JFNK within the semi-implicit scheme in NIMROD

Ben Jamroz, Scott Kruger, Travis Austin

Tech-X Corporation

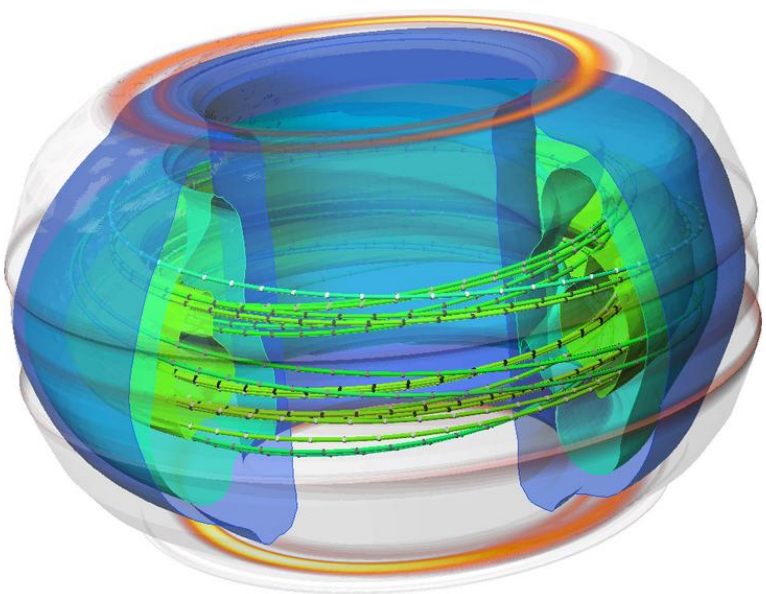
jamroz@txcorp.com

CEMM Meeting: Nov. 7, 2010

Hyatt Regency, Chicago Illinois

Outline

- Previous work: Fully Implicit JFNK
  - Analysis of time-stepping Schemes
- JFNK within the semi-implicit scheme
- Additive Schwarz method
- Conclusion & Future work



NIMROD currently uses a staggered in time method for advancing MHD eqns

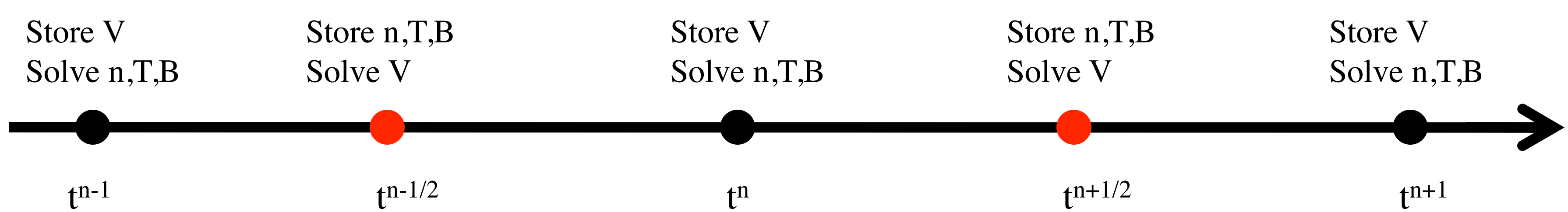
- Current scheme for MHD equations (excluding extended physics)

$$m_i n^{j+1/2}\left[\frac{\Delta\vec{V}}{\Delta t}+\frac{1}{2}\vec{V}^j\cdot\vec{\nabla}\Delta\vec{V}+\frac{1}{2}\Delta\vec{V}\cdot\vec{\nabla}\vec{V}^j\right]-\Delta t\mathcal{L}_{ideal}^{j+1/2}(\Delta\vec{V})+\vec{\nabla}\cdot\vec{\Pi}_i(\Delta\vec{V})$$
$$=\vec{J}^{j+1/2}\times\vec{B}^{j+1/2}+m_i n^{j+1/2}\vec{V}^j\cdot\vec{\nabla}\vec{V}^j-\vec{\nabla}p^{j+1/2}-\vec{\nabla}\cdot\vec{\Pi}_i(\vec{V}^j)$$

