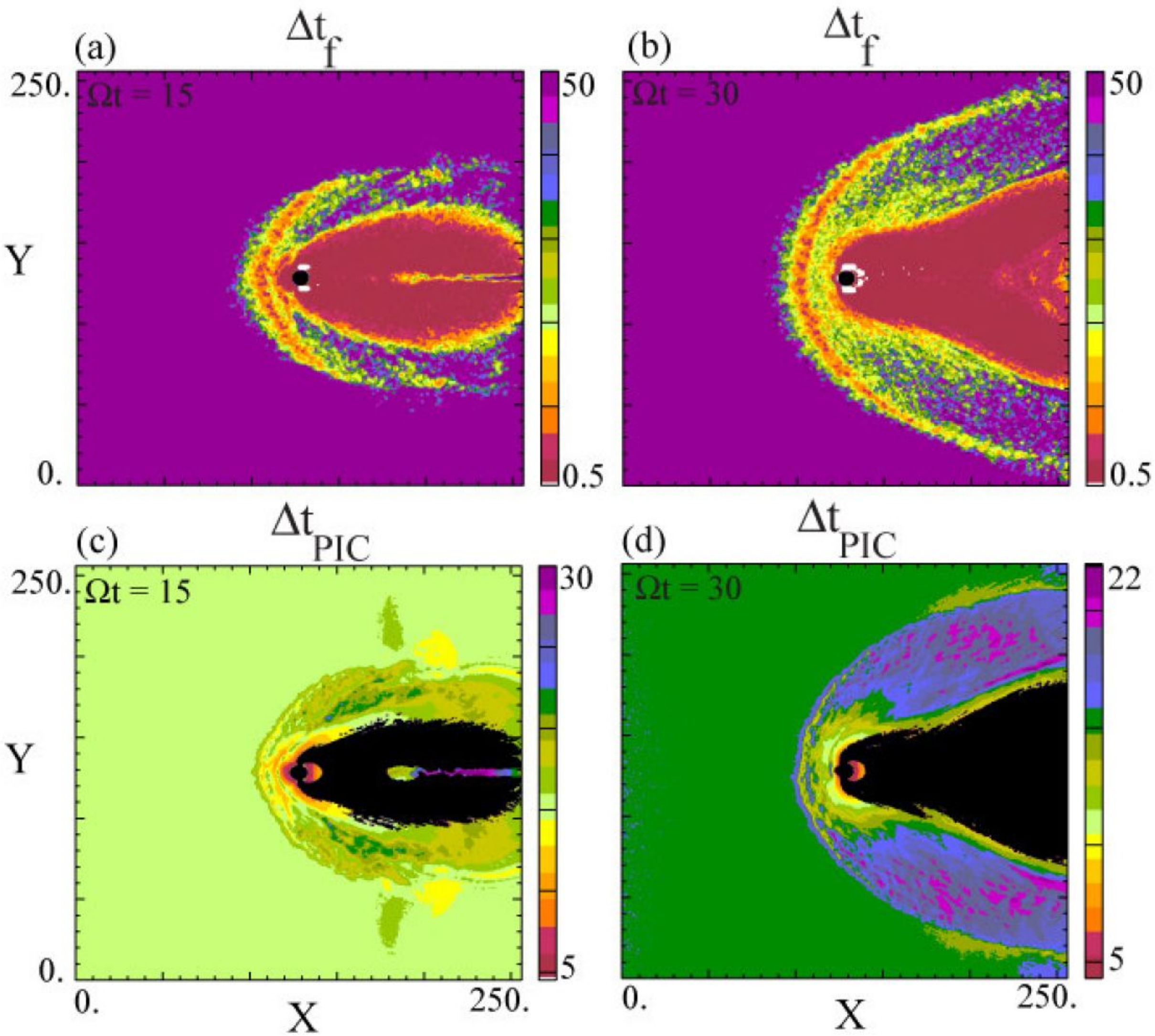


Spacecraft Data



Theory

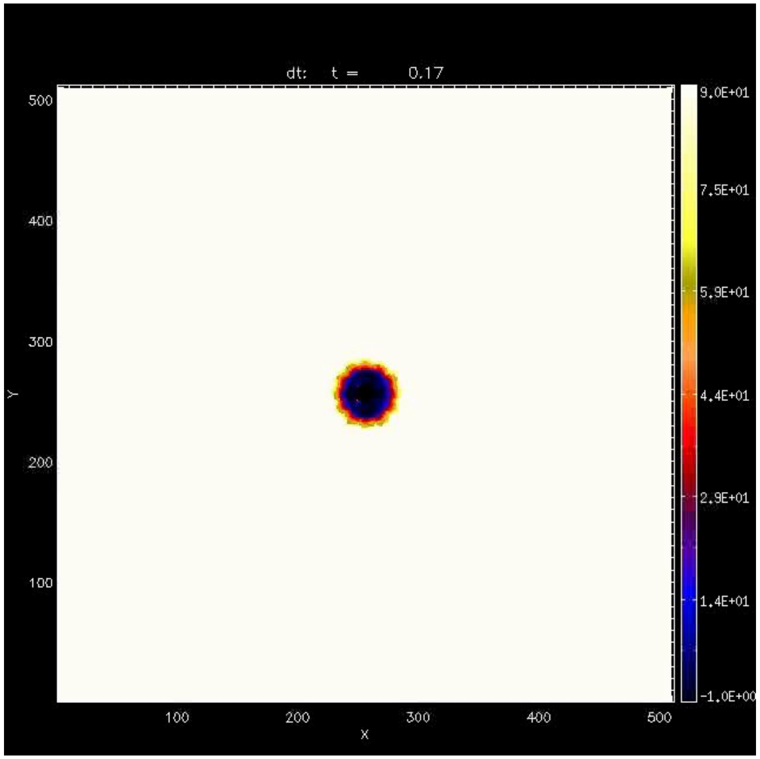
Event-Driven Simulations



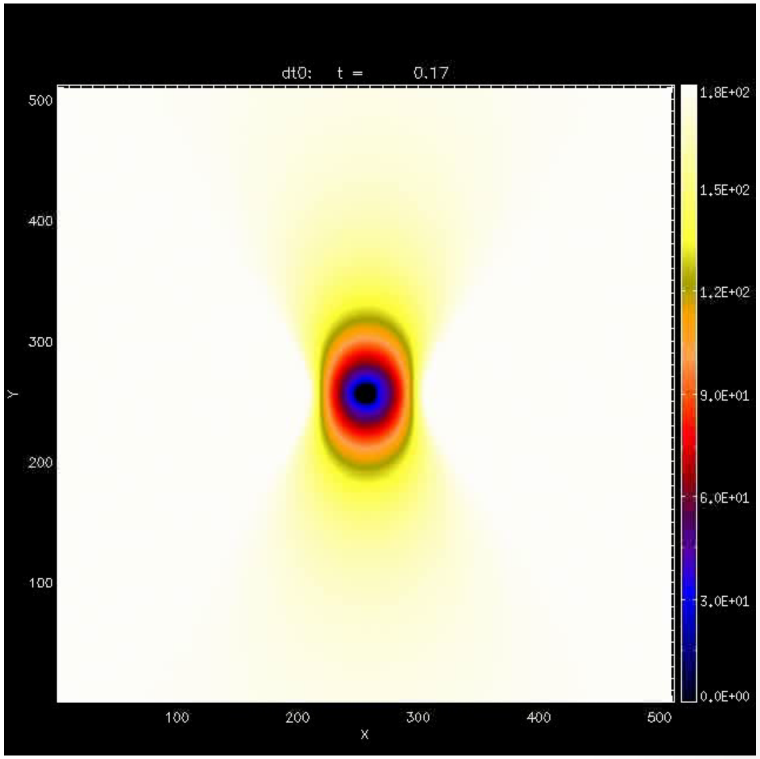
Omelchenko and Karimabadi, JCP, 2011

Field-Cells and Particles Self-Adapt Their Timesteps

Field Δt

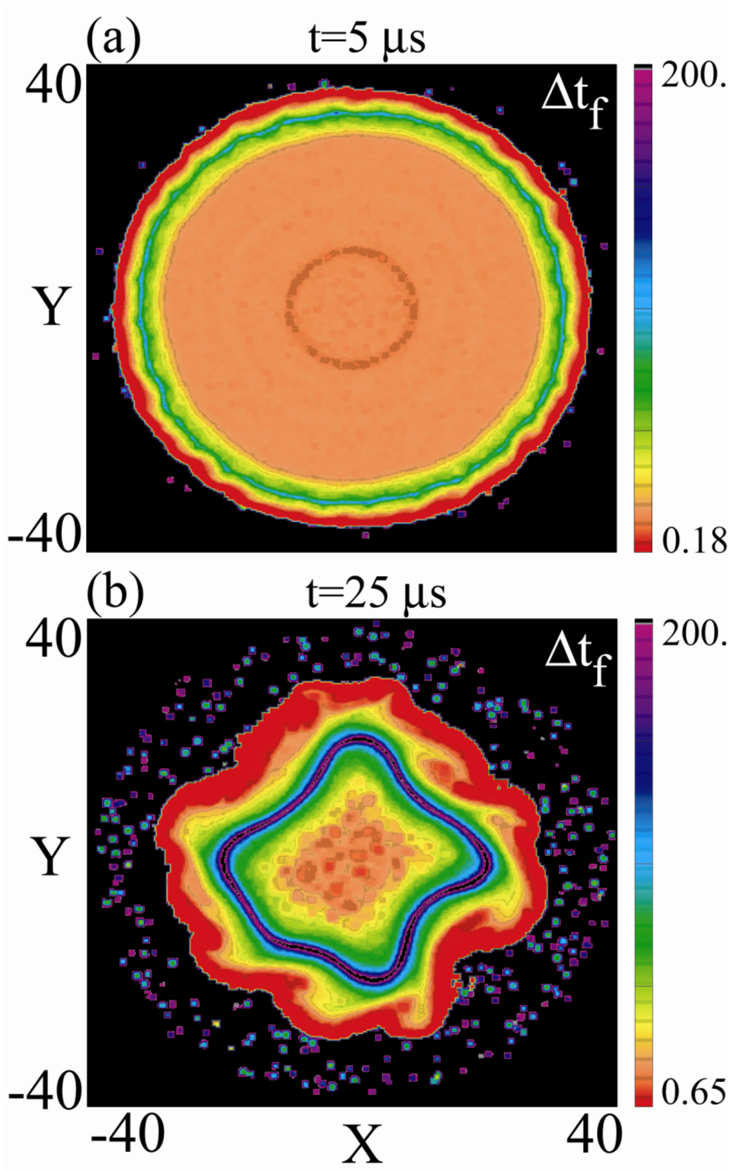


PIC Δt



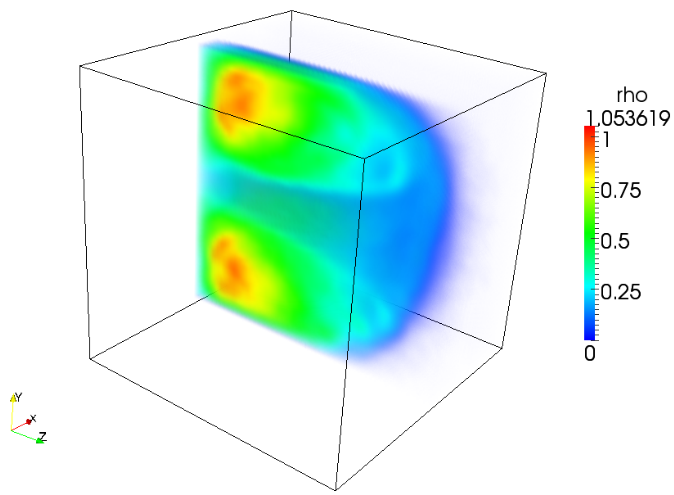
The only code in the world that can do this!

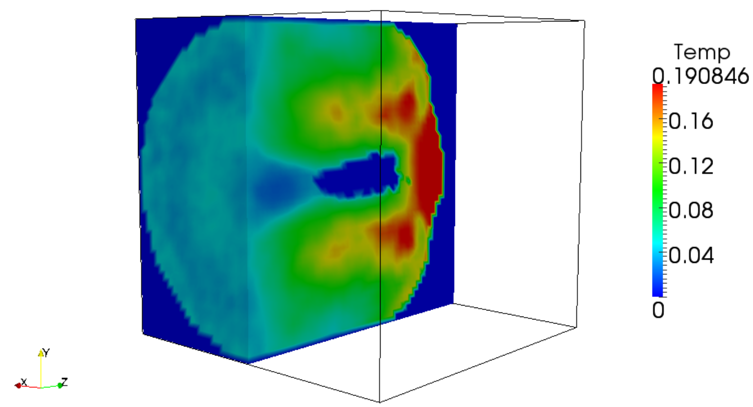
Field-Reversed θ-Pinch Discharge

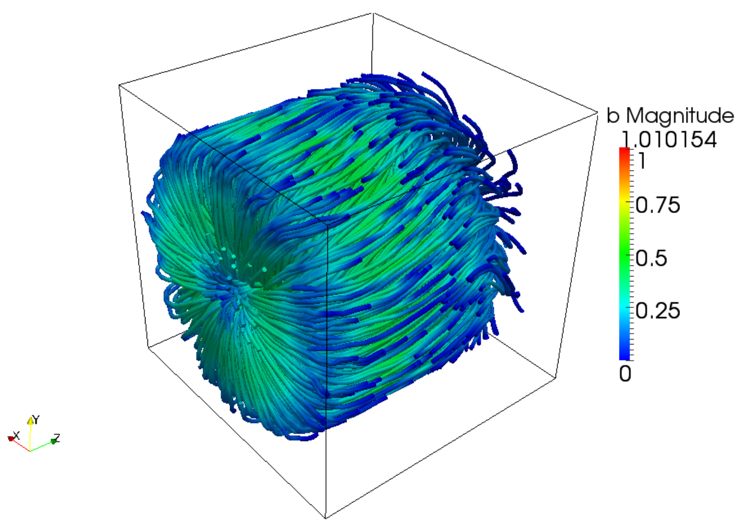


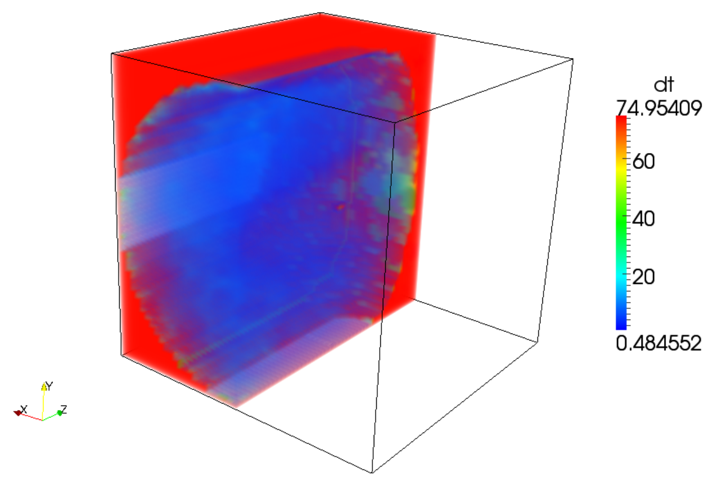
