(Extended) MHD Case Study

Stephen C. Jardin
Princeton Plasma Physics Laboratory

In consultation with

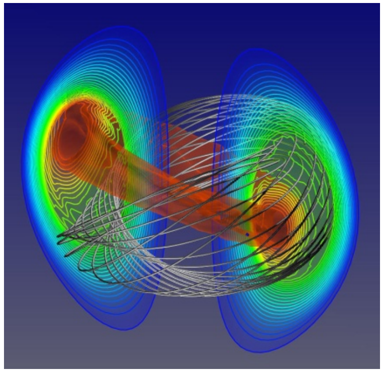
J. Breslau, J. Chen, N. Ferraro, G. Fu, S. Kruger,
D. Schnack, C. Sovinec, H. Strauss, L. Sugiyama

Large Scale Computing and Storage Requirements for
Fusion Energy Sciences
FES/ASCR/NERSC Workshop
August 3-4, 2010

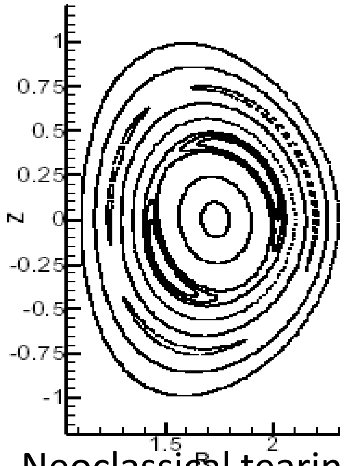
outline

- Types of calculations performed
- Equations solved
- Which codes are used
- General properties of Implicit MHD codes
- Scaling studies
- Kinetic closures
- Future trends
- Summary

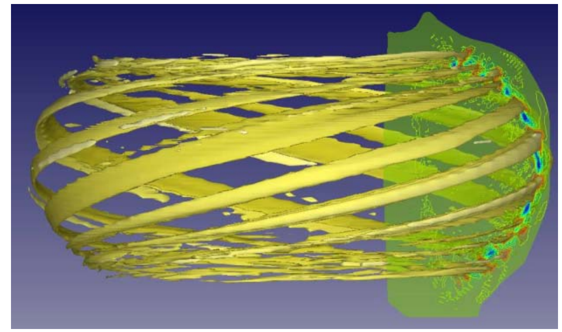
MHD codes are used for studying a variety of instabilities in tokamaks: Understanding these is very high priority for ITER



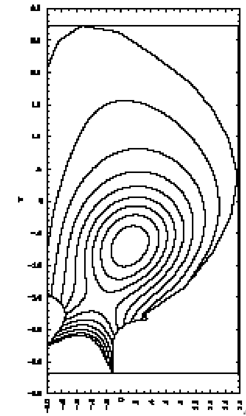
"sawtooth oscillations"



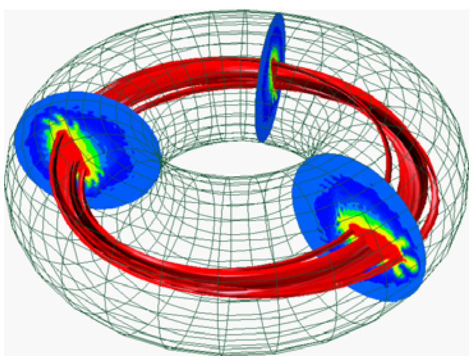
Neoclassical tearing modes and interaction of coupled island chains.



