Scalable implicit algorithms for stiff hyperbolic PDE systems

L. Chacón

Oak Ridge National Laboratory

J. N. Shadid, E. Cyr, P. T. Lin, R. Tuminaro, R. Pawlowski

Sandia National Laboratories

DOE Applied Mathematics Program Meeting

October 17-19, 2011

Washington, DC

Outline

- Motivation: the tyranny of scales
- Block-factorization preconditioning of hyperbolic PDEs
- Compressible resistive MHD
- Compressible extended MHD
- Incompressible Navier-Stokes and MHD (infinite sound-speed limit)

“The tyranny of scales”
(2006 NSF SBES report)

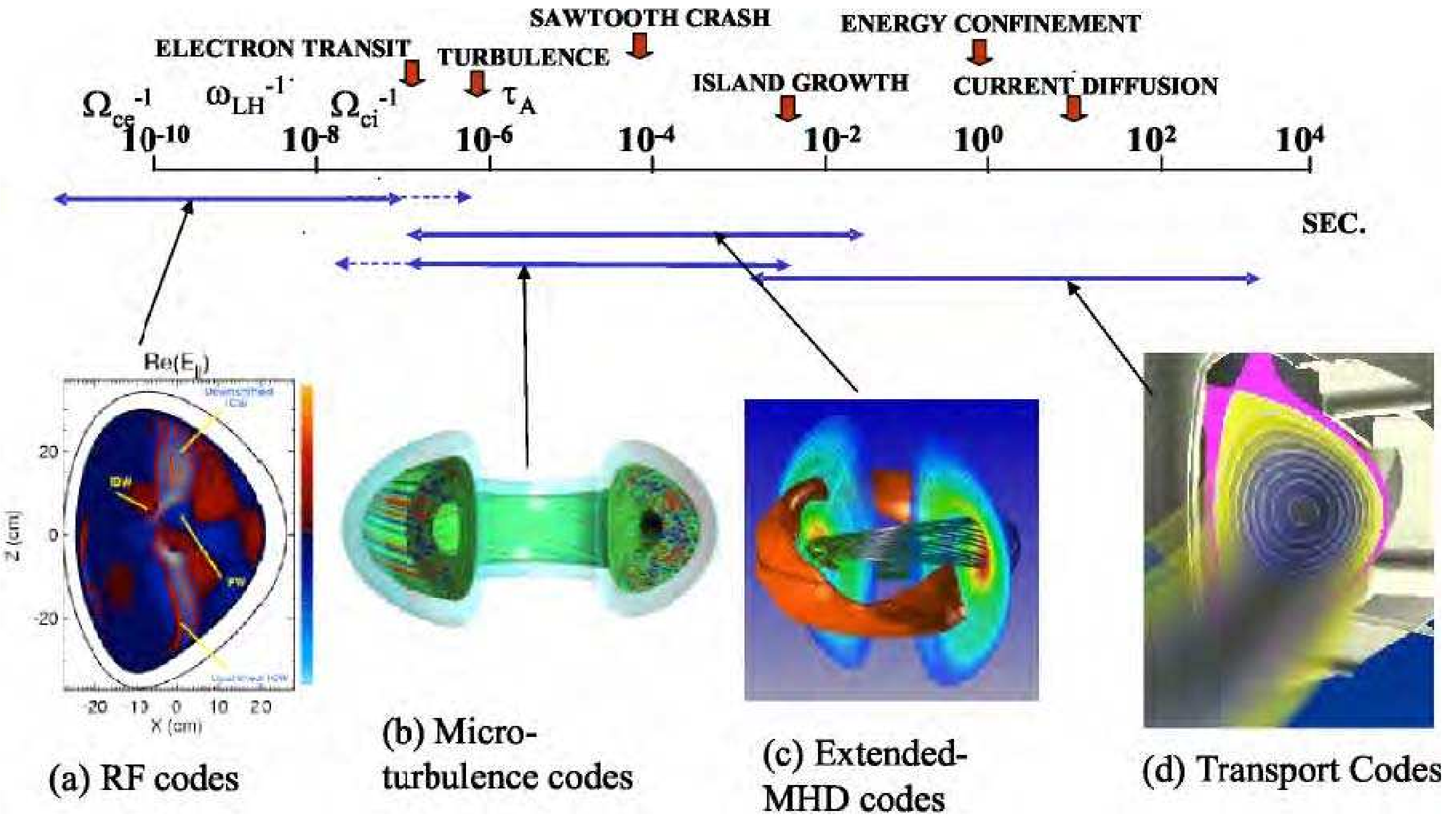


Figure 1: Time scales in fusion plasmas (FSP report)

Algorithmic challenges in temporal scale-bridging

- PDE systems of interest typically have mixed character, with hyperbolic and parabolic components.
  - Hyperbolic stiffness (linear and dispersive waves): $\kappa(J) \sim \Delta t\, \omega_{fast} \sim \frac{\Delta t}{\Delta t_{CFL}} \gg 1$
  - Parabolic stiffness (diffusion): $\kappa(J) \sim \frac{\Delta t\, D}{\Delta x^2} \gg 1$
- In some applications, fast hyperbolic modes carry a lot of energy (e.g., shocks, fast advection of solution structures), and the modeler must follow them.
- In others, however, fast time scales are ***parasitic***, and carry very little energy.
  - These are the ones that are usually **targeted for scale-bridging**.
- Bridging the time-scale disparity requires a combination of approaches:
  - Analytical elimination (e.g., reduced models).
  - Well-posed numerical discretization (e.g., asymptotic preserving methods)
  - Some level of implicitness in the temporal formulation (for stability; accuracy requires care).
- Key algorithmic requirement: SCALABILITY

$$\boxed{CPU \sim \mathcal{O}\left(\frac{N}{n_p}\right)}$$

Luis Chacón, chaconl@ornl.gov

Algorithmic scalability vs. parallel scalability

> "The tyranny of scales will not be simply defeated by building bigger and faster computers"
> (NSF SBES 2006 report, p. 30)

- Optimal algorithm: $\boxed{CPU \sim N/n_p}$.

$$\boxed{CPU \sim \frac{N^{1+\alpha}}{n_p^{1-\beta}}} \; ; \; N = \left(\frac{L}{\delta}\right)^d \begin{cases} \alpha \geq 0, \text{ algorithmic scalability} \\ \beta \geq 0, \text{ parallel scalability} \end{cases}$$

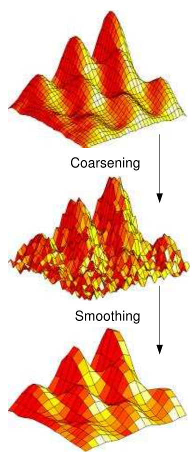
- Much emphasis has been placed on parallel scalability (β).
- However, parallel (**weak**) scalability is limited by the lack of algorithmic scalability:
  - $N \propto n_p \Rightarrow CPU \sim n_p^{\alpha+\beta} \Rightarrow$ **requires** $\alpha = \beta = 0!$

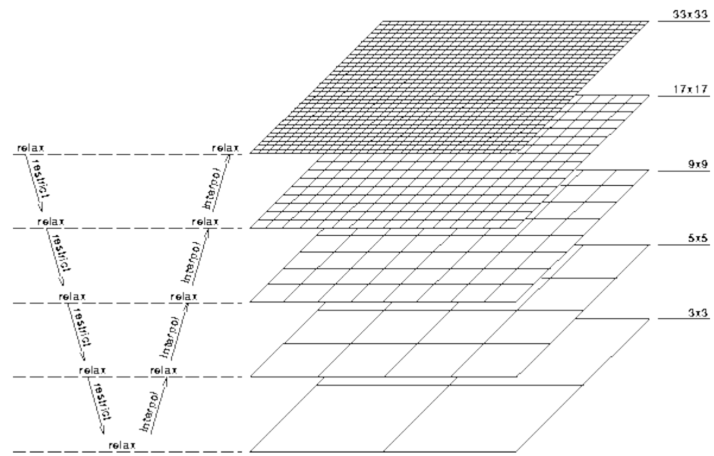
Explicit	Implicit (direct)	Implicit (Krylov iterative)	Implicit (multilevel)
$\alpha = 1/d$	$\alpha = 2 - 2/d$	$\alpha > 1$ (varies)	$\alpha \approx 0$

How do multilevel (multigrid) methods work?

- MG employs a divide-and-conquer approach to attack error components in the solution.
  - Oscillatory components of the error are "EASY" to deal with (if a SMOOTHER exists)
  - Smooth components are DIFFICULT.