Omelchenko and Karimabadi, PRL, 2012

Spheromak Expansion







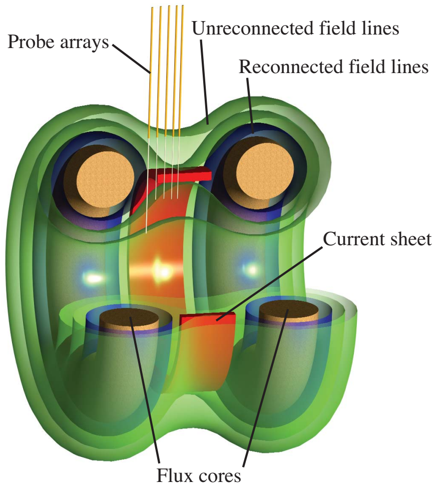


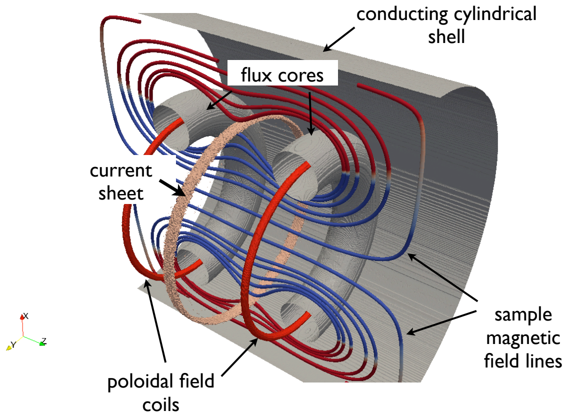
Challenge: Extreme Multi-scale Nature of the Physics

Modeling of Laboratory Experiments

Example : MRX (Magnetic Reconnection eXperiment, DOE/PPPL)

H. Ji & M. Yamada





Why we need such experiments: they are controlled, reproducible, well diagnosed; essential to validate and challenge models and codes

Simulation/theory challenge: “order of magnitude” answers are not good enough anymore