$$\frac{\Delta n}{\Delta t}+\frac{1}{2}\vec{V}^{j+1}\cdot\vec{\nabla}\Delta n=-\vec{\nabla}\cdot\left(n^{j+1/2}\vec{V}^{j+1}\right)$$

$$\frac{3n}{2}\left[\frac{\Delta T}{\Delta t}+\frac{1}{2}\vec{V}^{j+1}\cdot\vec{\nabla}\cdot\Delta T\right]+\frac{1}{2}\vec{\nabla}\cdot\left(\kappa_\perp\vec{\nabla}\Delta T\right)$$
$$=\frac{3n}{2}\vec{V}^{j+1}\cdot\vec{\nabla}T^{j+1/2}-nT\vec{\nabla}\cdot\vec{V}^{j+1}+\vec{\nabla}\cdot\left(\kappa_\perp\vec{\nabla}T^{j+1/2}\right)$$

$$\frac{\Delta\vec{B}}{\Delta t}+\frac{1}{2}\vec{V}^{j+1}\cdot\vec{\nabla}\Delta\vec{B}+\frac{1}{2}\vec{\nabla}\times\left(\frac{\eta}{\mu_0}\vec{\nabla}\times\Delta\vec{B}\right)+\frac{1}{2}\vec{\nabla}\left(\kappa_{div}\vec{\nabla}\cdot\Delta\vec{B}\right)$$
$$=-\vec{\nabla}\times\left(-\vec{V}^{j+1}\times\vec{B}^{j+1/2}+\frac{\eta}{\mu_0}\vec{\nabla}\times\vec{B}^{j+1/2}\right)-\kappa_{div}\vec{\nabla}\vec{\nabla}\cdot\vec{B}^{j+1/2}$$



TECH-X CORPORATION

Fully Implicit JFNK potentially has advantages over current scheme

- Fully Implicit – Time stepping uses Crank-Nicholson (CN)
  - Overcomes time step limitations due to nonlinearities such as advection, temperature-dependent diffusivities, etc.
  - Larger time steps -> Faster time to solve?
  - More accurate for a given dt, but
    does it matter for problems of interest?
- JFNK - Iterative (Newton type) method to solve nonlinear F(u)=0
- Action of the Jacobian (in building Krylov subspace) is approximated

$$\mathbf{F}'|_{\vec{u}}\vec{v} \approx \frac{\mathbf{F}(\vec{u}+\epsilon\vec{v}) - \mathbf{F}(\vec{u})}{\epsilon}$$

  - Don't need to form the analytical Jacobian
- Preconditioning needed to attain reasonable convergence rates
  - Preconditioner usually a simple approximation to the full Jacobian
  - Right preconditioned GMRES
  - Physics-based preconditioning (Chacon 2008)

Goal is fully implicit solve for all equations

- Apply Crank-Nicholson to discretized equations to solve for updates
- Evaluate all fields at the same time value

$$\mathbf{F}_n(\Delta\vec{x}) = \frac{\Delta n}{\Delta t} + \frac{1}{2}\nabla\cdot\left[\left(\mathbf{v}^j + \Delta\mathbf{v}\right)\left(n^j + \Delta n\right)\right] + \frac{1}{2}\nabla\cdot\mathbf{v}^j n^j$$

$$\begin{aligned}\mathbf{F}_T(\Delta\vec{x}) &= \frac{3}{2}\frac{\Delta T}{\Delta t} + \frac{3}{2}\left(\mathbf{v}^j + \Delta\mathbf{v}\right)\cdot\nabla\left(T^j + \Delta T\right) + \frac{3}{2}\mathbf{v}^j\cdot\nabla T^j \\ &+ \frac{1}{2}\left(T^j + \Delta T\right)\cdot\nabla\left(\mathbf{v}^j + \Delta\mathbf{v}\right) + \frac{1}{2}T^j\cdot\nabla\mathbf{v}^j\end{aligned}$$

