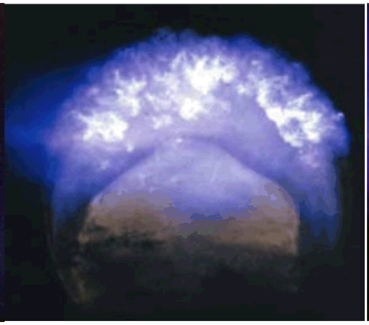
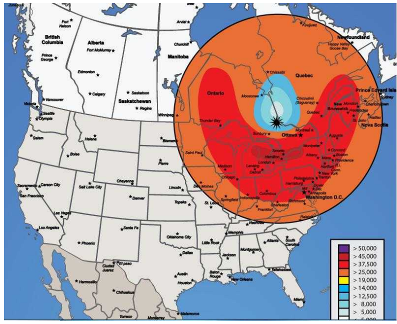
SAND2017-8471C







EMPIRE – EM/PIC/Fluid Simulation Code

M. Bettencourt and E. C. Cyr (presenting)

R. Kramer, S. Miller, R.P. Pawlowski, E.G. Phillips, A. Robinson, J.N. Shadid

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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.

Agenda

- Overview of goals
- EMPIRE-EM
  - Flexibility of design
  - Performance portability
- EMPIRE-PIC
- EMPIRE-Fluid
- Future work
- Summary

Goals

- EMPIRE – ElectroMagnetic Plasma In Realistic Environments
  - Open source application being developed for next generation platforms
- Modular design
  - Electromagnetics/electrostatics and alternate EM closures
  - Multi-fluild and Particle in Cell plasma modules
- Performance portability
  - Ability to run on modern computing platforms with a single code base
  - Kokkos abstraction to parallel backends
    - Cuda, OpenMP, Pthreads
- Advanced time integration
  - Implicit-Explicit (IMEX) methods
- Beyond forward simulation
  - Embedded sensitivities, adjoints, etc

