Time-Advance Algorithms, Solvers, and Extended MHD

Carl Sovinec,[1] Ben Jamroz,[2*] Travis Austin,[2*]
Scott Kruger[2]
[1]*University of Wisconsin-Madison*
[2]*Tech-X Corporation*

CEMM Meeting, April 18, 2010
Seattle, Washington

*DOE SBIR Grant #s:
DE-SC0000838, DE-FG02-08ER85154

Time-Advance Accomplishments over the Current Funding Cycle

- The implicit leapfrog algorithm has been analyzed with differential approximation.
- Nonlinear Newton solves have been applied to center $\mathbf{V}\cdot\nabla\mathbf{V}$ in the velocity advance and $\mathbf{J}\times\mathbf{B}$ in magnetic-field advance for the implicit leapfrog implementation.
- New preconditioning capability incorporates selective Fourier-component coupling.
- An implicit solve for the full system has been implemented for comparison.

The relative efficiency of time-centered and staggered advances needs to be tested.

- NIMROD's staggered advance often requires $\gamma\Delta t \cong 0.03$ for 1% accuracy on non-ideal modes.
  - Physical fields are solved separately, so the algebraic systems are relatively small.
- Time-centered advances just need $\gamma\Delta t \cong 0.35$ for 1% accuracy on all modes.
  - All fields are solved simultaneously, so algebraic systems are larger and yet less well conditioned.
- Recent computational work is starting to provide apples-to-apples comparisons.
  - U-WI group is implementing a θ-centered advance for the linear two-fluid system.
  - Tech-X is coupling NIMROD to PETSc's nonlinear and linear algebraic solvers.

It is possible to use NIMROD's 'framework' for θ-centered computations.

- The generic θ-implicit time-advance is

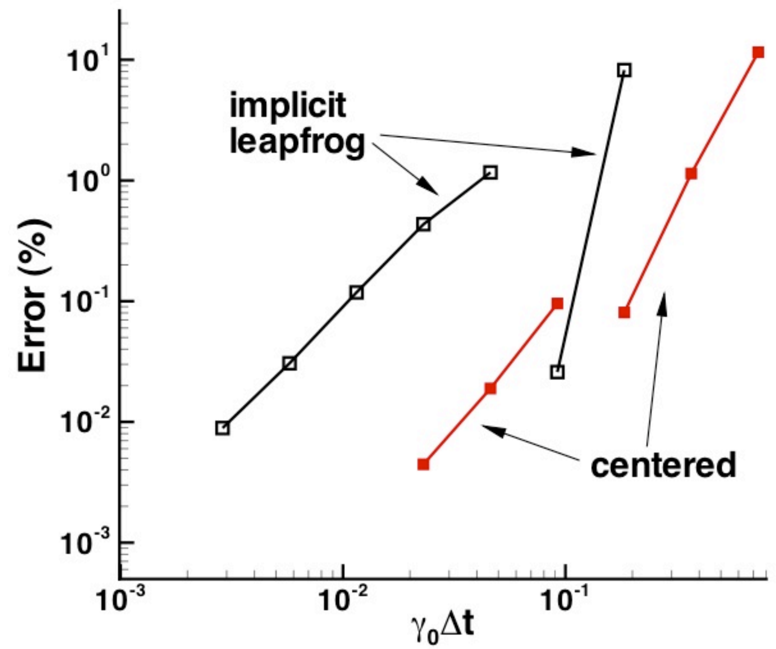
$$\frac{\partial}{\partial t}\underline{f} = \underline{G}\left(t,\underline{f}\right) \quad \Rightarrow \quad \Delta\underline{f} = \Delta t\left[\theta\underline{G}\left(t^{n+1},\underline{f}^{n+1}\right) + (1-\theta)\underline{G}\left(t^{n},\underline{f}^{n}\right)\right]$$

- The NIMITH code is a reorganized and augmented version of NIMROD for advancing $\underline{f} = \left(V_r, V_z, V_\phi, B_r, B_z, B_\phi, n, T\right)^T$
- At present, NIMITH is being developed for linear computations.
    - Several options are incomplete (aniso therm. cond.; GV+flow, etc.).
- The second part of the presentation covers the coupling to PETSc for nonlinear implicit solves.
    - The linear operator developed for NIMITH will provide alternative possibilities for preconditioning the nonlinear solve in PETSc.