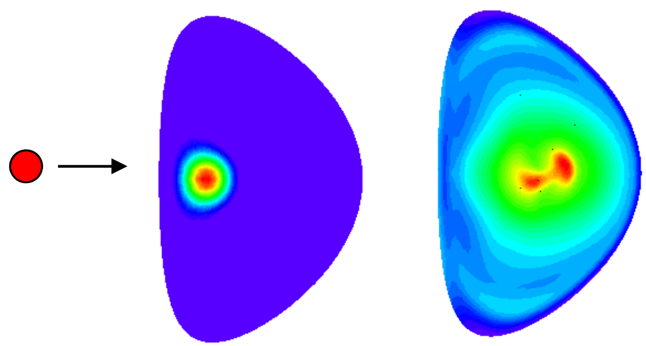
Edge Localized Modes



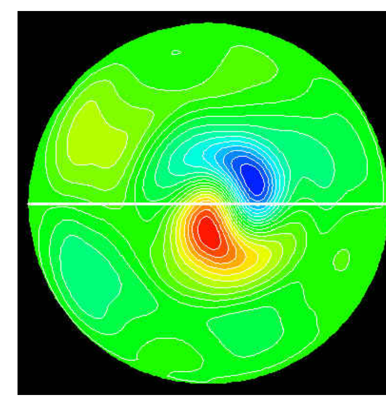
Disruption forces, RE, and heat loads during disruption



Disruptions caused by short wave-length modes interacting with helical structures.

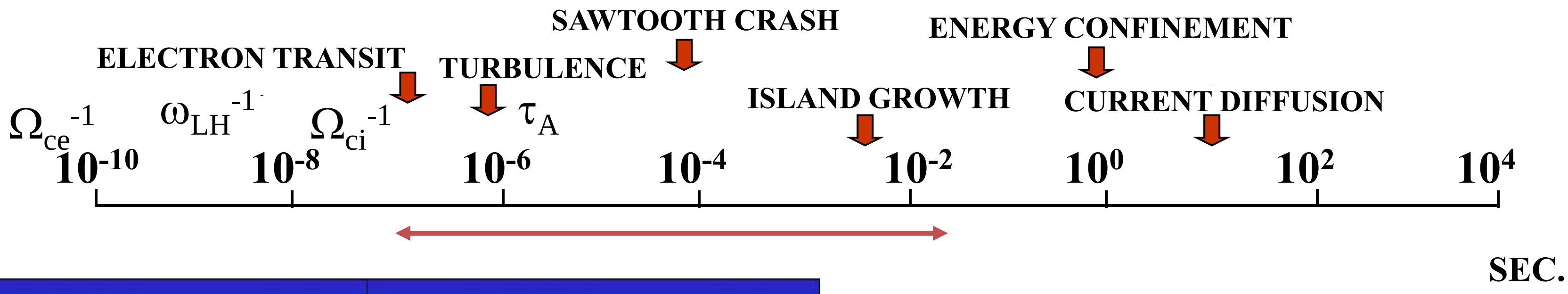


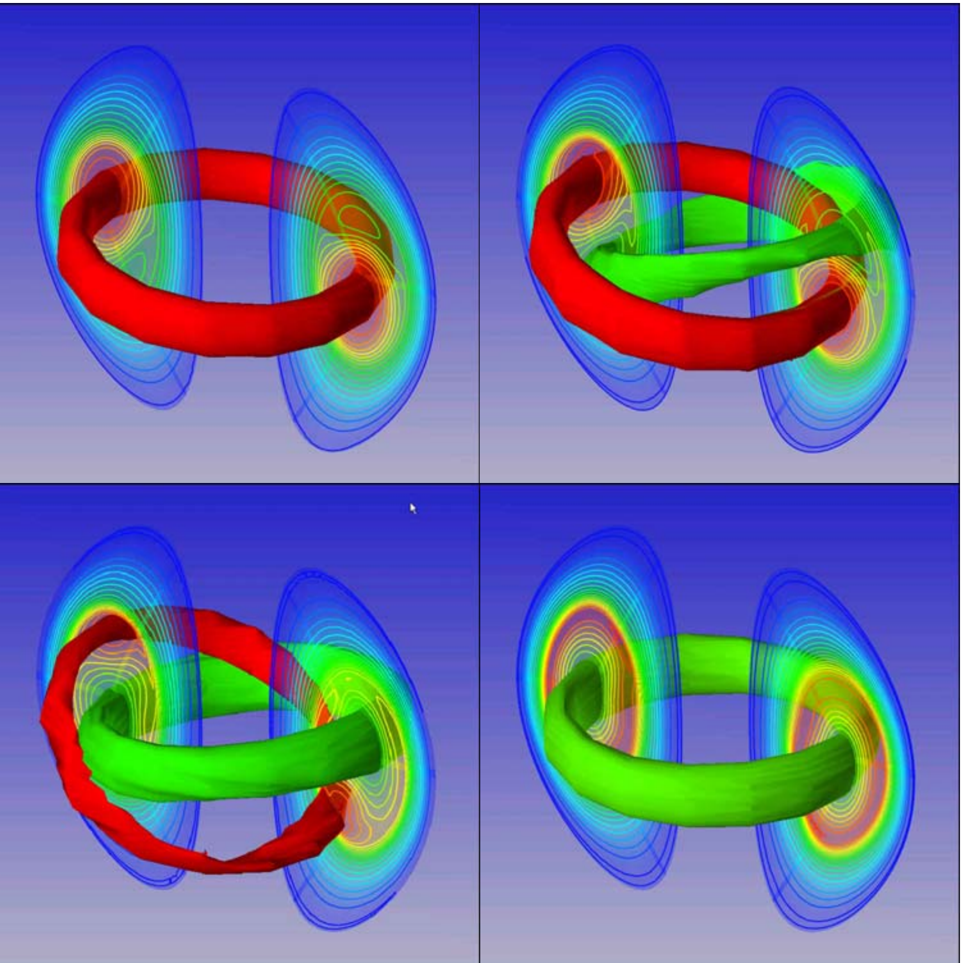
Mass redistribution after pellet injection