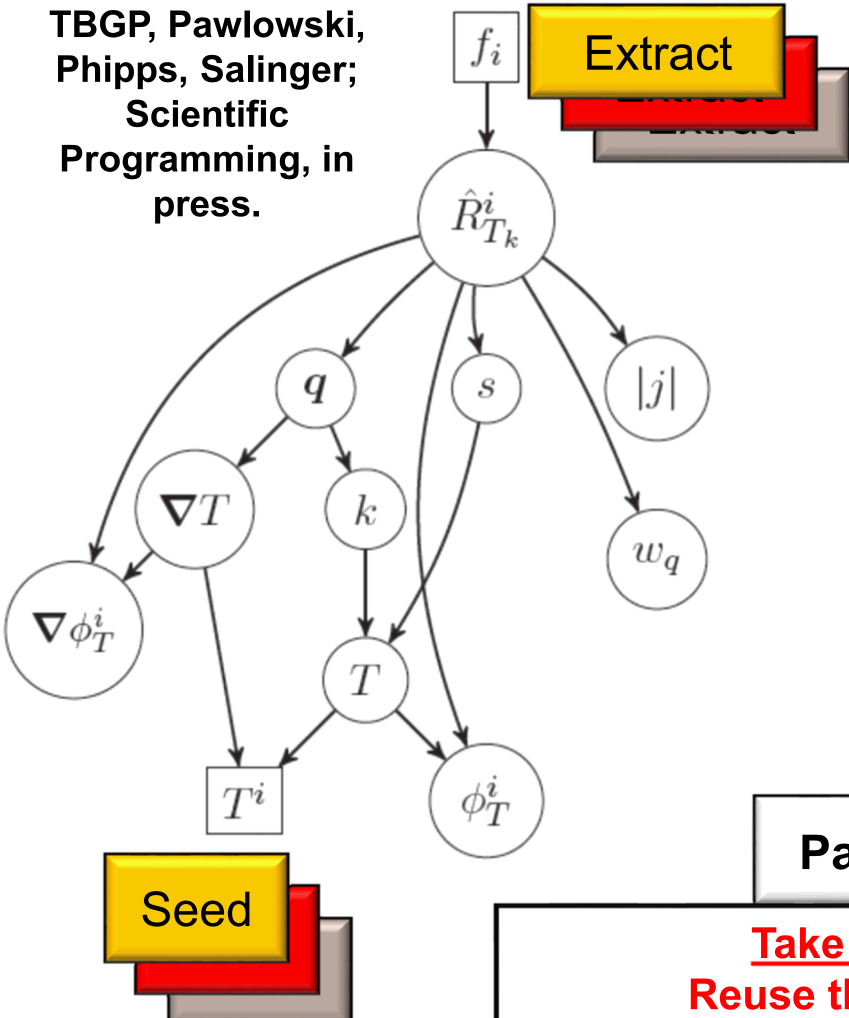
EMPIRE-EM

- EMPIRE-EM is designed to provide solvers for Maxwell's equations
  - Currently two approximations
    - Full Maxwell
    - Electrostatic
  - Could add magnetostatic or Darwin approximations
- Built upon Trilinos and the Panzer library
  - Directed acyclic graph assembly using Phalanx
  - Linear and nonlinear model descriptions through Thyra model-evaluators
  - Block based linear solvers/preconditioners through Thyra/Teko
  - Sensitivities - automatic differentiation through Sacado

Handling Complexity in Analysis Requirements

Sandia National Laboratories

TBGP, Pawlowski, Phipps, Salinger; Scientific Programming, in press.



$$f(x) = \sum_{k=1}^{N_w} f_k = \sum_{k=1}^{N_w} Q_k^T \hat{R}^i_{T_k}(P_k x)$$

$$\hat{R}^i_T = \sum_{e=1}^{N_e} \sum_{q=1}^{N_q} \left[-\nabla \phi^i_T \cdot q + \phi^i_T s \right] w_q |j| = 0$$

Evaluation Type	Scalar Type
$f(x,p)$	double
$J = \frac{\partial f}{\partial x}$	DFad<double>
$\frac{\partial^2 f}{\partial x_i \partial x_j}$	DFad< DFad<double> >

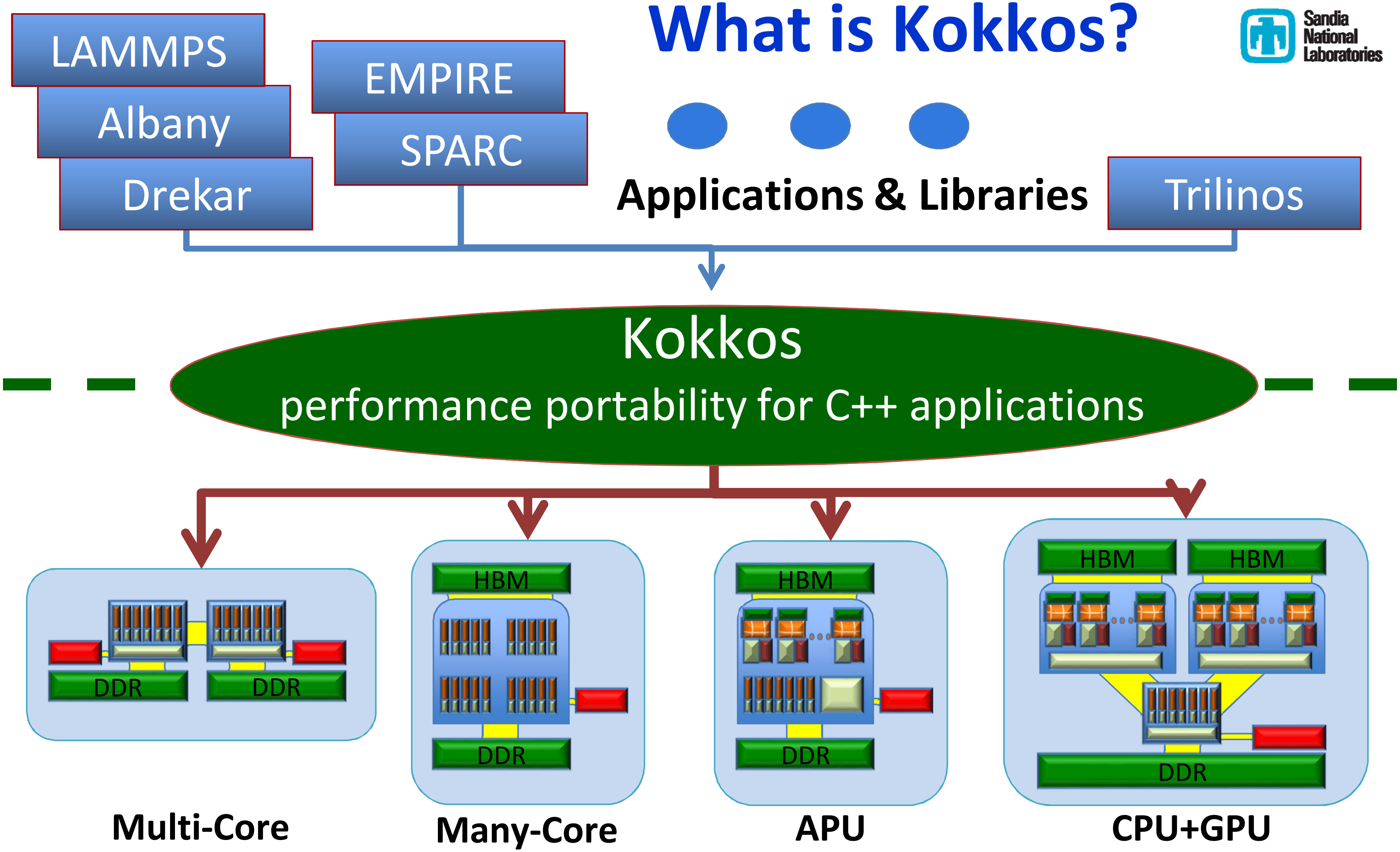
Param. Sens., Jv, Adjoint, PCE (SGF, SGJ), AP