$$\begin{aligned}\mathbf{F}_{\mathbf{B}}(\Delta\vec{x}) = \frac{\Delta\mathbf{B}}{\Delta t} &+ \frac{1}{2}\nabla\times\left(\left(\mathbf{v}^j + \Delta\mathbf{v}\right)\times\left(\mathbf{B}^j + \Delta\mathbf{B}\right)\right) + \frac{1}{2}\nabla\times\left(\mathbf{v}^j\times\mathbf{B}^j\right) \\ &- \frac{1}{2}\nabla\times\left(\frac{\eta}{\mu_0}\nabla\times\left(\mathbf{B}^j + \Delta\mathbf{B}\right)\right) - \frac{1}{2}\nabla\times\left(\frac{\eta}{\mu_0}\nabla\times\mathbf{B}^j\right) \\ &+ \frac{\kappa_{divB}}{2}\nabla\nabla\cdot\left(\mathbf{B}^j + \Delta\mathbf{B}\right) + \frac{\kappa_{divB}}{2}\nabla\nabla\cdot\mathbf{B}^j\end{aligned}$$

$$\begin{aligned}\mathbf{F}_{\mathbf{v}}(\Delta\vec{x}) &= m_i\left[\frac{(n^j + \Delta n)(\mathbf{v}^j + \Delta\mathbf{v}) - n^j\mathbf{v}^j}{\Delta t}\right] + \frac{m_i}{2}(n^j + \Delta n)(\mathbf{v}^j + \Delta\mathbf{v})\cdot\nabla(\mathbf{v}^j + \Delta\mathbf{v}) + \frac{m_i}{2}n^j\mathbf{v}^j\cdot\nabla\mathbf{v}^j \\ &+ \frac{m_i}{2}(\mathbf{v}^j + \Delta\mathbf{v})\nabla\cdot\left[(n^j + \Delta n)(\mathbf{v}^j + \Delta\mathbf{v})\right] + \frac{m_i}{2}\nabla\cdot(n^j\mathbf{v}^j) \\ &+ \frac{k}{2}\nabla\left[(n^j + \Delta n)(T^j + \Delta T)\right] + \frac{k}{2}\nabla(n^jT^j) - \frac{1}{2}\nabla\times(\mathbf{B}^j + \nabla\mathbf{B})\times(\mathbf{B}^j + \Delta\mathbf{B}) - \frac{1}{2}\nabla\times\mathbf{B}^j\times\mathbf{B}^j\end{aligned}$$

Symbolic Form for Fully Implicit Solve for MHD system

- Linearize to compute the Jacobian
  - An approximation is used for the preconditioner

$$J\Delta\vec{X} = \left[\begin{array}{cccc} D_n & 0 & 0 & U_{n\vec{V}} \\ 0 & D_T & 0 & U_{T\vec{V}} \\ 0 & 0 & D_{\vec{B}} & U_{\vec{B}\vec{V}} \\ L_{\vec{V}n} & L_{\vec{V}T} & L_{\vec{V}\vec{B}} & D_{\vec{V}} \end{array}\right] \left(\begin{array}{c} \Delta n \\ \Delta T \\ \Delta\vec{B} \\ \Delta\vec{V} \end{array}\right)$$

- Define

$$M = \left[\begin{array}{ccc} D_n & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_{\vec{B}} \end{array}\right] \qquad \Delta\vec{Y} = (\Delta n, \Delta T, \Delta\vec{B})$$

$$J\Delta\vec{X} = \left[\begin{array}{cc} M & U \\ L & D_{\vec{V}} \end{array}\right] \left(\begin{array}{c} \Delta\vec{Y} \\ \Delta\vec{V} \end{array}\right)$$

Extended MHD => M is not diagonal

Physics-based preconditioning method follows Chacon 2008

- Following Chacon (2008) apply LDU on 2x2 matrix and invert

$$\begin{bmatrix} M & U \\ L & D_{\Delta\vec{V}} \end{bmatrix}^{-1} = \begin{bmatrix} I & -M^{-1}U \\ 0 & I \end{bmatrix} \begin{bmatrix} M^{-1} & 0 \\ 0 & P_{\mathrm{schur}}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -LM^{-1} & I \end{bmatrix}$$

where $P_{\mathrm{schur}} = D_{\Delta\vec{V}} - LM^{-1}U$

- Approximate P_{schur} with

$$P_{\mathrm{sf}}\Delta\vec{V} = n^j \left[\frac{\Delta\vec{V}}{\Delta t} + \vec{V}\cdot\nabla\Delta\vec{V} + \Delta\vec{V}\cdot\nabla\vec{V}^j \right] - \Delta t \mathcal{L}^j_{\mathrm{ideal}}(\Delta\vec{V})$$

