The M3D-C1 Extended-MHD Code

Nathaniel Ferraro

A) Project Overview

- M3D-C1
- Goals: to understand the large-scale equilibria and stability of fusion plasmas
- Part of CEMM SciDAC (DOE)
  - Steve Jardin (PI), J. Breslau, J. Chen
- Collaboration with ITAPS for meshing software; TOPS for solvers
  - Mark Shephard (RPI)

B) Science Lesson

- Extended MHD is a fluid model of plasma
  - Disparate wave speeds
  - Highly anisotropic thermal transport
  - Very stiff equations!

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{v}) = 0$$

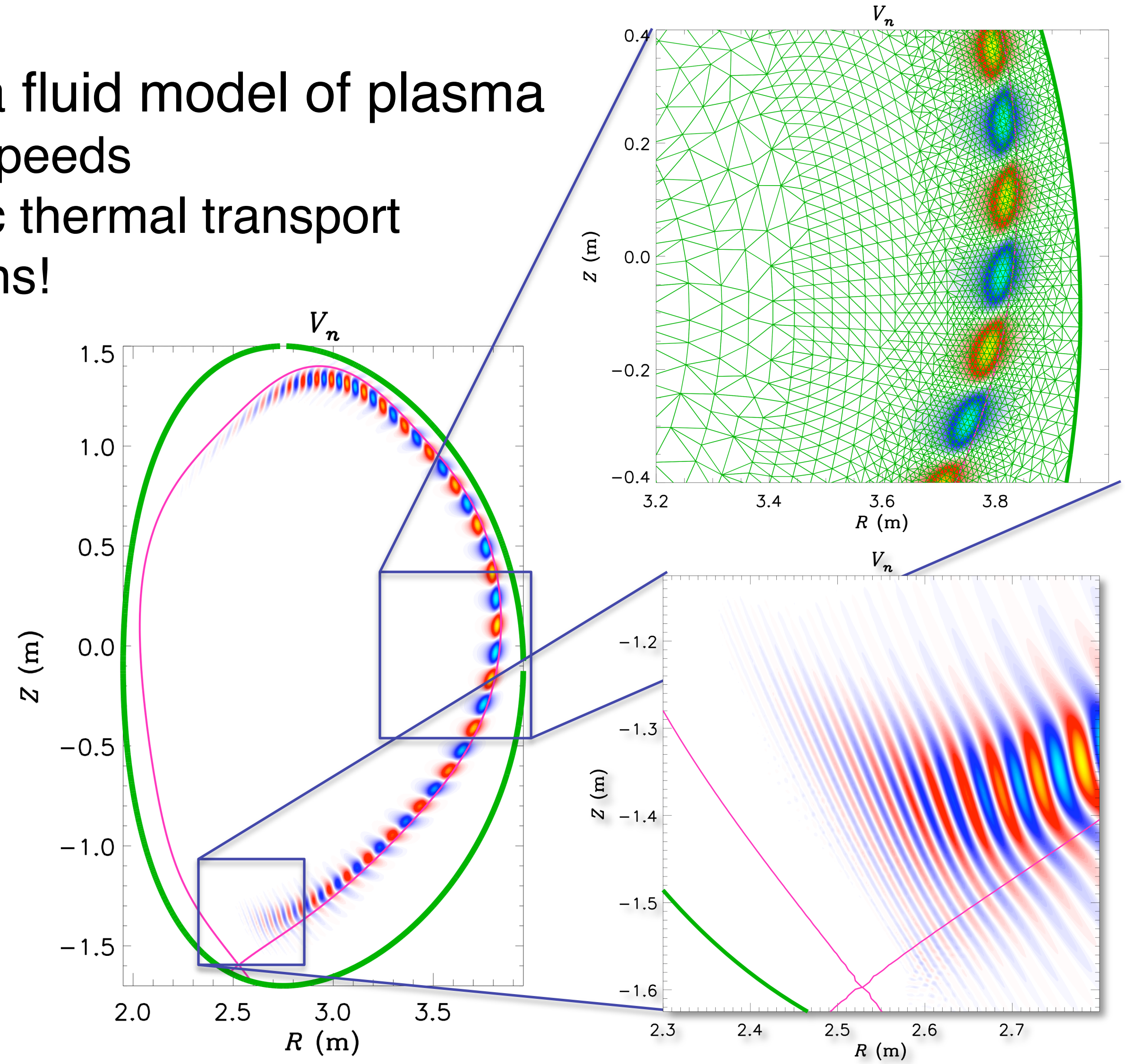
$$n\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \mathbf{J} \times \mathbf{B} - \nabla p - \nabla \cdot \Pi$$