Take Home Message:
Reuse the same code base!
Equations decoupled from algorithms!
Machine precision accuracy!

Performance Portability

- Once a graph is built we need to assemble and solve the graph efficiently on all different platforms
  - For linear problems we can build our matrices, preconditioners once and solve them many times
  - Solve application needs to be completed on device, however, setup can be done on the CPU
- Utilize Kokkos library for performance portability
  - Abstracts the hardware and programming models from the developer
  - Allows for compile time optimizations for memory layouts and access patterns
- Protects from future architecture changes, somewhat…

What is Kokkos?



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

Abstractions
Patterns, Policies, and Spaces

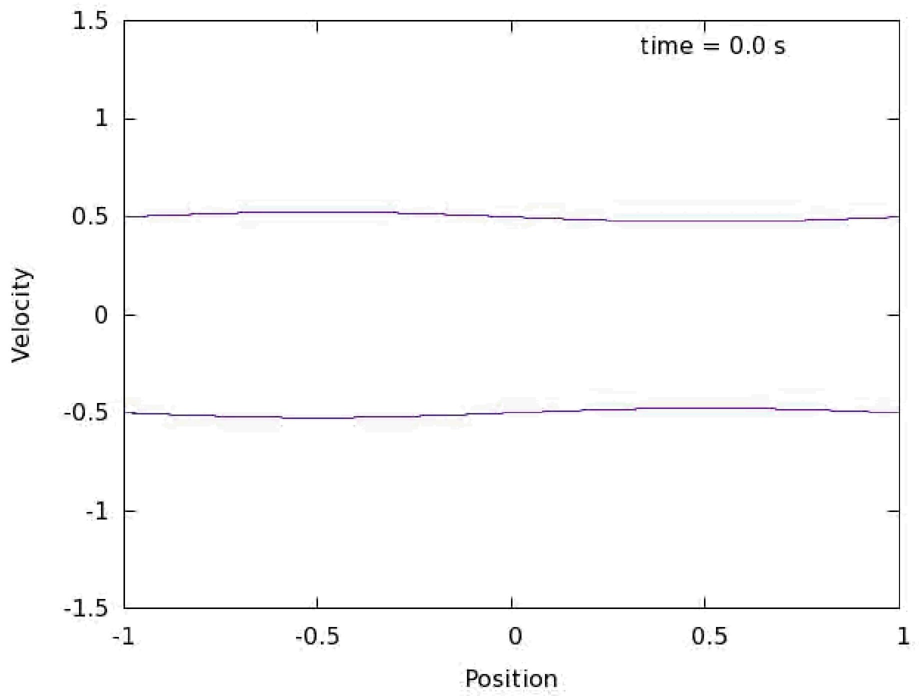
- Parallel Pattern of user's computations
  - parallel_for, parallel_reduce, parallel_scan, task-graph, ... *(extensible)*
- Execution Policy tells *how* user computation will execute
  - Static scheduling, dynamic scheduling, thread-teams, ... *(extensible)*
- Execution Space tells *where* computations 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

- Goal – develop a plasma simulation tool using the Particle In Cell (PIC) technique
- Built off the mini-PIC PIC algorithm
  - All field solve modified to use the Panzer based code
- Current status
  - Works on unstructured meshes for both EM/ES
    - Charge conserving current weighting for electromagnetic
    - Higher order fields for electrostatic
    - Lumped and consistent mass matrix for electric and magnetic field projections
  - MPI+X parallelization
    - Platform independent
  - General leap-frog time integration
  - Simple beam injection boundary conditions