- Where $\mathcal{L}_{\mathrm{ideal}}$ is the ideal MHD operator which contains all of the wave propagation information
  - P_{sf}, M^{-1} matrices computed in NIMROD
  - Physics-based preconditioning is same physics as our semi-implicit operator

Scaling/Nondimensionalization required for Newton solve

- NIMROD is coded in dimensional units
- Physical quantities have a disparate range of scales

Quantity	Dimensional Factor
n	$n^* = 2.5 \times 10^{17}\, m^{-3}$
T	$T^* = 5 \times 10^3\, eV$
$\mathbf{B}$	$B^* = 1$ tesla
$\mathbf{v}$	$v_A^* = \sqrt{\frac{B^{*2}}{\mu_0 m_i n^*}}\, ms^{-1}$
t	$t^* = L^*/v_A^*$

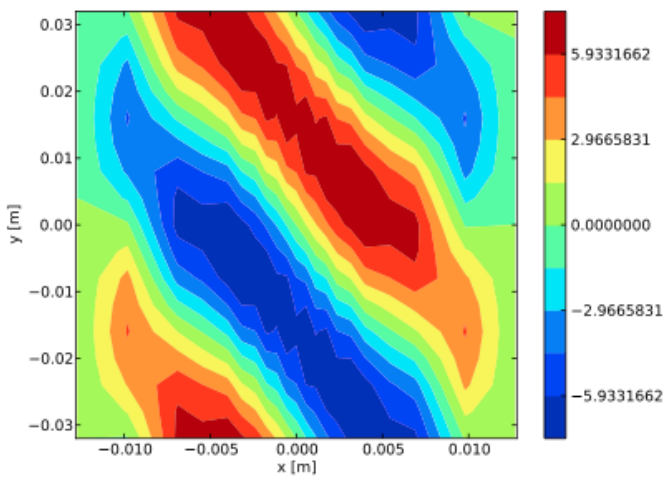
- In JFNK, using GMRES functional evaluations are used to determine the nonlinear iteration update
  - Magnitude of each residual determines the magnitude of nonlinear update
  - Differences in the mag. of residual and mag. of solution prevent convergence
- Conversion between nondimensional/dimensional variables and residuals

$$D_1 = \begin{bmatrix} n^* & & & \\ & T^* & & \\ & & B^* & \\ & & & v_A^* \end{bmatrix} \qquad D_2 = \begin{bmatrix} \frac{t^*}{n^*} & & & \\ & \frac{t^*}{n^* T^*} & & \\ & & \frac{t^*}{B^*} & \\ & & & \frac{t^*}{m_i n^* v_A^*} \end{bmatrix}$$

TECH-X CORPORATION

Fully Implicit JFNK with CN is unable to achieve reasonable convergence

- Poor GMRES convergence
  - Unable to reasonably advance in time
  - Ineffective preconditioner?
- High wave-number noise
- Coupled, noise affects preconditioner

