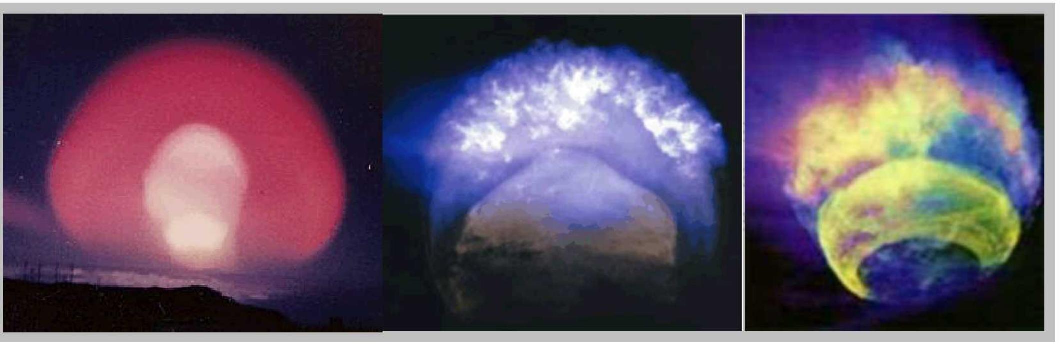
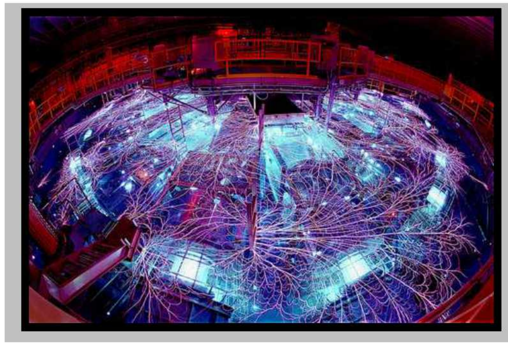
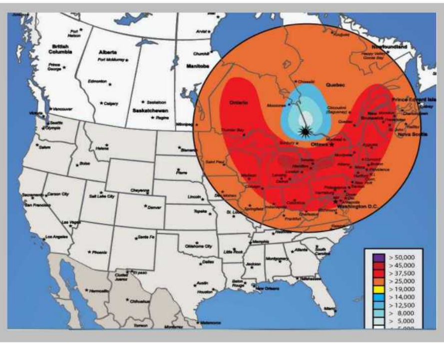
SAND2019-3233PE







EMPIRE: Sandia's Next Generation Plasma Tool

Feburary 6, 2019

Matt Bettencourt, Sidney Shields, Kris Beckwith, Keith Cartwright, Eric C. Cyr, Richard Kramer, Paul Lin, Billy McDoniel, Sean Miller, Roger P. Pawlowski, Edward Phillips, Nathan V. Roberts

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Background

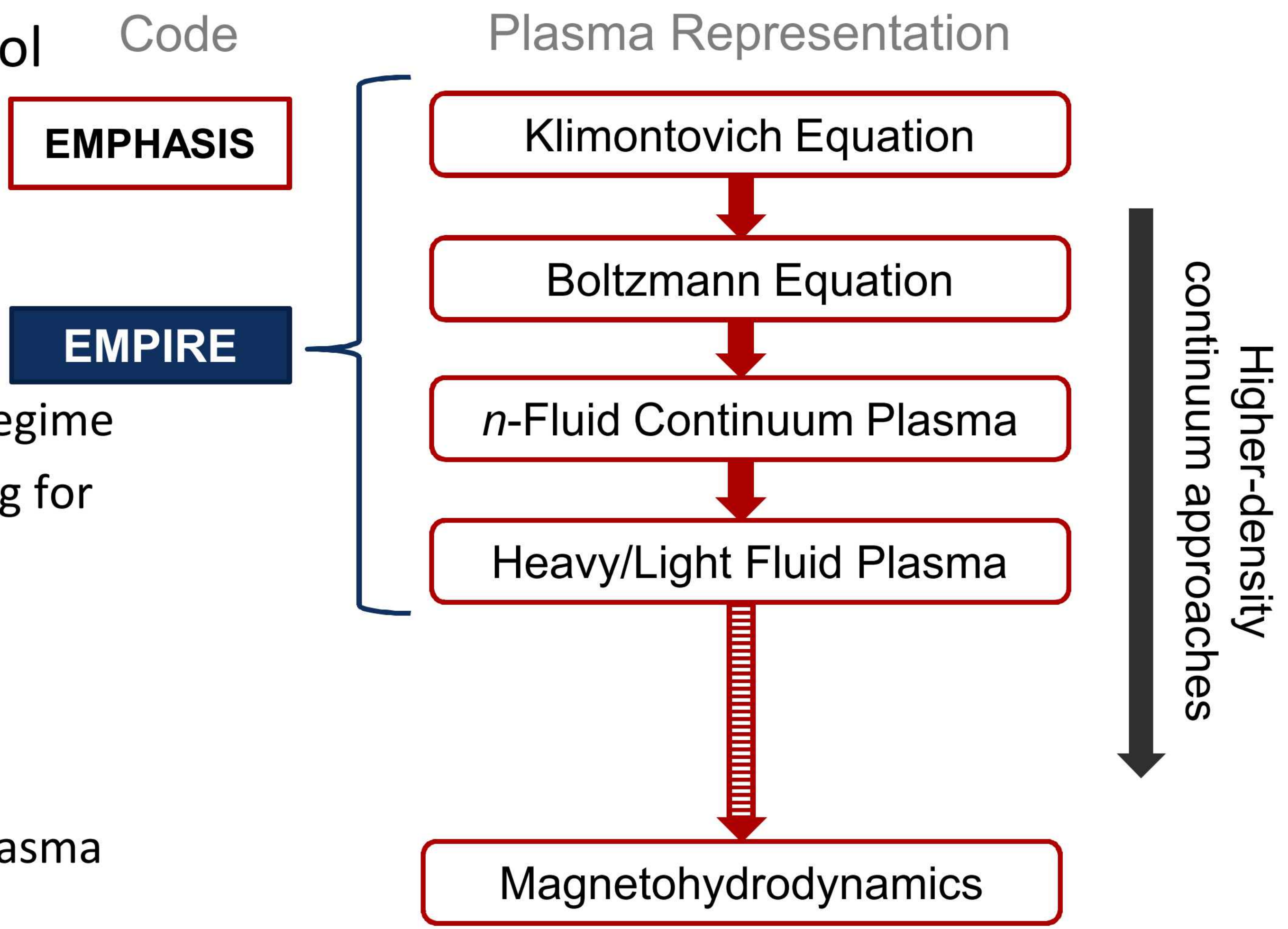
- US DoE Advanced Scientific Computing (ASC) is refreshing their major computing resources
  - FY16 Trinity – Split Intel Haswell and KNL cluster
  - FY19 Sierra – IBM Power with Nvidia accelerators
  - FY21 ATS-3 – In bidding process
- The DoE realized that current production codes wouldn't run on these modern architectures
- New program element to develop new applications
  - Advanced Technology Demonstration and Mitigation – ATDM
  - ATDM has since moved into the exascale computing initiative – ECP
- ATDM culminates in a level 1 milestone in FY20

EMPIRE

- Sandia has two applications which are designed to meet the goals of the ATDM project
  - EMPIRE – Sandia's electromagnetic plasma simulation tool
  - SPARC – Sandia's hypersonic reentry code
- EMPIRE – ElectroMagnetic Plasma In Realistic Environments
- Code being built up from scratch starting in 2015
  - Well, built using an existing component infrastructure inside of Trilinos
- Goal – Hybrid PIC-Fluid simulation for a 30B element mesh simulation on half of Sierra and one other capability cluster (Trinity or Astra)
- Build on the SNL component architecture

EMPIRE expands simulation capability across the plasma physics spectrum

- EMPHASIS: Current production tool
- EMPIRE adds new capabilities:
  - Written for advanced computing architectures
  - Expanded particle-based modeling regime
  - Full continuum fluid plasma modeling for high density plasmas
  - Hybrid particle-fluid modeling for intermediate densities
- EMPIRE will enable:
  - Higher fidelity modeling of critical plasma applications
  - Towards exascale simulation

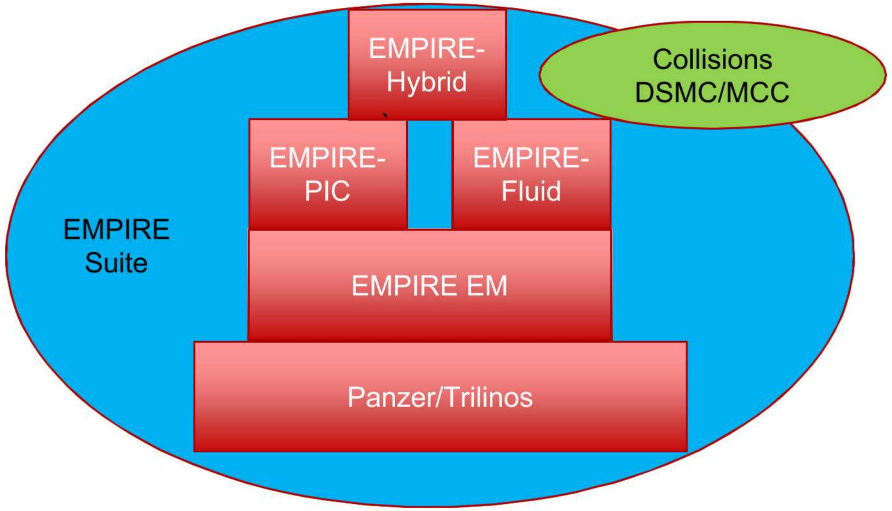


EMPIRE Overview

- EMPIRE is Sandia's "next generation" plasma simulation tool
- Unstructured mesh
- Finite element discretization
- Particle in cell or multi fluid discretizations
- Developed for performance portability
- Uses implicit compatible discretization for Maxwell's equation
  - Different time integration methods
  - Requires a matrix inversion each time-step
- Uses projected electric fields to push particles
  - L2 or lumped
- Supports 2 and 3 dimensional models
- Particles scale to ¼M cores on Trinity

EMPIRE-EM

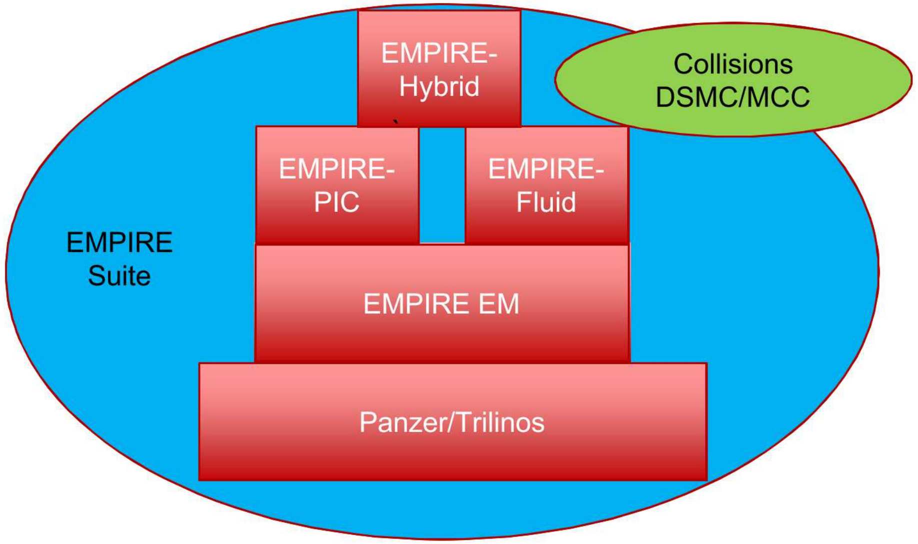
- EMPIRE-EM is the electromagnetic solver, and …
  - Time integration routines
  - Diagnostics
  - Parsing utilities
  - Output utilities
  - Utility classes



- EMPIRE-EM holds everything that is used by multiple modules
- EMPIRE-PIC and EMPIRE-Fluid build upon this core piece of code

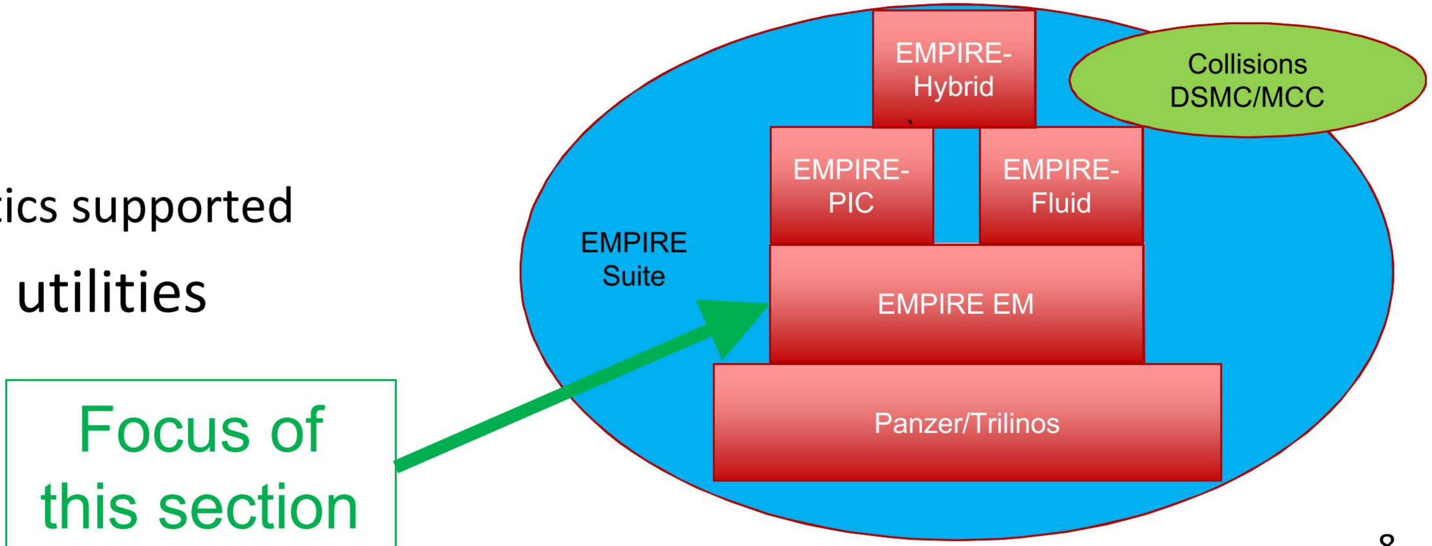
EMPIRE Hierarchy

- EMPIRE is built upon the Trilinos library
  - Panzer is the top level library which provides all the FEM discretization tools
    - Simplifies the assembly of linear and non-linear FEM problems
    - Assembles residual and Jacobian systems
  - Trilinos also provides the linear solver technology
    - Belos – Krylov solvers
    - MueLu – Multilevel preconditioner
    - Teko – Blocked linear system library
    - Tpetra – Sparse distributed matrices and vectors
    - Tempus – General time integration package
  - Kokkos – Portable threading library for CPU/GPU systems



EMPIRE Base Component

- EMPIRE base component provides the basic infrastructure for EMPIRE
- Electromagnetic and electrostatic discretizations and solvers
  - Matrices are built using the Panzer component of Trilinos
  - Decompose a complex model into a graph of simple kernels
  - Assemble these operations into a directed acyclic graph (DAG)
  - Evaluate the graph building Jacobian terms through automatic differenation
- Time integration infrastructure
- Diagnostic infrastructure
  - Point, Line, and volumetric diagnostics supported
- Parsing and input deck validation utilities



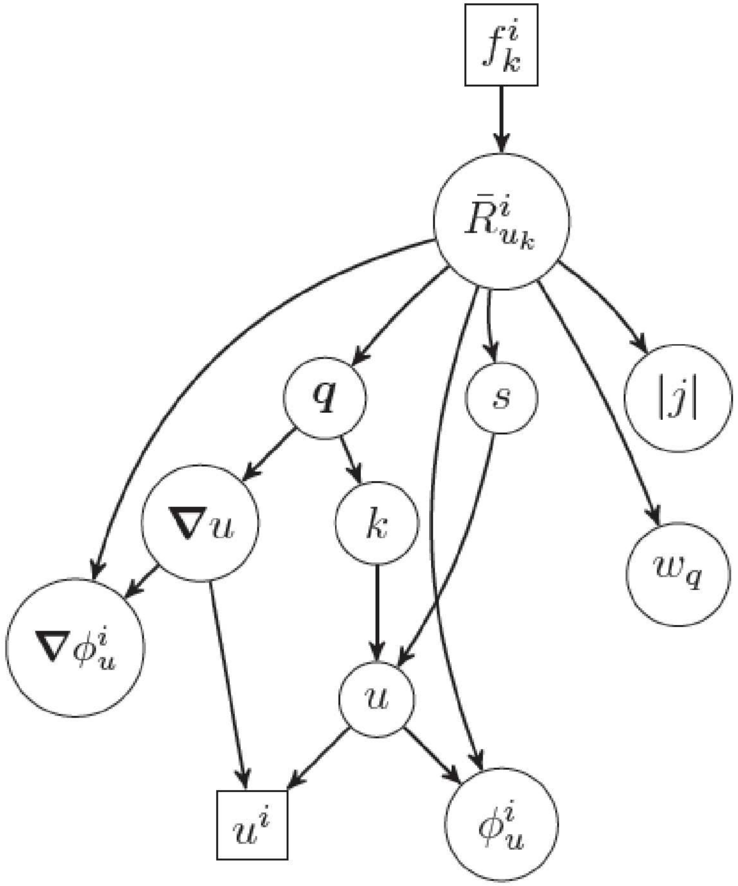
Equation Sets
DAG-based Expression Evaluation

- Decompose a complex model into a graph of simple kernels (functors)
  - Decomposition is NOT unique
- Supports rapid development, separation of concerns and extensibility.