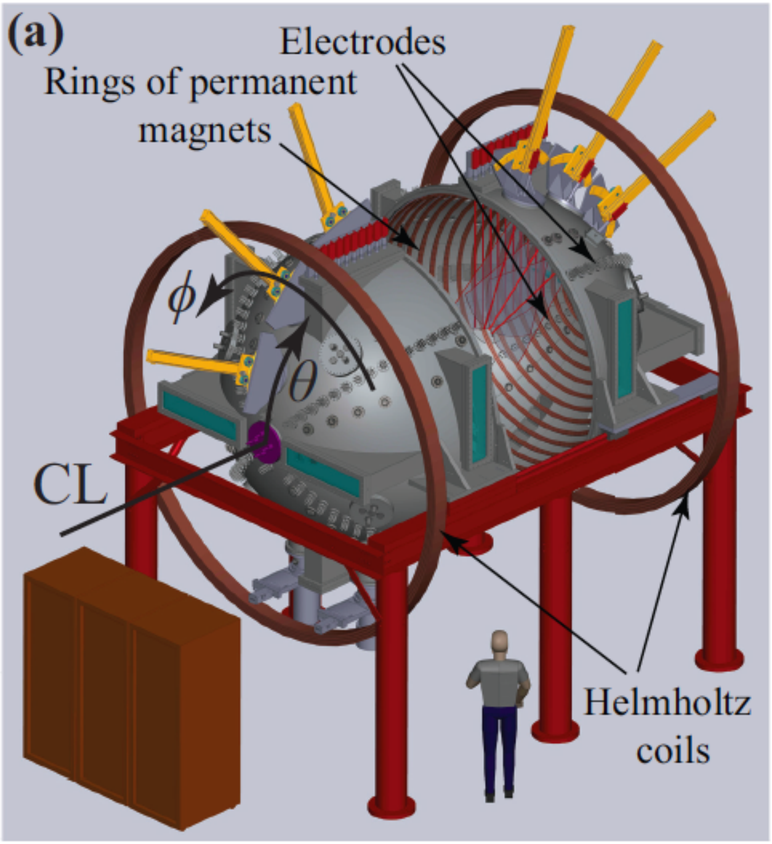
What does it take to model an experiment?

Parameters/size	physics	Cost, CPU-hrs
2D: M/m=100; $\Omega_{pe/}\Omega_{ce}$=2 1024 x 2048 cells	Basic physics of the diffusion region, role of the external drive	1.5 x 10^4
2D: M/m=1836; $\Omega_{pe/.}\Omega_{ce}$=2 5300 x 10000 cells	Realistic influence of binary collisions, realistic kinetic physics of trapping, etc	6.5 x 10^6
3D: M/m=300; $\Omega_{pe/}\Omega_{ce}$=2 2000 x 4000x4000 cells	Influence of current-aligned instabilities	9 x 10^8
3D: M/m=1836; $\Omega_{pe/}$ Ω_{ce}=80 (2x10^5) x (4x10^5) x (4x10^5) cells	Realistic physical parameters	9 x 10^{18}

Madison Plasma Dynamo Experiment (MPDX)

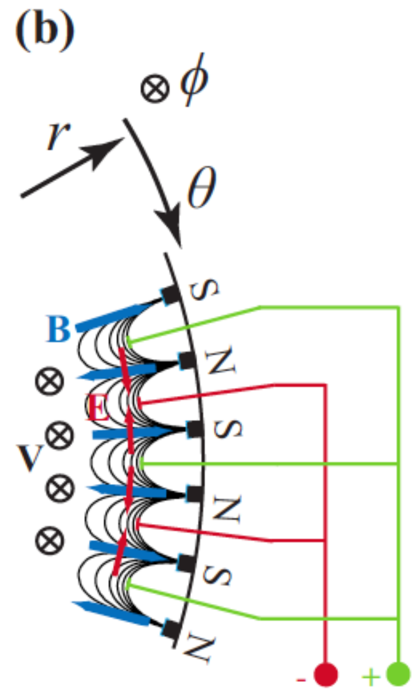
C. Forrest & I. Khalzov

Sketch of MPDX



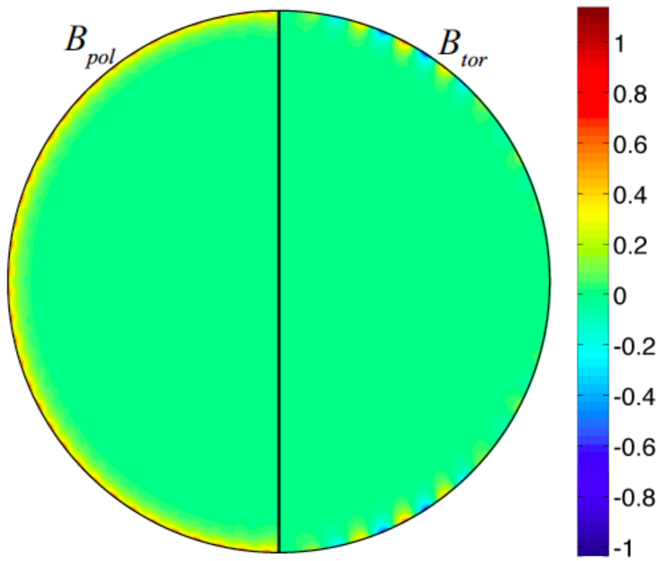
Multicusp magnetic field confines plasma

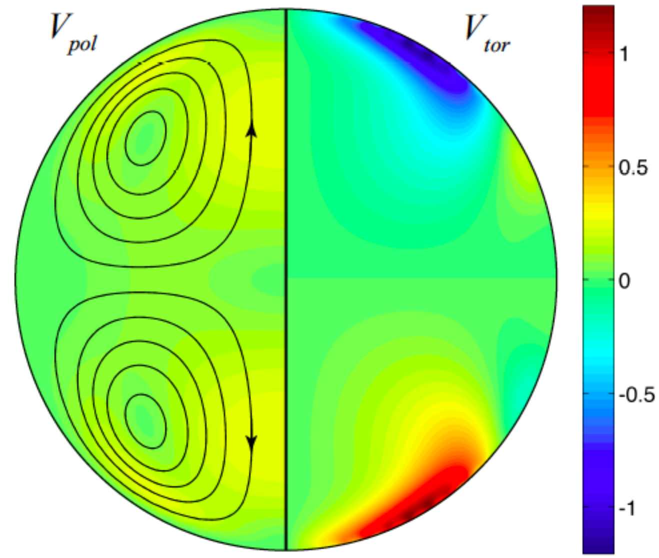
Electrode configuration near the wall



Arbitrary flow profiles $v_\varphi(\theta)$ can be driven near the wall by adjusting the bias of the discrete electrodes

2D MHD simulations (NIMROD)





- Multicusp magnetic field is localized near the wall

- This counter-rotating plasma flow results in dynamo excitation

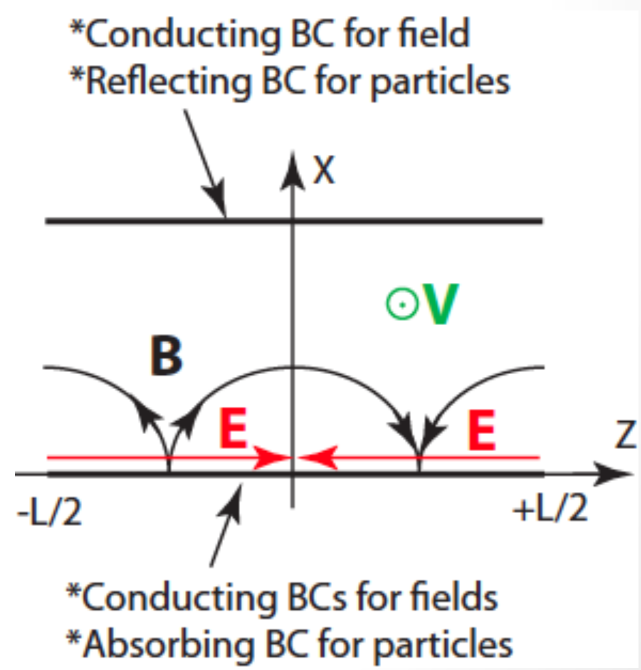
- The MHD simulations are missing important plasma physics:

- actual boundary conditions for velocity (free-slip or no-slip?)
- dependence of viscosity on magnetic field (Braginskii model)
- two-fluid effects (Hall term, electron inertia, etc.)