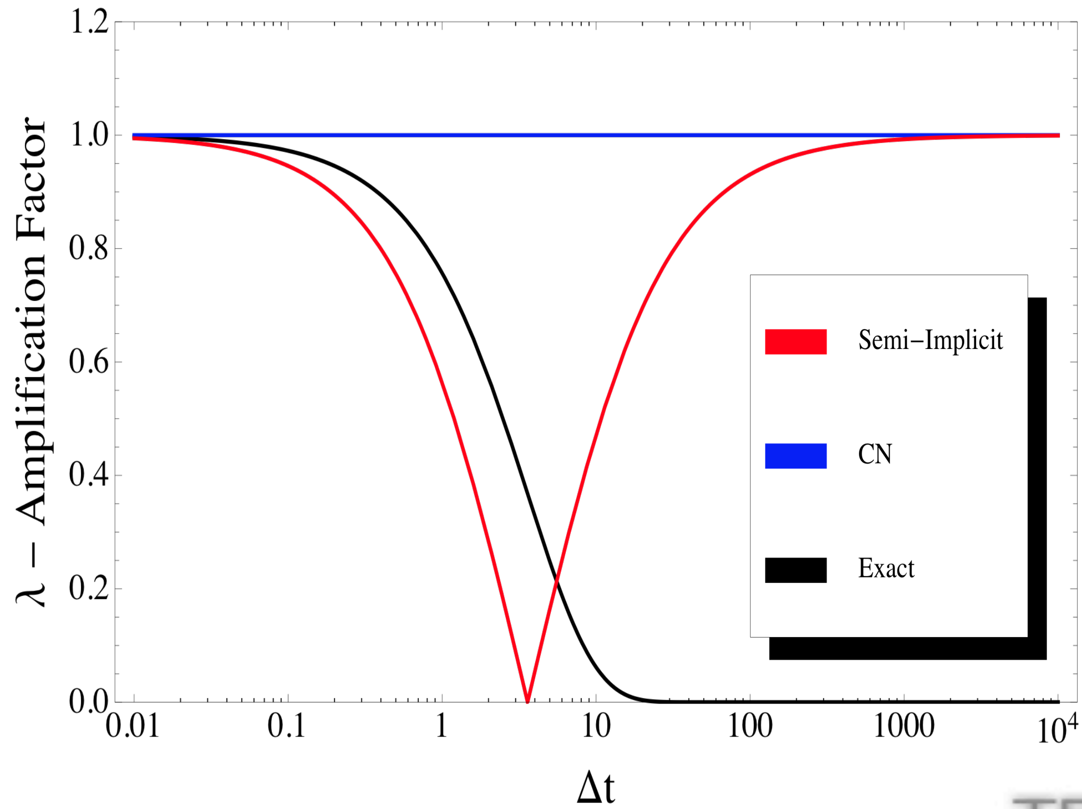


- y-component of velocity after one time step

CN may be responsible for inadequate damping

- Model Problem: 1D sound wave with damping
- Reinterpret sound speed as Alfven speed
- CN: No damping for large Alfven speed

$$\begin{aligned} \partial_t v &= -v_A \partial_x p + \nu \partial_x^2 v \\ \partial_t p &= -v_A \partial_x v \end{aligned}$$



Parameters taken from test case

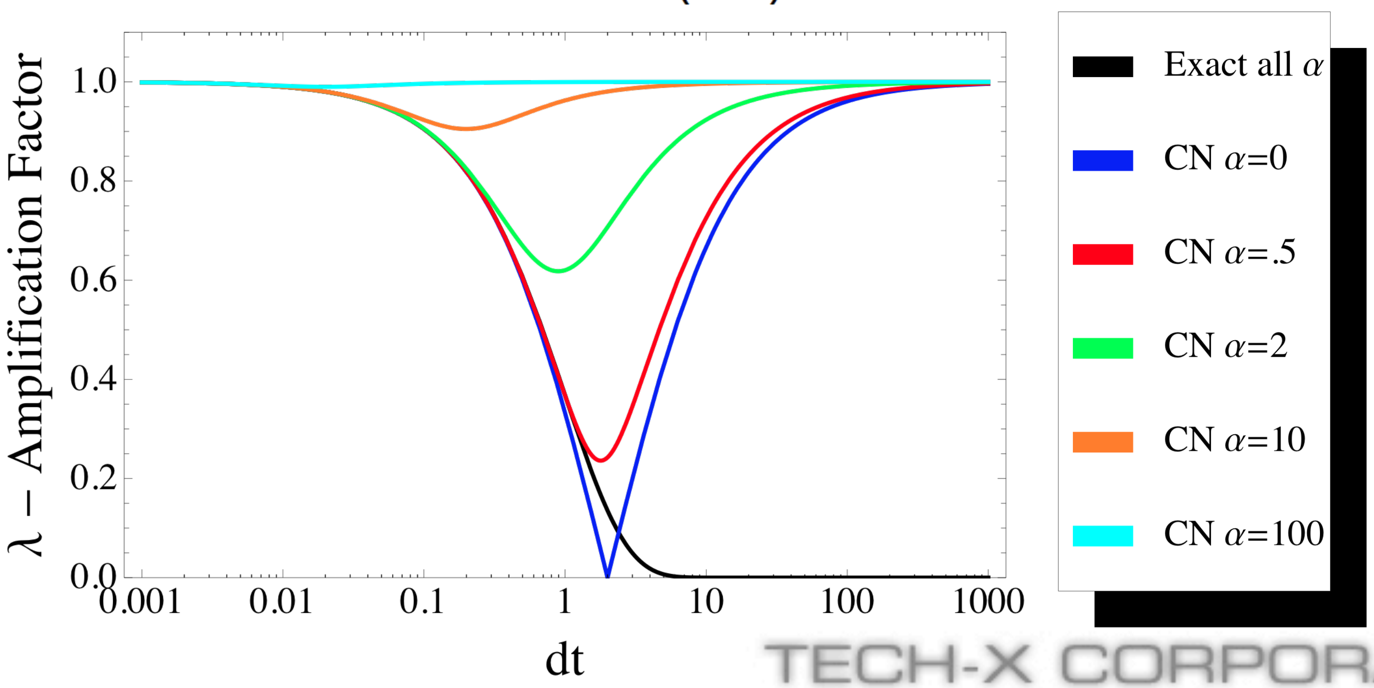
$$\begin{aligned} \nu &= 22 \\ k &= 2.4 \times 10^4 \\ v_A &= 3 \times 10^7 \end{aligned}$$