- A node in the graph evaluates one or more **fields**:
  - Declare fields to evaluate
  - Declare dependent fields
  - Function to perform evaluation

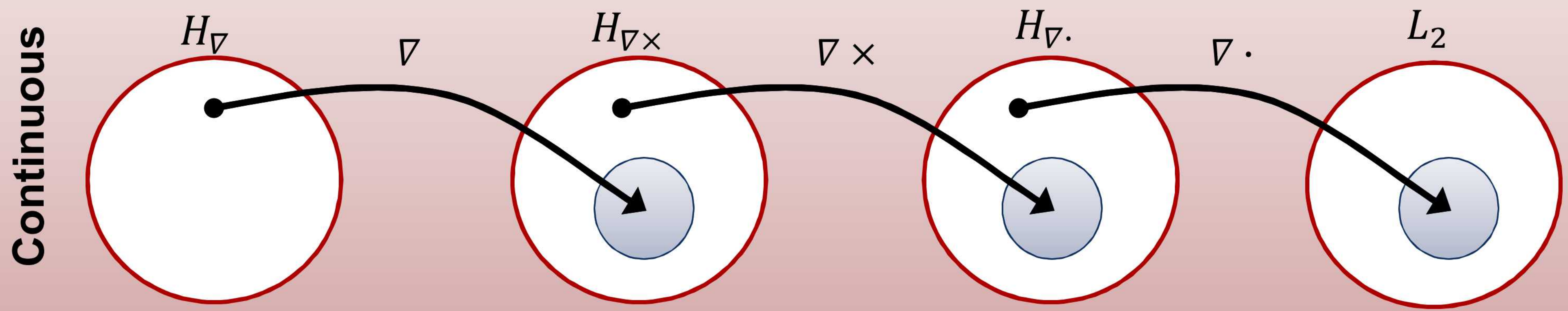
- Separation of data (Fields) and kernels (Expressions) that operate on the data
  - Fields are accessed via multidimensional array interface (shards or kokkos)

$$R_u^i = \int_\Omega \left[\phi_u^i \dot{u} - \nabla \phi_u^i \cdot \boldsymbol{q} + \phi_u^i s\right] \, d\Omega$$

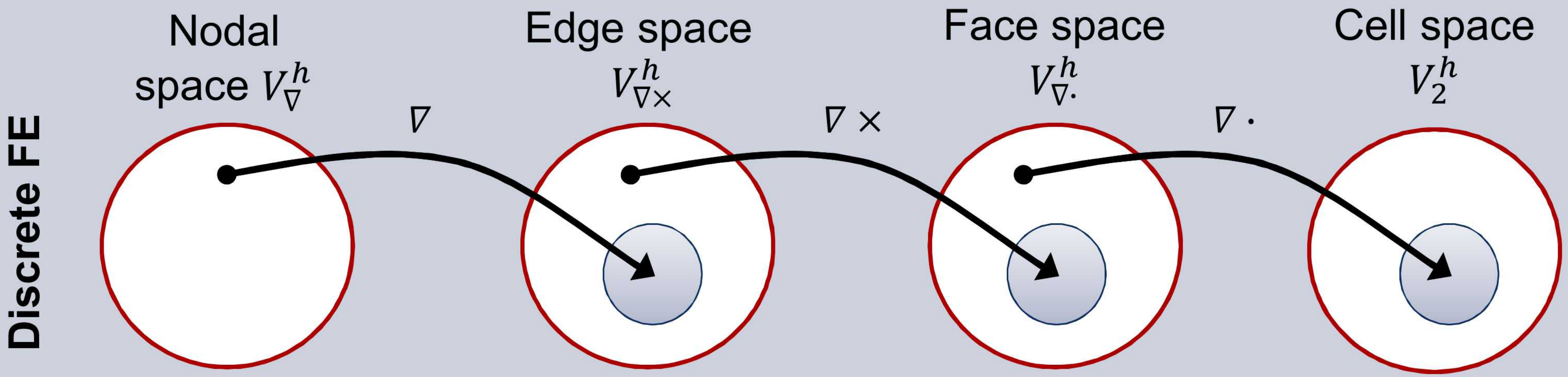


Handling Maxwell Equation Involutions

Function spaces posses an exact sequence property where the derivative maps into the next space, e.g.:



Exact sequence finite elements have been constructed[1] (note $V_*^h \subset H_*$):



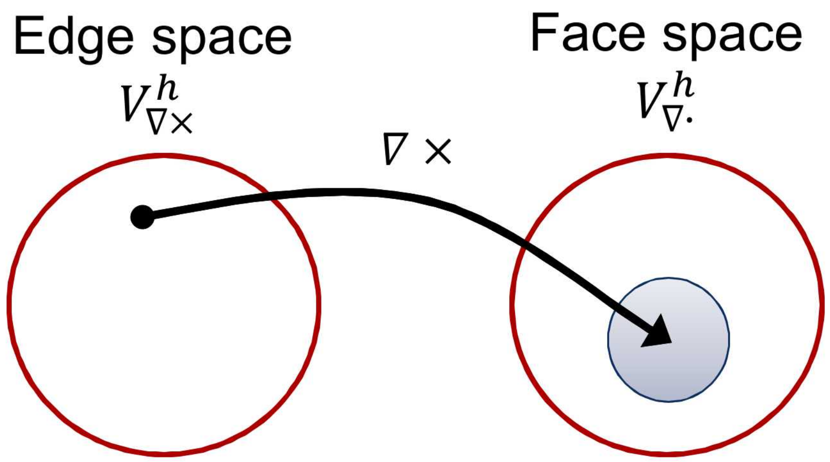
[1]Bochev, P., Edwards, H. C., Kirby, R. C., Peterson, K., & Ridzal, D. (2012). Solving pdes with intrepid. *Scientific Programming*, *20*(2), 151-180.

Enforcing no magnetic monopoles

Continuous:

$$\nabla \cdot \left(\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} \right) = 0 \Rightarrow \nabla \cdot \partial_t \mathbf{B} = 0$$

$$\Rightarrow \nabla \cdot \mathbf{B} = 0 \text{ (assuming satisfied at } t = 0)$$



Discrete FE:

Let $\mathbf{B}^h \in V^h_{\nabla\cdot}$ and $\mathbf{E}^h \in V^h_{\nabla\times}$, then the argument is straightforward and follows the continuous case:

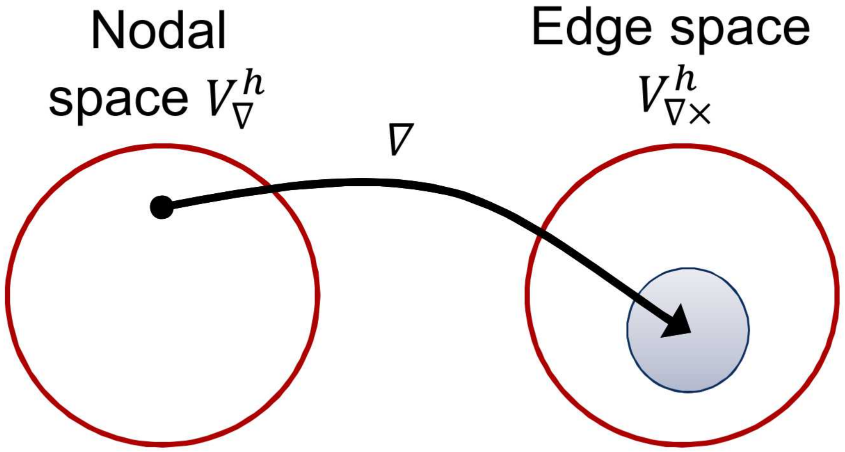
$$\nabla \cdot \left(\frac{\partial \mathbf{B}^h}{\partial t} + \nabla \times \mathbf{E}^h \right) = 0 \Rightarrow \nabla \cdot \partial_t \mathbf{B}^h = 0$$

$$\Rightarrow \nabla \cdot \mathbf{B}^h = 0 \text{ (assuming satisfied at } t = 0)$$

Note that no magnetic monopoles is enforced strongly

Enforcing Gauss Law (CG fluids, DG in progress)

Continuous:



$$\nabla \cdot (\partial_t \mathbf{E} - c^2 \nabla \times \mathbf{B} = -\frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \rho_\alpha \mathbf{u}_\alpha)$$

$$\Rightarrow \partial_t (\nabla \cdot \mathbf{E}) = -\frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)$$

Use continuity

$$\Rightarrow \partial_t (\nabla \cdot \mathbf{E}) = \frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \partial_t \rho_\alpha \Rightarrow \nabla \cdot E = \frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \rho_\alpha$$

Discrete FE (ignoring BCs):

Let $\mathbf{E}^h \in V^h_{\nabla\times}$ and $\rho^h_\alpha, \rho_\alpha \mathbf{u}^h_\alpha, \mathcal{E}^h_\alpha \in V^h_\nabla$, then using the weak forms:

Ampere's Law — Continuity

$$\int \partial_t \mathbf{E}^h \cdot \psi^h - \mathbf{B}^h \cdot \nabla \times \psi^h = -\frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \int \rho^h_\alpha \mathbf{u}^h_\alpha \cdot \psi^h \ \forall \psi^h \in V^h_{\nabla\times}, \quad \int \partial_t \rho^h_\alpha \phi^h = \int \rho^h_\alpha \mathbf{u}^h_\alpha \cdot \nabla \phi^h \ \forall \phi^h \in V^h_\nabla$$

Use exact sequence on the test space and substituting continuity into Ampere's:

Apply Exact Sequence: $\nabla \phi^h \in V^h_{\nabla\times}$ — Apply Continuity

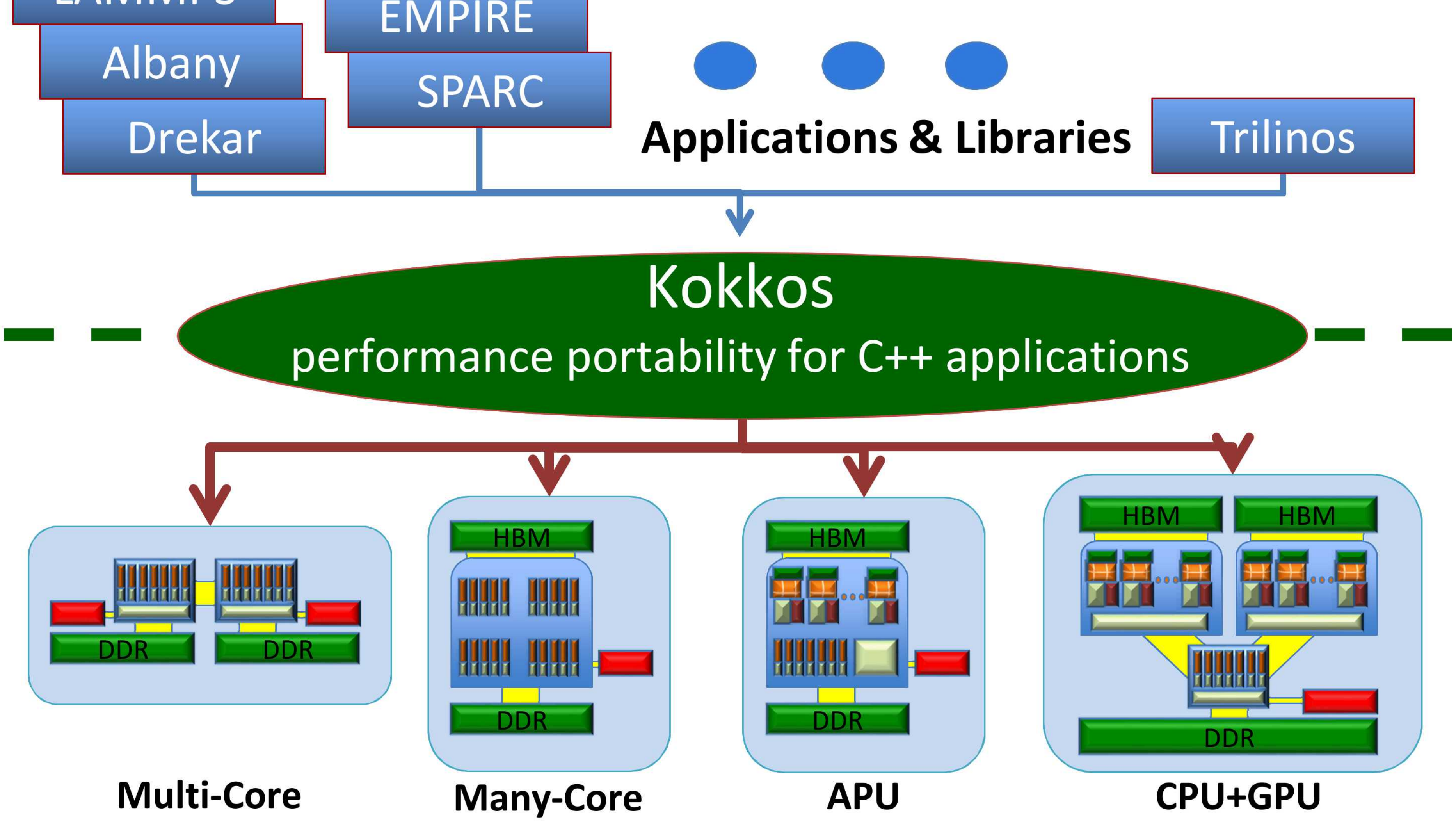
$$-\int \partial_t \mathbf{E}^h \cdot \nabla \phi^h = \frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \int \rho^h_\alpha \mathbf{u}^h_\alpha \cdot \nabla \phi^h = \frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \int \partial_t \rho^h_\alpha \phi^h \Rightarrow -\int \mathbf{E}^h \cdot \nabla \phi^h = \frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \int \rho^h_\alpha \phi^h$$

Note that Gauss' law is enforced weakly

General Performance Strategy

- EMPIRE has adopted an MPI+X approach
  - MPI is used for across node or sockets
  - Some threading approach is used on node
- Kokkos provides an abstraction layer for X
  - In Kokkos one can choose Pthreads, OpenMP or Cuda as X
- This allows us to write a single code and achieve portable execution
  - Work and specialization is still required to achieve performance on different systems
- Different aspects of the algorithm are broken into a per-particle or per-element block
  - Blocks are then distributed across different threads via kokkos

What is Kokkos?