Test results show both promise and problems at this point.

- Aside: normalization is important!
- Test cases (all linear) include sheared-slab and cylindrical tearing, and internal kink in cylindrical and toroidal geometry.
- The test of two-fluid tearing in a sheared slab is the kd_i=0.238 computation from the benchmark with the Ahedo-Ramos theory.
- Comparison of error in growth rates confirms 1% error at $\gamma\Delta t \cong 0.35$ with the centered computation.
- The implicit leapfrog consistently requires a time-step that is ~10 times smaller for the same accuracy, and each step runs ~2.5 times faster.
- Computations with hyperbolic pressure profiles and $\omega >> \gamma$ are problematic at this point:

Centered computations seem to be more prone to developing noise and divergence error.



Error in 2-fluid growth rates from impl. leapfrog and θ=1/2 computations.

Basis functions: sensitivity to the $\nabla\cdot B$ control parameter and other observations in recent tests motivate further consideration.

- Incompressible FE and spectral computations use separate, discontinuous representations for pressure that are of lower polynomial degree than flow-velocity.
    - We have tested the use of different polynomial degree for different fields, but all representations were continuous.
- Using continuous representations for fields that can be discontinuous causes a variety of problems: depending on details, spectral pollution, noise, slow convergence, etc.
- NIMROD's basis is designed for non-ideal systems, where there is enough smoothing to prevent discontinuity in the physical fields.

A potentially important use of discontinuous bases in NIMROD is to improve magnetic divergence control.

- The diffusive correction may be used with a discontinuous auxiliary field ϕ: $\frac{\partial}{\partial t}\mathbf{B} = -\nabla\times\mathbf{E} + \kappa_{divb}\nabla\phi \quad ; \quad \phi = \nabla\cdot\mathbf{B}$

$$\frac{\partial}{\partial t}\int \mathbf{A}\cdot\mathbf{B}\,dVol = -\int \mathbf{E}\nabla\times\mathbf{A}\,dVol + \oint d\mathbf{S}\cdot\mathbf{A}\times\mathbf{E} - \kappa_{divb}\int \phi\nabla\cdot\mathbf{A}\,dVol$$

$$\int v\phi\,dVol = \int v\nabla\cdot\mathbf{B}\,dVol \quad \text{for all } v, \mathbf{A} \text{ in the appropriate space}$$

- The discontinuous auxiliary field can be eliminated in the static condensation step prior to the linear solve.
- This is equivalent to penalty methods for incompressible flow.
- It should make the magnetic-field matrix less stiff and better conditioned.
- Using discontinuous *n* and continuous (*n*$\mathbf{V}$) can be applied to isothermal high-beta conditions.

JFNK Provides Nonlinear Implicit Capability

- 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 is needed to attain reasonable convergence rates
  - Preconditioner usually a simple approximation to the full Jacobian
  - Right preconditioned GMRES
  - Physics-based preconditioning (Chacon 2008)

Approach to Applying JFNK within NIMROD

- Non-invasive
  - Don’t change the structure of the code
  - Adapt to the existing routines
- Use as much functionality as possible
  - Less work, faster code
- Interface with PETSc
  - KSP library for linear solves
  - SNES library for nonlinear solves
- Staged approach:
  - N=0
    - Fully Implicit solve for velocity
    - Fully Implicit MHD
  - N>0, extended MHD, …

Goal is fully implicit solve for all equations

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

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

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

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

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

Symbolic Form for Fully Implicit Solve for MHD system

- Linearize to compute the Jacobian

$$J\Delta\vec{X} = \begin{bmatrix} D_n & 0 & 0 & U_{n\vec{V}} \\ 0 & D_T & 0 & U_{T\vec{V}} \\ 0 & 0 & D_{\vec{B}} & U_{\vec{B}\vec{V}} \\ L_{\vec{V}n} & L_{\vec{V}T} & L_{\vec{V}\vec{B}} & D_{\vec{V}} \end{bmatrix} \begin{pmatrix} \Delta n \\ \Delta T \\ \Delta\vec{B} \\ \Delta\vec{V} \end{pmatrix}$$