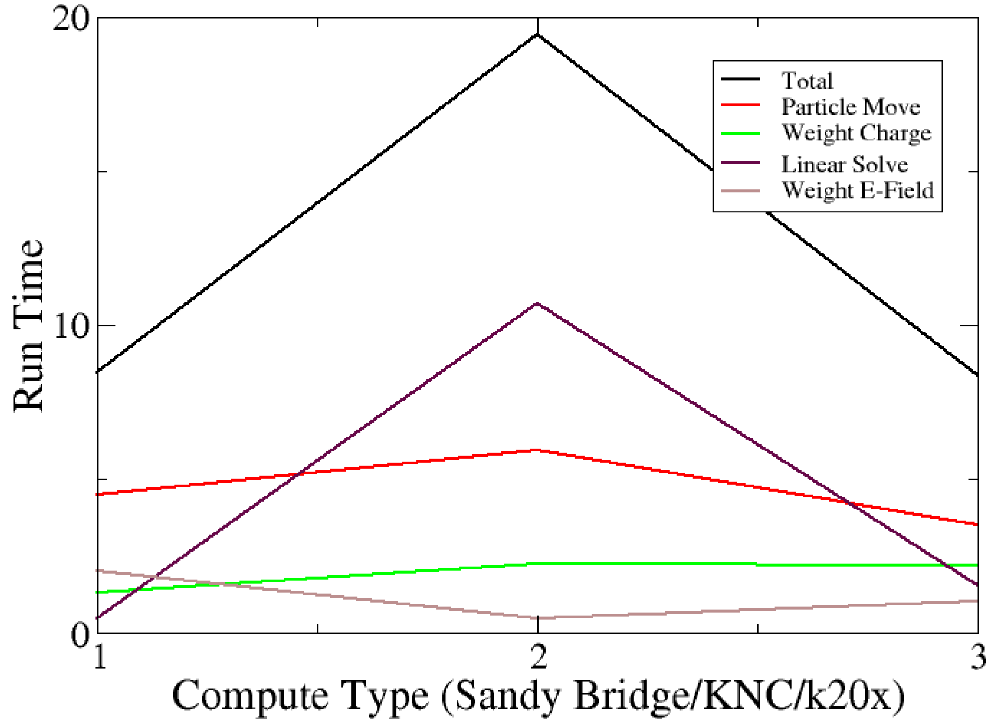
Initial verification

- We have demonstrated initial convergence results for simple problems
  - Plasma oscillation (EM/ES)
  - TM mode through a plasma (EM)
  - Growth rate for two stream and decay rate for Landau damping (ES)



Portable Performance

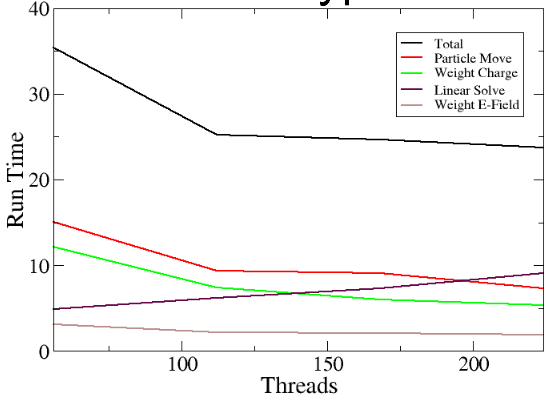
- Computed time per node on several machines
  - 18k unknowns and 11M particles, 20 timesteps CFL = 1.5
  - 16 cores of Sandy Bridge – 2 Knights Corner – 1NVidia k20x