> Idea: coarsen grid to make "smooth" components appear oscillatory, and proceed recursively





- SMOOTHER is make or break of MG!
- Smoothers are hard to find for hyperbolic systems, but fairly easy for parabolic ones:

CAN ONE MAKE HYPERBOLIC PDEs MG-FRIENDLY?

Luis Chacón, chaconl@ornl.gov

Implicit discretization of hyperbolic PDEs: a case study

$$\partial_t u = \frac{1}{\epsilon}\partial_x v \quad , \quad \partial_t v = \frac{1}{\epsilon}\partial_x u \quad ; \quad \omega = \pm\frac{k}{\epsilon}$$

- ϵ is a measure of hyperbolic stiffness. Discretize implicitly in time:

$$u^{n+1} = u^n + \frac{1}{\epsilon}\partial_x v^{n+1} \quad , \quad v^{n+1} = v^n + \frac{1}{\epsilon}\partial_x u^{n+1}.$$

Very ill conditioned as $\epsilon \longrightarrow 0$! However, if one combines equations:

$$\boxed{\left[I - \left(\frac{\Delta t}{\epsilon}\right)^2 \partial_x^2\right] u^{n+1} = u^n + \frac{\Delta t}{\epsilon}\partial_x v^n}$$

- Equation is now well-posed when $\epsilon \rightarrow 0$ (i.e., it is **asymptotic-preserving**)!
  - Limit system is elliptic/parabolic (MG-friendly!)
  - *Temporally unresolved* hyperbolic time scales have been "parabolized."
    - No further manipulation of PDE than implicit differencing (no terms added to PDE)!
    - This fact can be exploited to devise optimal solution algorithms (block factorization)!

Block-factorization of hyperbolic PDEs

$$u^{n+1} = u^n + \frac{\Delta t}{\epsilon}\partial_x v^{n+1} \; , \; v^{n+1} = v^n + \frac{\Delta t}{\epsilon}\partial_x u^{n+1}$$

- Coupling structure:

$$\begin{bmatrix} \mathrm{I} & -\frac{\Delta t}{\epsilon}\partial_x \\ -\frac{\Delta t}{\epsilon}\partial_x & \mathrm{I} \end{bmatrix} \begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

- 2×2 block can be formally inverted via block factorization:

$$\begin{bmatrix} D_1 & \frac{1}{\epsilon}U \\ \frac{1}{\epsilon}L & D_2 \end{bmatrix} = \begin{bmatrix} I & \frac{1}{\epsilon}UD_2^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} D_1 - \frac{1}{\epsilon^2}UD_2^{-1}L & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ \frac{1}{\epsilon}D_2^{-1}L & I \end{bmatrix}$$

- Only inverse of $D_1 - UD_2^{-1}L$ (Schur complement) is required!

$$\boxed{D_1 - \frac{1}{\epsilon^2}UD_2^{-1}L = I - \left(\frac{\Delta t}{\epsilon}\right)^2 \partial_x^2}$$

Nonlinear hyperbolic PDEs:
JFNK and block factorization preconditioning

- Objective: solve nonlinear system $\vec{G}(\vec{x}^{n+1}) = \vec{0}$ efficiently (scalably).
- Converge nonlinear couplings using Newton-Raphson method: $\left.\frac{\partial \vec{G}}{\partial \vec{x}}\right|_k \delta \vec{x}_k = -\vec{G}(\vec{x}_k)$.
- Jacobian-free implementation: $\left(\frac{\partial \vec{G}}{\partial \vec{x}}\right)_k \vec{y} = J_k \vec{y} = \lim_{\epsilon \to 0} \frac{\vec{G}(\vec{x}_k + \epsilon \vec{y}) - \vec{G}(\vec{x}_k)}{\epsilon}$
- Krylov method of choice: GMRES (nonsymmetric systems).
- Right preconditioning: solve equivalent Jacobian system for $\delta y = P_k \delta \vec{x}$:

$$J_k P_k^{-1} \underbrace{P_k \delta \vec{x}}_{\delta \vec{y}} = -\vec{G}_k$$

- Approximations in preconditioner do not affect accuracy of converged solution; only efficiency!
- **Block-factorization+MG will be our preconditioning strategy.**

Luis Chacón, chaconl@ornl.gov

Implicit *resistive* MHD solver

L. Chacon, *Phys. Plasmas* (2008)

Resistive MHD model equations

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} &+ \nabla \cdot (\rho \vec{v}) = 0, \\
\frac{\partial \vec{B}}{\partial t} &+ \nabla \times \vec{E} = 0, \\
\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left[\rho \vec{v}\vec{v} - \vec{B}\vec{B} \right. &- \left. \rho \nu \nabla \vec{v} + \overleftrightarrow{I} (p + \frac{B^2}{2}) \right] = 0, \\
\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T &+ (\gamma - 1) T \nabla \cdot \vec{v} = 0,
\end{aligned}
$$

- Plasma is assumed polytropic $p \propto n^{\gamma}$.
- Resistive Ohm's law:

$$\vec{E} = -\vec{v} \times \vec{B} + \eta \nabla \times \vec{B}$$

Resistive MHD Jacobian block structure

- The linearized resistive MHD model has the following couplings:

$$\begin{aligned}
\delta\rho &= L_\rho(\delta\rho, \delta\vec{v}) \\
\delta T &= L_T(\delta T, \delta\vec{v}) \\
\delta\vec{B} &= L_B(\delta\vec{B}, \delta\vec{v}) \\
\delta\vec{v} &= L_v(\delta\vec{v}, \delta\vec{B}, \delta\rho, \delta T)
\end{aligned}$$

- Therefore, the Jacobian of the resistive MHD model has the following coupling structure:

$$J\delta\vec{x} = \begin{bmatrix} D_\rho & 0 & 0 & U_{v\rho} \\ 0 & D_T & 0 & U_{vT} \\ 0 & 0 & D_B & U_{vB} \\ L_{\rho v} & L_{Tv} & L_{Bv} & D_v \end{bmatrix} \begin{pmatrix} \delta\rho \\ \delta T \\ \delta\vec{B} \\ \delta\vec{v} \end{pmatrix}$$

- Diagonal blocks contain advection-diffusion contributions, and are “easy” to invert using MG techniques. Off diagonal blocks L and U contain all hyperbolic couplings.

Luis Chacón, chaconl@ornl.gov

Block factorization of resistive MHD

- We consider the block structure:

$$J\delta\vec{x} = \begin{bmatrix} M & U \\ L & D_v \end{bmatrix} \begin{pmatrix} \delta\vec{y} \\ \delta\vec{v} \end{pmatrix} \;;\; \delta\vec{y} = \begin{pmatrix} \delta\rho \\ \delta T \\ \delta\vec{B} \end{pmatrix} \;;\; M = \begin{pmatrix} D_\rho & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_B \end{pmatrix}$$

- M is "easy" to invert (advection-diffusion, not very stiff, MG-friendly).

Schur complement analysis of 2x2 block J yields:

$$\begin{bmatrix} M & U \\ L & D_v \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -LM^{-1} & I \end{bmatrix} \begin{bmatrix} M^{-1} & 0 \\ 0 & P_{Schur}^{-1} \end{bmatrix} \begin{bmatrix} I & -M^{-1}U \\ 0 & I \end{bmatrix},$$

$$\boxed{P_{Schur} = D_v - LM^{-1}U}.$$

- EXACT Jacobian inverse only requires M^{-1} and P_{Schur}^{-1}.

Luis Chacón, chaconl@ornl.gov

Physics-based preconditioner (PBP)

- 3-step EXACT inversion algorithm:

$$
\begin{aligned}
\text{Predictor} &: \quad \delta\vec{y}^* = -M^{-1}G_y \\
\text{Velocity update} &: \quad \delta\vec{v} = P_{Schur}^{-1}[-G_v - L\delta\vec{y}^*], \quad \boxed{P_{Schur} = D_v - LM^{-1}U} \\
\text{Corrector} &: \quad \delta\vec{y} = \delta\vec{y}^* - M^{-1}U\delta\vec{v}
\end{aligned}
$$

- MG treatment of P_{Schur} is impractical due to M^{-1}.

WE CONSIDER HERE THE SMALL-FLOW LIMIT: $v \ll v_A \Rightarrow$ $M^{-1} \approx \Delta t\,\mathbb{I}$ ("CHEAP")

- We have extended the formulation to arbitrary-flows, $v \sim v_A$ based on commutation ideas[1] (more expensive, but more robust[2]).