PIC simulations

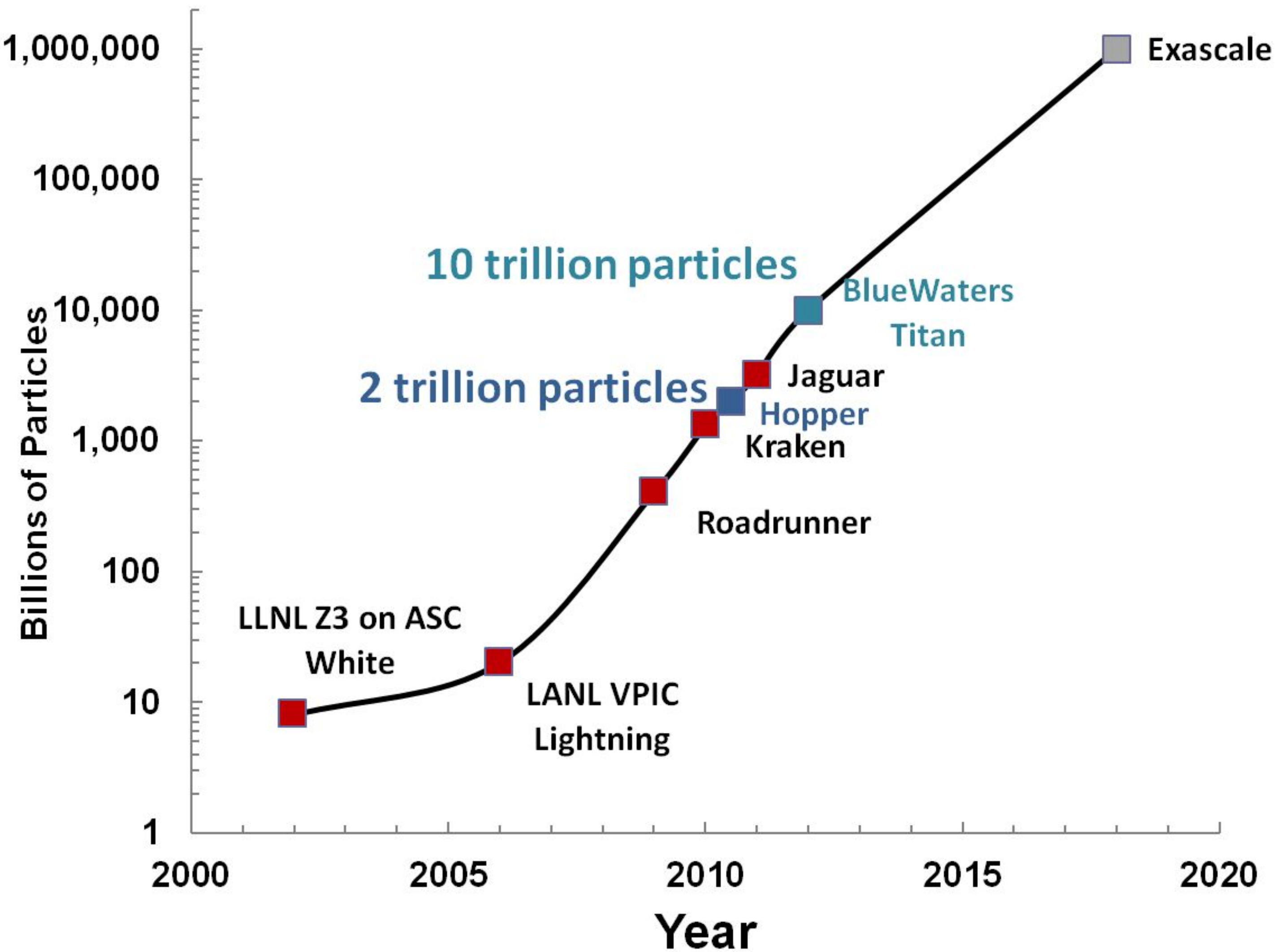
- Fully kinetic simulations are required to adequately resolve plasma dynamics in boundary region
- In experiment, cusp size is L=25 cm and typical Debye length is $D=2.5\times10^{-3}$ cm.
- The required resolution for 3D PIC run is about 1000^3 (1000 cells in each direction)
- Size of the device is 3 meters

  - Geometry for PIC simulations

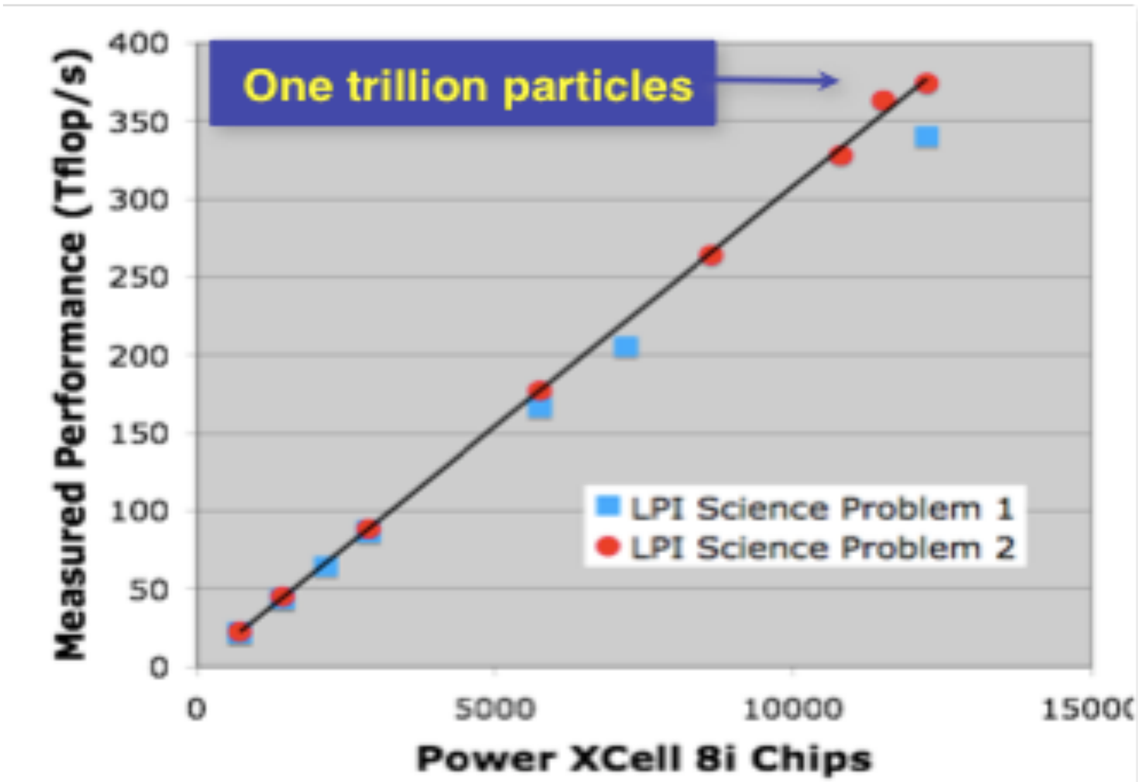


Simulation Characteristics – HPC Usage

Progress in Particle Simulations
(measured in terms of number of particles)



Scaling Properties



Performance obtained with VPIC on Roadrunner [*Bowers et al*, 2008b] demonstrating 0.374 Pflop/s sustained performance with ~1 trillion particles

Expected Scientific Breakthroughs Using Global Simulations on Exascale

2008	2012	2017	*Too many cores to count*
2D Hybrid *Breakthroughs* Physics of ion fore shock / bow shock	**3D Hybrid** *Breakthroughs* Transport, formation of boundaries such as PSBL, FTE generation, PDL, cusp physics, …	**3D Hybrid - Real time** *Expected Breakthroughs* Modeling of events over MHD scales	
	2D full PIC – reduced mass ratio *Breakthroughs* The first study of reconnection under realistic configuration	**3D full PIC – reduced mass ratio** *Expected Breakthroughs* Closure of long standing issues related to onset and time dependence of reconnection in the tail and dayside	**3D full PIC Simulation - high mass ratio** *Expected Breakthroughs* Proper capture of instabilities such as LHDI or electron kinetic effects associated with bow shock

Simulation Characteristics – Single Run

- 8 variables/particle =32 bytes
- $N_p = 10^{12}$ particles on 300 K cores
- N_b = Buffer factor ~ 2
- $N_c = N_{cores}/300{,}000$
- Memory = $32 * N_p * N_b * N_c / 10^{15}$ ~ **0.65 N_c PB**
- N_{check} = # of checkpointing files = 2
- Disk ~ **1.5*N_c PB**

Run time required on full scale machine ~ 72 hours

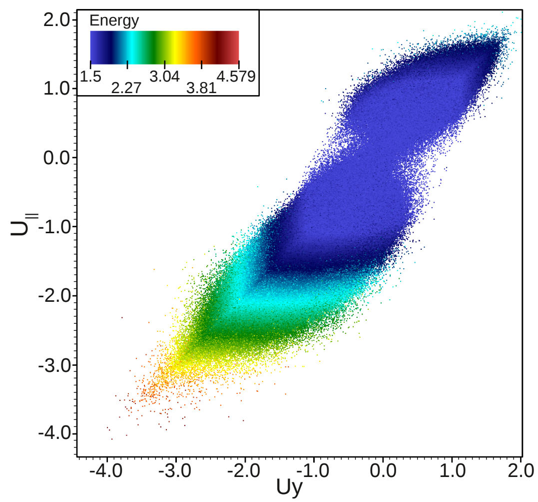
CPU Hours ~ $72 * N_{cores}$~**21.6*N_c Million Hours**