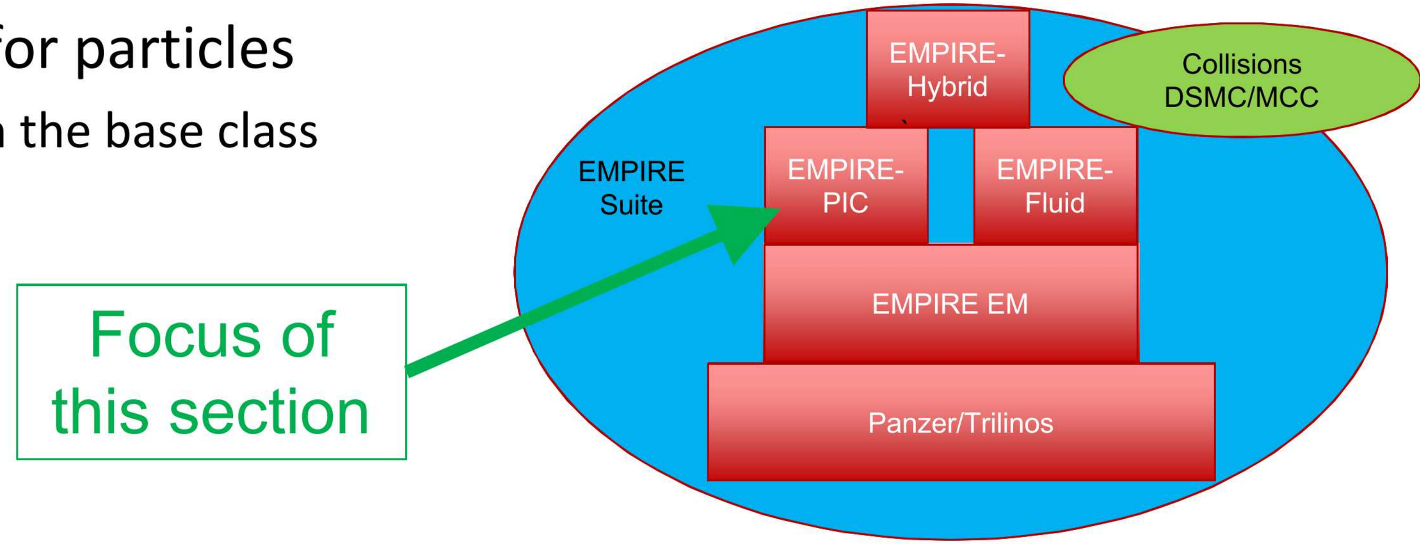
Cornerstone for performance portability across next generation HPC architectures at multiple DOE laboratories, and other organizations.

Abstractions
Patterns, Policies, and Spaces

- Parallel Pattern of user's computations
  - parallel_for, parallel_reduce, parallel_scan, task-graph, ... *(extensible)*
- Execution Policy tells *how* user computation will execute
  - Static scheduling, dynamic scheduling, thread-teams, ... *(extensible)*
- Execution Space tells *where* computations will execute
  - Which cores, numa region, GPU, ... *(extensible)*
- Memory Space tells *where* user data resides
  - Host memory, GPU memory, high bandwidth memory, ... *(extensible)*
- Layout (policy) tells *how* user array data is laid out
  - Row-major, column-major, array-of-struct, struct-of-array … *(extensible)*
- Differentiating: Layout and Memory Space
  - Versus other programming models (OpenMP, OpenACC, …)
  - Critical for performance portability …

EMPIRE-PIC Component

- EMPIRE-PIC builds on the base components adding a particle discretization
- Structure mimics the base component
- Augments parsing and diagnostics routines
- Augments the time integration routines
  - Standard leap-frog implemented
- Contains all the boundary, loading for particles
  - Field based boundary conditions live in the base class



Particle Formulation for Plasmas

Particle in Cell (PIC) is a solution technique which uses the particle formulation for plasmas

Assumes that the particle distribution is represented as a delta function of particles in both space and time

$$f = \sum \delta(x - x_i)\delta(v - v_i)$$

Then Newton's laws are applied to these particles

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i; \qquad \frac{d\vec{v}_i}{d\,t} = \frac{q_i}{m_i}\left(\vec{E}(x_i) + v_i \times \vec{B}(x_i)\right)$$

Then the distribution is updated

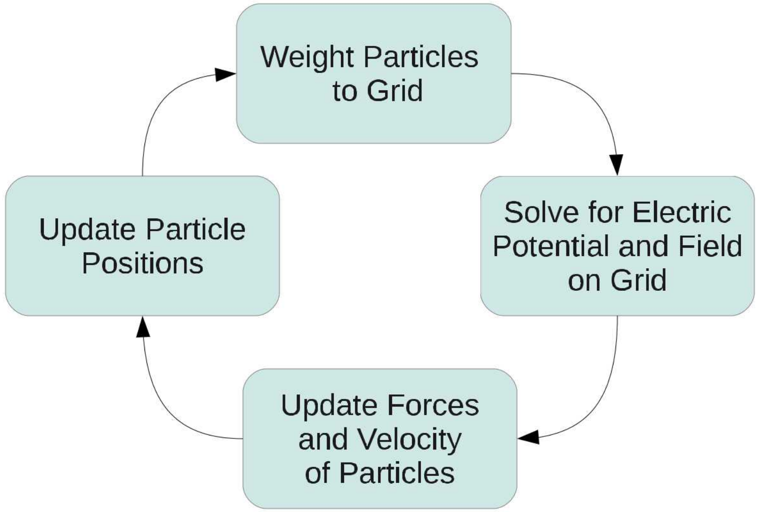
$$\frac{\partial f_i}{\partial t} + \frac{\vec{v}_i}{m} \cdot \nabla f_i + \frac{q_i}{m_i}\left(\vec{E}(x_i) + v_i \times \vec{B}(x_i)\right)\frac{\partial f_i}{\partial \vec{v}} = 0$$

Resulting in the Klimontovich equation

This formulation is used by PIC

Particle In Cell

- PIC starts with the Klimontovich equation and simplifies
  - Fields are computed on a mesh and interpolated to the particles
  - Particles represent a large number of physical particles
  - Particle motion coupled back to the field solve through charges and currents
- Typically these equations are solved using leap-frog integration on a regular mesh via MPI
- EMPIRE uses:
  - Unstructured mesh
  - Finite element method (FEM)
  - General time integration (tempus)
  - MPI+Kokkos threading



Finite Element and PIC

- The basics of the particle parts of PIC are unchanged between FEM and FDTD
  - Particles are accelerated and moved the same way
  - Current is still weighted to the mesh
- Field solve is totally different though - Use the weak form
  - Starting with Ampere's equation
  - Multiply by a test function and integrate
  - Expand solutions in test function spaces
    - Integrate curl by parts
  - Generates a matrix equation
  - Similar matrix for B equation

$$\frac{\partial D}{\partial t} = \nabla \times H - J$$

$$\int_V \left(\frac{\partial D}{\partial t} - \nabla \times H + J\right)\hat{e_i}dv$$

$$\int_V \sum\left(\frac{\partial d_j}{\partial t}\hat{e_j} - h_j\hat{b}_j\nabla \times +J\right) \quad \hat{e_i}dv$$

$$M_D\frac{\partial}{\partial t}d = K_H h - \int_V J\hat{e_i}dv$$

FEM Current weighting

Sandia National Laboratories

- The original charge conserving PIC current weighting used a "charge through faces" argument to develop weighing
- Weak form gives a more formal idea of current weighting

$$\int_V J\hat{e}_i dv = \sum_p \int_t^{t+\Delta t} \int_V q_p v_p \cdot \hat{e}_i dv\, dt$$

- But the velocity is a delta function in space, making the integral a simple evaluation

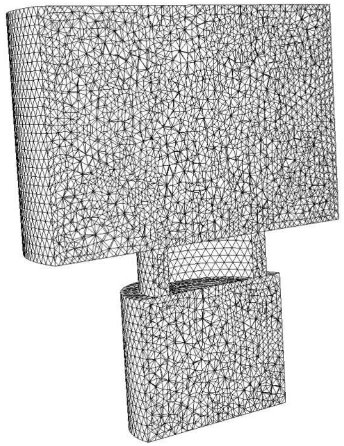
$$\int_V J\hat{e}_i dv = \sum_p \int_t^{t+\Delta t} q_p v_p \hat{e}_i(x_p(t))\, dt$$

- For tets, this reduces to the midpoint rule of integration
- Artifact for the current weighting
  - If you place a particle in the domain and move it, an immobile ghost charge is left behind
  - Fluid code as well

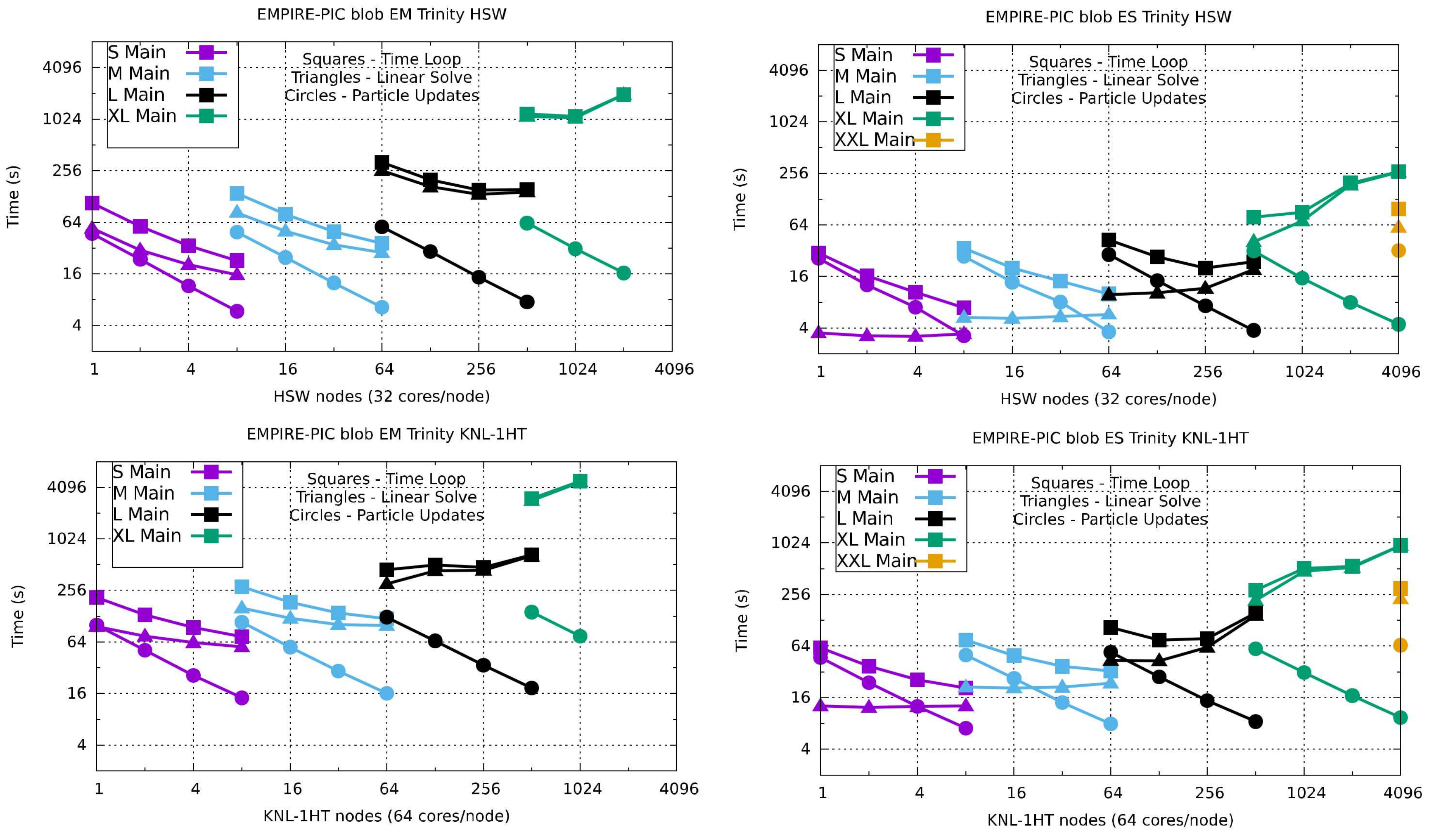
EMPIRE Performance Setup

- Requirement – EMPIRE needs 10 time-steps/second
- EMPIRE was ran on Trinity at the scale of 4096 nodes for 100 steps
  - Haswell - 2 MPI ranks/node, 16 threads per rank – 128k cores
  - Knights Landing – 4 MPI ranks/node, 16 threads per rank – 256k cores
- Problem used the same geometry
  - Cubit generated tetrahedral mesh "blob"
  - Problem was scaled up in particle and element count
  - Problem was run electrostatically and electromagnetically

Size	# of Elements	# of Nodes	# of Edges	# of Faces	# of Particles
S	337k	60k	406k	683k	16M
M	2.68M	462k	3.18M	5.40M	128M
L	20.7M	3.51M	24.4M	41.6M	1B
XL	166M	27.9M	195M	333M	8.2B
XXL	1.332B	223M	1.56B	2.67B	65.6B



EMPIRE Overall Performance

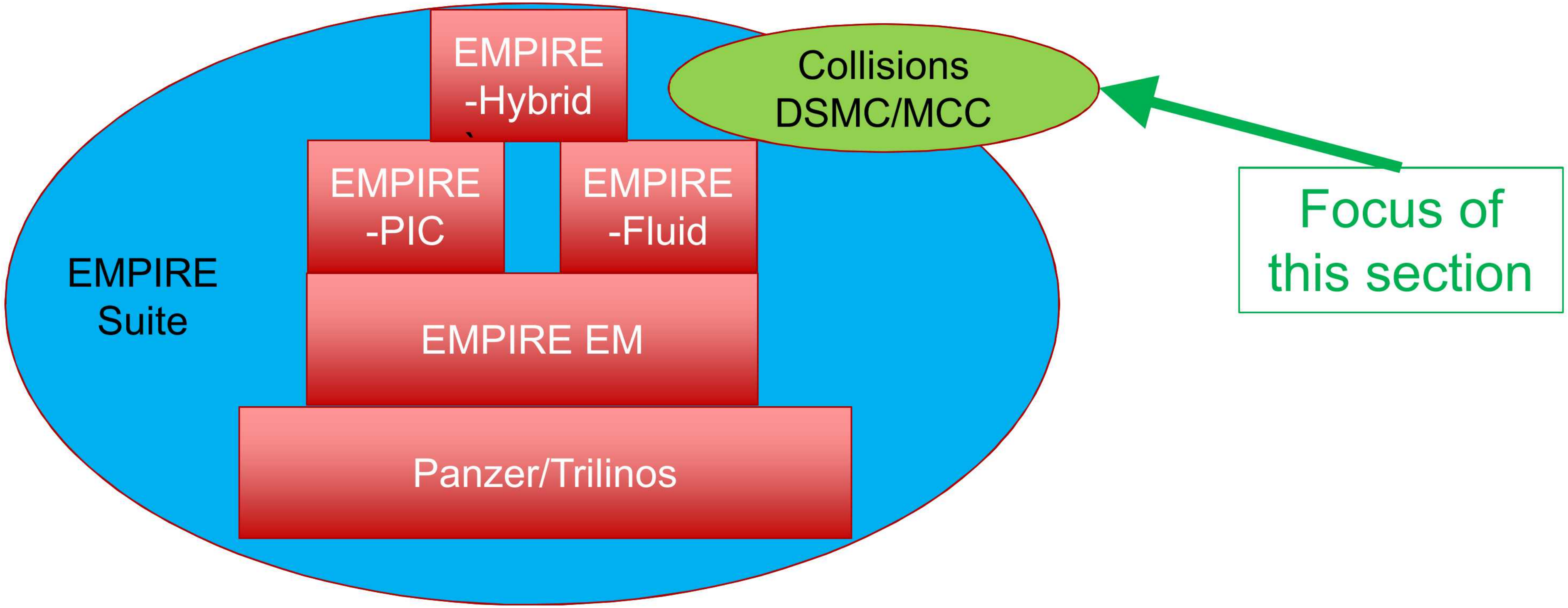


Plots of EMPIRE scaling, left EM, right ES, top HSW, bottom KNL

EMPIRE-PIC Summary

- EMPIRE-PIC is approaching the capability of our current production code
  - Many many important features and diagnostics are yet implemented
- EMPIRE-PIC has been shown to scale to 1/4M cores
- EMPIRE-PIC is performance portable
  - Recompile for different backends, main code remains the same
- Verification suite is building
  - Not shown but linear problems showing correct convergence
- Validation effort underway

EMPIRE: A hierarchy of capabilities



- EMPIRE's goal: Accurately simulate plasmas across regimes on next-generation exascale computing platforms
- Expands the range of critical plasma applications that we can address with high confidence and fidelity