---

[1] Elman, *SISC* **27**, 1651 (2006)

[2] L. Chacón, *J. Physics: Conf. Series*, **125**, 012041 (2008)

PBP: Small-flow limit

- Small flow approximation: $M^{-1} \approx \Delta t\, \mathbb{I}$ in steps 2 & 3 of Schur algorithm:

$$
\begin{aligned}
\delta\vec{y}^* &= -M^{-1} G_y \\
\delta\vec{v} &\approx P_{SI}^{-1}\left[-G_v - L\delta\vec{y}^*\right] \; ; \; P_{SI} = D_v - \Delta t LU \\
\delta\vec{y} &\approx \delta\vec{y}^* - \Delta t U \delta\vec{v}
\end{aligned}
$$

where:

$$
\boxed{P_{SI} = \rho^n \left[\mathbb{I}/\Delta t + \theta(\vec{v}_0 \cdot \nabla\mathbb{I} + \mathbb{I} \cdot \nabla\vec{v}_0 - \nu^n \nabla^2 \mathbb{I})\right] + \Delta t \theta^2 W(\vec{B}_0, p_0)}
$$

$$
W(\vec{B}_0, p_0) = \vec{B}_0 \times \nabla \times \nabla \times [\mathbb{I} \times \vec{B}_0] - \vec{j}_0 \times \nabla \times [\mathbb{I} \times \vec{B}_0] - \nabla[\mathbb{I} \cdot \nabla p_0 + \gamma p_0 \nabla \cdot \mathbb{I}]
$$

- Operator $W(\vec{B}_0, p_0)$ is ideal MHD energy operator, which has real eigenvalues!
- P_{SI} is parabolic, and hence block diagonally dominant by construction!
- We employ multigrid methods (MG) to approximately invert P_{SI} and M: 1 V(4,4) cycle

Luis Chacón, chaconl@ornl.gov

PBP: 2D *serial* performance (tearing mode)

Grid convergence study ($\Delta t = 1.0\,\tau_A$)

N	GMRES/Δt	CPU_{exp}/CPU	$\Delta t/\Delta t_{CFL}$
32x32	14	2.43	159
64x64	11.8	5.8	322
128x128	11.2	13.3	667
256x256	11.4	28.5	1429

$CPU \sim \mathcal{O}(N)$ OPTIMAL SCALING!

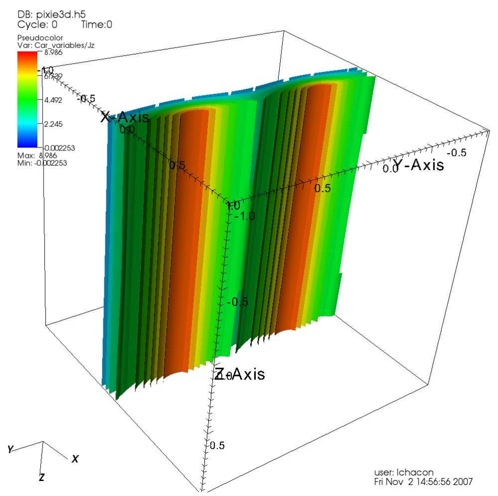
Δt convergence study (128x128)

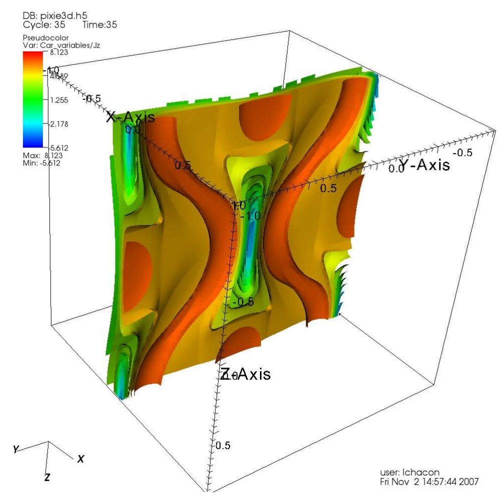
Δt	GMRES/Δt	CPU_{exp}/CPU	$\Delta t/\Delta t_{CFL}$
0.5	8.0	8.0	380
0.75	9.5	10.0	570
1.0	11.2	12.7	760
1.5	14.6	14.6	1140

$CPU \sim \mathcal{O}(\Delta t^{-0.6})$ FAVORABLE SCALING!

Luis Chacón, chaconl@ornl.gov

PBP: 3D *serial* performance (island coalescence)





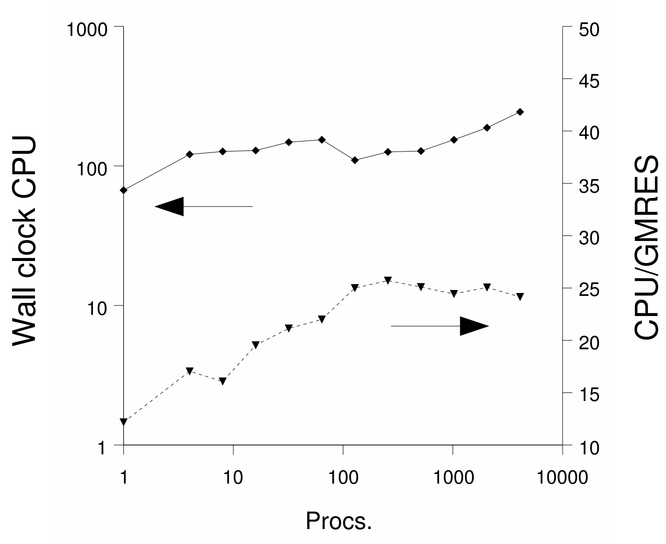
10 time steps, $\Delta t = 0.1$, V(3,3) cycles, mg_tol=1e-2

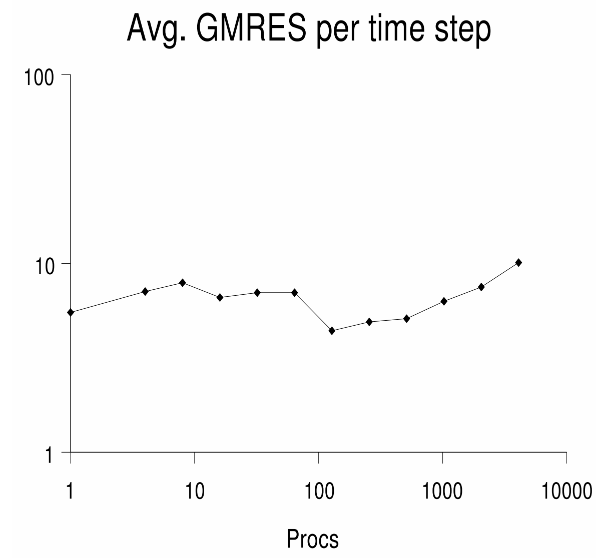
Grid	GMRES/Δt	CPU
16^3	5.5	81
32^3	7.9	1176
64^3	7.0	11135

Luis Chacón, chaconl@ornl.gov

PBP: 3D *parallel* performance (island coalescence)
(Weak scaling, 16^3 points per processor, Cray XT4)

$$\Delta t = 0.1 \gg \Delta t_{CFL}$$





- Key to parallel performance:
  - Matrix-light multigrid, where only diagonals are stored; residuals are calculated matrix-free.
  - Operator coarsening via rediscretization: avoids forming/communicating a matrix.
- Current limitations: we do not feature a coarse-solve beyond the processor skeleton grid.
  - This eventually degrades algorithmic scalability (only shows at $>$ 1000-processor level).