- Define $M = \begin{bmatrix} D_n & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_{\vec{B}} \end{bmatrix}$ $\qquad \Delta\vec{Y} = (\Delta n, \Delta T, \Delta\vec{B})$

$$J\Delta\vec{X} = \begin{bmatrix} M & U \\ L & D_{\vec{V}} \end{bmatrix} \begin{pmatrix} \Delta\vec{Y} \\ \Delta\vec{V} \end{pmatrix}$$

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

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

- Where $\mathcal{L}_{\mathrm{ideal}}$ is the ideal MHD operator which contains all of the wave propagation information
  - P_{sf}, M^{-1} matrices already exists in NIMROD (2D)
  - Physics-based pre-conditioning is same physics as our semi-implicit operator

Existing NIMROD infrastructure can be reused in performing the PETSc calls

$$\begin{bmatrix} M & U \\ L & D_V \end{bmatrix}^{-1} = \begin{bmatrix} I & -M^{-1}U \\ 0 & I \end{bmatrix} \begin{bmatrix} M^{-1} & 0 \\ 0 & P_{\mathrm{sf}}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -LM^{-1} & I \end{bmatrix}$$

- Diagonal inversions already coded in NIMROD
- Need to apply the upper (U) and lower (L) parts
  - Use existing matrix-free rhs routines in NIMROD
- For L part:
  - Temporarily set variables to zero and use the functional for V

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

- Similarly for U terms
  - Temporarily set variables to zero and use the functional for n,T,B

Status

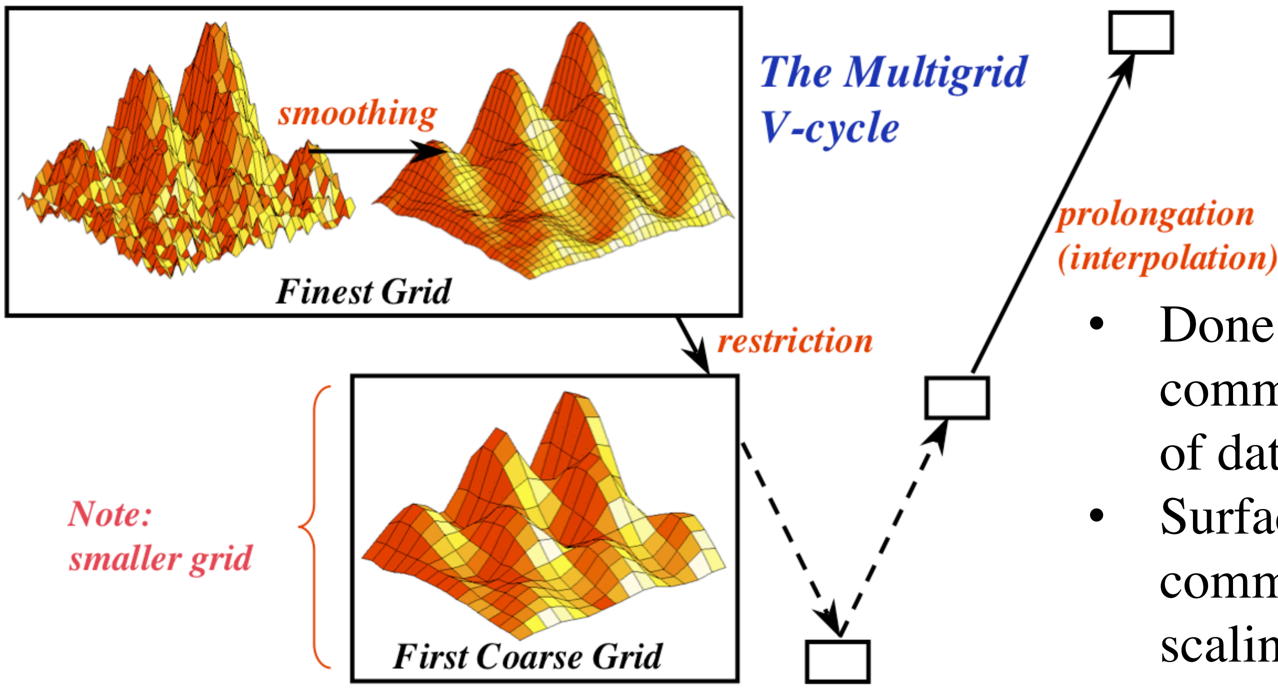
- Nonlinear Functional is computed
  - Copies:
    - NIMROD vectors into PETSc vector
    - Put NIMROD residuals into PETSc functional structure
    - PETSc vector into NIMROD vectors
  - Currently have many copies: Not optimized presently
    - E.g., need specialized copies for Schur complement-reduced vectors in preconditioning step
- On the fly nondimensionalization works
  - Produces residuals all within one order of magnitude
  - Error equally distributed across equations
  - All variables are equally modified by nonlinear updates
- GMRES with no preconditioning
  - Terribly slow convergence
- In progress: Finishing preconditioner
  - L and U terms