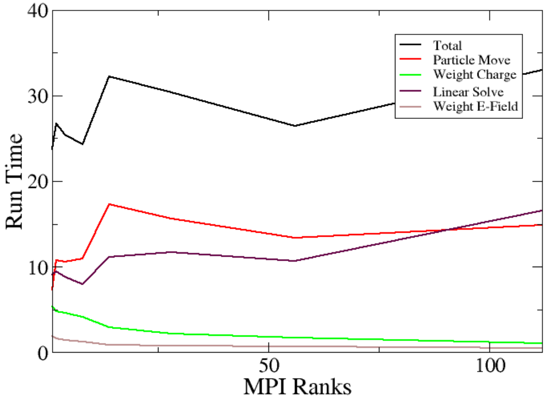
MPI vs Threads

- Comparisons of distributions between threads and MPI ranks

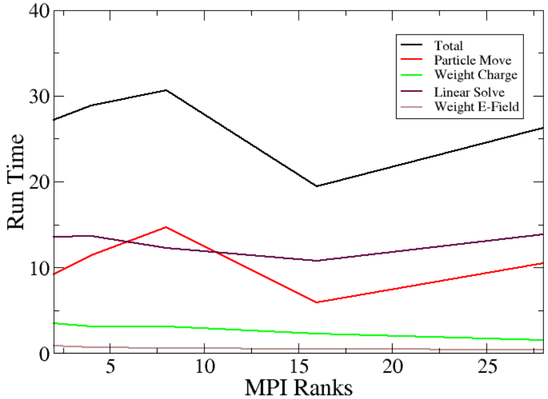
KNC 1 - 4 Hyperthreads



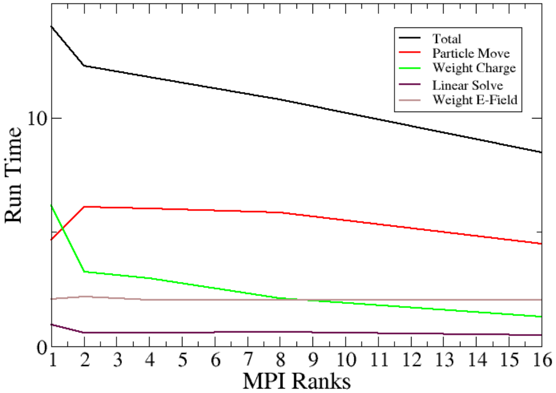
KNC 224 threads 1 -112 MPI ranks



2 KNC 456 threads 2 - 28 MPI ranks



Sandy Bridge 1-16 MPI ranks



EMPIRE-Fluid

- Goal – develop a PDE based multi-fluid plasma simulation tool building off EMPIRE-EM capability
- Leveraging research done in the Drekar code
  - Both codes are based on Panzer/Trilinos library
- Design points
  - Discretizion approach
    - IMEX time integration to handle multiple time scales (discussed today)
    - Continuous Galerkin Maxwell equations for enforcing involutions: solenoidal magnetic field and Gauss' law
    - Discontinuous Galerkin for fluid equations
  - Implicit block preconditioning
  - Beyond forward simulation
  - MPI+X parallelization using Kokkos

Multi-fluid plasma model

Coupling 5-moment fluid models (for each species) to Maxwell equations

5-Moment Fluid

$$\partial_t \rho_\alpha + \boldsymbol{u}_\alpha \cdot \nabla \rho_\alpha = -\rho_\alpha \nabla \cdot \boldsymbol{u}_\alpha$$

$$\partial_t \boldsymbol{u}_\alpha + \boldsymbol{u}_\alpha \cdot \nabla \boldsymbol{u}_\alpha = -\boldsymbol{u}_\alpha \nabla \cdot \boldsymbol{u}_\alpha - \frac{1}{\rho_\alpha} \nabla P_\alpha + \frac{1}{\rho_\alpha} \nabla \cdot \left(\mu_\alpha \left(\nabla \boldsymbol{u}_\alpha + \nabla \boldsymbol{u}_\alpha^T - \frac{2}{3} \boldsymbol{I} \nabla \cdot \boldsymbol{u}_\alpha \right) \right)$$
$$+ \frac{q_\alpha}{m_\alpha} (\boldsymbol{E} + \boldsymbol{u}_\alpha \times \boldsymbol{B}) - \sum_\beta \nu_{\alpha\beta} (\boldsymbol{u}_\alpha - \boldsymbol{u}_\beta)$$

$$\partial_t P_\alpha + \boldsymbol{u}_\alpha \cdot \nabla P_\alpha = -\gamma P_\alpha \nabla \cdot \boldsymbol{u}_\alpha + \nabla \cdot \big((\gamma - 1) k_\alpha \nabla T_\alpha \big)$$
$$- \sum_\beta \frac{(\gamma - 1) \nu_{\alpha\beta} \rho_\alpha}{m_\alpha + m_\beta} \left(3 \big(T_\alpha - T_\beta \big) - m_\beta \left(\boldsymbol{u}_\alpha - \boldsymbol{u}_\beta \right) \text{^} 2 \right)$$

Maxwell Equations

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

$$\partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E} = 0$$

Fluid challenge: multiple time scales

Sandia National Laboratories

- Scaling is strongly dependent on a species α's mass, density, and temperature.
- Can be broken into frequency scales, velocity scales, and diffusion scales:

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

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

Momentum diffusivity

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

Thermal diffusivity

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

IMEX time integration

- Splitting the model up based on stiffness allows us to choose what goes into the implicit solve:
  - Explicit for slow, non-stiff terms
  - Implicit for fast, stiff terms