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = \eta \mathbf{J}$$

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p = -\Gamma p \nabla \cdot \mathbf{v} - (\Gamma - 1)\nabla \cdot \mathbf{q}$$

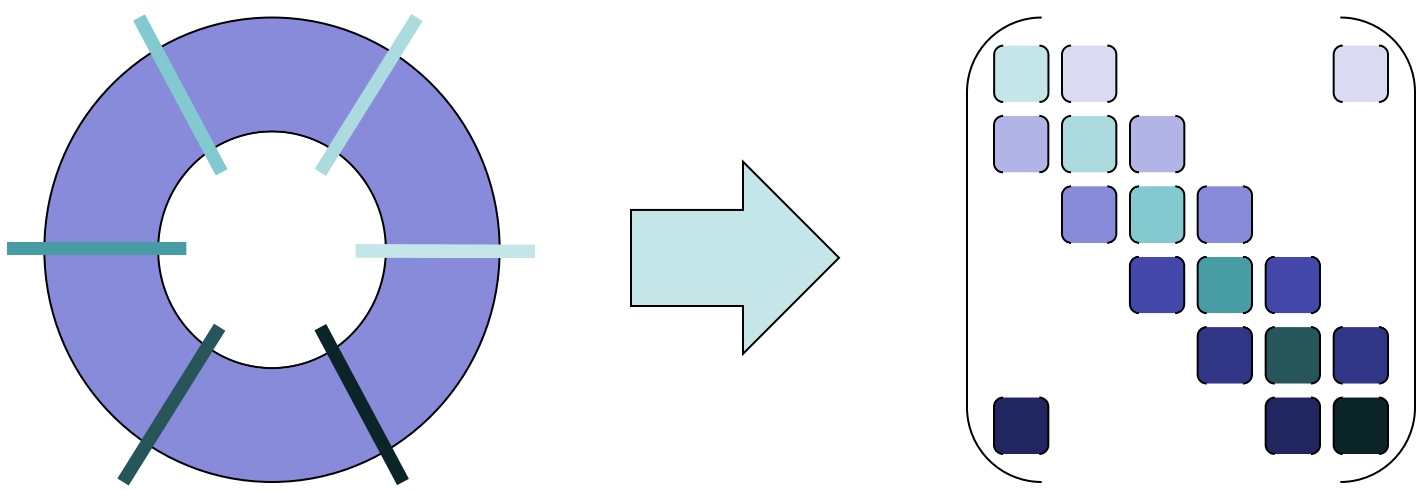
$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

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



C) Parallel Programming Model

- Fortran 95 on MPI
- PETSc, HDF5, SCOREC meshing software
- Runs on Hopper and several clusters at Princeton
- No plans for major changes to programming model



D) Computational Methods

- Lots and lots of PDEs
- Major matrices are block cyclic tridiagonal
  - Block-Jacobi preconditioning using SuperLU/ MUPMS
  - Preconditioned matrix solved using GMRES
- Future:
  - Particles?
  - Non-local operators?