Current HPC Usage

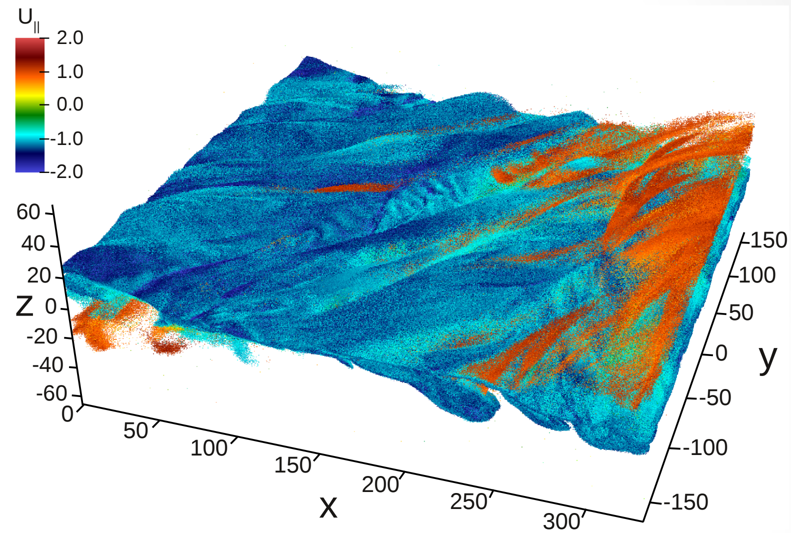
- Hopper ~ 3 million hours
- Jaguar/Titan ~ 20 million hours
- Kraken ~ 7 million hours
- Pleiades ~ 10 million hours
- Blue Waters ~ 30 million hours

Examples of New Physics Uncovered Due to Increase in Computational Power + Data Analytics Support at LBNL

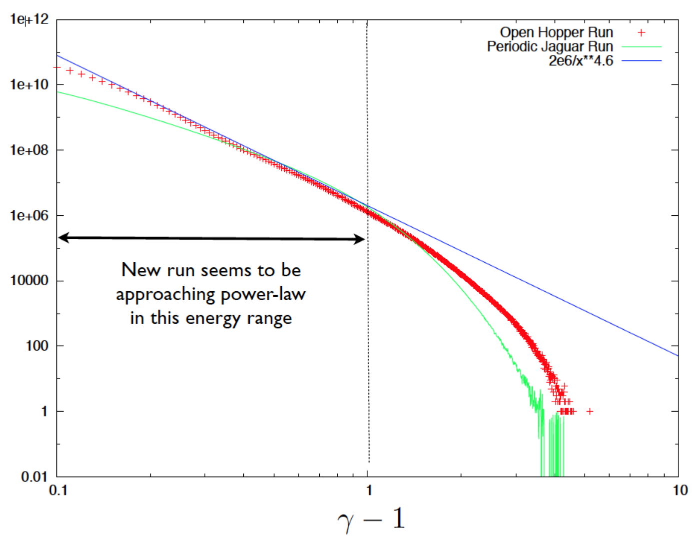
ExaHDF5/VPIC Science Impact



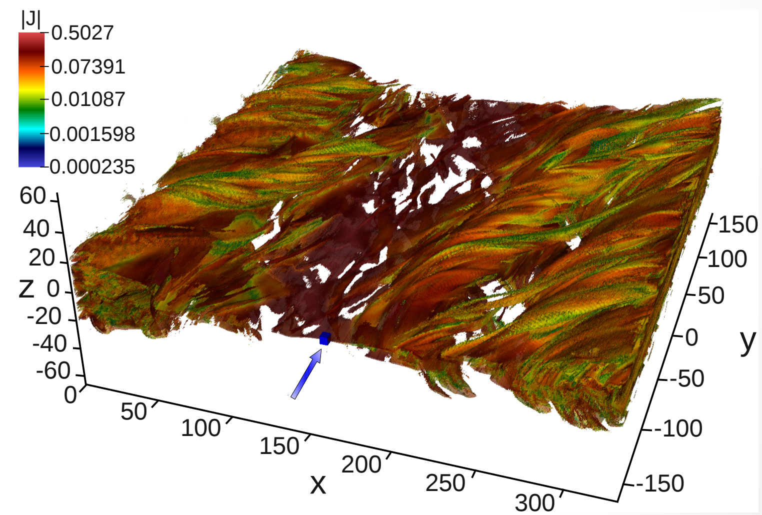
Preferential acceleration along magnetic field



Energetic particles are correlated with flux ropes



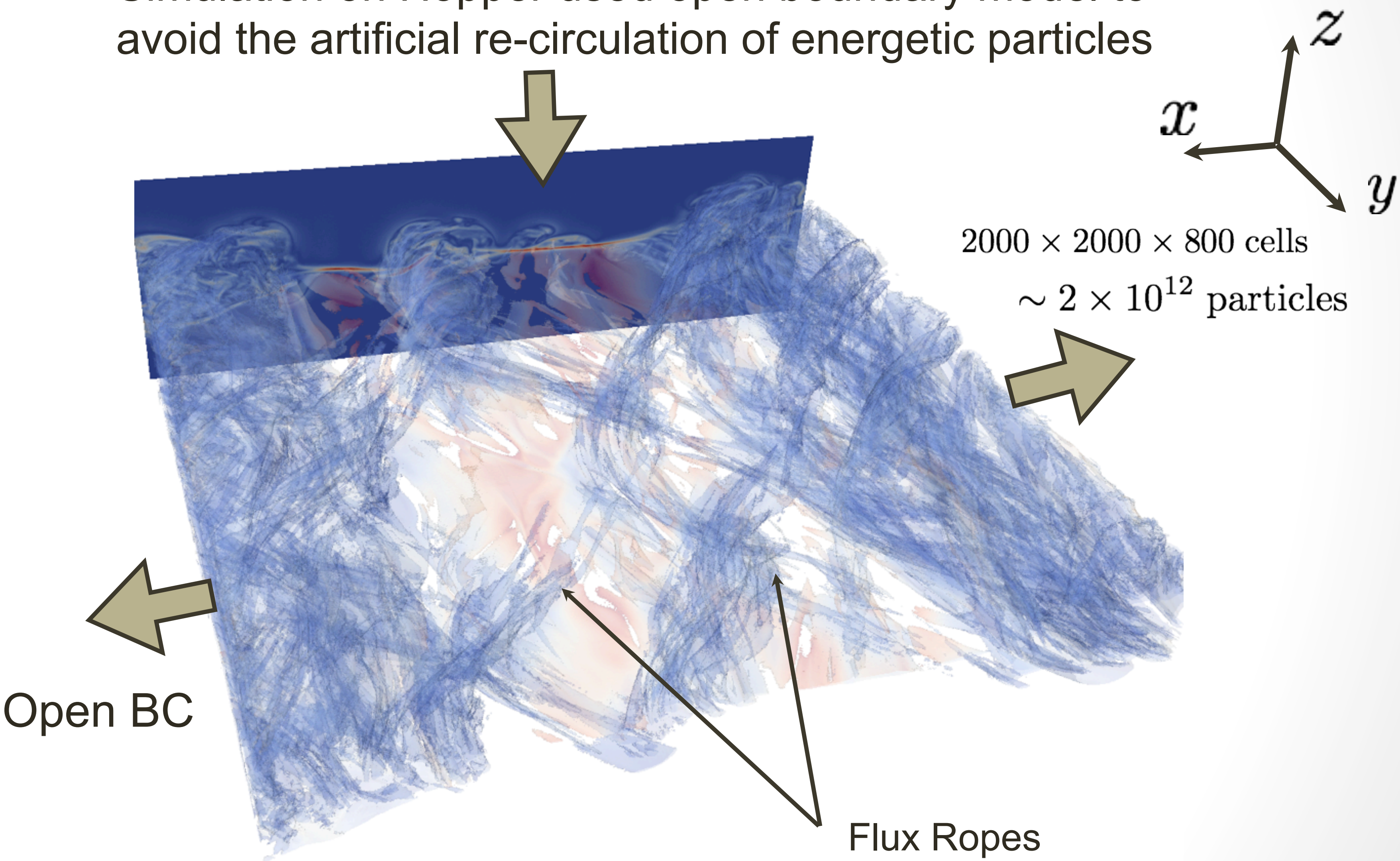
Discovered power-law distribution in energy spectrum



Discovered agyrotropy near the reconnection hot-spot

Kinetic simulations of magnetic reconnection feature the turbulent interaction of flux ropes

Simulation on Hopper used open boundary model to avoid the artificial re-circulation of energetic particles