Energetic Particle modes

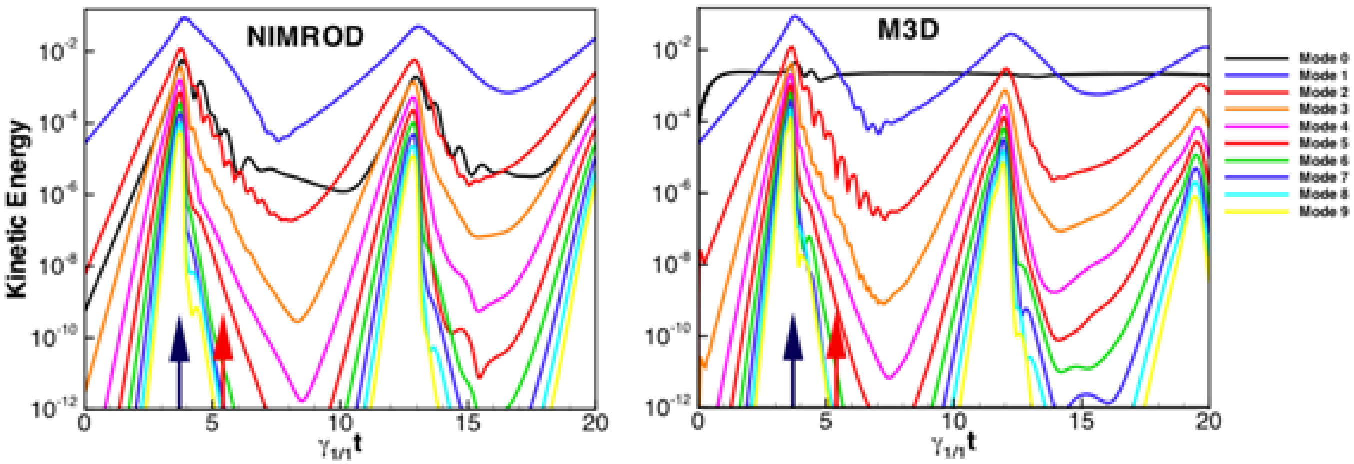
Extended MHD Codes solve 3D fluid equations for device-scale stability