Implicit tableau	Explicit tableau
$\begin{array}{c\|c} c & A \\ \hline & b^t \end{array}$	$\begin{array}{c\|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^t \end{array}$

$$\partial_t u = f(u,t) + g(u,t)$$

$$u^{(i)} = u^n + \Delta t \sum_{j=0}^{j<i} \hat{A}_{ij} f\left(u^{(j)}, t_n + \hat{c}_j \Delta t\right) + \Delta t \sum_{j=0}^{j\leq i} A_{ij} g\left(u^{(j)}, t_n + c_j \Delta t\right)$$

$$u^{n+1} = u^n + \Delta t \sum_{i=0}^{i<s} \hat{b}_i f\left(u^{(i)}, t_n + \hat{c}_i \Delta t\right) + \Delta t \sum_{i=0}^{i\leq s} b_i g\left(u^{(i)}, t_n + c_i \Delta t\right)$$

Example 3-stage IMEX algorithm

Implicit Solves

Explicit Solves

$$\mathrm{u}^{(0)} = \mathrm{u}^{\mathrm{n}} + \Delta t\left(A_{00}\, g^{(0)}\right)$$

$$g^{(0)} = g(u^{(0)}, t_n + c_0 \Delta t)$$
$$f^{(0)} = f\left(u^{(0)}, t_n + \hat{c}_0 \Delta t\right)$$

$$\mathrm{u}^{(1)} = \mathrm{u}^{\mathrm{n}} + \Delta t\left(\hat{A}_{10} f^{(0)} + A_{10} g^{(0)} + A_{11}\, g^{(1)}\right)$$

$$g^{(1)} = g\left(u^{(1)}, t_n + c_1 \Delta t\right)$$
$$f^{(1)} = f(u^{(1)}, t_n + \hat{c}_1 \Delta t)$$

$$\mathrm{u}^{(2)} = \mathrm{u}^{\mathrm{n}} + \Delta t\, (\hat{A}_{20} f^{n} + \hat{A}_{21} f^{(1)} + A_{20} g^{(0)} + A_{21} g^{(1)} + A_{22} g^{(2)})$$

$$g^{(2)} = g\left(u^{(2)}, t_n + c_2 \Delta t\right)$$
$$f^{(2)} = f(u^{(2)}, t_n + \hat{c}_2 \Delta t)$$

$$\mathrm{u}^{\mathrm{n+1}} = \mathrm{u}^{\mathrm{n}} + \Delta t\, (\hat{b}_0 f^{n} + \hat{b}_1 f^{(1)} + \hat{b}_2 f^{(2)} + b_0 g^{(0)} + b_1 g^{(1)} + b_2 g^{(2)})$$

Splitting multi-fluid plasma model

- For most applications, multi-fluid plasma model's can be broken into fast and slow components based on the associated time scales:

$\partial_t \rho_\alpha$ [illegible]

$\partial_t \boldsymbol{u}_\alpha$ [illegible]

[illegible]

$\partial_t P_\alpha$ [illegible]

[illegible]

$\partial_t \boldsymbol{E}$ [illegible] $= $ [illegible]

$\partial_t \boldsymbol{B}$ [illegible] $= 0$

- The stiffness of these terms are **problem dependent**

Adaptable block preconditioning

- Efficient Implicit (in IMEX) time integration requires effective preconditioning
  - Effective means: “Stiffness arising from fast time scales ishandled by the preconditioner”
- Our approach is to use **Block preconditioning** to identify and isolate scales by grouping terms together