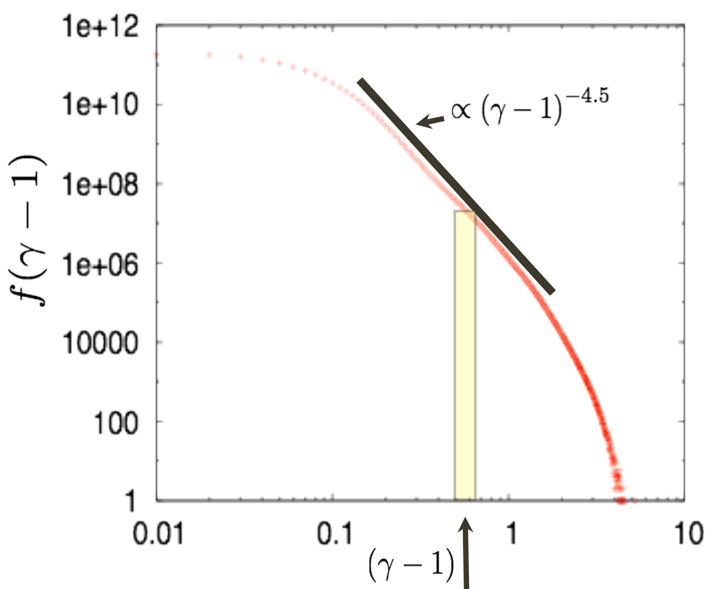


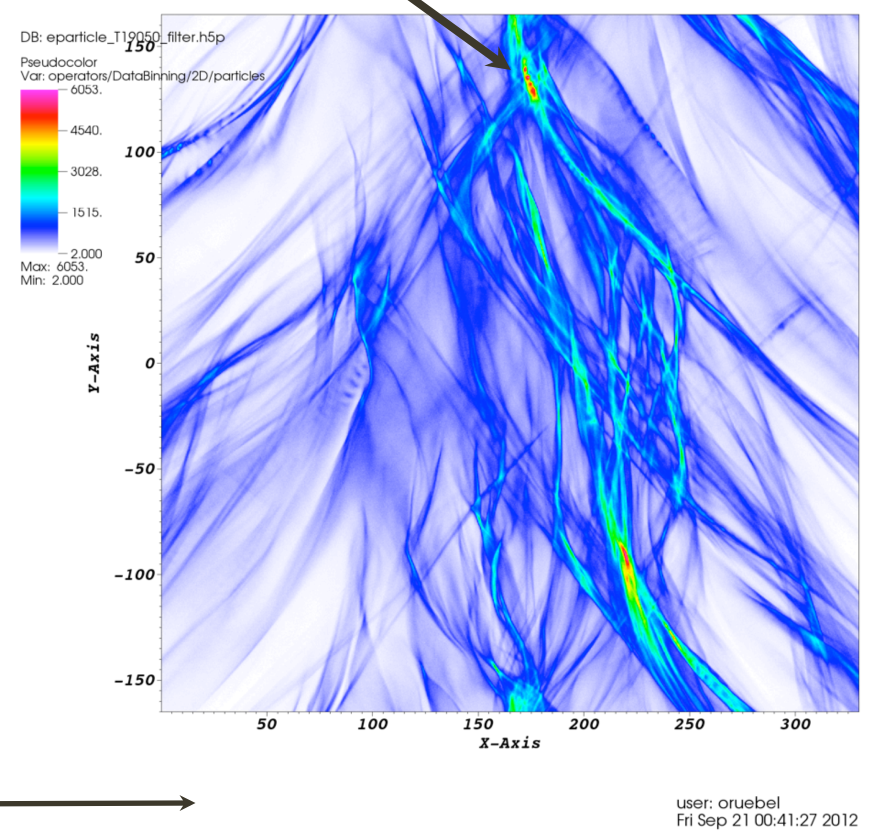
$2000 \times 2000 \times 800$ cells

$\sim 2 \times 10^{12}$ particles

First self-consistent 3D simulations to demonstrate a power-law in the spectrum of energetic particles

Acceleration mechanism is due to parallel electric fields associated with reconnecting flux ropes

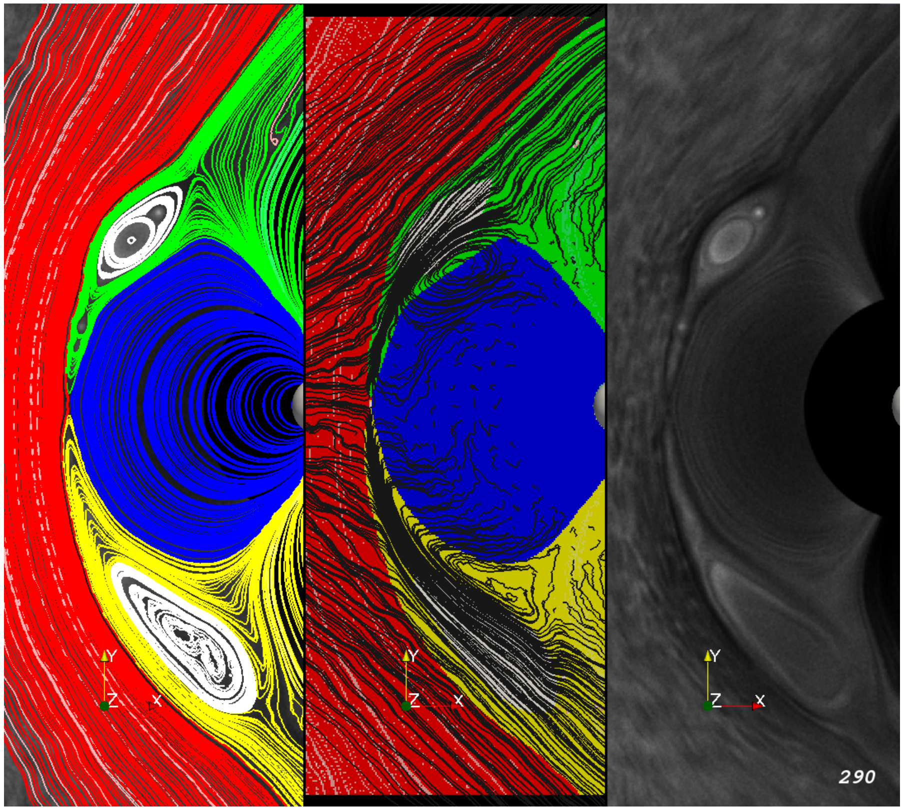




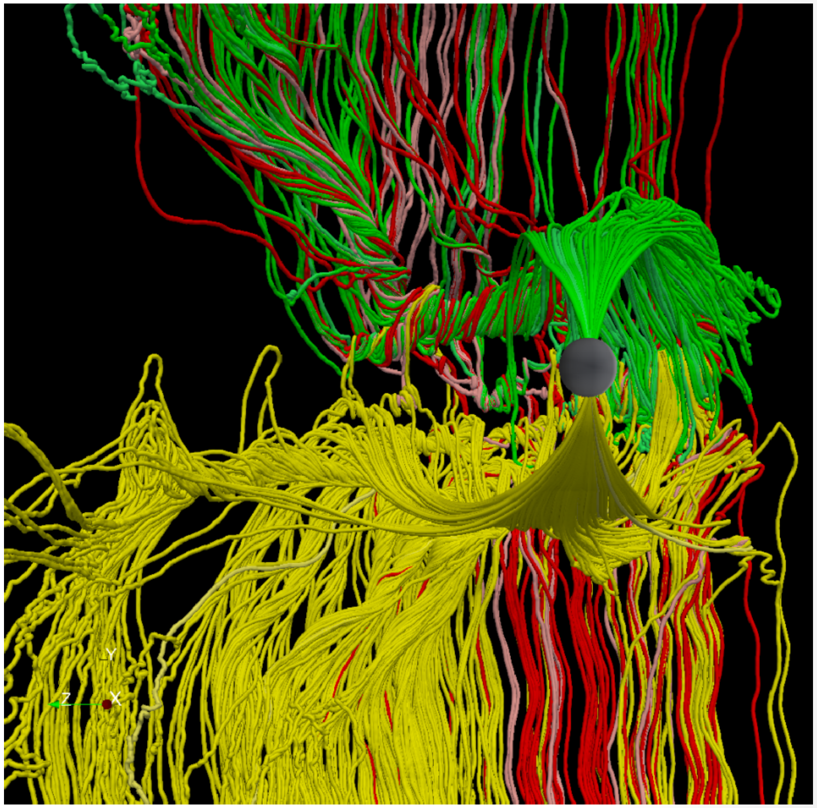
Density of particles within this energy band