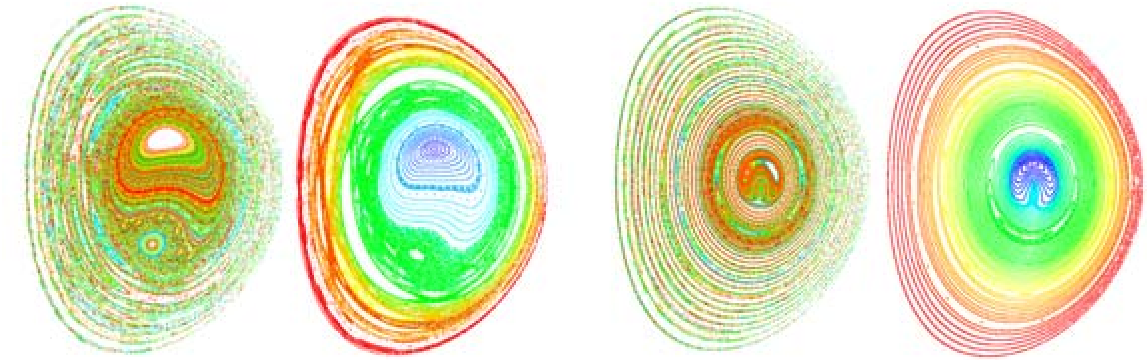


- Sawtooth cycle is one example of global phenomena that need to be better understood
- Can cause degradation of confinement, or plasma termination if the amplitude is too large and it couples with other modes
- There are several codes in the US and elsewhere that are being used to study this and related phenomena:

Excellent Agreement between NIMROD and M3D throughout the nonlinear cycle



Kinetic energy vs time in lowest toroidal harmonics



M3D **NIMROD** **M3D** **NIMROD**

Flux Surfaces during crash at 2 times

2-Fluid MHD Equations:

$$\frac{\partial n}{\partial t} + \nabla \bullet (n\mathbf{V}) = 0$$ continuity

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \qquad \nabla \bullet \mathbf{B} = 0 \qquad \mu_0 \mathbf{J} = \nabla \times \mathbf{B}$$ Maxwell

$$nM_i\left(\frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \bullet \nabla \mathbf{V}\right) + \nabla p = \mathbf{J} \times \mathbf{B} - \nabla \bullet \mathbf{\Pi}_{GV} + \mu \nabla^2 \mathbf{V}$$ momentum

$$\mathbf{E} + \mathbf{V} \times \mathbf{B} = \eta \mathbf{J} + \frac{1}{ne}\left(\mathbf{J} \times \mathbf{B} - \nabla p_e\right)$$ Ohm's law

$$\frac{3}{2}\frac{\partial p_e}{\partial t} + \nabla \bullet \left(\frac{3}{2} p_e \mathbf{V}\right) = -p_e \nabla \bullet \mathbf{V} + \eta J^2 - \nabla \bullet \mathbf{q}_e + Q_\Delta + S_{Fe}$$ electron energy

$$\frac{3}{2}\frac{\partial p_i}{\partial t} + \nabla \bullet \left(\frac{3}{2} p_i \mathbf{V}\right) = -p_i \nabla \bullet \mathbf{V} + \mu \left|\nabla V\right|^2 - \nabla \bullet \mathbf{q}_i - Q_\Delta + S_{Fi}$$ ion energy

Ideal MHD

Resistive MHD

2-fluid MHD

n number density

$\mathbf{B}$ magnetic field

$\mathbf{J}$ current density

$\mathbf{E}$ electric field

$nM_i \equiv \rho$ mass density

$\mathbf{V}$ fluid velocity

p_e electron pressure

p_i ion pressure

$p \equiv p_e + p_i$

e electron charge

μ viscosity

η resistivity

$\mathbf{q}_i, \mathbf{q}_e$ heat fluxes

Q_Δ equipartition

μ_0 permeability

$S_{Fe,i}$ Fusion power

Which Codes are being used?