Luis Chacón, chaconl@ornl.gov

Implicit *extended* MHD solver

Extended (two-fluid, Hall) MHD model equations

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &+ \nabla \cdot (\rho \vec{v}) = 0, \\
\frac{\partial \vec{B}}{\partial t} &+ \nabla \times \vec{E} = 0, \\
\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left[\rho \vec{v}\vec{v} - \vec{B}\vec{B} \right. &+ \left. \overleftrightarrow{\Pi} + \overleftrightarrow{I}(p + \frac{B^2}{2}) \right] = 0, \\
\frac{\partial T_e}{\partial t} + \vec{v}^* \cdot \nabla T_e &+ (\gamma - 1) T_e \nabla \cdot \vec{v}^* = (\gamma - 1) \frac{Q - \nabla \cdot \vec{q}}{(1 + \alpha)\rho},
\end{aligned}$$

$$\boxed{\overleftrightarrow{\Pi} = \overleftrightarrow{\Pi}_i + \overleftrightarrow{\Pi}_e \; ; \; \overleftrightarrow{\Pi}_e = -\nu_e \nabla \vec{v}_e \; ; \; \vec{v}_e = \vec{v} - d_i \frac{\vec{j}}{\rho} \; ; \; \vec{v}^* = \vec{v} - \frac{d_i}{1 + \alpha} \frac{\vec{j}}{\rho} \; ; \; \alpha = \frac{T_i}{T_e}}$$

$$\text{Ohm's Law}: \begin{cases} \vec{E} = -\vec{v} \times \vec{B} + \eta \vec{j} + \frac{d_i}{\rho}(\vec{j} \times \vec{B} - \nabla p_e - \nabla \cdot \overleftrightarrow{\Pi}_e) & \text{electron EOM} \\ \vec{E} = -\vec{v} \times \vec{B} + \eta \vec{j} + d_i[\partial_t \vec{v} + \vec{v} \cdot \nabla \vec{v} + \frac{1}{\rho}(\nabla p_i + \nabla \cdot \overleftrightarrow{\Pi}_i)] & \text{ion EOM} \end{cases}$$

Note that EOM_i - EOM_e = EOM. Admits an energy principle.

THIS MODEL SUPPORTS FAST DISPERSIVE WAVES $\omega \sim k^2$.

Luis Chacón, chaconl@ornl.gov

Extended MHD Jacobian block structure: electron EOM (standard choice)

$$\vec{E} = -\vec{v} \times \vec{B} + \eta \vec{j} + \boxed{\frac{d_i}{\rho}(\vec{j} \times \vec{B} - \nabla p_e - \nabla \cdot \overleftrightarrow{\Pi}_e)}$$

- Linearized induction equation $\delta \vec{B} = -\nabla \times \delta \vec{E}$ has the following couplings:

$$\delta \vec{B} = L_B(\delta \vec{B}, \delta \vec{v}, \delta \rho, \delta T)$$

- Jacobian coupling structure:

$$J\delta \vec{x} = \begin{bmatrix} D_\rho & 0 & 0 & U_{v\rho} \\ L_{TB} & D_T & U_{BT} & U_{vT} \\ L_{\rho B} & L_{TB} & \boxed{D_B} & U_{vB} \\ L_{\rho v} & L_{Tv} & L_{Bv} & D_v \end{bmatrix} \begin{pmatrix} \delta \rho \\ \delta T \\ \delta \vec{B} \\ \delta \vec{v} \end{pmatrix}$$

- We have added off-diagonal couplings to block M.
- Stiffest block is D_B $\Rightarrow$ breaks approximations in block-factorization approach. UNSUITABLE!

Extended MHD Jacobian block structure: ion EOM

$$\vec{E} \approx -\vec{v} \times \vec{B} + \eta \vec{j} + \boxed{d_i[\partial_t \vec{v} + \vec{v} \cdot \nabla \vec{v} + \frac{1}{\rho}(\nabla p_i + \nabla \cdot \overleftrightarrow{\Pi}_i [\vec{v}])]}$$

- Hall coupling is mainly via $\partial_t \vec{v}$.
- Jacobian coupling structure becomes:

$$J\delta\vec{x} \approx \begin{bmatrix} D_\rho & 0 & 0 & U_{v\rho} \\ 0 & D_T & 0 & U_{vT} \\ 0 & 0 & D_B & U_{vB}^R + U_{vB}^H \\ L_{\rho v} & L_{Tv} & L_{Bv} & D_v \end{bmatrix} \begin{pmatrix} \delta\rho \\ \delta T \\ \delta\vec{B} \\ \delta\vec{v} \end{pmatrix}$$

We can therefore reuse ALL resistive MHD PC framework!

Extended MHD preconditioner

- Use same block factorization approach.
- M block contains ion time scales only $\Rightarrow$ $M^{-1} \approx \Delta t\, \mathbb{I}$ is a *very* good approximation
- Additional block U_{vB}^{H}:

$$\boxed{P_{SI}\delta\vec{v} = \rho^n \left[\delta\vec{v}/\Delta t + \theta(\vec{v}_0 \cdot \nabla\delta\vec{v} + \delta\vec{v} \cdot \nabla\vec{v}_0 + \nabla \cdot \delta\overleftrightarrow{\Pi})\right] + \Delta t\theta^2 W(\vec{B}_0, p_0)\delta\vec{v}}$$

$$W(\vec{B}_0, p_0) = \vec{B}_0 \times \nabla \times \nabla \times [\mathbb{I} \times \vec{B}_0 - \frac{d_i}{\theta\Delta t}\mathbb{I}\ ] - \vec{j}_0 \times \nabla \times [\mathbb{I} \times \vec{B}_0] - \nabla[\mathbb{I} \cdot \nabla p_0 + \gamma p_0 \nabla \cdot \mathbb{I}]$$

- Additional term brings in dispersive waves $\omega \sim k^2$!
- We can show analytically that additional term (yellow) is amenable to simple damped Jacobi smoothing!

We can use classical MG!

On the issue of dissipation in extended MHD

- Dispersive waves $\omega \sim k^2$ require higher order dissipation.
- Resistivity is unable to provide a dissipation scale.
- Dissipation scale defined by electron viscosity, $\nabla \cdot \delta\overleftrightarrow{\Pi}_e$:

$$\nabla \cdot \delta\overleftrightarrow{\Pi}_e \sim -\nu_e \nabla^2 (\nabla \times \nabla \times \delta\vec{v}) \sim \nu_e \nabla^4 \delta\vec{v}$$

- Viscosity coefficient can be determined to provide adequate dissipation of dispersive waves

$$\omega \sim v_A d_i k^2 : \nu_e k^4 \;\Rightarrow\; \boxed{\nu_e > C \frac{d_i v_A k_{\parallel,max}}{k_{max}^3}}$$

- In the preconditioner, we deal with $\nabla \cdot \delta\overleftrightarrow{\Pi}_e$ by considering 2 second-order systems, and solving them coupled within MG.

Luis Chacón, chaconl@ornl.gov

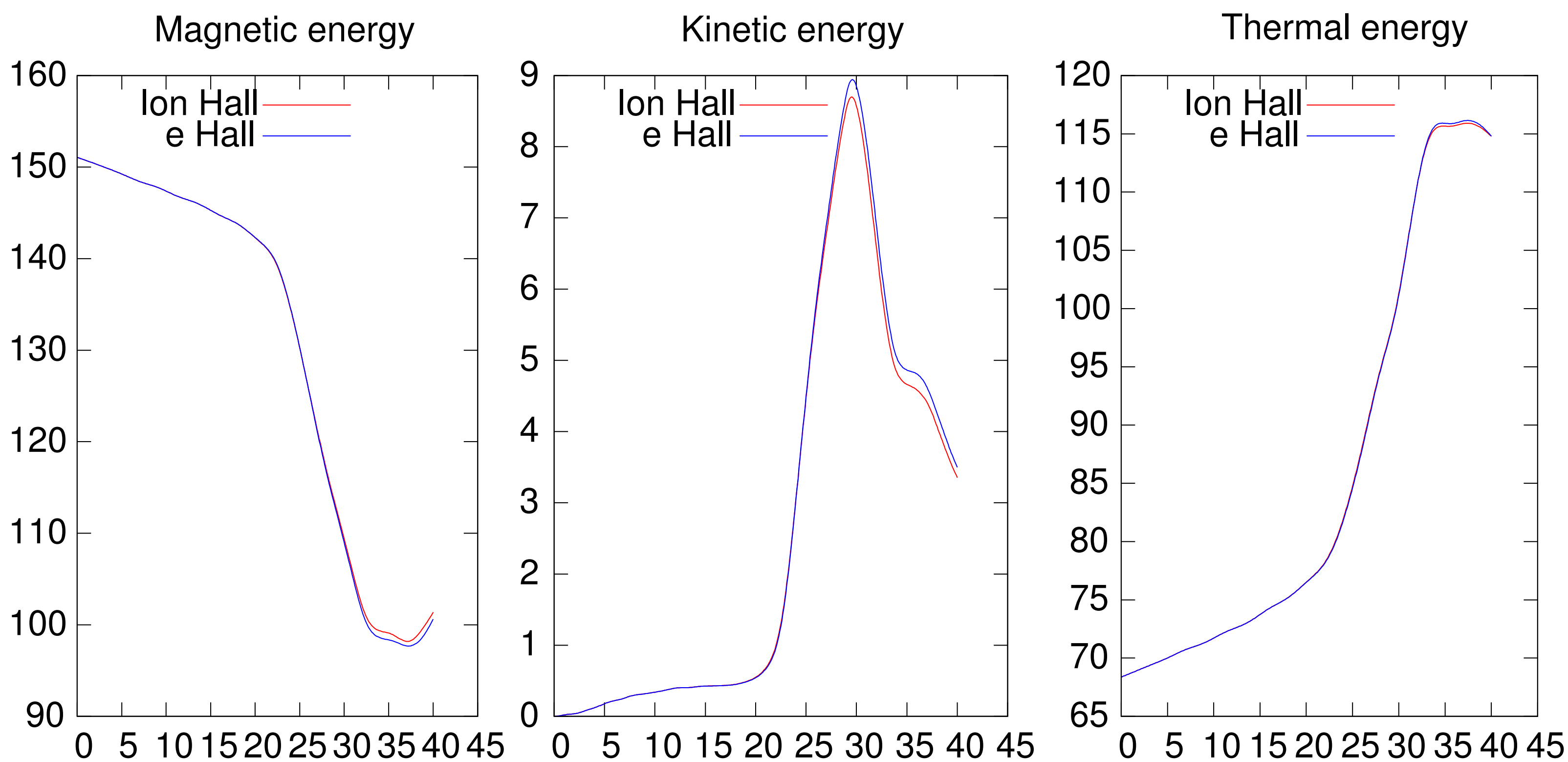
Extended MHD performance results (2D tearing mode)