Summary/To Do / Future Work for nonlinear solves in NIMROD

- Current Challenges
  - Complete Preconditioning for the Full MHD system
- Future Work
  - 3D
    - Complex (Fourier) coefficients
    - Same (axisymmetric) preconditioner
  - Efficient method of applying preconditioner
    - Multigrid to apply D^{-1}
  - Upwinding-like smoothing for preconditioner
  - Including closures+ (anisotropic closures, Hall terms)

Preconditioning: Why use multigrid methods?

Multigrid methods treat all scales of the problem with the combination of smoothing and coarse grid corrections



- Done properly, each level communicate small amount of data
- Surface/volume of computation/ communication gives good scaling properties
- HYPRE's BoomerAMG has scaled to 125K processors for 3D 7-Pt Finite Difference Method.

Extended MHD contains many operators that challenge linear solvers

- Full extended MHD system in full matrix notation:

$$J\Delta\vec{X} = \begin{bmatrix} D_n & 0 & 0 & U_{n\vec{V}} \\ 0 & D_T & 0 & U_{T\vec{V}} \\ 0 & 0 & D_{\vec{B}} & U_{\vec{B}\vec{V}} \\ L_{\vec{V}n} & L_{\vec{V}T} & L_{\vec{V}\vec{B}} & D_{\vec{V}} \end{bmatrix} \begin{pmatrix} \Delta n \\ \Delta T \\ \Delta\vec{B} \\ \Delta\vec{V} \end{pmatrix}$$

- Within these sub-matrices, contain difficult linear matrices:

Operator	**Physics**	**Eqn.**	**Properties**
$\mathcal{D}_{thermal}$	Anisotropic Thermal Diffusion	T_α	HPD and Non-Symmetric
$\mathcal{D}_{res}$	Resistive Diffusion	$\vec{B}$	HPD
$\mathcal{L}_{ideal}$	MHD Waves	$\vec{V}$	HPD
$\mathcal{L}_{whistler}$	Whistler Waves	$\vec{B}$	Non-Symmetric

Why are these operators particularly challenging?

- Consider anisotropic heat conduction:

$$\mathcal{D}_{thermal}(\Delta T_\alpha) = \vec{\nabla} \cdot \left((\kappa_{\|} - \kappa_{\perp}) \hat{\mathbf{b}}\hat{\mathbf{b}} \cdot \vec{\nabla} \Delta T_\alpha + \kappa_{\wedge} \hat{\mathbf{b}} \times \vec{\nabla} \Delta T_\alpha + \kappa_{\perp} \vec{\nabla} \Delta T_\alpha \right)$$

- Extreme anisotropy causes extreme condition numbers
  - Also places constraint on spatial discretization => high-order FE
- High-order FE's generally do not satify div-curl identities exactly
- This admits small but nonlocal finite eigenvalues to curl-curl operators
  => Standard iterative methods will not work well for these operators

$$\mathcal{D}_{res}(\Delta \vec{B}) = \vec{\nabla} \times \left(\frac{\eta}{\mu_0} \vec{\nabla} \times \Delta \vec{B} \right)$$

- AMG methods for curl-curl operators require spatial discretization schemes that satisfy div-curl instabilities (e.g., staggered meshes) and yield local curl-free components eliminated by smoothing.