Code Name	Developers/Major Users
NIMROD	C. Sovinec, S. Kruger, D. Schnack, C. Kim, many others
M3D	J. Breslau, L. Sugiyama, H. Strauss, G. Fu, J. Chen, others
M3D-C^1	N. Ferraro, S. Jardin, J. Breslau, J. Chen
XMHD	L. Chacon, others
HiFi	S. Lukin, A. Glasser, others

Note: M3D-C^1 is an extension of M3D that uses higher-order finite elements and is fully implicit

Center for Extended MHD Modeling (CEMM)

S. Jardin PI
2001-2010

a SciDAC activity…
Partners with:
TOPS
ITAPS
APDEC
SWIM
CPES

GA: V. Izzo, N. Ferraro
U. Washington: A. Glasser, C. Kim
MIT: L. Sugiyama, J. Ramos
NYU: H. Strauss
PPPL: J. Breslau, M. Chance, J. Chen, S. Hudson
TechX: S. Kruger, T. Jenkins, A. Pletzer
U. Colorado: S. Parker
U. Wisconsin: C. Sovinec , D. Schnack
Utah State: E. Held

NIMROD and M3D codes (+ new code development such as M3D-*C*[1] code)

General features of tokamak implicit MHD

~ 8-9 variables per element (or mesh point) **V**, **B**, ρ, p_e, p_i

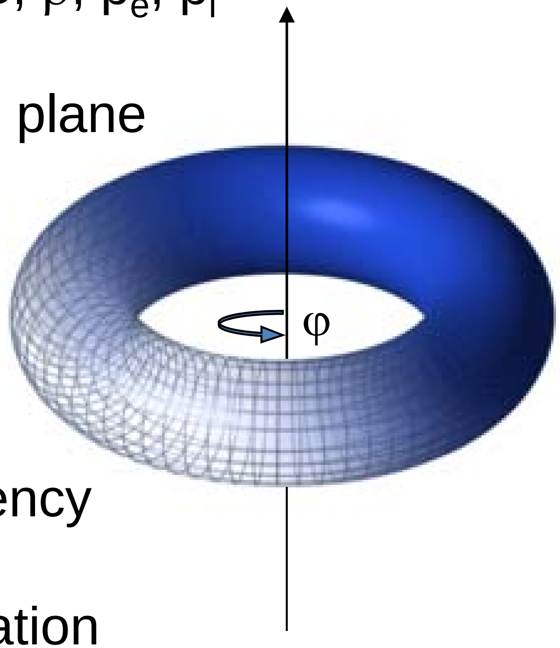
~ 10^4 -10^5 element DOF per variable per toroidal plane

~ 10^2 toroidal planes (or Fourier modes)

➔ 10^7 – 10^8 DOF per problem

➔ Large sparse matrix equations require low latency

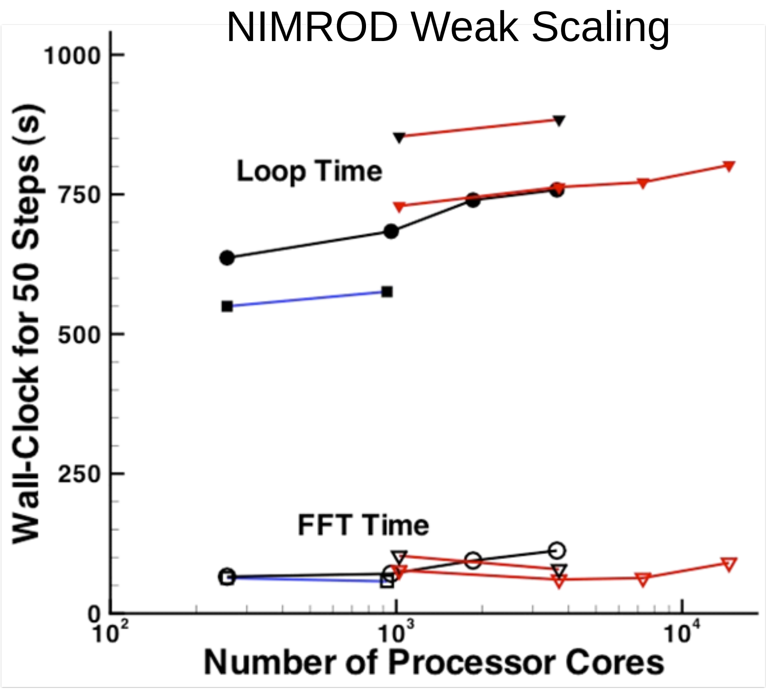
➔ Typically store all DOF ~ 100 times per calculation