E) I/O Patterns and Strategy

- HDF5 is used for main output
- One file/process is used for restart files (this needs to change)!
- Biggest file is HDF5 output (~10s of GB)
- HDF5 appears to scale well so far. Need to rethink restart file I/O
  - HDF5 output doesn't necessarily contain all the data necessary for restart

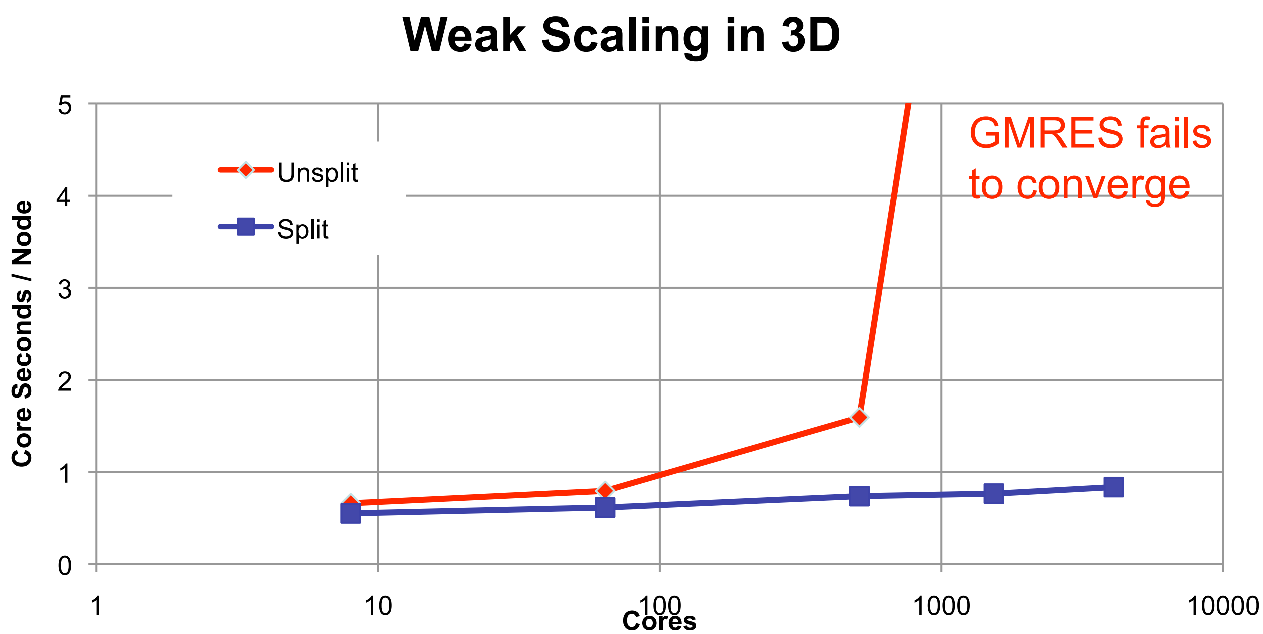
F) Visualization and Analysis

- HDF5 data is not in an easy format to visualize
  - Unstructured mesh
  - Coeffs of high order elements are written

- Custom IDL routines are used
- Capability to read data into VisIt has recently been provided (A. Sanderson)

G) Performance

- Profiling done using internal timers and TAU
- Computation/distribution of matrix elements are slower than expected
- Sparse linear solves are scaling bottleneck
  - Stiff system; needs very good preconditioning



H) Tools

- Debugging done with print statements
  - Most “bugs” are physics bugs; not coding bugs
- TAU is used for profiling
  - Memory profiling?

I) Status and Scalability

- Scaling depends enormously on problem
- Cases without strongly anisotropic thermal transport scale well, to at least 4096 cores
- We want to be scaling to 100,000 cores with strongly anisotropic thermal transport.
- Top pains:
  - Getting necessary software installed on computers
  - Comparing matrices
  - Errors that occur after 10,000 cpu-hours
  - Memory profiling with multiple external libraries

- Scaling has been improved by:
  - Split time step
  - Physics-based preconditioning
  - Block-Jacobi preconditioning

J) Roadmap

- Need to improve scaling with anisotropic thermal transport
  - Can we do better at preconditioning?
  - Should we change our algorithm?

- Can we identify and correct bottlenecks in creation of matrix?