Anisotropic operators with curl-curl are unique to MHD community

- Linear wave operators have elements of curl-curl but with

$$
\begin{aligned}
\mathcal{L}_{ideal}(\Delta\vec{V}) &= \frac{1}{\mu_0}\left[\vec{\nabla}\times\vec{B}\times\vec{\nabla}\times\left(\Delta\vec{V}\times\vec{B}\right)-\vec{B}\times\vec{\nabla}\times\left[\vec{\nabla}\times\left(\Delta\vec{V}\times\vec{B}\right)\right]\right] \\
&\quad -\vec{\nabla}\left[\Delta\vec{V}\cdot\vec{\nabla}p+\frac{5}{3}p\vec{\nabla}\cdot\Delta\vec{V}\right]
\end{aligned}
$$

$$
\mathcal{L}_{whistler}(\Delta\vec{B})=\vec{\nabla}\times\frac{1}{ne}\left[\left(\vec{\nabla}\times\vec{B}^{j+1/2}\right)\times\Delta\vec{B}+\left(\vec{\nabla}\times\Delta\vec{B}\right)\times\vec{B}^{j+1/2}\right]
$$

- Many approaches exist for MG including those tailored to each operator (e.g., most AMG methods, many ML methods)

- Approach here: focus first on handling the complexity of high-order FEs for diffusion problems and later consider more complicated curl-curl operators.

Proposal: Use low-order system as pre-conditioner for high-order system

$$-\nabla \cdot \left(\vec{D}\nabla\phi\right) = f$$

$$\left(\vec{D}\nabla\phi, \nabla\psi\right) = (f, \psi) \;\; \forall\, \psi \in V$$

$$\left(\vec{D}\nabla\phi_H, \nabla\psi_H\right) = (f, \psi_H) \;\; \forall\, \psi_H \in V^H$$

Denser High-Order System

$$\left(\vec{D}\nabla\phi_L, \nabla\psi_L\right) = (f, \psi_L) \;\; \forall\, \psi_L \in V^L$$

Sparse Low-Order System

Automatic Preconditioner

Let $\mathbf{A}_i$ be the i^{th} element stiffness matrix associated with matrix high-order finite element matrix, $\mathbf{A}_H$. **Goal**: Find $\mathbf{C}_i$ that minimizes

$$\sum_{\mathbf{s}_k \notin \mathcal{N}(\mathbf{A}_i)} \frac{1}{\lambda_k^2} \|\lambda_k \mathbf{s}_k - \mathbf{C}_i \mathbf{s}_k\|_0^2$$

where $eig(\mathbf{A}_i) = \{(\mathbf{s}_i, \lambda_i)\}_{i=1,n}$ and $\mathcal{N}(\mathbf{C}_i) \equiv \mathcal{N}(\mathbf{A}_i)$. $\mathbf{C}_i$ is defined to have a nonzero pattern (i.e., sparsity) similar to employing bilinear finite elements. We then place the nonzeros of $\mathbf{C}_i$ in a vector, $\mathbf{z}$, define a matrix, $\mathbf{G}_k$, and redefine the system as $\mathbf{C}_i\ \mathbf{s}_k = \mathbf{G}_k\ \mathbf{z}$. We then solve for $\mathbf{z}$ with $\mathbf{H}$ representing the null space of $\mathbf{A}_H$.