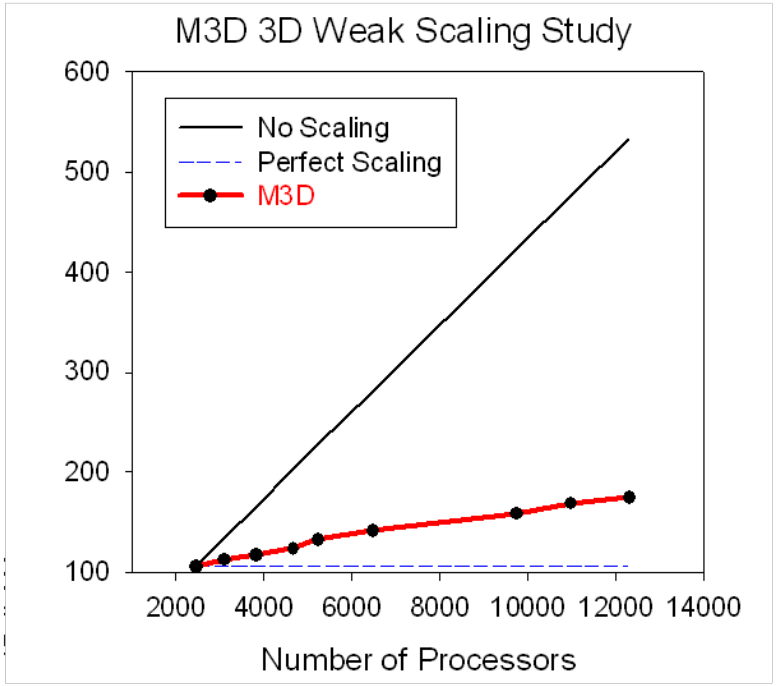


Codes vary in:

- single (big) matrix equation or several smaller equations
- non-linearly implicit (NK), linearly implicit, or partial implicit
- spectral, finite element, or finite differences in toroidal direction

Recent scaling studies on Franklin to over 12,000 processors





- Some limited Fourier coupling in preconditioner
- Data and loop reordering

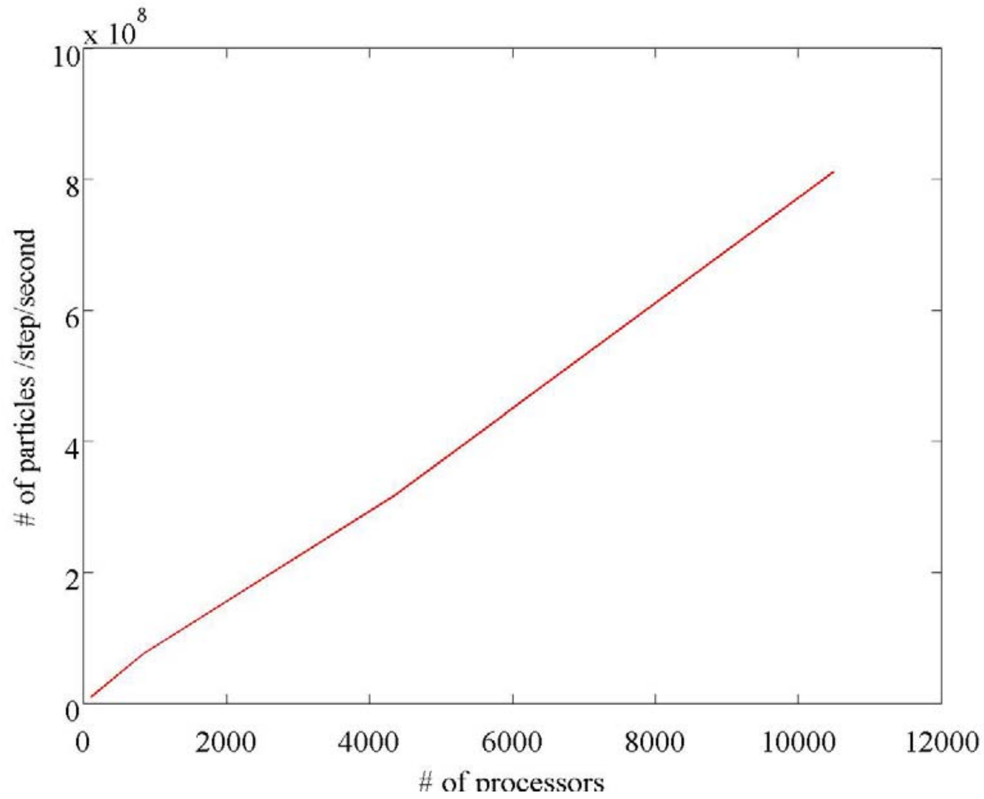
- AMG preconditioned CG
- RCM matrix reordering