3D Simulations of the Magnetosphere

5,000 cores

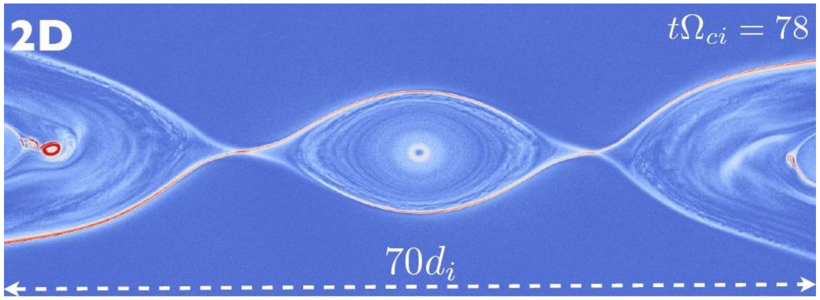


100 K cores

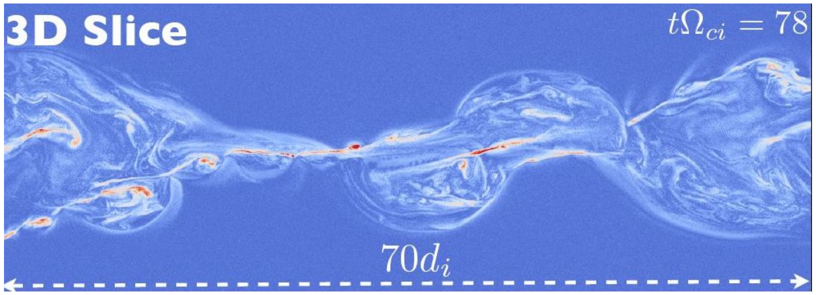


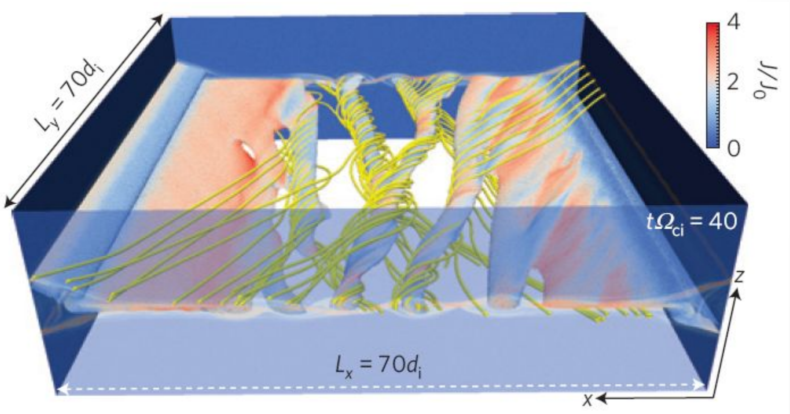
First Glimpse of 3D Effects

2D: Billion Particles



3D: Trillion Particles





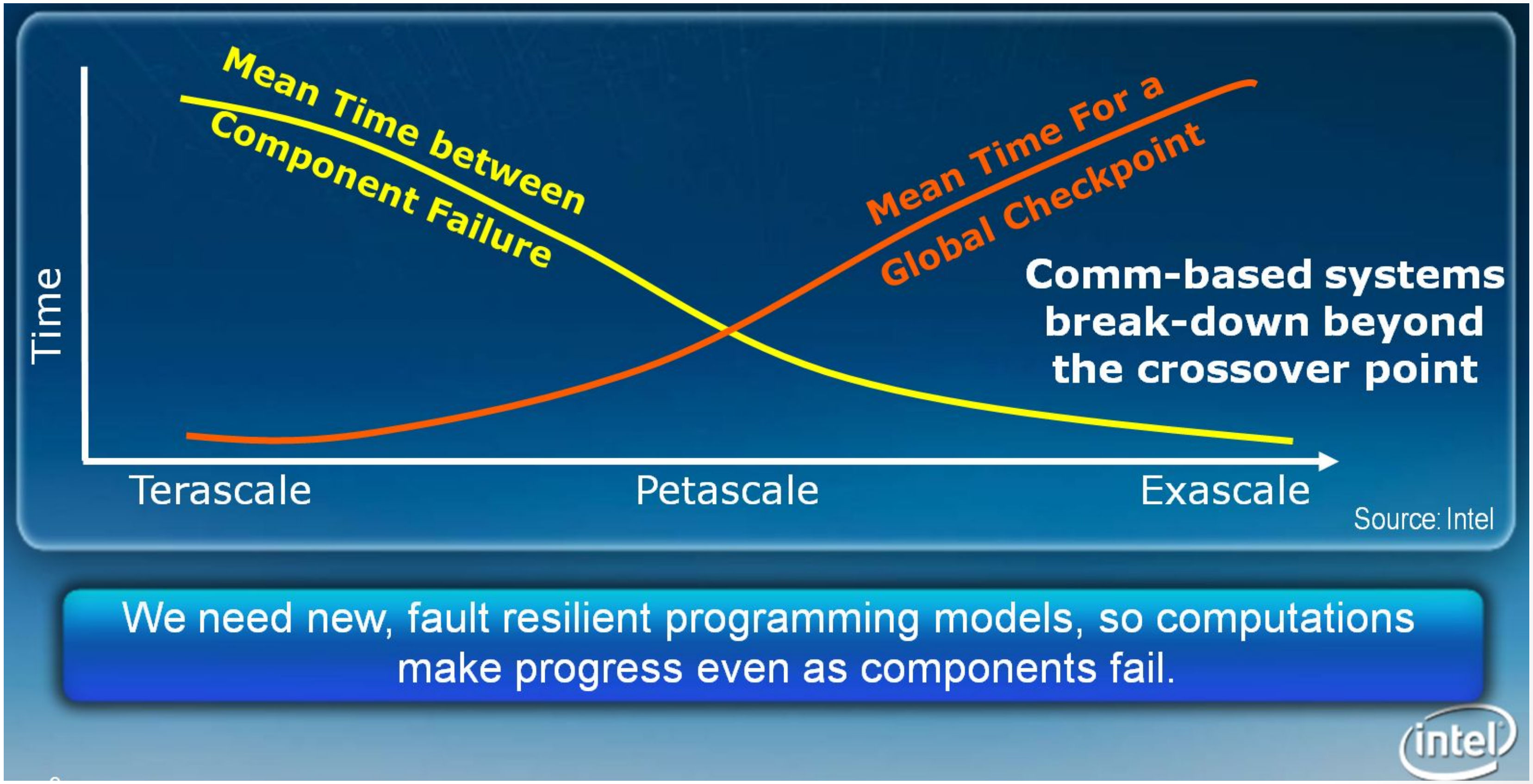
Daughton et al., Nature Physics, 2011

Obstacles in Extending to Exascale

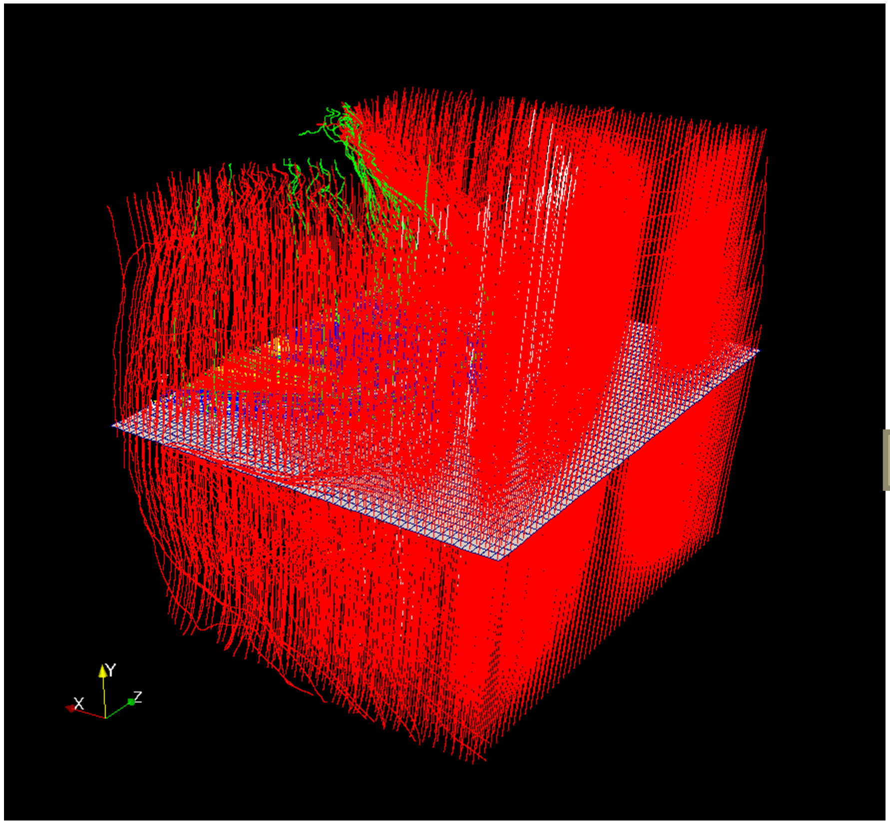
- Dynamic load balancing of particles
- Efficient checkpointing
- Fast, scalable priority queues
- Data Analysis / Visualization / Sharing
- Intelligent Fault Tolerance
- Archiving / Scratch Space

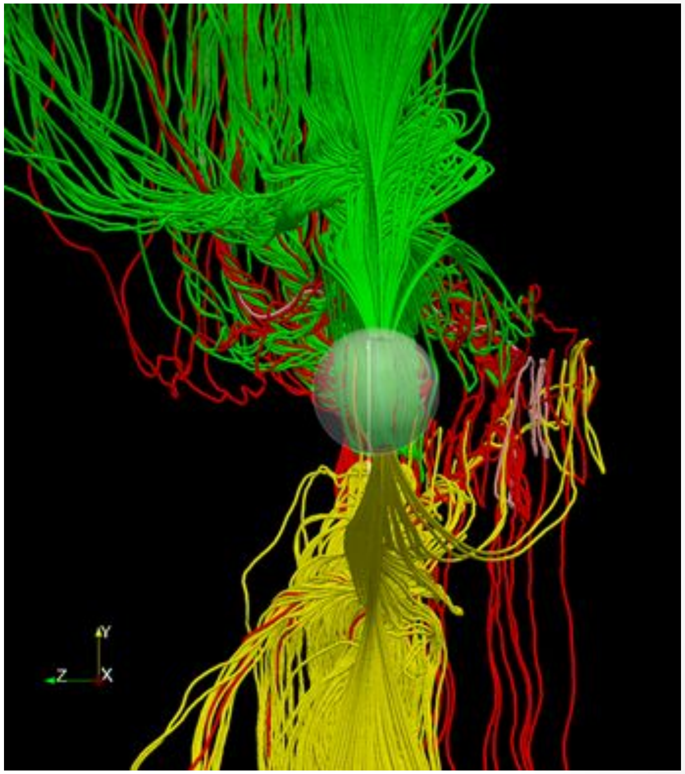
Checkpointing

One restart dump took nearly 1 hour for our 98 K core run.
Checkpointing/restart not viable at exascale.