Reformulation of $\lambda_k \mathbf{s}_k - \mathbf{C}_i \mathbf{s}_k$

$$\begin{pmatrix} \mathbf{G} \\ \mathbf{H} \end{pmatrix} \mathbf{z} = \begin{pmatrix} \frac{1}{\lambda_1}\mathbf{G}_1 \\ \frac{1}{\lambda_1}\mathbf{G}_2 \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_{n-1}}\mathbf{G}_{n-1} \\ \mathbf{G}_n \end{pmatrix} \mathbf{z} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{s}_{n-1} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{s} \\ 0 \end{pmatrix}$$

Least-Squares System for the Coefficients of $\mathbf{C}_i$

$$\begin{pmatrix} \mathbf{G}^T\mathbf{G} & \mathbf{H}^T \\ \mathbf{H} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ \ell \end{pmatrix} = \begin{pmatrix} \mathbf{G}^T\mathbf{s} \\ 0 \end{pmatrix}$$

Conclusions

- Initial linear results with a time-centered advance are mixed.
    - Some cases show second-order convergence and 1% error at $\gamma\Delta t \cong 0.35$.
    - Other computations are more prone to divergence. Control via discontinuous bases should help.
- Nonlinear Newton solves have been accomplished with minimal changes to NIMROD.
    - Planned work will bring the nonlinear PETSc coupling to production-level computations.
- Preconditioning based on low-order discretization and on spectral decomposition of submatrices is being tested.
    - Efficiency in the construction of the automatic preconditioner is being improved.

Extra slides

Proof-of-principle case focused just on advective operator

- Discretized velocity equation

$$m_i n^{j+1/2}\left(\frac{\Delta\vec{V}}{\Delta t}+\frac{1}{2}\vec{V}^j\cdot\vec{\nabla}\Delta\vec{V}+\frac{1}{2}\Delta\vec{V}\cdot\vec{\nabla}\vec{V}^j+\frac{1}{4}\Delta\vec{V}\cdot\vec{\nabla}\Delta\vec{V}\right)$$
$$-\Delta t\mathcal{L}_{ideal}^{j+1/2}(\Delta\vec{V})+\vec{\nabla}\cdot\vec{\Pi}_i(\Delta\vec{V})=\vec{J}^{j+1/2}\times\vec{B}^{j+1/2}$$
$$-m_i n^{j+1/2}\vec{V}^j\cdot\vec{\nabla}\vec{V}^j-\vec{\nabla}p^{j+1/2}-\vec{\nabla}\cdot\vec{\Pi}_i(\vec{V}^j)$$

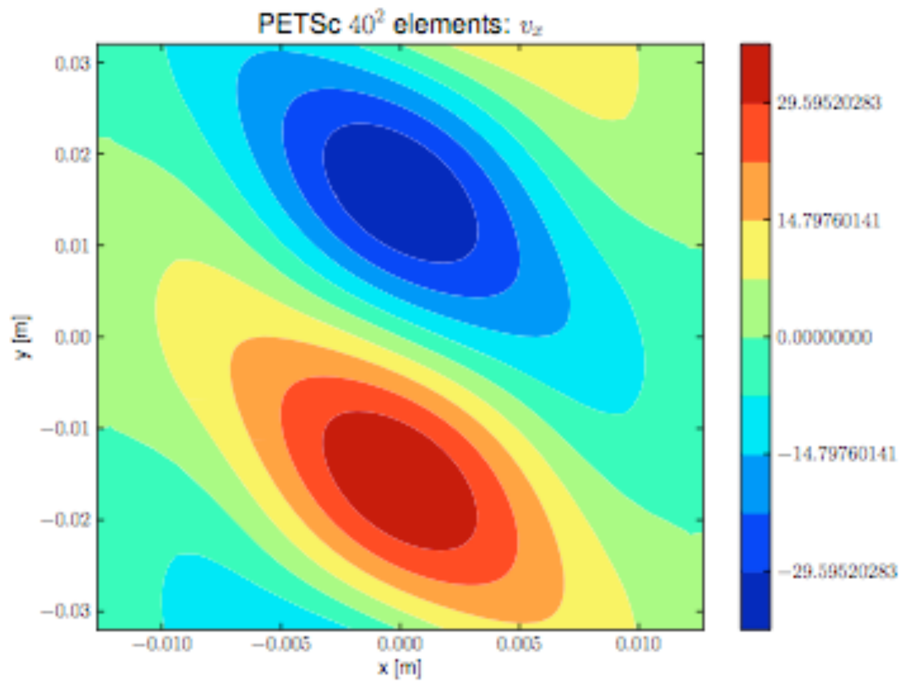
  - Sovinec: Newton method implemented within NIMROD's infrastructure exploiting the bi-linear nature of the operator.
- Our approach:
  - Include the nonlinear term $\mathbf{F}(\Delta\vec{V})=\mathbf{L}\Delta\vec{V}+\mathbf{N}(\Delta\vec{V})+\mathbf{R}$
  - Precondition GMRES using $\mathbf{L}$
  - Calculate the action of the Jacobian using finite differencing on $\mathbf{F}$