CN can't handle damped advection

- CN has difficulty accurately capturing physical diffusion in the presence of strong advection
- Model problem $\partial_t u = \alpha \partial_x u + \nu \partial_x^2 u$
  - Exact Solution

$$u = \hat{u} e^{ikx} e^{(\alpha i k - \nu k^2)t} \qquad |\lambda| = e^{-dt k^2 \nu}$$

- For CN, as α gets large $\lambda \equiv \lambda(dt) \to 1 \;\; \forall\, dt$



TECH-X CORPORATION

Side-stepping: JFNK within the semi-implicit scheme

- Perhaps noise in solution produced by CN time-stepping
- Implement JFNK within the semi-implicit time stepping scheme
  - Velocity solved at t=j*dt,
  - n, T, B solved at t=(j+1/2)*dt
  - Velocity decoupled, n decoupled – System for B,T only
- Nonlinear thermal conductivity, resistivity

$$
\begin{aligned}
\frac{\partial n}{\partial t} &= -\nabla \cdot (n\mathbf{v}) \\
\frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \left(-\mathbf{v} \times \mathbf{B} + \eta(T)\mathbf{J} + \frac{1}{ne}\mathbf{J} \times \mathbf{B} \right) \\
n\frac{dT}{dt} &= -(\gamma - 1)\left[-n\nabla \cdot \mathbf{v} + \nabla \cdot \kappa_{\parallel}(T)\hat{\mathbf{b}}\hat{\mathbf{b}} \cdot \nabla T + \nabla \cdot n\kappa_{\perp}(T)\nabla T + \eta(T)J^2 \right]
\end{aligned}
$$

Examples of Nonlinear coupling

- T coupled to B
  - Thermal conductivity depends upon B
  - B is independent of T

$$J = \begin{bmatrix} D_B & 0 \\ L_{TB} & D_T \end{bmatrix}$$

- B coupled to T
  - T independent of B
  - Temperature dependent resistivity

$$J = \begin{bmatrix} D_B & L_{BT} \\ 0 & D_T \end{bmatrix}$$

- Fully coupled
  - Thermal conductivity depends upon B and T
  - Temperature dependent resistivity

$$J = \begin{bmatrix} D_B & L_{BT} \\ L_{TB} & D_T \end{bmatrix}$$

- Currently these couplings are integrated using a predictor-corrector method
  - Predict T based on explicit B
  - Update B based on predicted value of T
  - Correct T based on "implicit" value of B

Split JFNK: Implementation Details

- Use semi-implicit scheme
  - Solve for v, then solve for n
- Preconditioner for T,B system $\kappa_{||}(T), \kappa_{\perp}(T), \eta(T), \hat{b}\hat{b}$
  - Assume small differences in
  - Full variances of these terms retained in the functional

$$\begin{bmatrix} D_B & L_{BT} \\ L_{TB} & D_T \end{bmatrix} \qquad \begin{bmatrix} D_B & 0 \\ 0 & D_T \end{bmatrix}$$