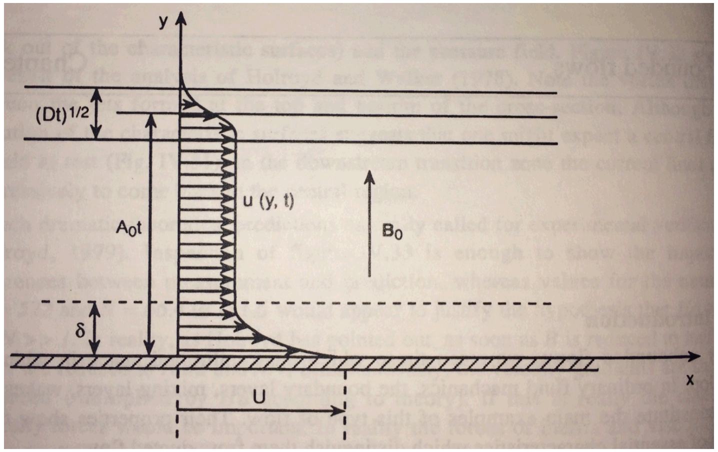
$$
\left[\begin{array}{ccc:ccc:cc}
\mathbf{D}_{\rho_i} & \mathbf{K}^{\rho_i}_{\rho_i u_i} & 0 & \mathbf{Q}^{\rho_i}_{\rho_e} & 0 & 0 & 0 & 0 \\
\mathbf{D}^{\rho_i u_i}_{\rho_i} & \mathbf{D}_{\rho_i u_i} & 0 & \mathbf{Q}^{\rho_i u_i}_{\rho_e} & \mathbf{Q}^{\rho_i u_i}_{\rho_e u_e} & 0 & \mathbf{Q}^{\rho_i u_i}_{E} & \mathbf{Q}^{\rho_i u_i}_{B} \\
\mathbf{D}^{\mathcal{E}_i}_{\rho_i} & \mathbf{D}^{\mathcal{E}_i}_{\rho_i u_i} & \mathbf{D}_{\mathcal{E}_i} & \mathbf{Q}^{\mathcal{E}_i}_{\rho_e} & \mathbf{Q}^{\mathcal{E}_i}_{\rho_e u_e} & \mathbf{Q}^{\mathcal{E}_i}_{\mathcal{E}_e} & \mathbf{Q}^{\mathcal{E}_i}_{E} & 0 \\
\hdashline
\mathbf{Q}^{\rho_e}_{\rho_i} & 0 & 0 & \mathbf{D}_{\rho_e} & \mathbf{K}^{\rho_e}_{\rho_e u_e} & 0 & 0 & 0 \\
\mathbf{Q}^{\rho_e u_e}_{\rho_i} & \mathbf{Q}^{\rho_e u_e}_{\rho_i u_i} & 0 & \mathbf{D}^{\rho_e u_e}_{\rho_e} & \mathbf{D}_{\rho_e u_e} & 0 & \mathbf{Q}^{\rho_e u_e}_{E} & \mathbf{Q}^{\rho_e u_e}_{B} \\
\mathbf{Q}^{\mathcal{E}_e}_{\rho_i} & \mathbf{Q}^{\mathcal{E}_e}_{\rho_i u_i} & \mathbf{Q}^{\mathcal{E}_e}_{\mathcal{E}_i} & \mathbf{D}^{\mathcal{E}_e}_{\rho_e} & \mathbf{D}^{\mathcal{E}_e}_{\rho_e u_e} & \mathbf{D}_{\mathcal{E}_e} & \mathbf{Q}^{\mathcal{E}_e}_{E} & 0 \\
\hdashline
0 & \mathbf{Q}^{E}_{\rho_i u_i} & 0 & 0 & \mathbf{Q}^{E}_{\rho_e u_e} & 0 & \mathbf{Q}_{E} & \mathbf{K}^{E}_{B} \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{K}^{B}_{E} & \mathbf{Q}_{B}
\end{array}\right]
\left[\begin{array}{c}
\rho_i \\ \rho_i \mathbf{u}_i \\ \mathcal{E}_i \\ \hdashline \rho_e \\ \rho_e \mathbf{u}_e \\ \mathcal{E}_e \\ \hdashline \mathbf{E} \\ \mathbf{B}
\end{array}\right]
$$

See for instance

- L. Chacon, Physics of Plasmas, 2008
- E.C. Cyr, J.N. Shadid, R.S. Tuminaro, R.P. Pawlowski, and L. Chacon, *SISC*, Vol. 35, 2013
- E.G. Phillips, J.N. Shadid, E.C. Cyr, H.C. Elman, and R.P. Pawlowski, *SISC*, Vol. 38, 2016
- E.G. Phillips, J.N. Shadid, and E.C. Cyr, Submitted to SISC, 2017

Resistive Alfven wave problem

- **Objective**: Run a full multi-fluid solver over MHD plasma scales.
- Solution is derived from resistive/viscous MHD which **ignores Hall effects**:
  - Hall parameter $\mathrm{H} = \frac{\omega_{ce}}{\nu_{ei}} = \frac{\eta\, B}{n_e e} \ll 1$
  - Reducing Hall effects in a magnetized multi-fluid model is tricky - requires large collision frequency
- Problem is used for verifying resistive, Lorentz force, and viscous operators:
  - Impulse shear due to a moving wall drives a **Hartmann layer**
  - Hartmann layer shear excites **Alfven wave** traveling along magnetic field
  - Alfven wave front diffuses due to momentum and magnetic diffusivity
  - Profile depends on the effective **Lundquist number** $\mathrm{S} = \frac{L\, v_A}{\lambda}$