SPIN Sandia's Particle Interaction Library

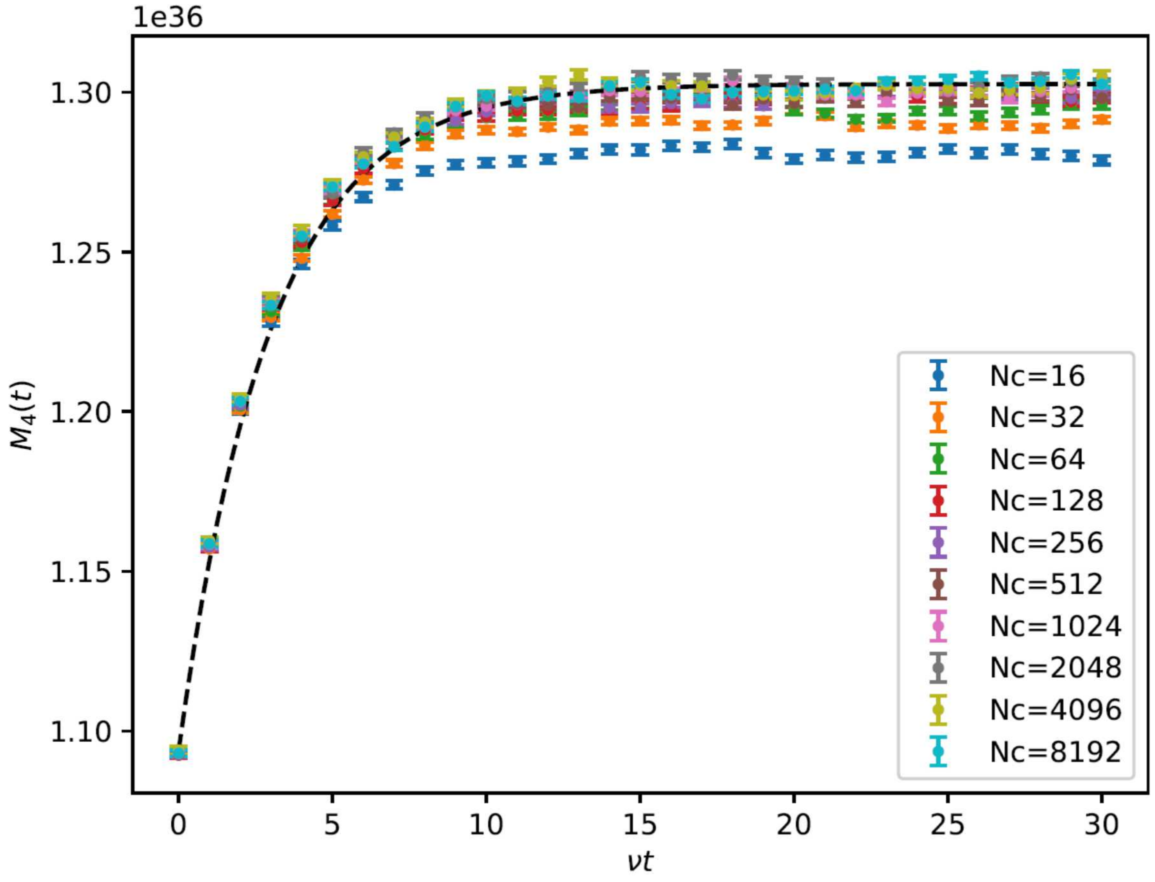
SPIN handles chemistry and inner particle forces

Complex collision types:

- Ionization
- Excitation
- Dissociation / chemical reaction with >2 products

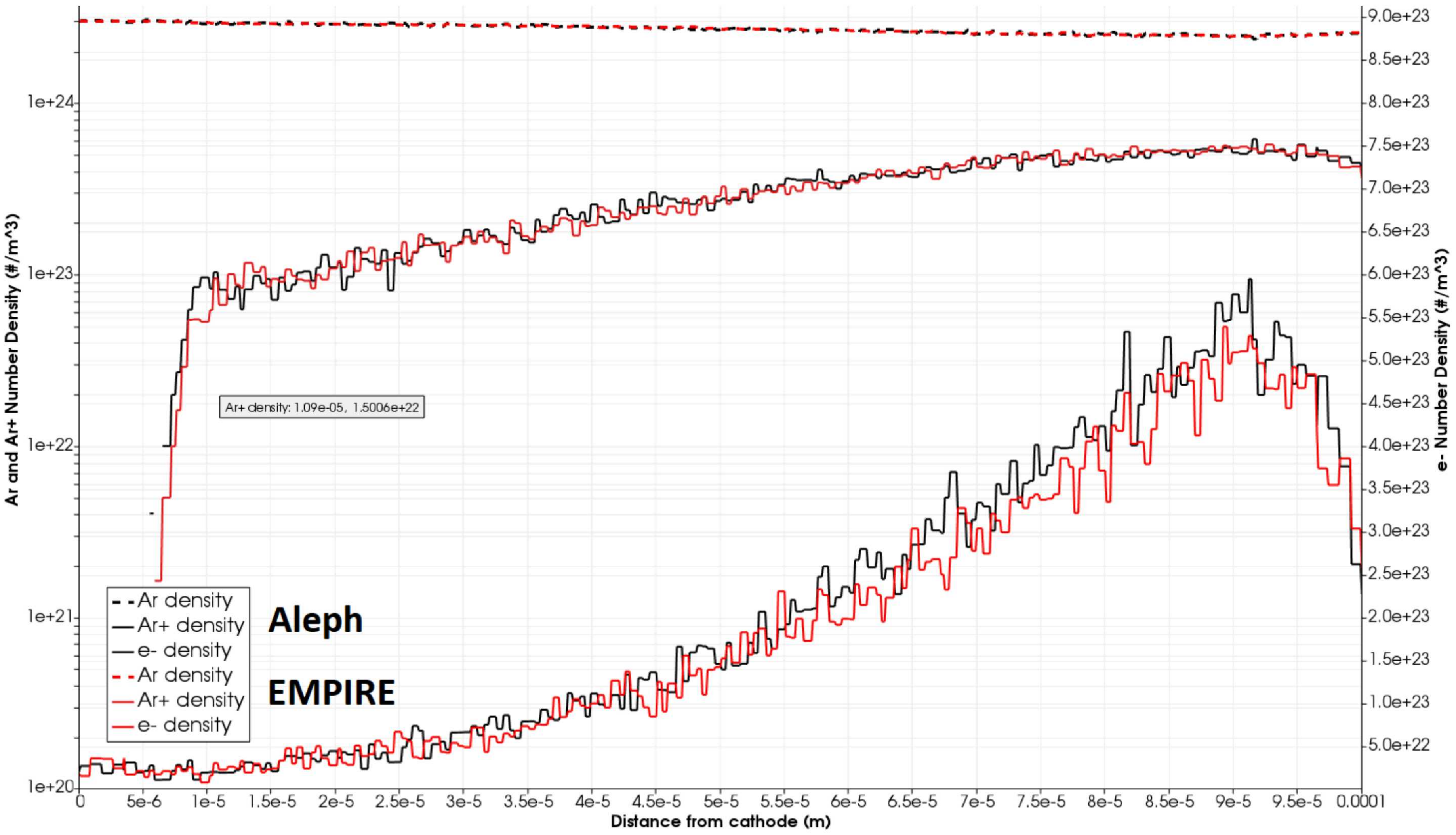
- Molecular cross-sections from an analytic model or read from a table

- Particles can have different weights, allowing accurate simulation of trace species



Convergence test to analytic solution of the 4th moment of the velocity distribution for Bobylev-Krook-Wu relaxation using different numbers of molecules per cell.

SPIN Comparison to Other Codes

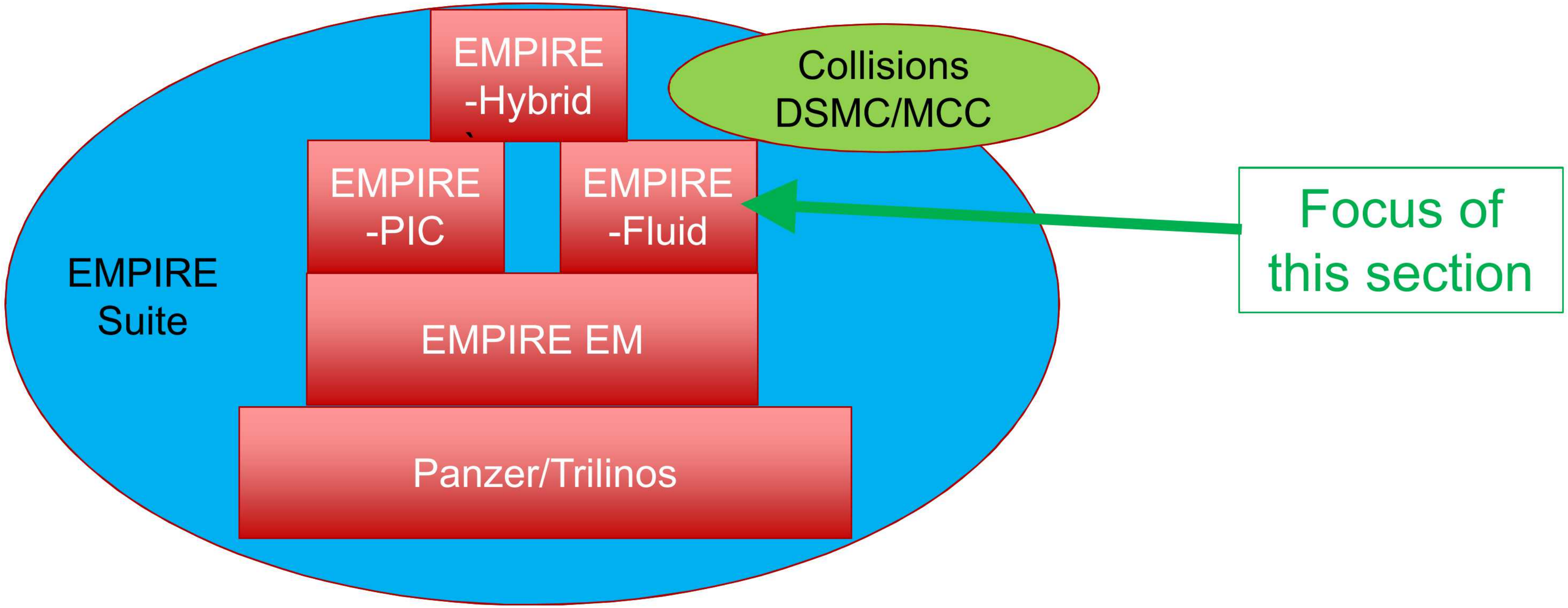


- 1D ionization of argon using EMPIRE with SPIN compared to Aleph

SPIN Summary Road Map

- SPIN is in its initial form but starting to do real problems
- Coming months:
  - Run on GPUs with Kokkos
  - MCC collisions with background fluid
  - Relativistic collisions
  - Provide Empire-Fluid with momentum/energy transfer rates derived from SPIN cross-sections
  - Merge particles to control total particle count
- Long-term:
  - Rotational/vibrational excitation and energy transfer
  - Particle-specific random numbers (for parallel reproducibility)
  - Non-isotropic scattering

EMPIRE: A hierarchy of capabilities



- <u>EMPIRE's goal</u>: Accurately simulate plasmas across regimes on next-generation exascale computing platforms
- Expands the range of critical plasma applications that we can address with high confidence and fidelity

Maxwell coupled to fluid formulation for plasmas

- Multi-species Euler coupled to Maxwell
- Strong collisions terms
- Lots of time scales
- Maxwell involutions must be enforced

5-Moment Fluid

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = \sum_{\text{srcs}} m_\alpha \Gamma^{\text{src}} - \sum_{\text{sinks}} m_\alpha \Gamma^{\text{sink}}$$

$$\frac{\partial (\rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha + p_\alpha I + \Pi_\alpha) = \frac{q_\alpha}{m_\alpha}\rho_\alpha (\mathbf{E} + \mathbf{u}_\alpha \times \mathbf{B}) + \sum_{\text{srcs}} m_\alpha \mathbf{u}_{\text{src}} \Gamma^{\text{src}} - \sum_{\text{sinks}} m_\alpha \mathbf{u}_\alpha \Gamma^{\text{sink}} + \sum_{\beta \neq \alpha} \mathbf{R}^{\alpha,\beta}$$

$$\frac{\partial \mathcal{E}_\alpha}{\partial t} + \nabla \cdot ((\mathcal{E}_\alpha + p_\alpha)\mathbf{u}_\alpha + \mathbf{u}_\alpha \cdot \Pi_\alpha + \mathbf{h}_\alpha) = \frac{q_\alpha}{m_\alpha}\rho_\alpha \mathbf{E} \cdot \mathbf{u}_\alpha + \sum_{\beta \neq \alpha} \left(\mathbf{u}_\alpha \mathbf{R}^{\alpha,\beta} + Q^{\alpha,\beta}\right) + \frac{1}{2}\sum_{\text{srcs}} m_\alpha u_{\text{src}}^2 \Gamma^{\text{src}} - \frac{1}{2}\sum_{\text{sinks}} m_\alpha u_\alpha^2 \Gamma^{\text{sink}}$$

Maxwell Equations

$$\frac{\partial \mathbf{E}}{\partial t} - c^2 \nabla \times \mathbf{B} = -\frac{1}{\epsilon_0}\sum_\alpha \frac{q_\alpha}{m_\alpha}\rho_\alpha \mathbf{u}_\alpha \qquad \nabla \cdot \mathbf{E} = \frac{1}{\epsilon_0}\sum_\alpha \frac{q_\alpha}{m_\alpha}\rho_\alpha$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0 \qquad \nabla \cdot \mathbf{B} = 0$$

Important to satisfy involutions numerically

Fluid plasma implementation

Developing a blended finite element discretization:

- Electromagnetic fields use conformal continuous-Galerkin (CG) finite elements
  - To enforce involutions by construction
  - Maximize shared code with EMPIRE-PIC
- Fluid fields use discontinuous-Galerkin (DG) finite elements
  - Couples to CG EM solvers
  - Handling shocks and steep gradients
  - Potential for high order to handle waves

Using IMEX time integration to split fast and slow time scales

- Many application dependent stiff time scales

Take home: These plasmas are hard to simulate!

Momentum diffusivity

$$\upsilon_\alpha = \frac{\mu_\alpha}{\rho_\alpha}$$

Thermal diffusivity

$$\kappa_\alpha \sim \frac{k_\alpha}{\rho_\alpha}$$

Plasma frequency

$$\omega_{p\alpha} = \sqrt{\frac{q_\alpha^2 n_\alpha}{m_\alpha \epsilon_0}}$$

Cyclotron frequency

$$\omega_{c\alpha} = \frac{q_\alpha B}{m_\alpha}$$

Collision frequency

$$\nu_{\alpha\beta} \sim \frac{n_\beta}{\sqrt{m_\alpha} T_\alpha^{\frac{3}{2}}} \frac{1 + \frac{m_\alpha}{m_\beta}}{\left(1 + \frac{m_\alpha}{m_\beta}\frac{T_\beta}{T_\alpha}\right)^{\frac{3}{2}}}$$

Flow velocity

$$u_\alpha$$

Speed of sound

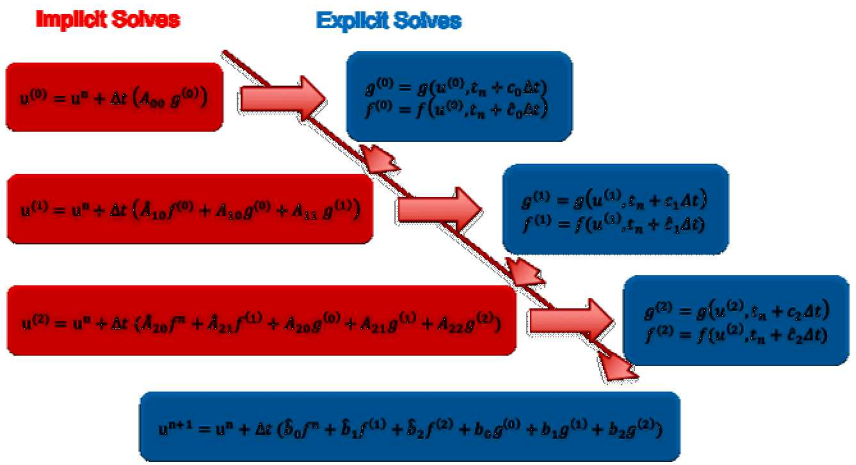
$$v_{s\alpha} = \sqrt{\frac{\gamma P_\alpha}{\rho_\alpha}}$$