M3D scaling properties can be improved by improving data layout…in progress, and being implemented in M3D-*C*[1]

Kinetic closures increase running time, but have very good scaling properties

- Both NIMROD and M3D have the option of including a population of high energy particles by using particle-based gyrokinetic dynamics

- This part of the calculation can be dominant, increasing the running time by factors of 3-5. However, it scales very well, as it is very similar to plasma turbulence codes (see C.S. Chang talk)



Future Trends

- Need for greater spatial resolution in all three spatial dimensions

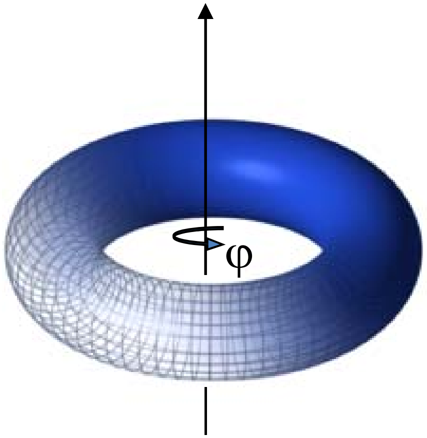
- Wall-clock time / calculation will not decrease as only # processors increases. Will give higher resolution calculations, but will require same or greater wall-clock time. **Some calculations now take months.**

- Certain pre-conditioners based on block-Jacobi with SuperLU would benefit from more memory per processor. We have shared memory machines at PPPL with 130GB of memory that get a lot of use.

- High-order finite elements in 3D may be able to use GPUs for integrations over 3D volume elements.

3D Nonlinear Solver Strategy for M3D-C^1

- In 2D, solve efficiently with direct solver up to $(200)^2$nodes
  - $10^5 - 10^6$ DOF
- In 3D, leads to block triangular structure



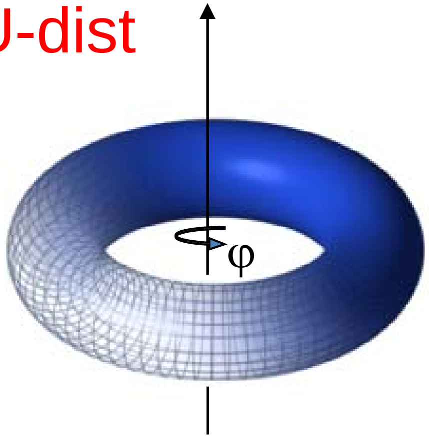
$$
\begin{bmatrix}
\mathbf{B}_1 & \mathbf{C}_1 & & & & & & \mathbf{A}_1 \\
\bullet & \bullet & \bullet & & & & & \\
 & \bullet & \bullet & \bullet & & & & \\
 & & \mathbf{A}_j & \mathbf{B}_j & \mathbf{C}_j & & & \\
 & & & \bullet & \bullet & \bullet & & \\
 & & & & \bullet & \bullet & \bullet & \\
 & & & & & \bullet & \bullet & \bullet \\
\mathbf{C}_N & & & & & & \mathbf{A}_N & \mathbf{B}_N
\end{bmatrix}
\bullet
\begin{bmatrix}
x_1 \\ . \\ x_{j-1} \\ x_j \\ x_{j+1} \\ . \\ . \\ x_N
\end{bmatrix}
=
\begin{bmatrix}
y_1 \\ . \\ y_{j-1} \\ y_j \\ y_{j+1} \\ . \\ . \\ y_N
\end{bmatrix}
$$