R. Moreau, Magnetohydrodynamics, 1990

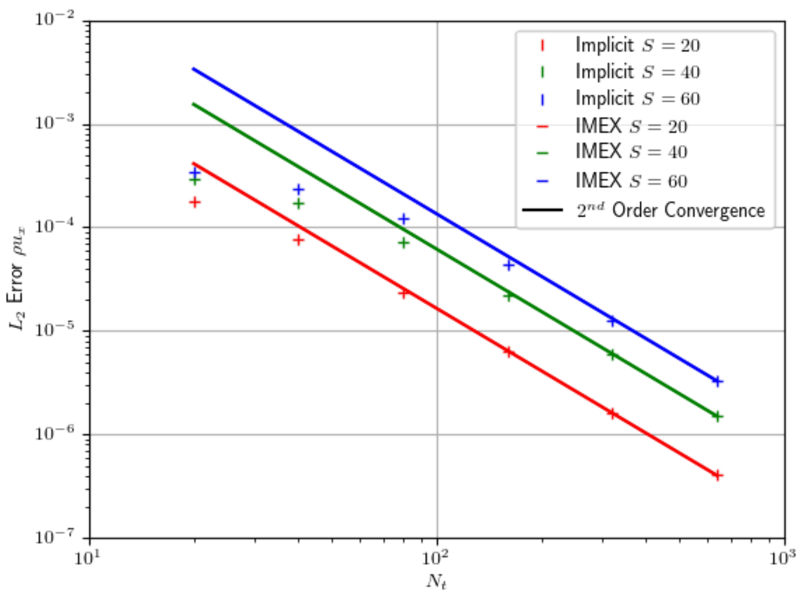
$$u_x = \frac{U}{4}\left(1 + \exp\left(\frac{v_A y}{\lambda}\right)\right)\mathrm{erfc}(\eta_+) + \frac{U}{4}\left(1 + \exp\left(-\frac{v_A y}{\lambda}\right)\right)\mathrm{erfc}(\eta_-)$$

$$B_x = \sqrt{\mu_0\rho}\,\frac{U}{4}\left(1 - \exp\left(\frac{v_A y}{\lambda}\right)\right)\mathrm{erfc}(\eta_+) - \sqrt{\mu_0\rho}\,\frac{U}{4}\left(1 - \exp\left(-\frac{v_A y}{\lambda}\right)\right)\mathrm{erfc}(\eta_-)$$

$$\eta_\pm = \frac{y \pm v_A t}{2\sqrt{\lambda t}}$$

Overstepping resistive plasmas*

- Convergence tests show expected convergence even when massively overstepping non-MHD plasma scales
- Roll-off at low resolutions due to under-resolving Hartmann layer
  - Large Lundquist number implies thin Hartmann layer



Plasma Scales for $S = 60$		
	Electrons	Ions
$\omega_p \Delta t$	$4 \cdot 10^7 - 1.3 \cdot 10^9$	$9.4 \cdot 10^5 - 3 \cdot 10^7$
$\omega_c \Delta t$	$1.7 \cdot 10^6 - 5.5 \cdot 10^7$	$9.4 \cdot 10^2 - 3 \cdot 10^4$
$\nu_{\alpha\beta} \Delta t$	$1.7 \cdot 10^{10} - 5.5 \cdot 10^{11}$	$9.4 \cdot 10^6 - 3 \cdot 10^8$
$v_s \frac{\Delta t}{\Delta x}$	$7 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
$u \frac{\Delta t}{\Delta x}$	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
$\frac{\mu}{\rho} \frac{\Delta t}{\Delta x^2}$	$0.4 - 12$	$0.01 - 0.3$
$c \frac{\Delta t}{\Delta x}$	167	

IMEX terms: implicit/explicit

Overstepping fast time scales is both stable and accurate. The inclusion of a resistive operator adds dissipation to the electron dynamics on top of the L-stable time integrator.

*Results computed using Drekar code

Future Work

EMPIRE-PIC

- FY18
  - Release open source
  - Realistic boundary conditions
    - SCL/FN/Secondary Emission
    - PML for EM
  - Implicit PIC algorithm research
  - Higher order EM and ES research
  - Performance optimization
  - Improved V&V suite
    - Automated convergence analysis
      Validation against B-dot experiment
- FY19
  - Implicit PIC working
  - Preproduction mode
  - Asynchronous Many Tasking (AMT) loadbalancing
  - Large Sierra scale simulation

EMPIRE-Fluid

- FY18
  - CG compatible Maxwell coupled to DG hydro with IMEX time integration
  - Linear wave verification suite **(see S. Miller's talk**)
  - Hardening of user interface
  - Test problems
    - Brio-Wu
    - Atmospheric blast
- FY19
  - Release open source
  - Performance portable implementation
  - Large scale simulation

Summary

- Foundation for next generation plasma model laid out
  - Basic electromagnetic capabilities
  - Multiple plasma modeling abilities (PIC and Fluid)
  - Kokkos for performance portability
- ASC and Sandia LDRD funding providing resources to develop
  - Extend the basic code to a preproduction application
  - Feature enhancements
- Future open source code base