$d_i = 0.05, \nu_e = 2.5 \times 10^{-6}$

100 time steps, $\Delta t = 1.0$, 1 V(4,4) MG cycle

Grid	GMRES/Δt	CPU_{exp}/CPU	$\Delta t/\Delta t_{exp}$	$\Delta t/\Delta t_{CFL}$
32x32	22.3	0.74	135	110
64x64	15.4	10.9	1582	384
128x128	10.6	214	23809	1436
256x256	13.1	3097	370370	5660

2D nonlinear verification: GEM challenge
Ion Hall vs. electron Hall



Luis Chacón, chaconl@ornl.gov

Incompressible Navier-Stokes solver

Cyr, Shadid, Tuminaro, JCP 2011

Elman, Howle, Shadid, Shuttleworth, Tuminaro, JCP 2008

Block preconditioning: CFD example

Consider discretized Navier-Stokes equations

$$\left.\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\nabla^2\mathbf{u} + \nabla p = f \\ \nabla\cdot\mathbf{u} = 0\end{aligned}\right\} \Leftrightarrow \begin{bmatrix} F & B^T \\ B & C \end{bmatrix}\begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

Fully Coupled Jacobian

$$\mathcal{A} = \begin{bmatrix} F & B^T \\ B & C \end{bmatrix}$$

Block Factorization

$$\mathcal{A} = \begin{bmatrix} I & \\ BF^{-1} & I \end{bmatrix}\begin{bmatrix} F & B^T \\ & S \end{bmatrix}$$

$$S = C - BF^{-1}B^T$$

- Coupling in Schur-complement

Preconditioner

$$\mathcal{A}^{-1} \approx \mathcal{M}^{-1} = \begin{bmatrix} \hat{F} & B^T \\ & \hat{S} \end{bmatrix}^{-1}$$

Required operators:

- $F^{-1} \approx \hat{F}^{-1} \rightarrow$ Multigrid
- $S^{-1} \approx \hat{S}^{-1} \rightarrow$ PCD, LSC, SIMPLEC

Properties of block factorization

1. Important coupling in Schur-complement
2. Better targets for AMG $\rightarrow$ leveraging scalability

Properties of approximate Schur-complement

1. "Nearly" replicates physical coupling
2. Invertible operators $\rightarrow$ good for AMG

Brief Overview of Block Preconditioning Methods for Navier-Stokes:
(A Taxonomy based on Approximate Block Factorizations, JCP – 2008)

Discrete N-S	Exact LDU Factorization	Approx. LDU
$\begin{pmatrix} F & B^T \\ \hat{B} & -C \end{pmatrix} \begin{pmatrix} \Delta \mathbf{u}_k \\ \Delta p_k \end{pmatrix} = \begin{pmatrix} \mathbf{g}_\mathbf{u}^k \\ g_p^k \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ \hat{B}F^{-1} & I \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -S \end{pmatrix} \begin{pmatrix} I & F^{-1}B^T \\ 0 & I \end{pmatrix}$ $S = C + \hat{B}F^{-1}B^T$	$\begin{bmatrix} I & 0 \\ \hat{B}H_1 & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & -\hat{S} \end{bmatrix} \begin{bmatrix} I & H_2 B^T \\ 0 & I \end{bmatrix}$

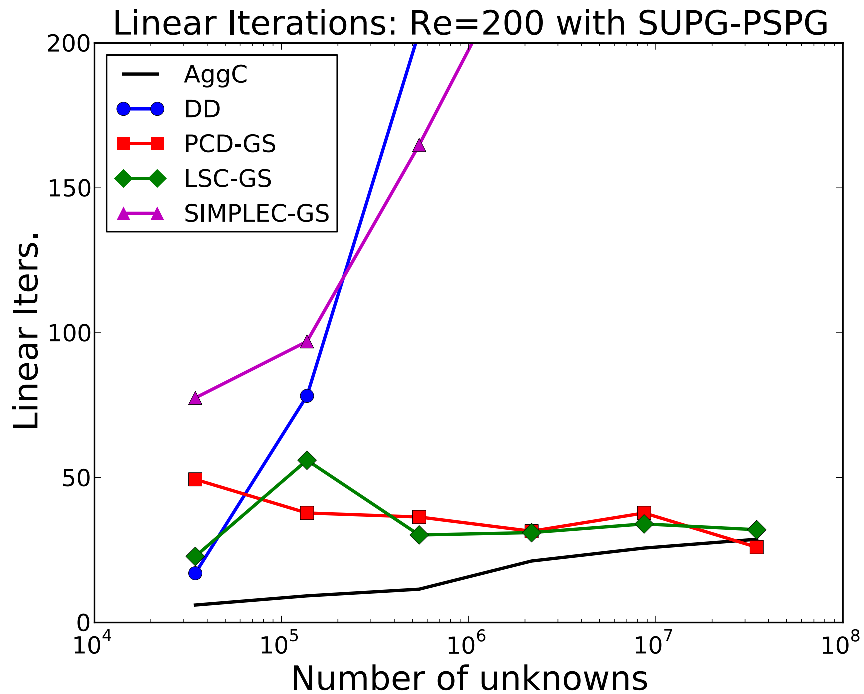
Precond. Type	H_1	H_2	$\hat{S}$	References
Pres. Proj; 1st Term Neumann Series	$\mathbf{F}^{-1}$	$(\Delta t \mathbf{I})^{-1}$	$\mathbf{C} + \Delta t \hat{\mathbf{B}} \mathbf{B}^\mathbf{T}$	Chorin(1967);Temam (1969); Perot (1993): Quateroni et. al. (2000) as solvers
SIMPLEC	$\mathbf{F}^{-1}$	$(\mathrm{diag}(\sum \lvert\mathbf{F}\rvert))^{-1}$	$\mathbf{C} + \hat{\mathbf{B}}(\mathrm{diag}(\sum \lvert\mathbf{F}\rvert))^{-1}\mathbf{B}^\mathbf{T}$	Patankar et. al. (1980) as solvers; Pernice and Tocci (2001) as smoothers/MG
Pressure Convection / Diffusion	0	$\mathbf{F}^{-1}$	$\mathbf{A_p F_p^{-1}}$	Kay, Loghin, Wathan, Silvester, Elman (1999 - 2006); Elman, Howle, S., Shuttleworth, Tuminaro (2003,2008)