Speed of light

$$c \gg u_\alpha, v_{s\alpha}$$

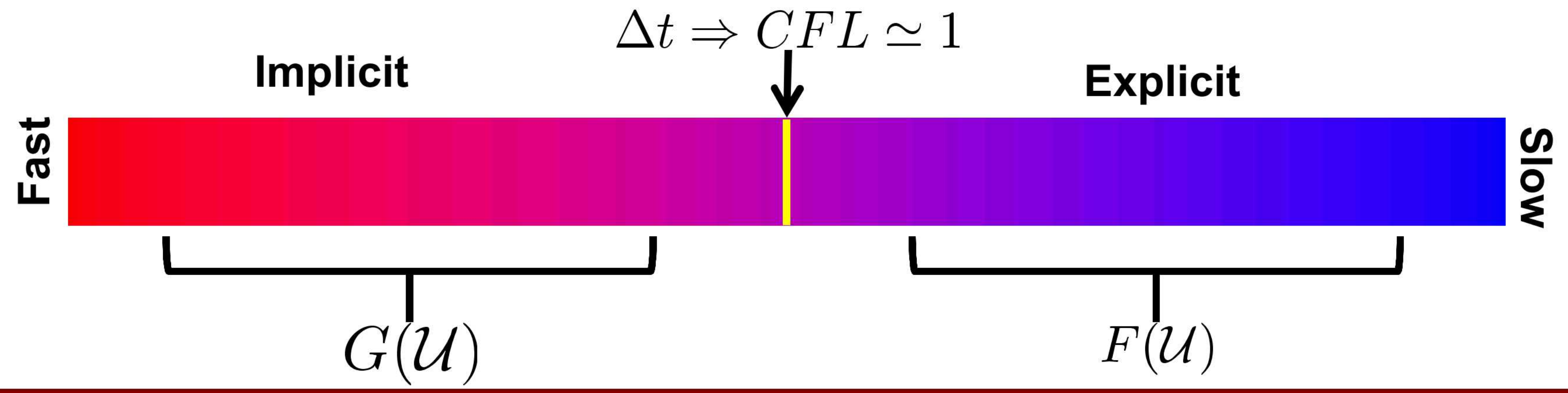
Implicit-Explicit (IMEX) Time Integration

3 Stage IMEX-RK Algorithm



IMEX methods split fast and slow modes

- Implicit terms solve for stiff modes (plasma oscillation, speed of light)
- Explicit terms are accurately resolved
- Combine with block/physics-based preconditioning for implicit solves
- IMEX assumes an additive decomposition: $\dot{\mathcal{U}} + F(\mathcal{U}) + G(\mathcal{U}) = 0$



Fast/Stiff/Implicit modes in plasma model

Stiff Modes:

- Speed of light
- Plasma Oscillation
- Collisions
- Cyclotron frequency

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = \sum_{\text{srcs}} m_\alpha \Gamma^{\text{src}} - \sum_{\text{sinks}} m_\alpha \Gamma^{\text{sink}}$$

$$\frac{\partial (\rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha + p_\alpha I + \Pi_\alpha) = \frac{q_\alpha}{m_\alpha} \rho_\alpha (\mathbf{E} + \mathbf{u}_\alpha \times \mathbf{B}) + \sum_{\text{srcs}} m_\alpha \mathbf{u}_{\text{src}} \Gamma^{\text{src}} - \sum_{\text{sinks}} m_\alpha \mathbf{u}_\alpha \Gamma^{\text{sink}} + \sum_{\beta \neq \alpha} \mathbf{R}^{\alpha,\beta}$$

$$\frac{\partial \mathbf{E}}{\partial t} - c^2 \nabla \times \mathbf{B} = -\frac{1}{\epsilon_0} \sum_\alpha \frac{q_\alpha}{m_\alpha} \rho_\alpha \mathbf{u}_\alpha$$

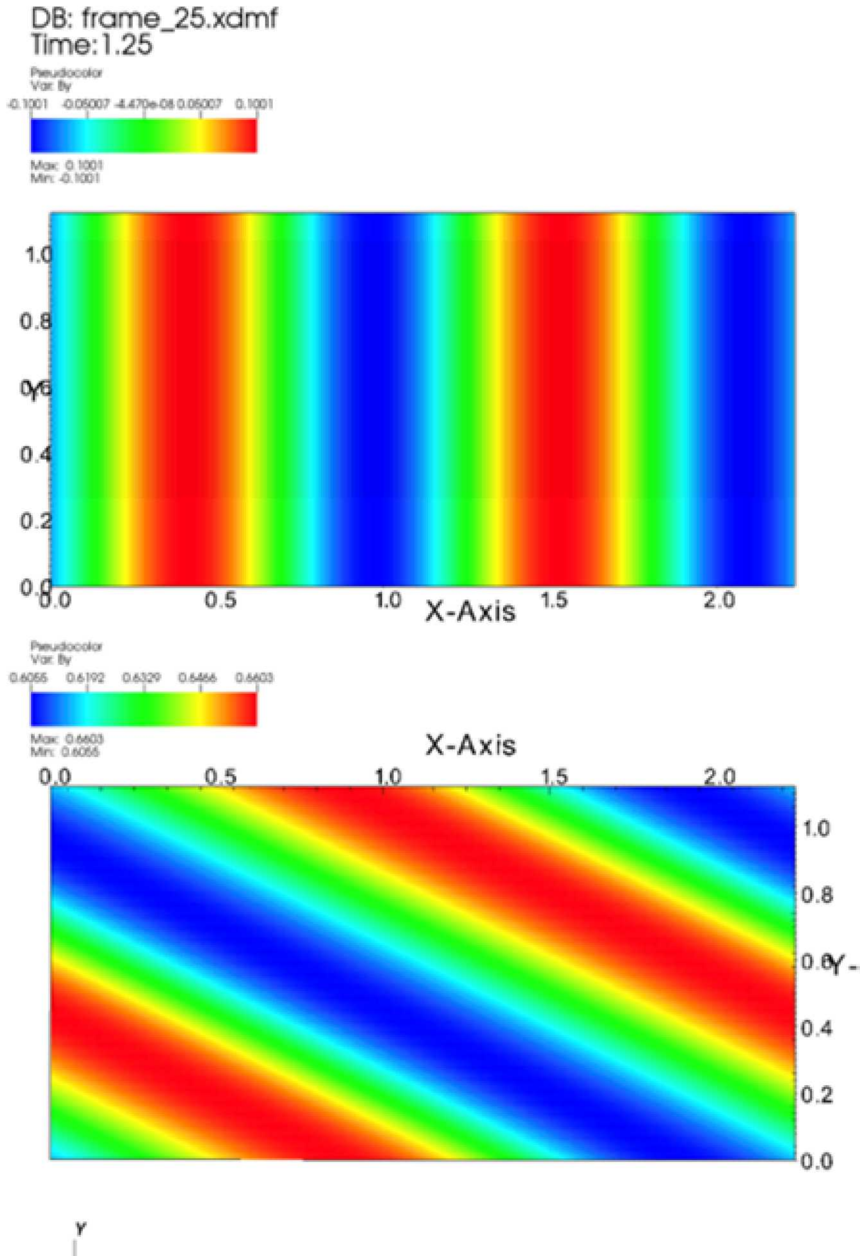
$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0$$

- Speed of light arises from coupling of electromagnetic field: explicit CFL $\sim c\Delta t/\Delta x$
- Plasma oscillation arises from Ampere’s law to momentum conservation: explicit CFL ~
- Collisions explicit CFL$\sim\Delta t$
- Cyclotron frequency explicit CFL$\sim|\mathrm{B}|\Delta t$

$\nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)$ If the plasma oscillation is implicit, then the mass flux needs to be implicit for maintenance of Gauss law

Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
  - EMPIRE-Fluid initial results:
    - Wave aligned with grid propagates stably
    - Wave oblique to grid develops checkerboard instability and fails
    - Possible solution: add divergence cleaning methodology
    - With divergence cleaning: oblique wave propagates in stable fashion



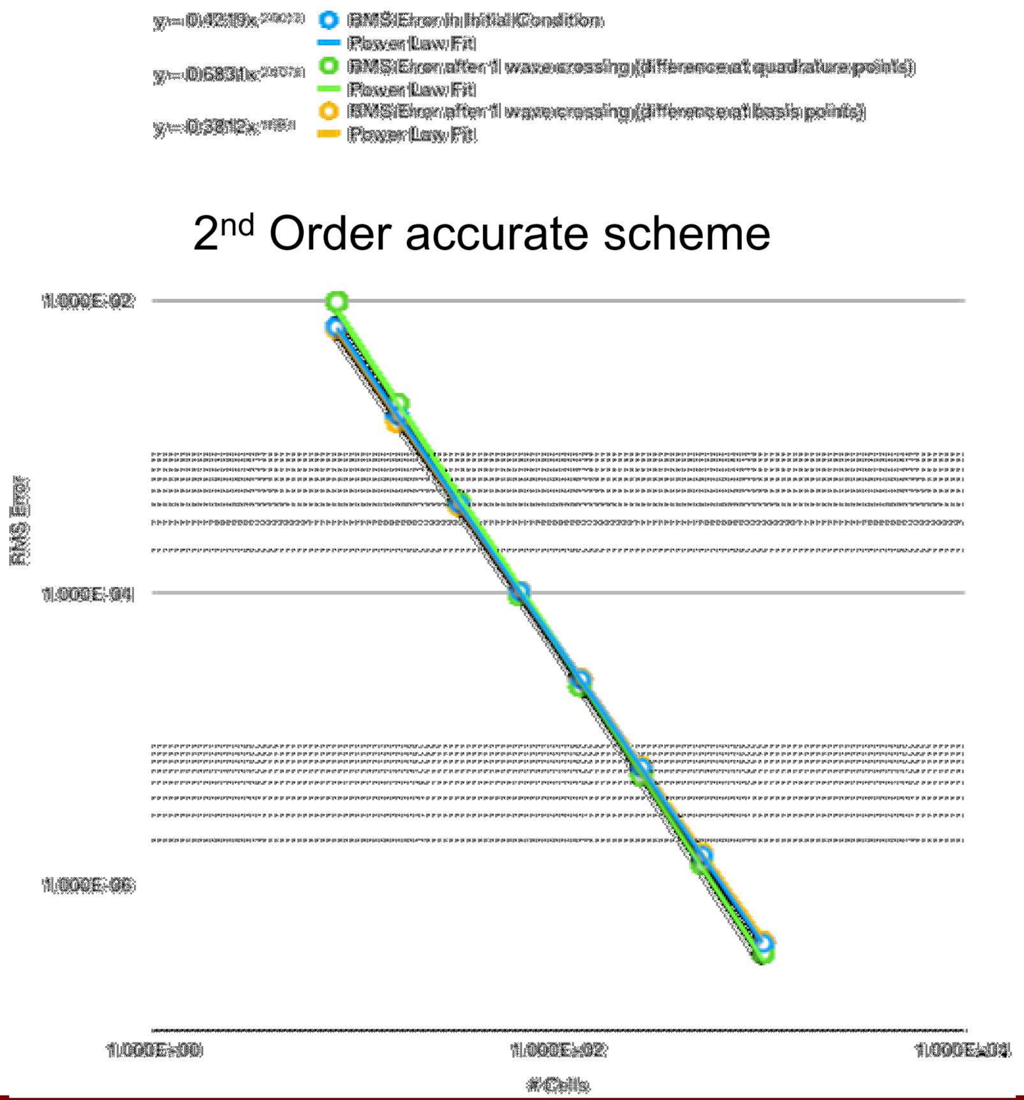
Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

Sandia National Laboratories

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
  - EMPIRE-Fluid initial results:
    - One-dimensional wave converges with expected order of accuracy in RMS-error after 1 wave crossing period

$$\|\delta q\| = \sqrt{\sum_k (\delta q_k)^2}$$

$$\delta q_k = \frac{1}{2N^2} \sum_i \sum_j |q^{\prime}_{i,j,k} - q^{0}_{i,j,k}|$$

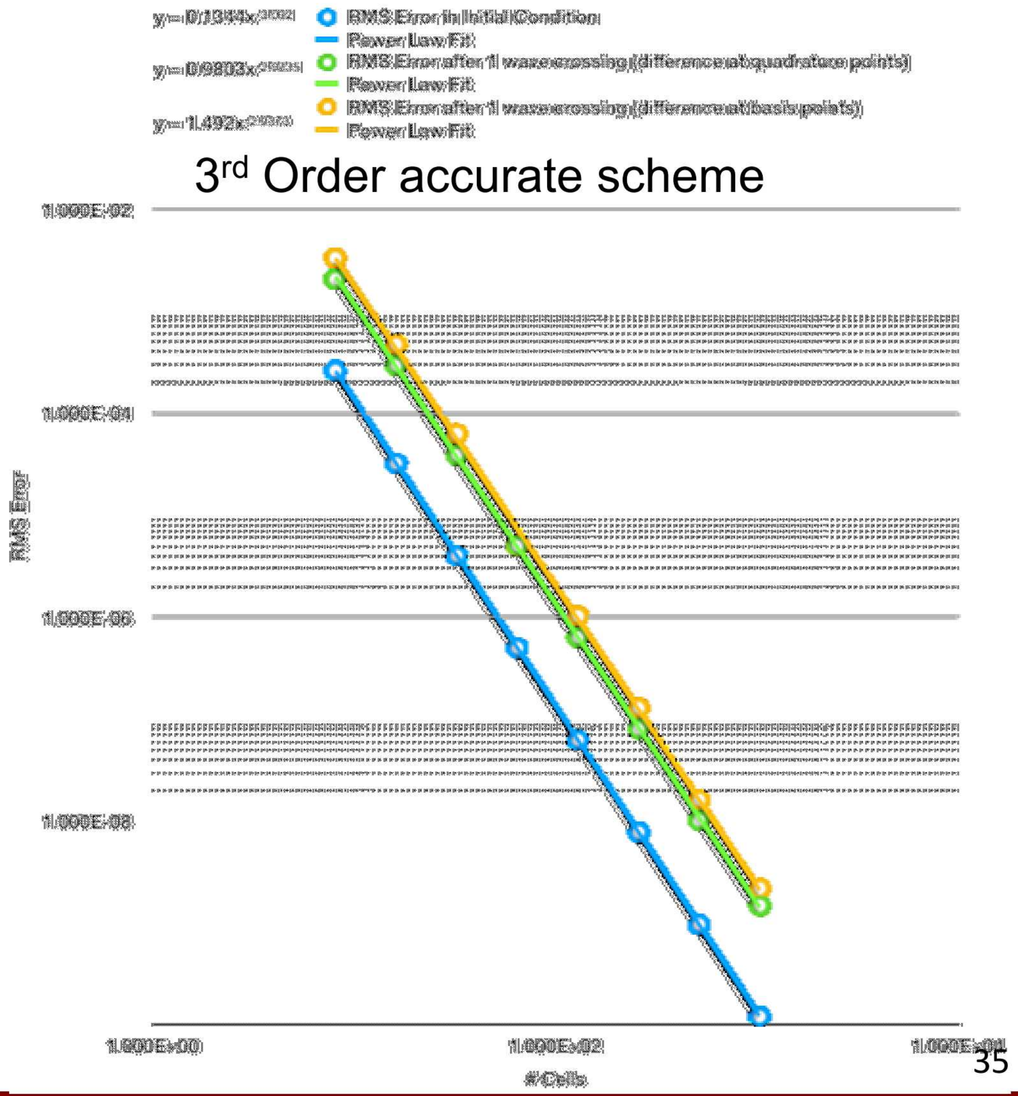


Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
  - EMPIRE-Fluid initial results:
    - One-dimensional wave converges with expected order of accuracy in RMS-error after 1 wave crossing period

$$\|\delta q\| = \sqrt{\sum_k (\delta q_k)^2}$$

$$\delta q_k = \frac{1}{2N^2} \sum_i \sum_j \left| q^t_{i,j,k} - q^0_{i,j,k} \right|$$