- Here D_B and D_T are the matrices for the non-coupled equations
  - Use NIMROD factorization routines

$$\begin{aligned}
\frac{\partial n}{\partial t} &= -\nabla \cdot (n\mathbf{v}) \\
\frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \left(-\mathbf{v} \times \mathbf{B} + \eta(T)\mathbf{J} + \frac{1}{ne}\mathbf{J} \times \mathbf{B} \right) \\
n\frac{dT}{dt} &= -(\gamma - 1)\left[-n\nabla \cdot \mathbf{v} + \nabla \cdot \kappa_{||}(T)\hat{\mathbf{b}}\hat{\mathbf{b}} \cdot \nabla T + \nabla \cdot n\kappa_{\perp}(T)\nabla T + \eta(T)J^2 \right]
\end{aligned}$$

Implicit evaluation of nonlinear coefficients is expensive

- Function evaluations are costly $\mathbf{F}'|_{\vec{u}}\vec{v} \approx \frac{\mathbf{F}(\vec{u}+\epsilon\vec{v}) - \mathbf{F}(\vec{u})}{\epsilon}$
  - Need to update $\kappa_{||}(T), \kappa_{\perp}(T), \eta(T), \hat{b}\hat{b}$ based on Krylov update
  - Vector transfers
    - NIMROD structures ⬄ Flat (PETSc) vectors

```
!Add DT to T to update k_\perp(T) k_\parallel(T)
DO ibl=1,nbl
    CALL vector_add(tion(ibl),work2(ibl),v2fac=0.5_r8)
ENDDO
CALL temp_store('ion end',newkarti)
CALL find_kappa_t
! Subtract DT back off
DO ibl=1,nbl
    CALL vector_add(tion(ibl),work2(ibl),v2fac=-0.5_r8)
ENDDO
CALL temp_store('ion end',newkarti)
```

Less iterations, more work…

	SI - GMRES	JFNK - GMRES	JFNK - fGMRES
Iterations	B 2.01, T 9.85	8.90	6.18
Time (s)	45.7	70.8	61.1

- Average over 100 time steps after evolving for 1000 time steps
  - Temperature dependent resistivity
  - Anisotropic thermal diffusion
    - Nonlinear dependence on temperature
- Less GMRES iterations
  - More work per GMRES it
  - Vector B is 3x larger than T
  - One iteration: Full system 4x more work than just T
- Potential benefit: many B and T iterations

Next Step: JFNK with "mixed" finite element formulation

- Introduce auxiliary variable for heat flux along mag. Field
  - Greater accuracy
- Remain within semi-implicit time-stepping scheme
  - Solve a system for q,B,T

$$\frac{\kappa_0 q_{\parallel}}{(\kappa_{\parallel} - \kappa_{\perp})} \equiv -\sqrt{\kappa_0}\hat{\mathbf{b}} \cdot \nabla T \qquad \hat{\mathbf{b}} = \frac{\mathbf{B} + \mathbf{B_{eq}} + \Delta\mathbf{B}}{||\mathbf{B} + \mathbf{B_{eq}} + \Delta\mathbf{B}||}$$

$$n\frac{dT}{dt} = -(\gamma - 1)\left[nT\nabla \cdot \mathbf{v} - \nabla \cdot \left(\kappa_{\perp}\nabla T + (\kappa_{\parallel} - \kappa_{\perp})\hat{\mathbf{b}}\hat{\mathbf{b}} \cdot \nabla T\right)\right]$$

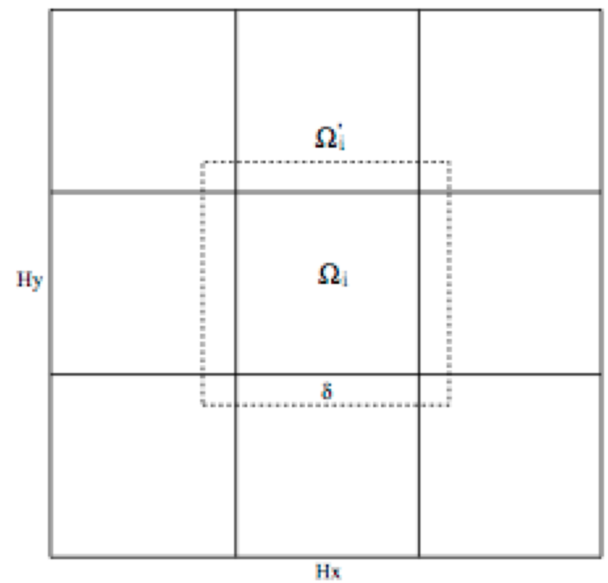
$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left(-\mathbf{v} \times \mathbf{B} + \eta(T)\mathbf{J} + \frac{1}{ne}\mathbf{J} \times \mathbf{B}\right)$$