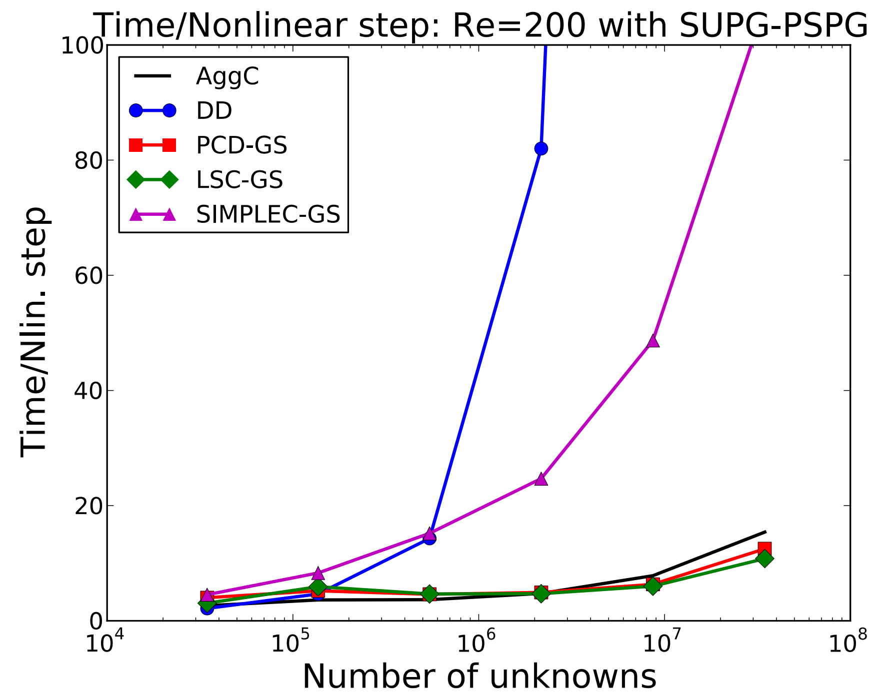
Now use AMG type methods on sub-problems.

Momentum transient convection-diffusion: $F\Delta\mathbf{u} = \mathbf{r_u}$

Pressure – Poisson type: $-\hat{S}\Delta p = \mathbf{r}_p$

CFD Weak Scaling: Steady Backward Facing Step





<u>Fully coupled Algebraic</u>

AggC: Aggressive Coarsening Multigrid

DD: Additive Schwarz Domain Decomposition

<u>Block Preconditioners</u>

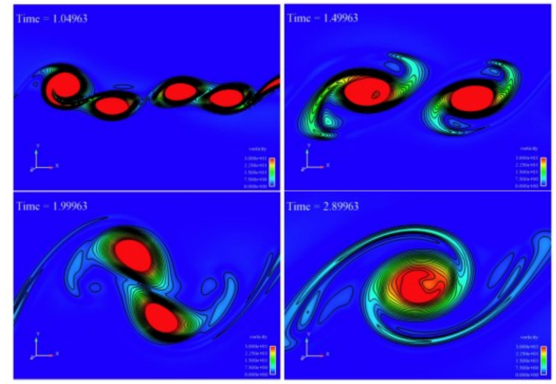
PCD & LSC: Commuting Schur complement

SIMPLEC: “Physics-based” Schur complement

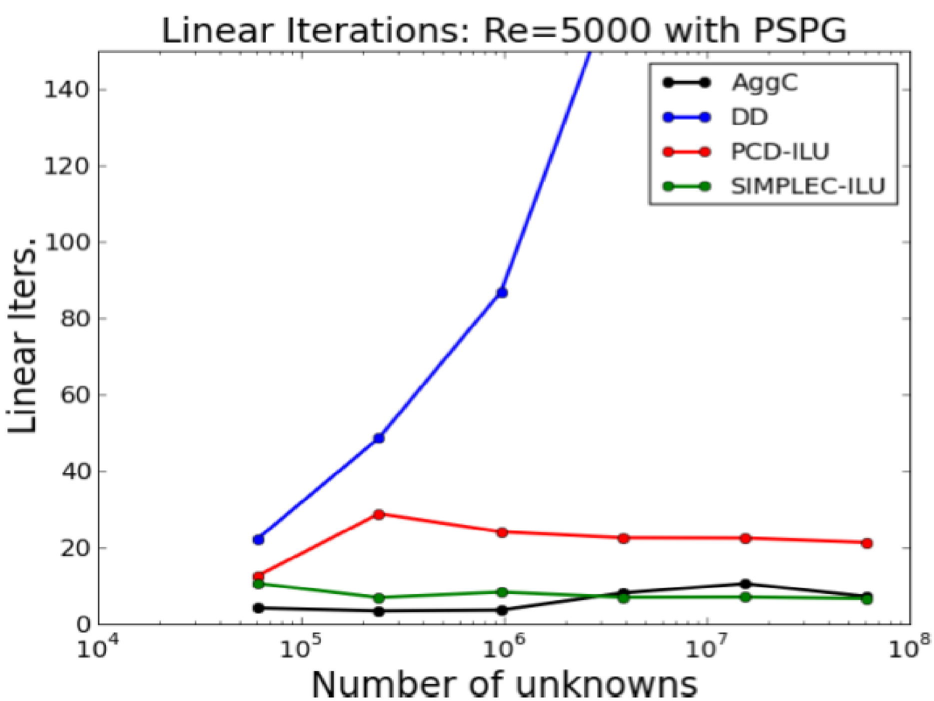
Take home: Block preconditioners competitive with fully coupled multigrid for CFD

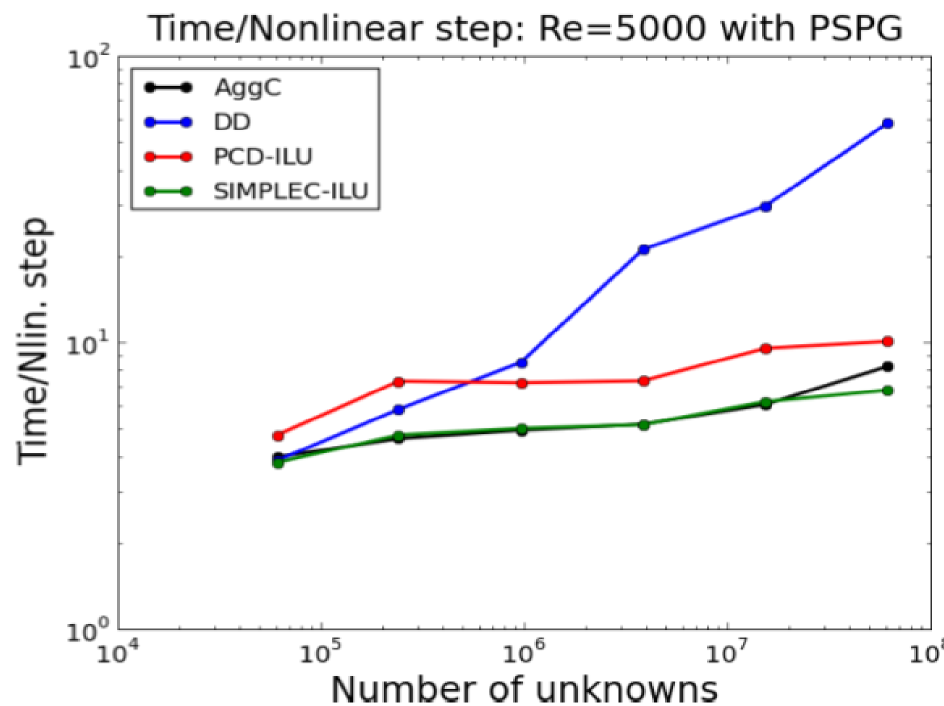
*** Paper accepted: E. C. Cyr, J. N. Shadid, R. S. Tuminaro, Stabilization and Scalable Block Preconditioning for the Navier-Stokes Equations, Accepted by J. Comp. Phys., 2011.**

Weak Scaling of NK Solver with Fully-coupled AMG and Approx. Block Factorization Preconditioners



Transient Kelvin-Helmholtz instability (Re = 5 x 10^3 shear layer, constant CFL = 2.5)





Quad-core Nehalems with Infini-band SNL Red Sky

Incompressible MHD solver

Incompressible MHD: 2D Vector Potential Formulation

Magnetohydrodynamics (MHD) equations couple fluid flow to Maxwell's equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p + \nabla \cdot \left(-\frac{1}{\mu_0} \mathbf{B} \otimes \mathbf{B} + \frac{1}{2\mu_0} \|\mathbf{B}\|^2 \mathbf{I} \right) = f$$

$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial A_z}{\partial t} + \mathbf{u} \cdot \nabla A_z - \frac{\eta}{\mu_0} \nabla^2 A_z = -E_z^0$$

$$\text{where} \;\; \mathbf{B} = \nabla \times \mathbf{A}, \;\; \mathbf{A} = (0, 0, A_z)$$