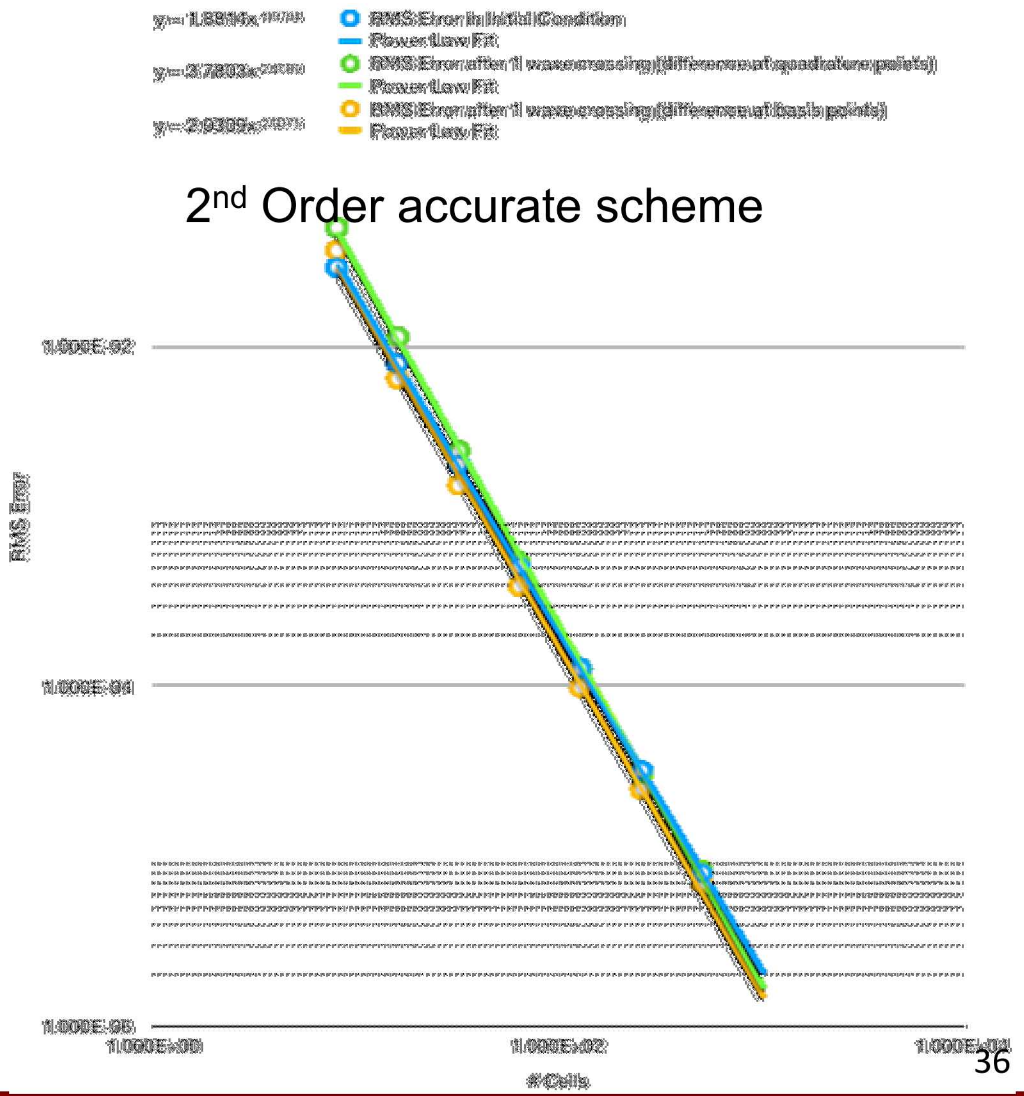


Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
  - EMPIRE-Fluid initial results:
    - Two-dimensional wave converges with expected order of accuracy in RMS-error after 1 wave crossing period

$$\|\delta q\| = \sqrt{\sum_k (\delta q_k)^2}$$

$$\delta q_k = \frac{1}{2N^2} \sum_i \sum_j \| q^t_{i,j,k} - q^0_{i,j,k} \|$$



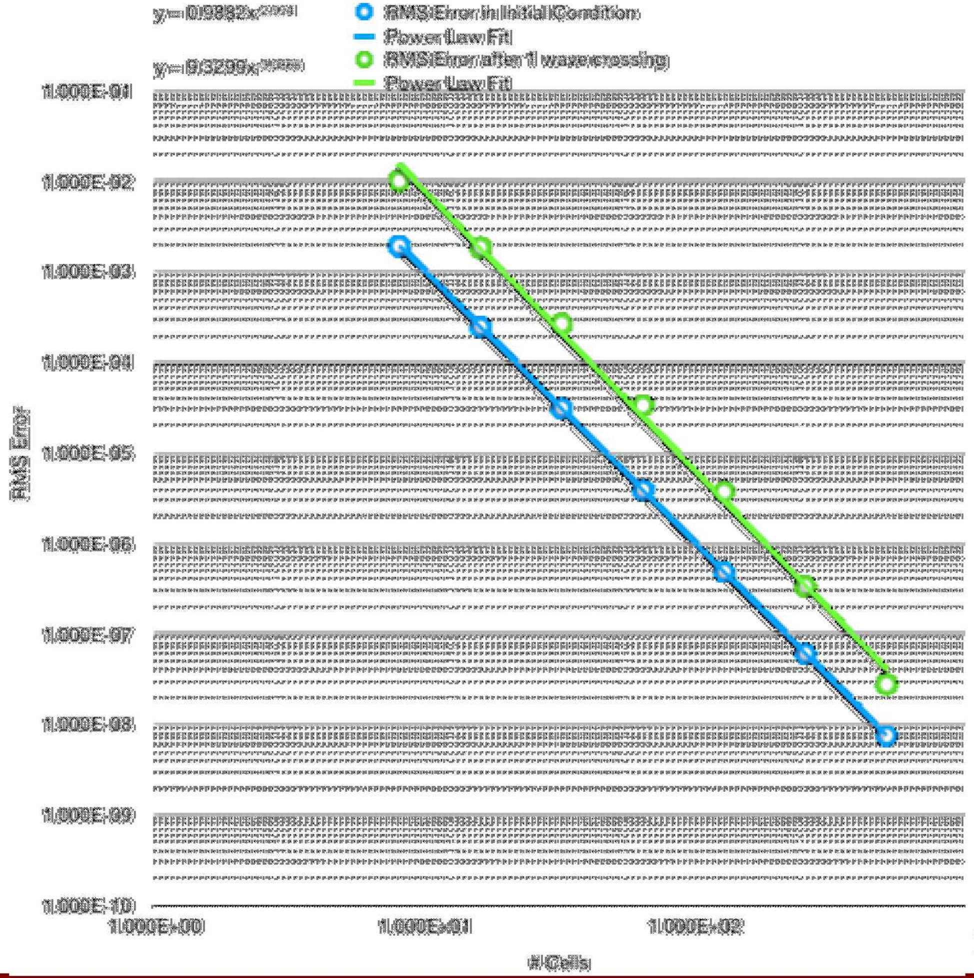
Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
  - EMPIRE-Fluid initial results:
    - Two-dimensional wave converges with expected order of accuracy in RMS-error after 1 wave crossing period

$$\|\delta q\| = \sqrt{\sum_k (\delta q_k)^2}$$

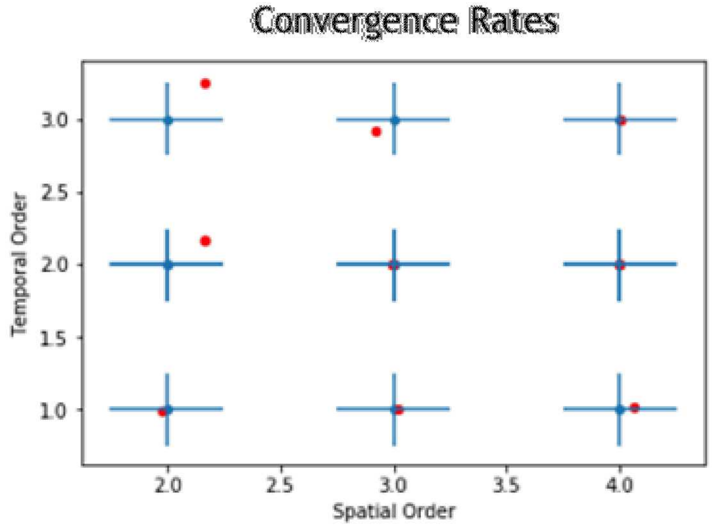
$$\delta q_k = \frac{1}{2N^2} \sum_i \sum_j \left| q^t_{i,j,k} - q^0_{i,j,k} \right|$$

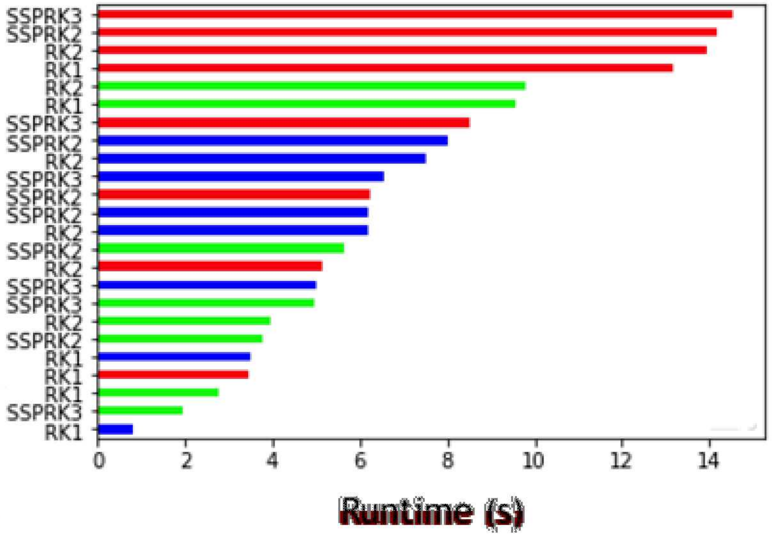
3rd Order accurate scheme



Non-Linear Circular Polarized Alfven Waves: EMPIRE-Fluid

- Circularly Polarized Alfven Wave:
  - Exact, non-linear solution to ideal MHD equations
- Extremely useful for:
  - Diagnosing faults in numerical scheme (see e.g. Beckwith & Stone, 2011)
- EMPIRE-Fluid results:
  - Now being used as part of the vvtest suite for EMPIRE-Fluid (M. Scott Swan)
  - Run for 12 different combinations of spatial basis order and time integrator.
  - Most combinations converge very close to the theoretical rate.

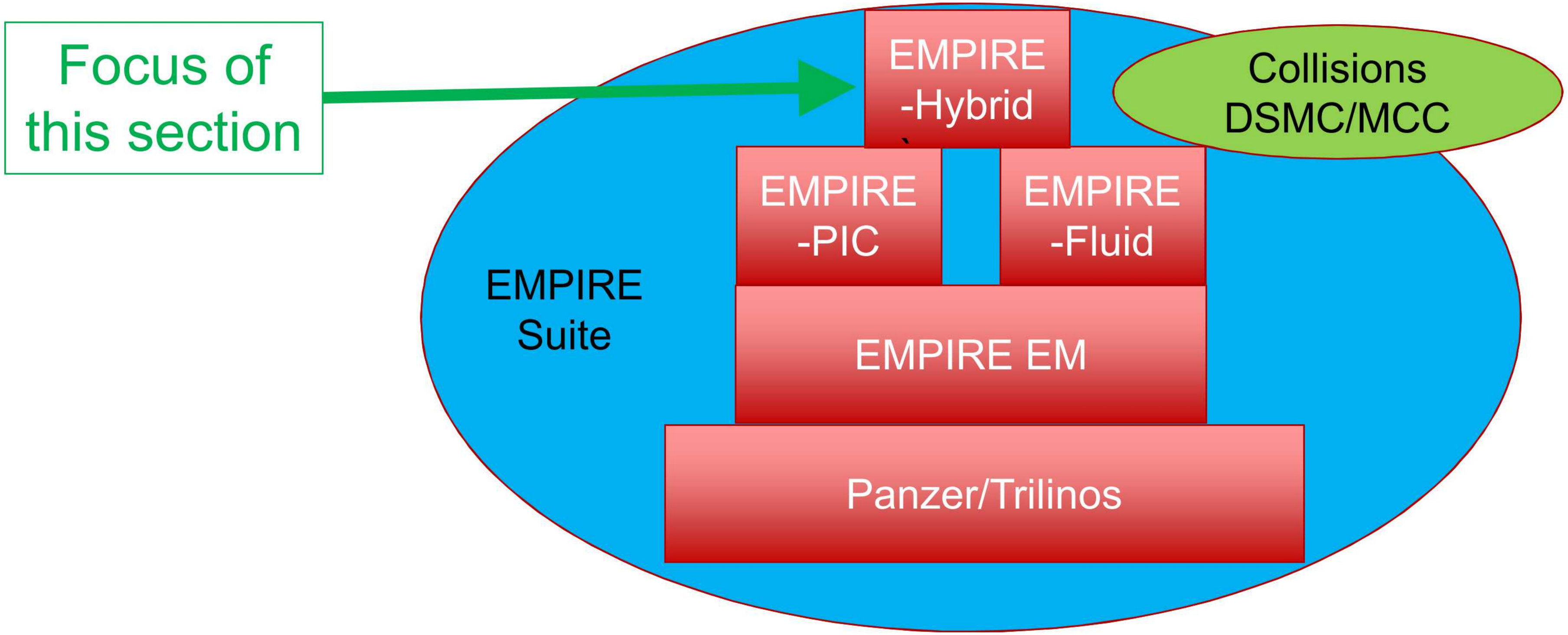




Summary: EMPIRE-Fluid

- EMPIRE-Fluid is the multi-fluid plasma simulation component of a broader plasma simulation tool being developed at Sandia to provide high fidelity modeling of critical plasma applications
- We have been working on benchmarking discretization approaches on a range of linear and non-linear problems:
  - Demonstrated that discretization can deliver 3rd order accuracy for non-linear problems
  - We have incorporated Implicit-Explicit (IMEX) methods to allow us to step over stiff time scales efficiently
  - We also are able to handle steep gradients and are beginning the comparison to PIC methods

EMPIRE: A hierarchy of capabilities



- <u>EMPIRE's goal</u>: Accurately simulate plasmas across regimes on next-generation exascale computing platforms
- Expands the range of critical plasma applications that we can address with high confidence and fidelity

EMPIRE—Hybrid: Plasma Oscillation

- E. Cyr has shown that a plasma oscillation can be setup assuming an neutralizing background fluid by setting an initial electric field to **E=0**. Writing the momentum and Ampere equations gives:

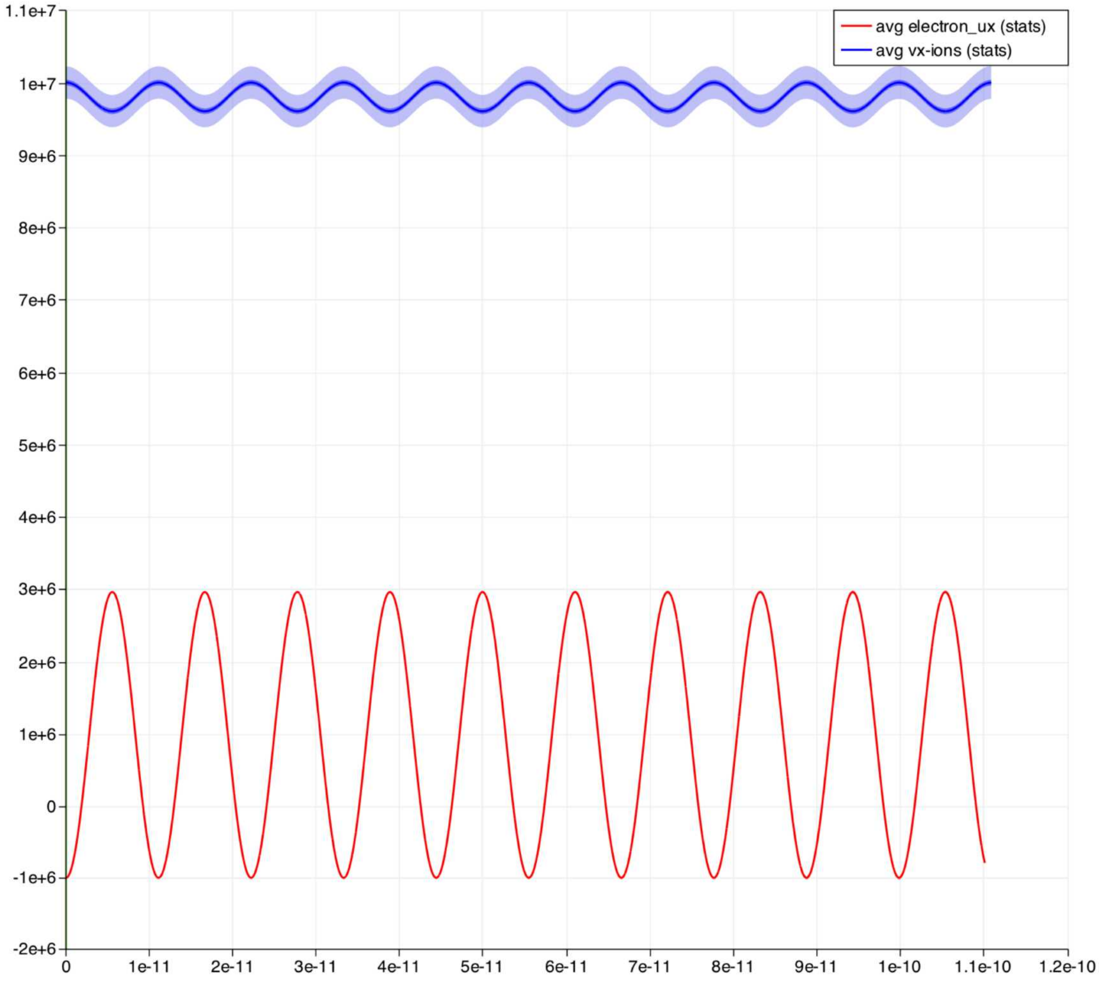
$$\partial_t \mathbf{v}_1 = \frac{q_1}{m_1}\mathbf{E} \qquad E = C sin(\gamma t)$$

$$\partial_t \mathbf{v}_2 = \frac{q_2}{m_2}\mathbf{E} \qquad v_1 = -\frac{q_1}{m_1\gamma} C cos(\gamma t) + A$$

$$\epsilon_0 \partial_t \mathbf{E} = -q_1 n_1 \mathbf{v}_1 - q_2 n_2 \mathbf{v}_2 \qquad v_2 = -\frac{q_2}{m_2\gamma} C cos(\gamma t) - \xi A$$

$$\gamma = \sqrt{\frac{q_1^2 n_1}{\epsilon_0 m_1} + \frac{q_2^2 n_2}{\epsilon_0 m_2}},\ \xi = \frac{q_1 n_1}{q_2 n_2},\ C = -\gamma \frac{\tilde{\mathbf{v}}_2 + \xi \tilde{\mathbf{v}}_1}{\left(\frac{q_2}{m_2} + \xi \frac{q_1}{m_1}\right)},\ A = \tilde{\mathbf{v}}_1 + \frac{q_1}{m_1 \gamma} C.$$

- Simulation
  - Kinetic (PIC) ions; fluid (DG) electrons; Maxwell (DG) with cleaning
  - *Spatially constant* by construction, though the fluid and PIC solvers are definitely not constrained in that way.
  - The time integration is explicit SSPRK3 for the fluid+Maxwell; operator split with the particle update.
  - The particles are initially randomly distributed throughout the domain
  - When the Debye length is sufficiently resolved, the analytic solution is recovered.