Velocity Results

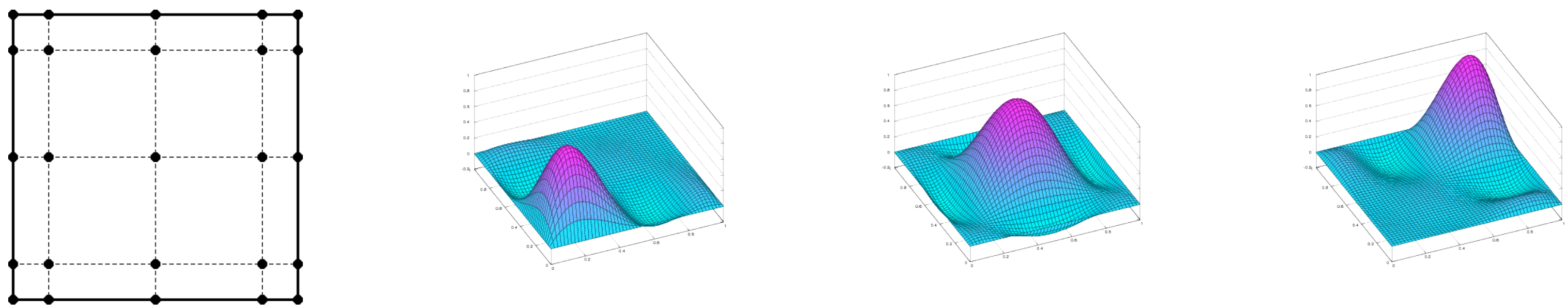
- N=0 Tearing Mode Instability

$$
\begin{aligned}
J_z &= 0.1e^{(x/.005)^2} \\
B_z(x=0) &= 1 \\
p(x) &= 0.001B_z^2/2
\end{aligned}
$$

- Convergence in 2-3 GMRES its
  - Similar to Sovinec's method
  - Roughly an order of magnitude slower (but not optimized, many caveats)



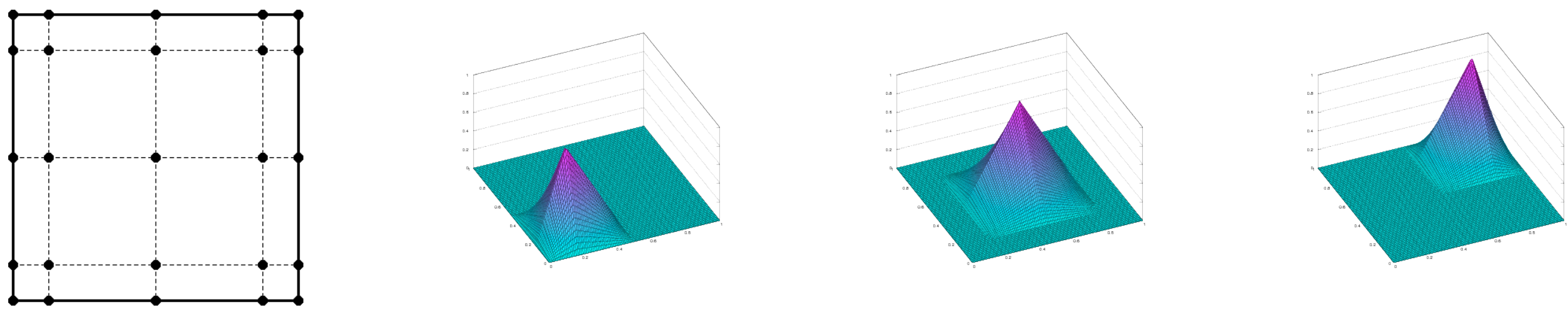
High-Order Finite Elements lead to dense sub-matrices