Discretized using a stabilized finite element formulation

Block LU Factorization

$$\begin{bmatrix} F & B^T & Z \\ B & C & 0 \\ Y & 0 & D \end{bmatrix} = \begin{bmatrix} I & & \\ BF^{-1} & I & \\ YF^{-1} & -YF^{-1}B^TS^{-1} & I \end{bmatrix} \begin{bmatrix} F & B^T & Z \\ & S & -BF^{-1}Z \\ & & P \end{bmatrix}$$

where

$$S = C - BF^{-1}B^T$$
$$P = D - YF^{-1}(I + B^TS^{-1}BF^{-1})Z$$

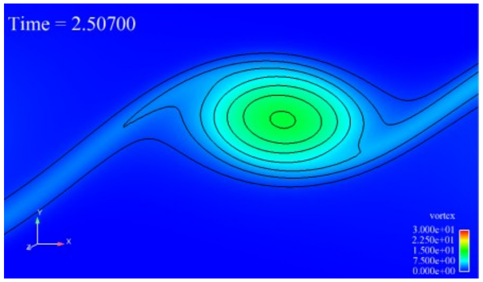
- ***C*** **is zero for mixed interpolation FE and staggered FV methods, nonzero for stabilized FE**
- **Indefinite system – hard to solve with incomplete factorizations without pivoting**
- **Block factorization of 3x3 system leads to nested Schur complements**

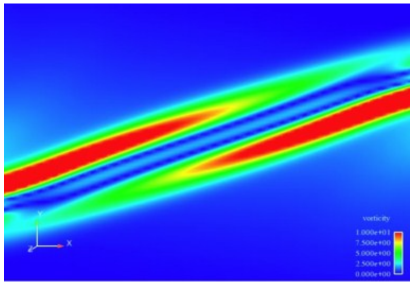
- **Use an operator splitting approximation to factor**

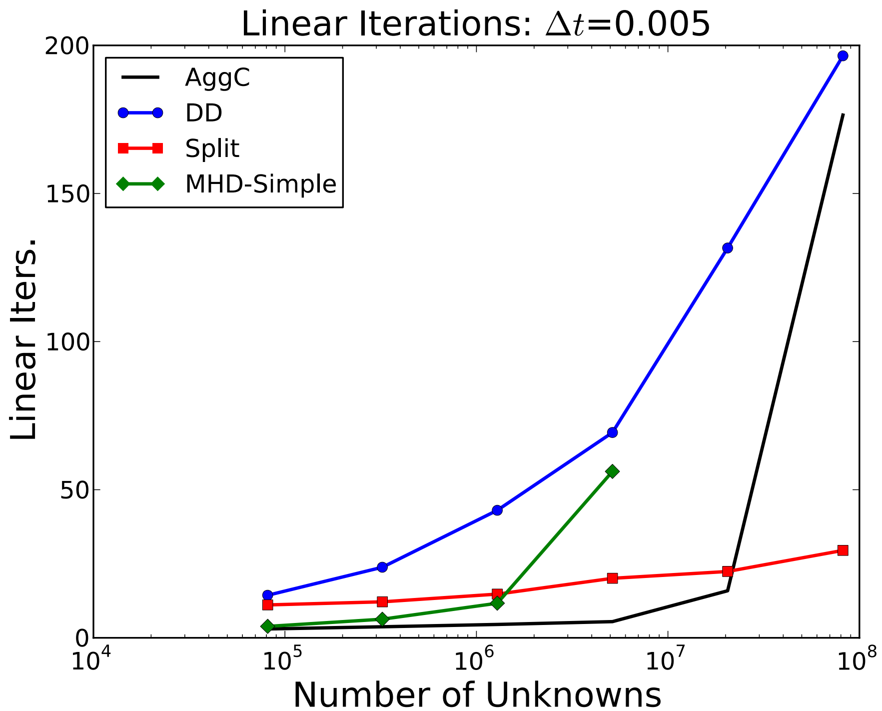
$$\begin{bmatrix} F & B^T & Z \\ B & C & 0 \\ Y & 0 & D \end{bmatrix} \approx \begin{bmatrix} F & & Z \\ & I & \\ Y & & D \end{bmatrix} \begin{bmatrix} F^{-1} & & \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} F & B^T & \\ B & C & \\ & & I \end{bmatrix} = \begin{bmatrix} F & B^T & Z \\ B & C & \\ Y & \boxed{YF^{-1}B^T} & D \end{bmatrix}$$

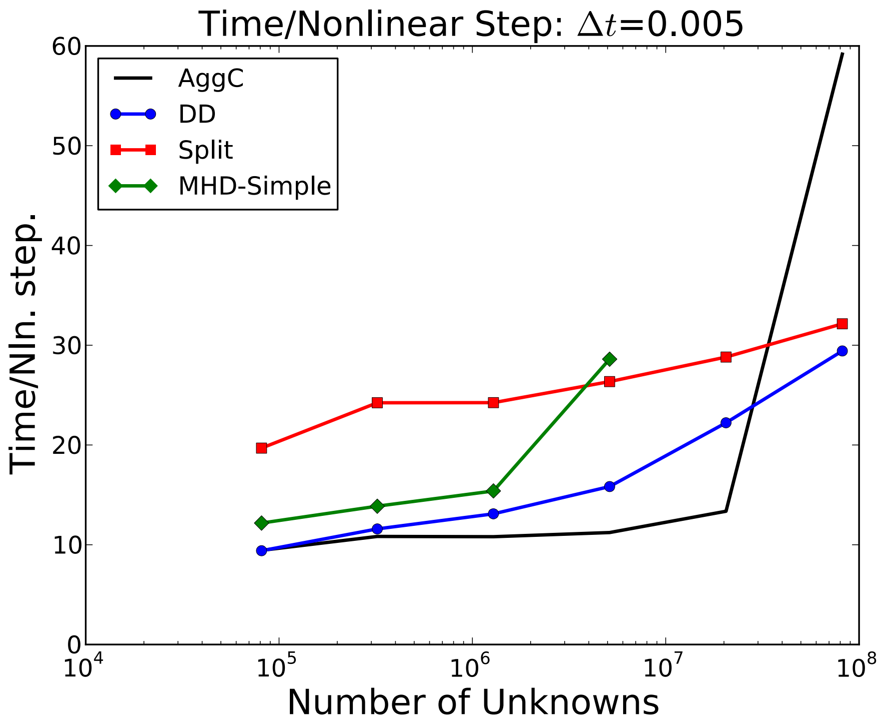
  - **Reduces to 2 – 2x2 systems for Navier-Stokes and magnetics-velocity blocks;**
  - **C need not be non-zero or invertible (*C*$^{-1}$ doesn't need to exist!)**

Transient Hydro-Magnetic Kelvin-Helmholtz Problem
(Re = 700, S = 700)









<u>Fully coupled Algebraic</u>

AggC: Aggressive Coarsening Multigrid
DD: Additive Schwarz Domain Decomposition

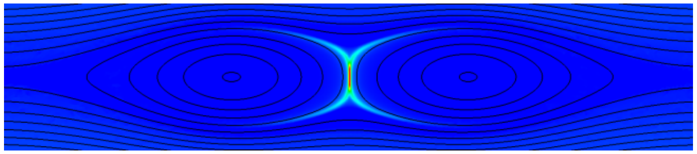
<u>Block Preconditioners</u>

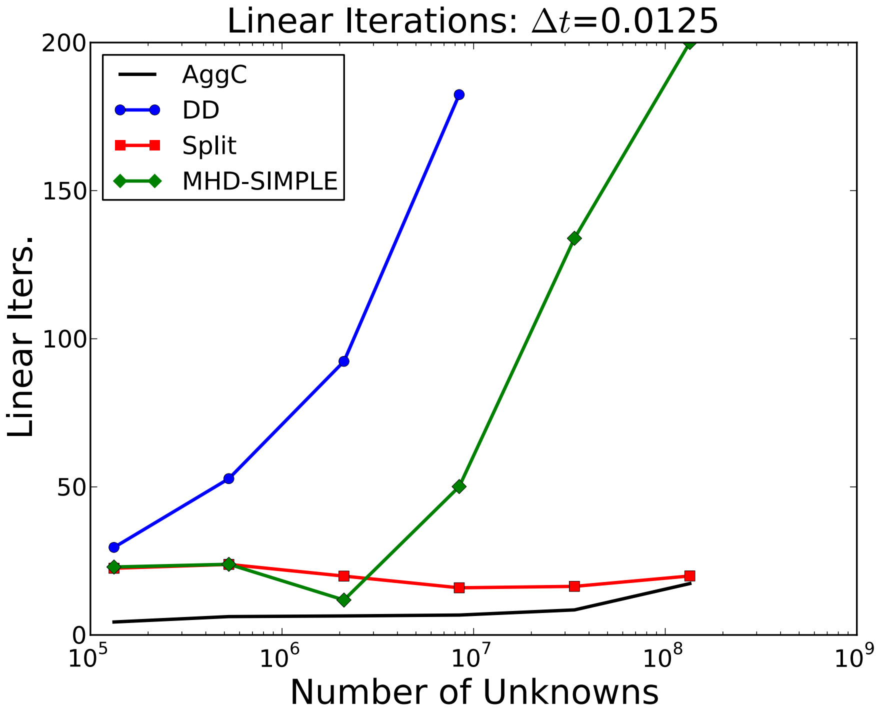
<u>Split</u>: New Operator split preconditioner
<u>SIMPLEC</u>: Extreme diagonal approximations