- Jacobian

$$\begin{bmatrix} D_B & 0 & L_{BT} \\ L_{qB} & D_q & L_{qT} \\ L_{TB} & L_{Tq} & D_T \end{bmatrix} \begin{pmatrix} \Delta B \\ \Delta q_{\parallel} \\ \Delta T \end{pmatrix}$$

Additive Schwarz method is a **scalable** preconditioner

- Additive Schwarz method (ASM) limits communication
  - Approximately applies matrix inversion
  - Domain decomposition/processor mapping
- "Overlapping" Block Gauss-Jacobi relaxation
  - Iterative
- Per-process preconditioner application
  - No global LU decomp
  - Local decomposition
- PETSc provides routines for ASM
  - Matrices already ported to PETSc
  - Can be set from command line options
- Systems investigated required many iterations
  - SuperLU is faster

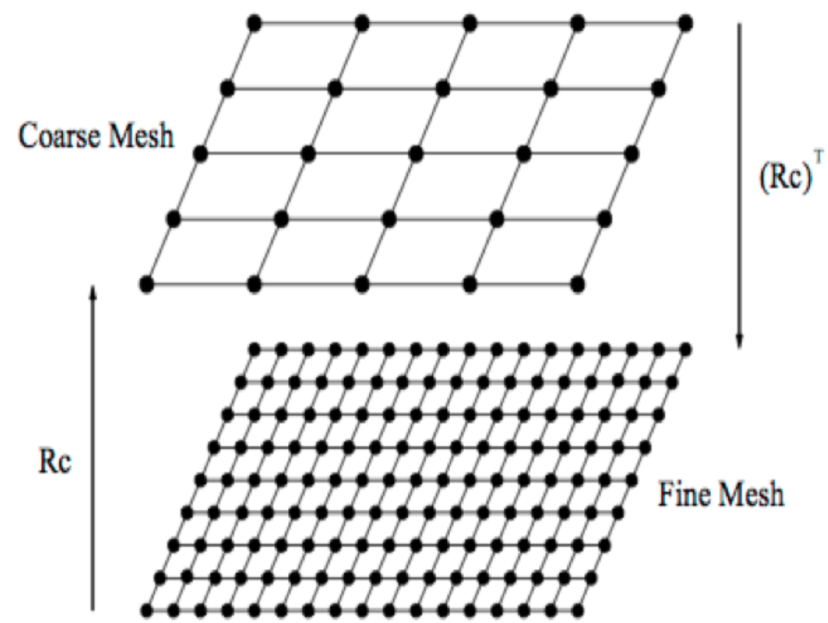


$$M^{-1} = \sum_{i=1}^{N} \left(R_i^0\right)^T A_i^{-1} R_i^\delta$$

TECH-X CORPORATION

Multi-level ASM required for an **effective** preconditioner

- Many iterations of ASM require a lot of communication
- Introduce a coarse level
  - Many iterations on the coarse level
    - Coarse level: linear elements
    - Less work since fewer points
    - Less communication on lower level
  - Use coarse solution as an initial guess
- Not directly implemented in PETSc
  - Define projection to coarse grid
  - Define interpolation from coarse grid
- Independent Solver on coarse grid
  - SuperLU_dist – smaller system



$$M_2^{-1} = R_c^T A_c^{-1} R_c + \sum_{j=1}^{N} (R_j^0)^T B_j^{-1} R_j^\delta$$

Conclusions/Future work

- Fully Implicit JFNK using CN wasn't effective
  - Use a different time-stepping scheme
- JFNK within the semi-implicit scheme
  - Less iterations, more work
  - May be competitive for some problems
  - Implement "mixed" finite element formulation
- Additive Schwarz Method
  - Scalable – available from interface to PETSc
    - Many iterations for MHD problems investigated
  - Implement multi-level method to be competitive