(a) (b) (c) (d)

Figure 1: (a) A single 2D biquartic element with Gauss-Legendre-Lobatto points used for the node locations and (b)-(d) three sample plots of the basis functions for the biquartic elements.

Low-Order Finite Elements lead to sparser matrices



(a) (b) (c) (d)

Figure 2: (a) A single 2D biquartic element with Gauss-Legendre-Lobatto points used for the node locations and (b)-(d) three sample plots of the basis functions for the bilinear finite elements on the new higher resolution mesh.

Extended MHD

- NIMROD has the capability to solve the equations of XMHD

$$\frac{Dn}{Dt} + n\vec{\nabla}\cdot\vec{V} = 0$$

$$m_i n\frac{D\vec{V}}{Dt} = -\vec{\nabla}p + \vec{J}\times\vec{B} - \underline{\vec{\nabla}\cdot\vec{\Pi}}$$

$$n\frac{DT_\alpha}{Dt} = -(\gamma-1)\left[nT_\alpha\vec{\nabla}\cdot\vec{V}_\alpha + \underline{\vec{\nabla}\cdot\vec{q}_\alpha + \Pi_\alpha : \vec{\nabla}\vec{V}_\alpha - \eta J^2 - Q_\alpha}\right]$$

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

$$\vec{E} + \vec{V}\times\vec{B} = \eta\vec{J} + \underline{\frac{1}{ne}\left[-\vec{\nabla}p_e + \vec{J}\times\vec{B} - \vec{\nabla}\cdot\vec{\Pi}_e\right] + \frac{1}{\epsilon_0\omega_{pe}^2}\left[\frac{\partial\vec{J}}{\partial t} + \vec{\nabla}\cdot\left(\vec{V}\vec{J} - \vec{J}\vec{V}\right)\right]}$$

… but we will only focus on MHD terms in discussing JFNK

Existing NIMROD infrastructure can be reused in performing the PETSc calls

- The nondimensional functional G yields residuals that are all within an order of magnitude
- Re-dimensionalizing ensures that each unknown is of the proper order
- Modified Jacobian for the dimensionless functional

$$G(\Delta\overline{\mathbf{x}}) = D_2 F(D_1 \Delta\overline{\mathbf{x}}).$$

$$\begin{aligned} G'(\Delta\overline{\mathbf{x}})\mathbf{y} &= \lim_{\epsilon\to 0} \frac{D_2\left[F\left(D_1\Delta\overline{\mathbf{x}}\right) + \epsilon F'\left(D_1\Delta\overline{\mathbf{x}}\right)D_1\mathbf{y}\right] - D_2 F\left(D_1\Delta\overline{\mathbf{x}}\right)}{\epsilon} \\ &= D_2 F'\left(D_1\Delta\overline{\mathbf{x}}\right)D_1\mathbf{y}. \end{aligned}$$

- The dimensional Jacobian is based on the dimensional equations
  - Already implemented in NIMROD
- Leverage functionals and matrices already implemented in NIMROD

Goal: create "automatic preconditioner" based on any high-order FE mesh

- Using this low-order finite element space as a preconditioner requires a rediscretization of the problem on the mesh constructed from high-order nodes (LO-DS).
- We are building an approach through PETSc where this idea can be used in an automatic sense by just passing off the element stiffness matrices (LO-LS) and solving a least-squares problem.
- We ensure the sparse matrix constructed from the least-squares problem approximates the smoother eigenvectors from the element stiffness matrices and gets exactly the nullspace.
- In the next few months we will be adding to NIMROD code that constructs the element stiffness matrices that allows us