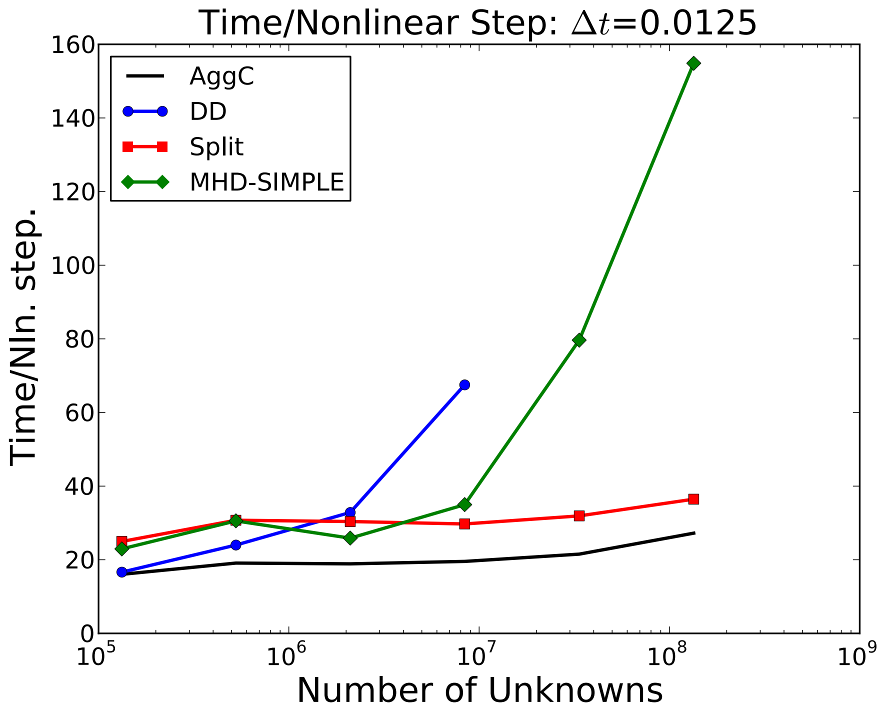
Take home: **AggC and Split preconditioner scale algorithmically**

1. **SIMPLE preconditioner performance suffers with increased CFL**
2. **Run times are for unoptimized code**
3. **AggC not applicable to mixed discretizations, block factorization is**

Driven Magnetic Reconnection: Magnetic Island Coalescence Half domain symmetry on [0,1]x[-1,1] with S = 10e+4







Fully coupled Algebraic

AggC: Aggressive Coarsening Multigrid

DD: Additive Schwarz Domain Decomposition

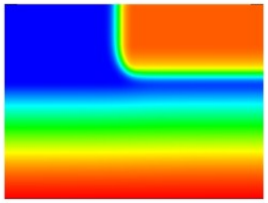
Block Preconditioners

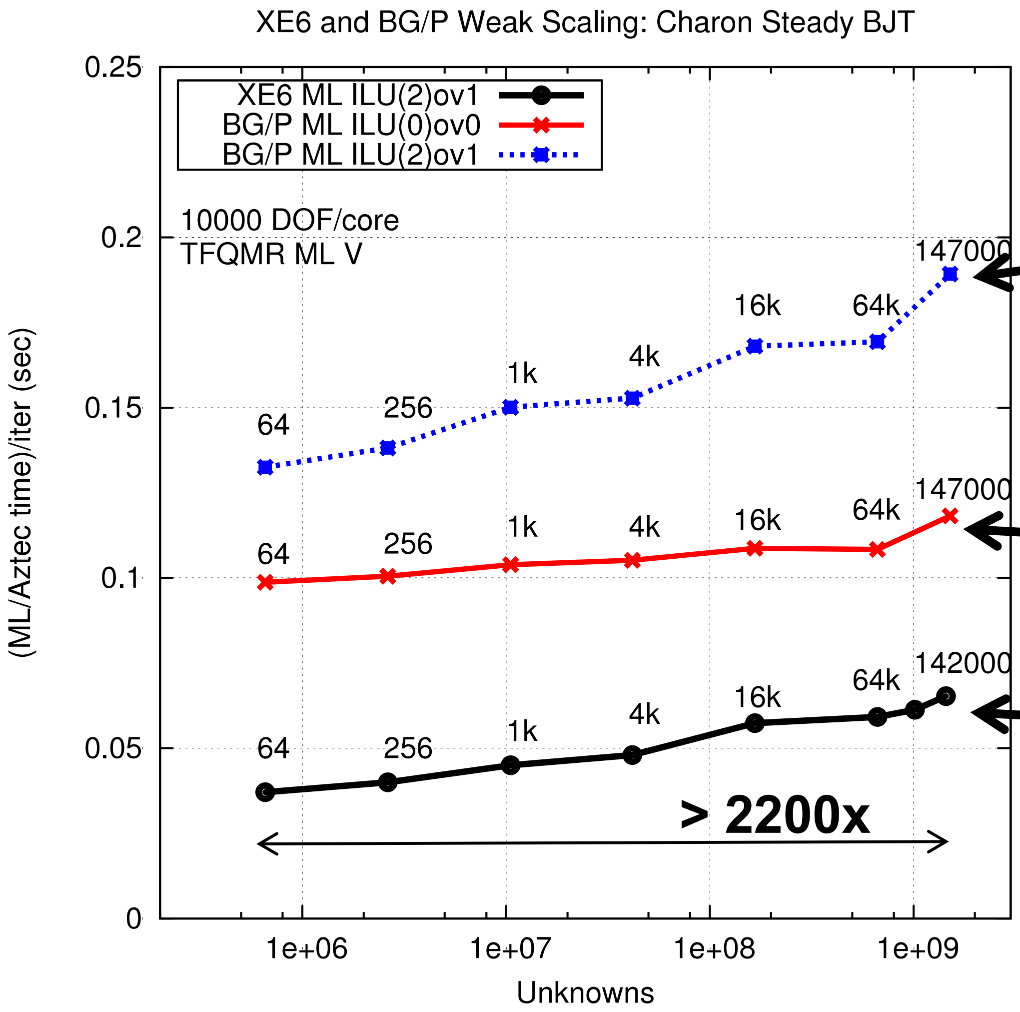
Split: New Operator split preconditioner

SIMPLEC: Extreme diagonal approximations

Take home: Split preconditioner scales algorithmically

Initial Weak Scaling Performance of AMG V-cycle on Leadership Class Machines
Cray XE6 and BG/P Weak Scaling
(Transport-reaction: Drift-diffusion simulations)





Sub-domain smoothers: Impact of data locality of smoother?

BG/P: ILU(2); overlap = 1

BG/P: ILU(0); overlap = 0
[Better scaling and faster time to solution than ILU(2),ov=1]

Cray XE6: ILU(2); overlap = 1

- **Steady-state drift-diffusion BJT**
- **TFQMR time per iteration**
- **Cray XE6 2.4GHz 8-core Magny-Cours**

(Paul Lin)

Summary and Conclusions

- Stiff hyperbolic PDEs describe many applications of interest to DOE.
- In applications where fast time scales are parasitic, an implicit treatment is possible to bridge time-scale disparity.
- A fully implicit solution may only realize its efficiency potential if a suitable scalable algorithmic route is available.
- Here, we have identified stiff-wave block-preconditioning (aka physics-based preconditioning) in the context of JFNK methods as a suitable algorithmic pathway.
  - An important property is that it renders the numerical system suitable for multilevel preconditioning.
- We have demonstrated the effectiveness of the approach in incompressible Navier-Stokes, incompressible MHD, and compressible resistive MHD and extended MHD.
  - In all these applications, the approach is robust and scalable, both algorithmically and in parallel.