$\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j$

are 2D sparse matrices at plane j

- Block Jacobi preconditioner for M3D-C^1 corresponds to multiplying each row by $\mathbf{B}_j^{-1}$
  - PETSc has the capability of doing this using SUPERLU_Dist, but it is very CPU and memory intensive ….compromise with incomplete LU preconditioner
- NIMROD uses Fourier representation in the third direction, which makes this matrix dense
  - However, it is also more block diagonally dominant because off-diagonal blocks scale with perturbation size

Block Jacobi preconditioner corresponds to factoring each of the ~100 2D sparse matrices every few timesteps…now using SuperLU-dist



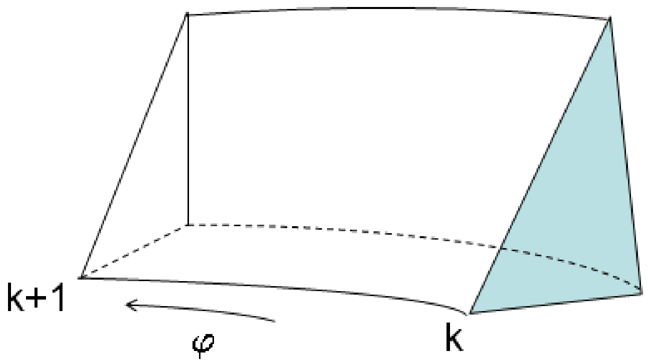
Largest case run to date corresponds to 801,378 DOF/plane with 129,413,052 non-zeros. Memory requirements are as follows:

#P on each 2D plane	Total Memory for 1 2D plane	Required per-processor memory
8	12GB	1.60GB
16	14GB	0.46GB
128	16GB	0.13GB
512	17GB	0.04GB

Thanks to X. Li for numbers

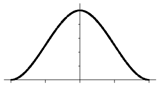
Problem size (and memory requirements) will increase in the future as resources permit.

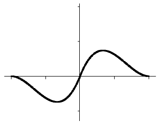
3D C^1 elements by combining Q_{18} triangles in (R,Z) Hermite cubic representation in the toroidal angle ϕ



Each toroidal plane has two Hermite cubic functions associated with it

$$\Phi_1(x) = \left(|x|-1\right)^2\left(2|x|+1\right); \qquad \Phi_2(x) = x\left(|x|-1\right)^2$$





Solution for each scalar function is represented in each triangular wedge as the product of Q_{18} and Hermite functions

$$U(R,Z,\varphi) = \sum_{j=1}^{18} \nu_j(R,Z)\left[U^1_{j,k}\,\Phi_1(\varphi/h) + U^2_{j,k}\Phi_2(\varphi/h) + U^1_{j,k+1}\Phi_1(\varphi/h-1) + U^2_{j,k+1}\Phi_2(\varphi/h-1)\right]$$

18 x 4=72 DOF for each scalar in integration volume. Products of 4 scalars evaluated at ~ 200 integration points for each term in equations.

Each Q_{18} basis function requires evaluation of quintic with 20 coefficients → 10^5 – 10^6 local multiplications and additions per element per timestep

→ May be possible to perform on GPUs

Summary

- Extended-MHD codes are addressing some of the most important problems for today's tokamaks and for ITER
- Codes solve a coupled system for 8 scalar variables using state-of-the art implicit techniques
- Codes have 10^7 – 10^8 DOF per problem … leads to large sparse matrix equations. Latency and data movement are key issues.
- Weak scaling has been demonstrated to over 12,000p.   This can be improved with better data layout
- Some calculations include kinetic closures which scale even better
- Real-world problems require more resolution than is now practical.   This implies more processors, but same or greater wall-clock time
- Storage and CPU-hour requirements are best extrapolated from past usage.
- Preconditioners of interest require some minimum memory per processor
- High-order finite elements may benefit from GPUs for local integrations over volumes.