Resolution: 64x64 mesh, with 256000 particles (total); Debye length = 7.234643e-7; Debye length / Cell length = 2.828

EMPIRE—Hybrid: Plasma Oscillation

- E. Cyr has shown that a plasma oscillation can be setup assuming an neutralizing background fluid by setting an initial electric field to **E=0**. Writing the momentum and Ampere equations gives:

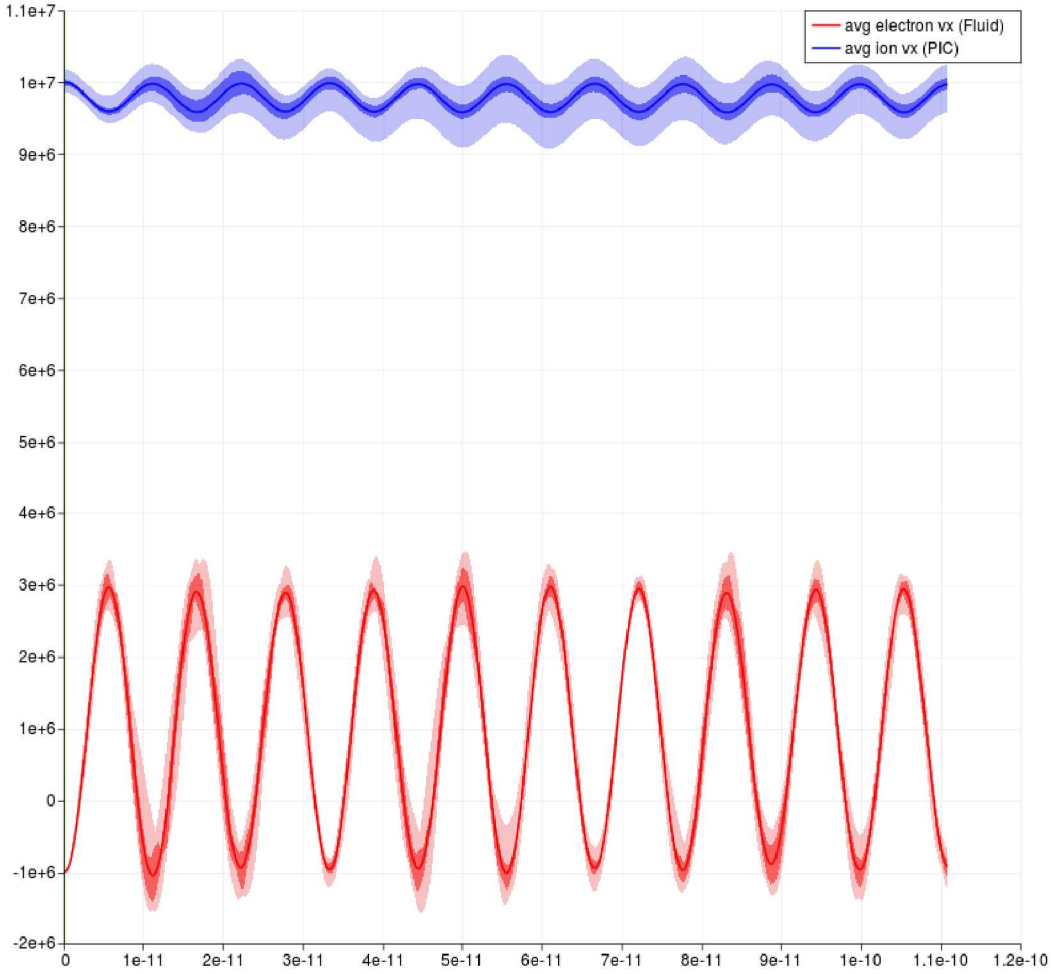
$$\partial_t \mathbf{v}_1 = \frac{q_1}{m_1}\mathbf{E}$$

$$\partial_t \mathbf{v}_2 = \frac{q_2}{m_2}\mathbf{E}$$

$$\epsilon_0 \partial_t \mathbf{E} = -q_1 n_1 \mathbf{v}_1 - q_2 n_2 \mathbf{v}_2$$

$$\gamma = \sqrt{\frac{q_1^2 n_1}{\epsilon_0 m_1} + \frac{q_2^2 n_2}{\epsilon_0 m_2}},\ \xi = \frac{q_1 n_1}{q_2 n_2},\ C = -\gamma \frac{\tilde{\mathbf{v}}_2 + \xi \tilde{\mathbf{v}}_1}{\left(\frac{q_2}{m_2} + \xi \frac{q_1}{m_1}\right)},\ A = \tilde{\mathbf{v}}_1 + \frac{q_1}{m_1 \gamma} C.$$

- Simulation setup (E. Cyr):
  - All simulations used the following parameters:
    - $n_1 = 1e18, n_2 = 1e20$
    - $q_1 = -10q_e, q_2 = q_e$
    - $m_1 = 100m_e, m_2 = m_e$
    - $v_1 = 1e7, v_2 = -1e6$
  - PIC and Fluid codes were coupled through EM fields only, no collisions
  - If the Debye length is not sufficiently resolved, the solution becomes very noisy and deviates from the analytic solution.

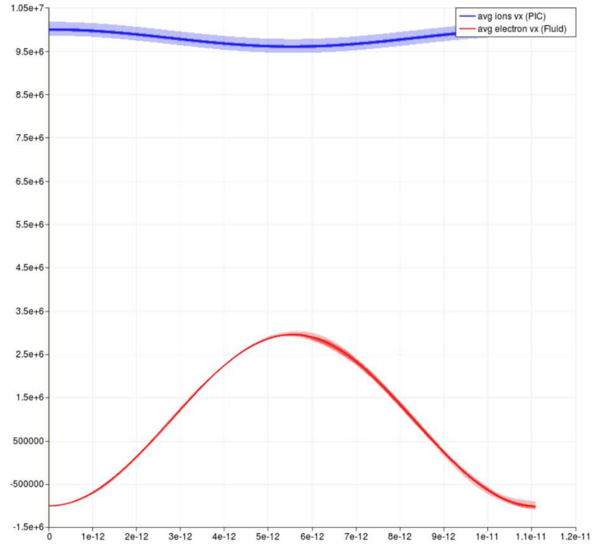


Resolution: 4x4 mesh, with 100 particles (total); Debye length = 7.234643e-7; Debye length / Cell length = 0.08838

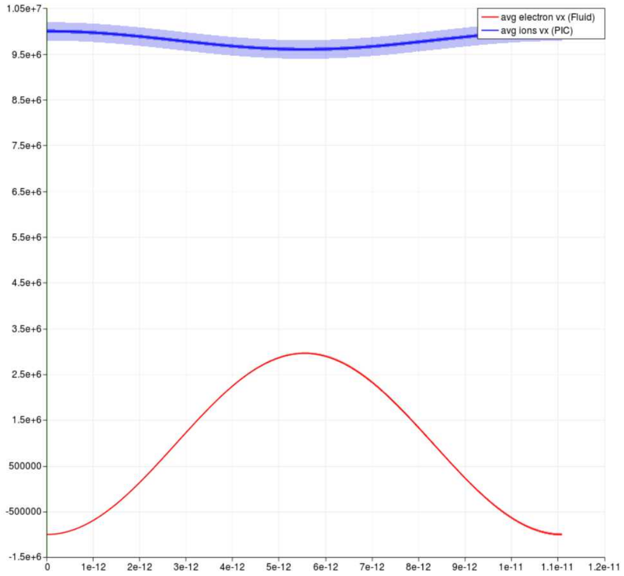
EMPIRE—Hybrid: Plasma Oscillation

- Initial condition:
  - *Spatially constant* by construction, though the fluid and PIC solvers are definitely not constrained in that way.
  - The particles are initially randomly distributed throughout the domain
  - Initial temperature given to the PIC and Fluid species of T = 21987 K to create a Debye length that can be sufficiently resolved.
- Algorithm:
  - Electrons + EM use DG spatial discretization (divergence cleaning)
  - Ions: particle-in-cell discretization
  - Time integration: electrons + EM use explicit SSPRK3; PIC is (first order) operator split
  - The time integration is explicit SSPRK3 for the fluid+Maxwell; operator split with the particle update.
- Simulation setup:
  - Resolution
    - Low: 4x4 mesh, with 100 particles (total); Debye length = 7.234643e-7; Debye length / Cell length = 0.17675
    - Medium: 16x16 mesh, with 1600 particles (total); Debye length = 7.234643e-7; Debye length / Cell length =0.717
    - High: 64x64 mesh, with 25600 particles (total); Debye length = 7.234643e-7; Debye length / Cell length = 2.828

Low resolution:



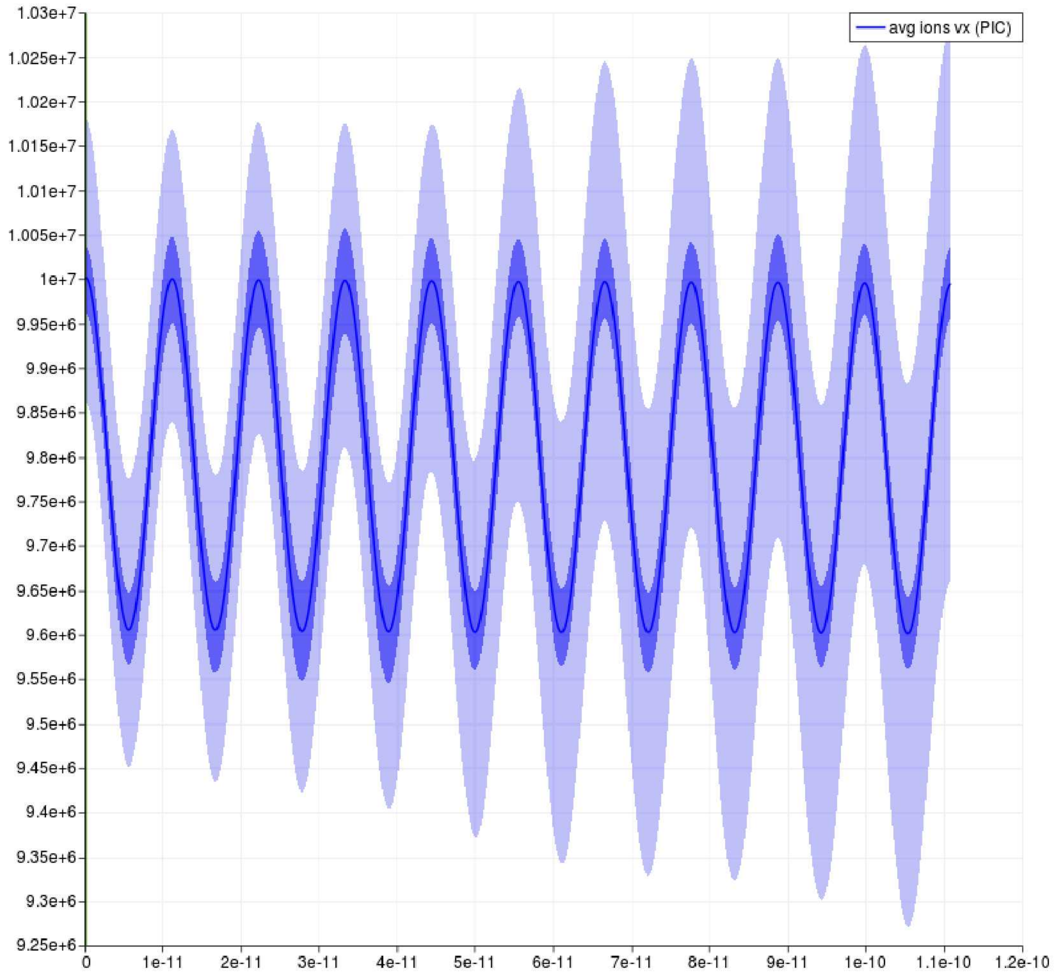
Medium resolution:



High resolution: not pictured

EMPIRE—Hybrid: Plasma Oscillation

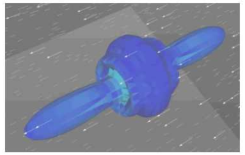
- PIC has a spatial stability limit, requiring the Debye length of the plasma to be sufficiently resolved by the mesh.
  - If the Debye length is under-resolved, the fluid will heat up until the Debye length is resolved.
  - Temperature in PIC is represented by particle velocity noise, so an increase in temperature translates to an increase in particle noise.
  - The Fluid code tracks temperature as a scalar value on each node, and noise is interpreted as just noise.
  - This heating in PIC translates to increasing error/noise in the Fluid part of Hybird
- The explicit time stepper in fluid has a CFL stability limit
  - Properly resolving the Debye length for PIC forces an inconveniently small time step value to satisfy the CFL condition
  - The CFL stability restriction in these simulations was much more restrictive than the plasma oscillation frequency.
  - Using an implicit time stepper with a CG electromagnetic solve fixes this problem and greatly increases performance.



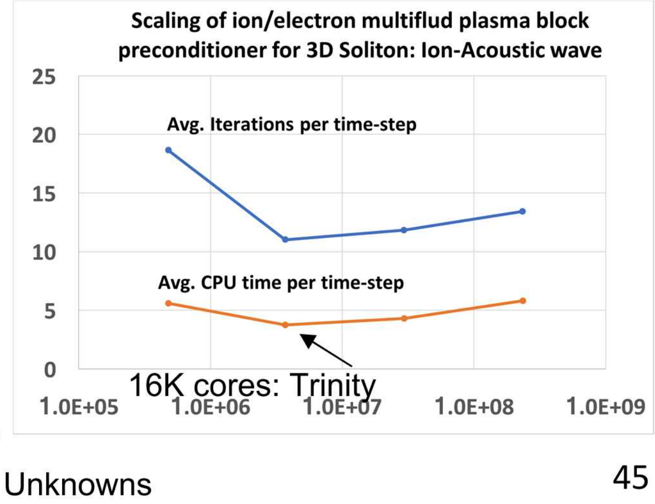
Initial Progress on R&D Hybrid Implicit/IMEX Multifluid Plasma and Kinetic – PIC Coupling

- Goal: Develop a R&D hybrid IMEX plasma code with 5-moment multifluid model + Electromagnetics (**Drekar**) coupled to particle-in-cell (PIC) model (**EMPIRE-PIC**).

- Essential character of modest R&D effort
  - **Flexible and extensible for complex multiphysics multispecies plasma systems**
  - **Robust, accurate, and scalable solution for longer-time-scale simulations.** Focus on coupling with higher-order multi-rate IMEX time-integration
  - **Performance Portable** on emerging HPC architectures: HSW, KNL, CUDA, …
  - **Component-based**: heavily leverage existing R&D plasma simulation capabilities and SNL/ECP software stack (e.g. Trilinos, Kokkos, Panzer, ….)