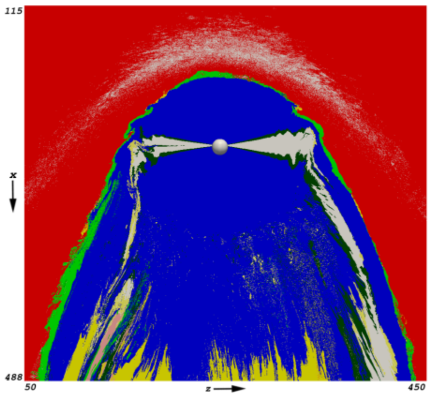


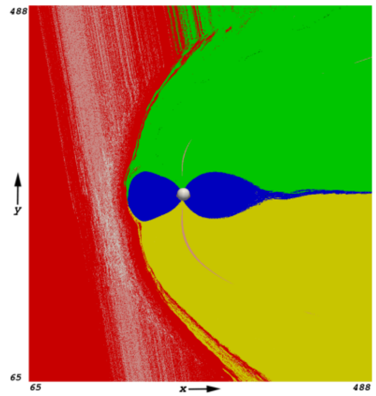
Information Overload : Looking for FTEs





SCIVIZ: A Physics Mining Tool
(scientific viz + data mining + computer vision)



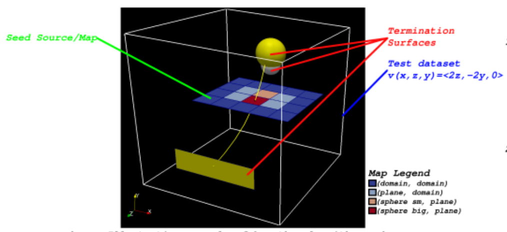


Class	Color	Description
d ⇔ d		solar wind
ø ⇔ d		
i ⇔ d		
n ⇔ s		closed magnetosphere
n ⇔ d		northern hemisphere connected
n ⇔ ø		
n ⇔ i		
n ⇔ n		
s ⇔ d		southern hemisphere connected
s ⇔ ø		
s ⇔ i		
s ⇔ s		
ø ⇔ ø		field null
i ⇔ ø		
i ⇔ i		

Legend

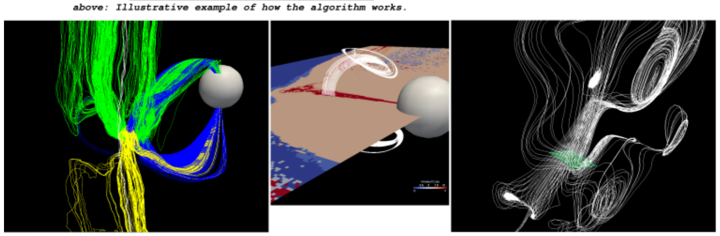
d=simulation domain, n=northern hemisphere, s=southern hemisphere, o=field null, i=short integration

Two magnetic field topology maps of the matgentosphere in a global hybrid simulation. Left: map in the equatorial plane. Right: Map in the noon-midnight meridional plane. The maps are generated by specifying two hemispherical surfaces centered on the Earth's north and south poles as termination surfaces and tracing field lines from each pixel in the map asigning a color based on which of the surfaces the lines intersect.

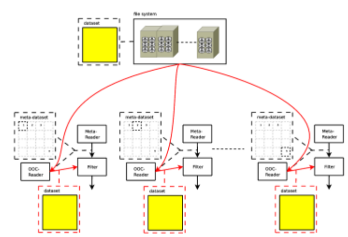


above: Illustrative example of how the algorithm works.

1) The user configures three inputs interactively in GUI
   a) A dataset containing a vector field.
   b) A set of termination surfaces. For magnetospheric visualization these will be the north and south polar caps.
   c) A seed source, typically a surface.
2) For each cell in the seed source, the center is used as a seed point and a stream line is traced. The cell is then asigned a color based on which termination surfaces (if any) the stream line intersects.

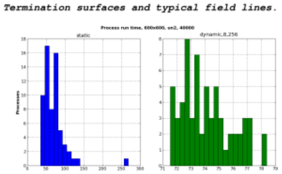


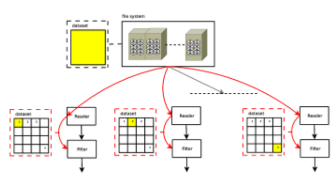
Examples:
left: Termination surfaces and typical field lines. center: Short integation. right: Nulls/short integation.



Left: ParaView data-parallel visualization pipeline. Filter makes a single IO request for data prior to its execution.

Top: Our out-of-core (OOC) data parallel pipeline. Meta reader sits as a proxy for the reader. ParaView thinks it's a reader, but it doesn't read any data. Instead the meta-reader inserts an OOC reader object in the pipeline information, downstream filters then can make multiple IO requests as needed during execution.





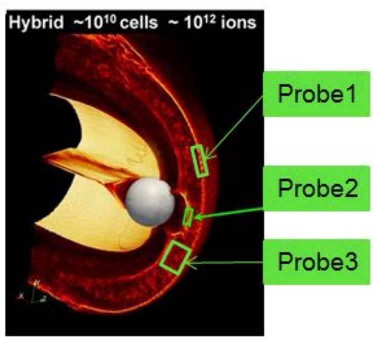
left: Static distribution of seed points doesn't ballance the work load well, one process has much more work than the others.

In Situ Visualization

- For many of our analysis, we need high time resolution data dumps. However, this is not practical since each data dump can take over 10 TB.

- In situ visualization provides a possible solution.

In Situ Visualization Using Intelligent Probes



Features of Intelligent Probes

Targeted data collection
Ability to mimic spacecraft-like trajectories
Creation of instrument-like products
Generation of high resolution data products

Selective Outputs

Images and visualizations
Data from probes (less raw data)
Statistics

Enables

Direct comparison with spacecraft data
Orbit optimization
Instrument design and testing

Barrier to Adoption of Current Technologies

- GPUs: CUDA implementation is a major effort
  Not clear whether CUDA will survive to exascale
  Will it be a standard across other accelerators?

- MIC: Codes work out of the box but will it be around?

- Data movement across host and accelerators
  That bandwidth is slower in current technologies

Experience With Hopper

- Website (nim.nersc.gov) is easy to use
- Francesca Verdier – Great POC
- Working at NERSC and collaborating with NERSC analytics and ExaHDF5 team has been very beneficial & has led to a number of collaborative scientific papers
- The work of visualization group is so essential to our progress that they are coauthors on our papers

Experience With Hopper

- NERSC is DOE's production computing facility, hence scheduling full-scale, I/O intensive runs are challenging on Hopper.

- In our work with the ExaHDF5 team, we have had to request NERSC staff for scheduling full-scale 120,000 core runs; which were rather hard to schedule and troubleshoot.

- We were able to do these runs only once in a 2 or 4-week period, which extended the science project over a period of 6-9 months.

- The ExaHDF5 team had to ask for repeated extensions for a 500TB disk quota on NERSC scratch system. We believe that it is challenging for routine NERSC users to push full-scale I/O intensive jobs through on hopper, without hand-holding from NERSC staff.

Backup Slides

The Hybrid Approximation

- Ions: kinetic particles
- Electrons: massless, quasi-neutral ($en_e = q_i n_i$) fluid
- Electromagnetic fields:

  - Faraday's law

  $$(\partial/\partial t)\,\mathbf{B} = -c\ \nabla \times \mathbf{E}$$

  - Ampere's law

  $$\nabla \times \mathbf{B} = 4\pi\,\mathbf{J}/c = 4\pi\, q_i n_i\,(\mathbf{v}_i - \mathbf{v}_e)/c$$

  - Electric field from electron momentum equation

  $$\mathbf{E} = -\mathbf{v}_i \times \mathbf{B}/c - \nabla \mathbf{p}_e/(q_i n_i) - \mathbf{B} \times (\nabla \times \mathbf{B})/(4\pi\, q_i n_i) - \eta \mathbf{J}$$

Recommendations

- 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
- Fast archiving systems
- Efficient I/O
- Establish a POC for large users
- Support from visualization experts is critical
  (support from Visualization group at LBNL including ExaHDF5 team has been essential to our progress)

Wish List For Current Programming Languages, Models & Software Abstraction

- Standardization/acceptance of PGAS type languages (for dealing with particle load imbalance) would be of interest

- General load balancing tools taking as input a dynamic "load distribution", including tools that would facilitate computing this distribution (i.e., CPU load as a function of cell) at runtime

- Optimized priority queues

- Fault-tolerance (e.g., intelligent checkpointing)