- Physics/Plasma Capability Components
  - **Drekar:** Scalable multifluid plasma solver
    - Implicit/IMEX & Newton-Krylov (sources and fast-waves)
    - FE H(grad) and structure preserving (nodal, edge, face)
    - Hyperbolic Systems: Algebraic flux limited CG
  - **EMPIRE-PIC:** Scalable PIC solver
    - Currently explicit, implicit under development



R. Pawlowski, J. Shadid, S. Conde & ATDM EMPIRE Team (E. Philips, E. Cyr, M. Bettencourt)

Initial Proof-of-principle: Simple Operator Split Coupled Model

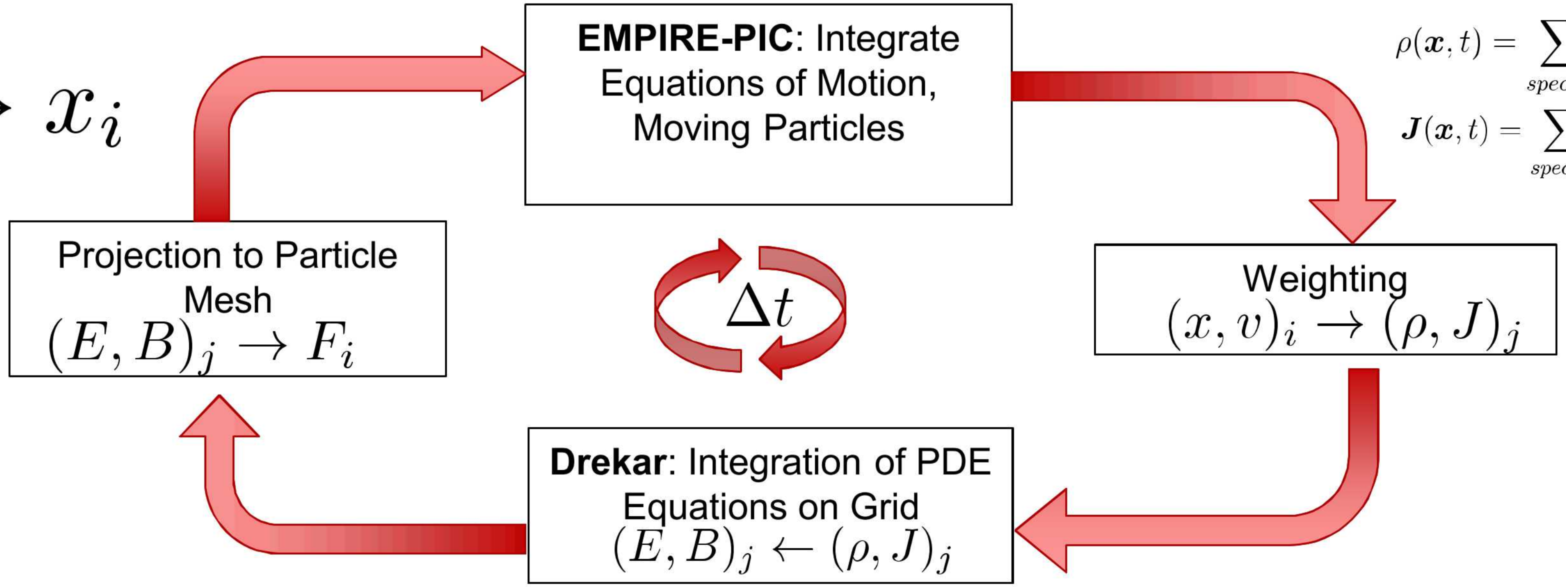
Sandia National Laboratories

$$m_s \frac{d}{dt}\left(\mathbf{V}_i(t)\gamma_i(t)\right) = q_s\left(\mathbf{E}(\boldsymbol{X}_i(t),t) + \boldsymbol{V}_i(t) \times \mathbf{B}(\boldsymbol{X}_i(t),t)\right)$$

$$F_i \rightarrow v_i \rightarrow x_i$$

$$\rho(\boldsymbol{x},t) = \sum_{species} q_s \int d\boldsymbol{u} N_s(\boldsymbol{x},\boldsymbol{u},t)$$

$$\boldsymbol{J}(\boldsymbol{x},t) = \sum_{species} q_s \int d\boldsymbol{u}\boldsymbol{u} N_s(\boldsymbol{x},\boldsymbol{u},t)$$

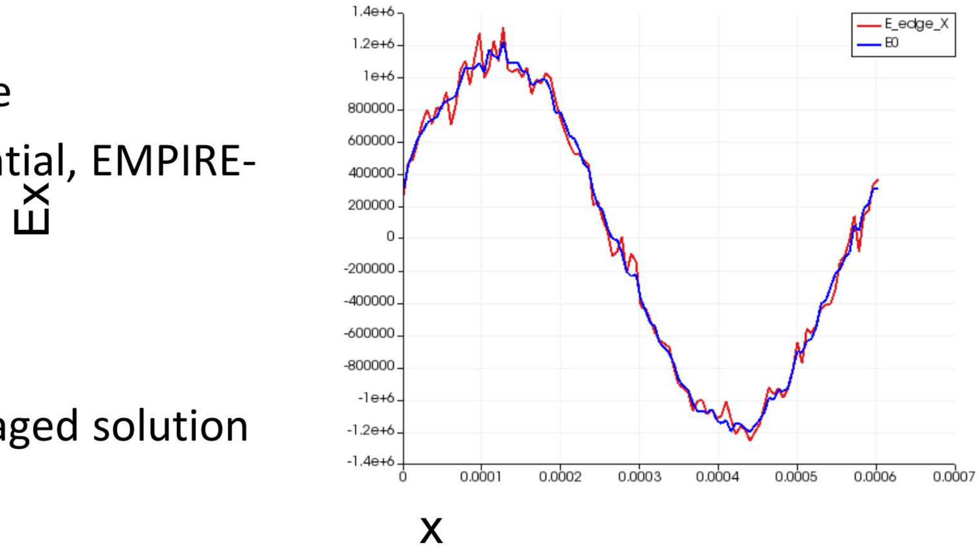


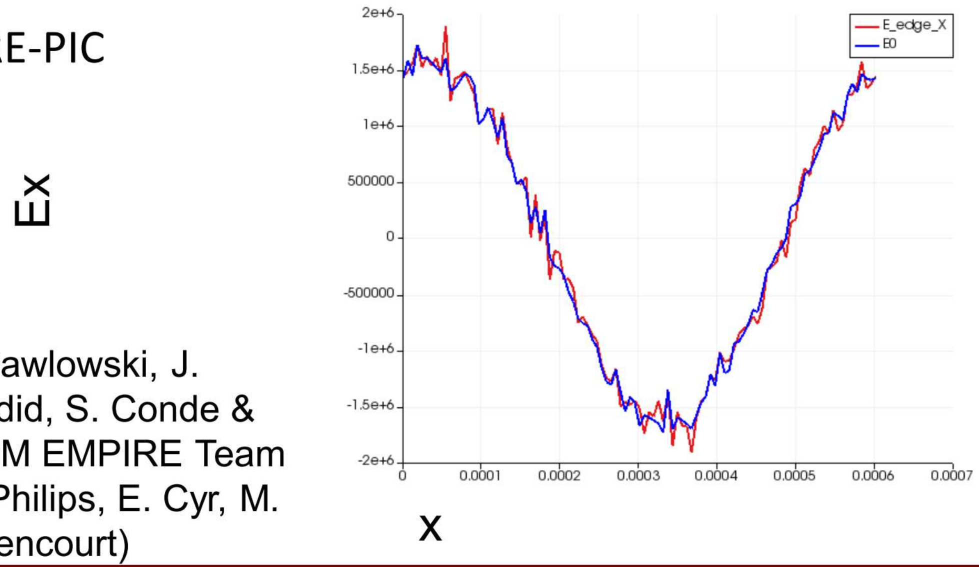
$\frac{\partial \rho_a}{\partial t} + \nabla \cdot (\rho_a \mathbf{u}_a) = \mathcal{R}_{\rho_a}$
$\frac{\partial (\rho_a \mathbf{u}_a)}{\partial t} + \nabla \cdot (\rho_a \mathbf{u}_a \otimes \mathbf{u}_a + p_a I + \Pi_a) - q_a n_a (\mathbf{E} + \mathbf{u}_a \times \mathbf{B}) = \mathcal{R}_{\rho_a u_a}$
$\frac{\partial \mathscr{E}_a}{\partial t} + \nabla \cdot ((\mathscr{E}_a + p_a)\mathbf{u}_a + \Pi_a \cdot \mathbf{u}_a + \mathbf{h}_a) - q_a n_a \mathbf{u}_a \cdot \mathbf{E} + Q_a^{src} = \mathcal{R}_{\mathscr{E}_a}$
$\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \frac{1}{\mu_0}\mathbf{B} + \mathbf{J} = \mathbf{0}; \quad \frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = \mathbf{0}; \quad \mathbf{J} = \sum_k q_k n_k \mathbf{u}_k$

R. Pawlowski, J. Shadid, S. Conde & ATDM EMPIRE Team (E. Philips, E. Cyr, M. Bettencourt)

1st Proof-of-principle for Drekar/EMPIRE-PIC Coupling

- First coupling of Drekar to EMPIRE-PIC for a Langmuir Wave
- Simple proof-of-principle (Drekar solves electrostatic potential, EMPIRE-PIC electrons)
- Replace PIC EM Solver with Drekar EM
- 1024 cells, 9K particles
- Plots show Ex line plot over spatial domain: red is cell averaged solution in Drekar, blue is EM-PIC standalone solution at nodes.
  - After projection Drekar solution matches EMPIRE's.
- Verified coupling of fields from Drekar PDE solver to EMPIRE-PIC particle push
- Results show good agreement as expected



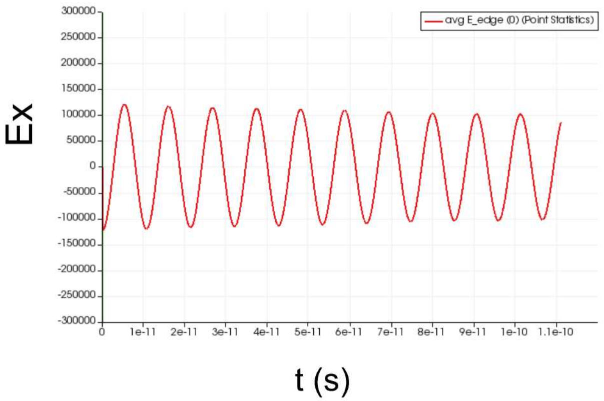


R. Pawlowski, J. Shadid, S. Conde & ATDM EMPIRE Team (E. Philips, E. Cyr, M. Bettencourt)

Preliminary Hybrid Fluid/Kinetic Capability in Drekar: E.g. Ion/Electron Plasma Oscillation

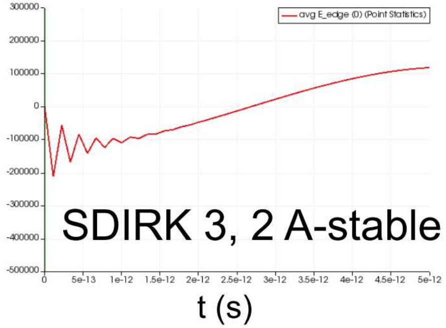
Coupled system time integration:

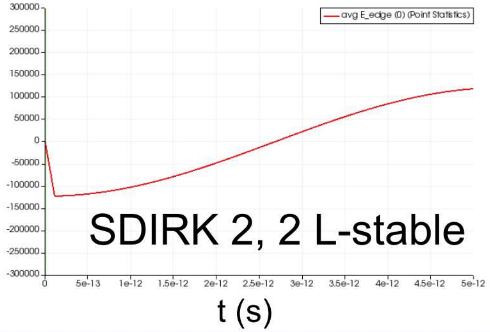
- Current / Proof-of-principle: 1st order operator split coupling (2nd order implementation straightforward)
  - Fluid electron Euler and electro-static Poisson system time integration
    - SDIRK 2nd order, 2 stage: L-Stable
    - SDIRK 3rd order, 2 stage: A-Stable
  - Ion kinetic system PIC (standard EMPIRE Particle push)



- Theory period: 1.06192e-11 $\omega_p = \sqrt{\sum_\alpha \omega_{p\alpha}^2}$ with $\omega_{p\alpha} = \sqrt{\frac{q_\alpha^2 n_\alpha}{m_\alpha \epsilon_0}}$
- Computed period: 1.0593e-11, 0.25% error

- Demonstrated L-stable fluid/EM solve can control high-frequency unresolved time-scales (during startup in this case)





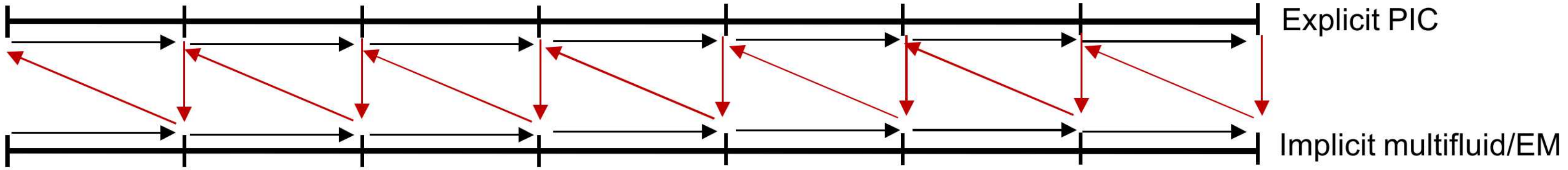
Integrator	Stability	Period	% Error
DIRK 2nd Order, 2 stage	L-Stable	1.1111E-11	4.63E+00
DIRK 3rd Order, 2 stage	A-Stable	1.1111E-11	4.63E+00

R. Pawlowski, J. Shadid, S. Conde & ATDM EMPIRE Team (E. Philips, E. Cyr, M. Bettencourt)

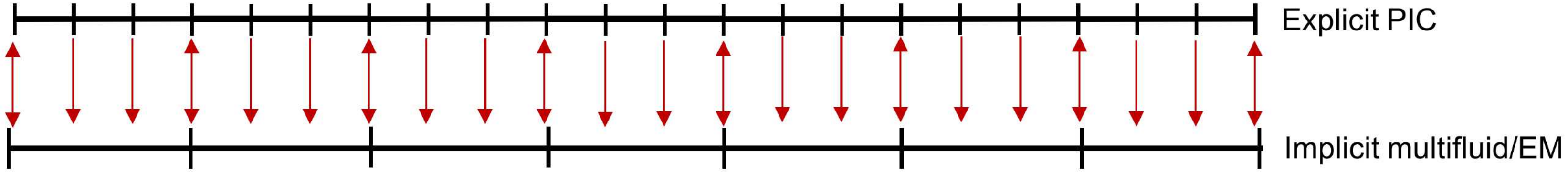
Time Integration for Hybrid Multifluid – PIC plasma capability

- Goal higher-order integration with flexible stability properties (R&D)
  - Multirate generalized-structure additive partitioned IMEX RK
    - Longer time-step implicit integrator for fluid continuum / EM system
    - Ion kinetic system smaller integral steps that match up with PDE solver (goal)

Current 1st order operator split:



Multirate generalized-structure additive partitioned IMEX RK:



Status:

- Demonstrated 1st order operator split with implicit A/L-stable PDE solves
- Demonstrated some control over high-frequency unresolved modes (Ex)
- Developing a multirate RK capability in Tempus time integration package
- Preparing hybrid software coupling for more advanced time integration capabilities

R. Pawlowski, J. Shadid, S. Conde & ATDM EMPIRE Team (E. Philips, E. Cyr, M. Bettencourt)

Summary: EMPIRE-Hybrid

- We have begun exploring methods for hybrid PIC-Fluid simulations:
  - Utilizing first order coupling strategies; PIC push separated from Fluid/Maxwell
  - Motivation: understand where simple temporal coupling strategies fail and develop methodology to mitigate
- EMPIRE-PIC/Fluid (particle push + DG Fluid/Maxwell):
  - Particle noise can induce shocks within fluid
  - Ensuring conservation of momentum/energy requires careful treatment of coupling of plasma to EM fluid
- EMPIRE-PIC/Drekar
  - Drekar provides mature multi-fluid scheme that enables experimentation with temporal discretizations
  - EM solver matches EMPIRE-EM results
  - Results indicate particle noise triggers plasma oscillation
    - Method gives accurate results
- Both hybrid approaches provide foundation to develop methodologies to address issues of noise, EM discretization, time stepping this FY.

Oh, and don't forget ICNSP