SAND2013-1798C

Solution Techniques for Large-scale Fully-Implicit Multi-Physics Systems Using Trilinos

Roger Pawlowski, Andy Salinger , Eric Phipps,
Eric Cyr, John Shadid,and Tom Smith
Sandia National Laboratories

Stuart Slattery and Paul Wilson
University of Wisconsin

Roscoe Bartlett, Kevin Clarno, Steve Hamilton,
Kate Evans and Tom Evans
Oak Ridge National Laboratory

European Trilinos User Group Meeting 2015
Tuesday March 3, 2015

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Mathematical Motivation

Achieving Scalable Predictive Simulations of Complex Highly Nonlinear Multi-physics PDE Systems

- Multiphysics systems are characterized by a myriad of complex, interacting, nonlinear multiple time- and length-scale physical mechanisms:
  - Dominated by **short dynamical time-scales** → **Explicit Methods**
  - Widely separated time-scales **(stiff system)**
  - Evolve a solution on a **long time scale relative to component time scales**
  - Balance to produce **steady-state** behavior.

  (Last three items) → **Typically requires some form of Implicit Methods**

e.g. Nuclear Fission / Fusion Reactors; Conventional /Alternate Energy Systems; High Energy Density Physics; Astrophysics; etc ….

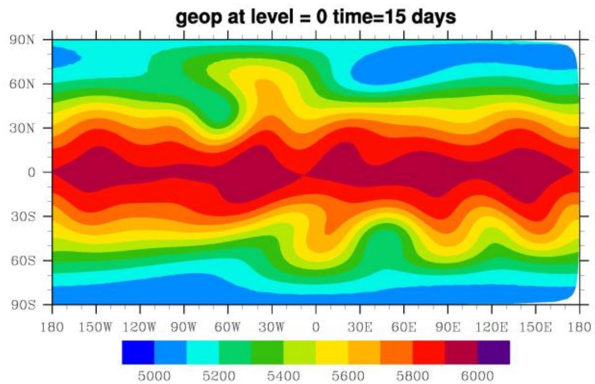
- Our approach:
  - Stable and higher-order accurate implicit formulations and discretizations
  - Robust, scalable and efficient prec. for fully-coupled Newton-Krylov methods
  - Integrate sensitivity and error-estimation to enable UQ capabilities.

Tools for Multiphysics Simulation
(Spanning Individual Applications and Coupled Systems)

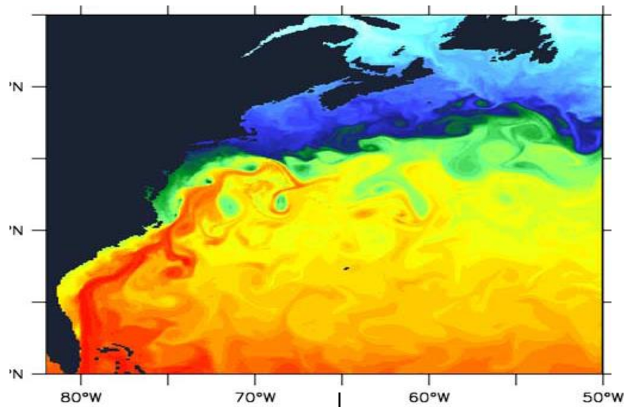
- **Domain Model (SAND2011-2195)**
- **Abstraction Layer ANAs**
  - Thyra::ModelEvaluator: Application Interface
  - Thyra: Operator, Vector
- **Implicit Nonlinear Solution Algorithms**
  - **NOX**: Globalized Newton-Krylov and JFNK
  - **LIME/PIKE**: Multiphysics coupling driver. Picard iteration and tools to assemble block aggregate systems to call with NOX
- **Linear Algebra and Linear Solution Algorithms**
  - Epetra, Tpetra: Concrete Linear Algebra
  - Stratimikos, Belos, AztecOO, Amesos: Linear Solvers
  - ML, MueLE, Ifpack,Teko: Preconditioners
- Examples

Implicit Climate Simulators can be Built on Trilinos Solvers and Software

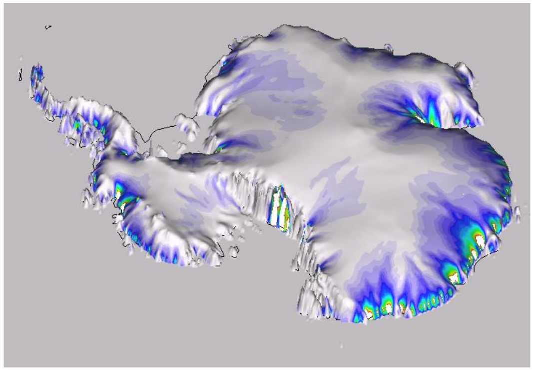
HOMME Atmospheric Model



POP Ocean Model
THCM Ocean Model



Glimmer Ice Sheet Model
FELIX Ice Sheet Model



+ IBECS, Sea Ice

Trilinos

Parallelization Tools
- Data Structures
- Partitioning
- Load Balancing
- Architecture-Dependent Kernels

Linear Solvers
- Iterative Solvers
- Direct Solvers
- Eigen Solver
- Preconditioners
- Multi-Level Algs

Analysis Tools
- Nonlinear Solver
- Time Integration
- Stability Analysis
- Optimization
- UQ Algorithms

Software Quality
- Version Control
- Regression Testing
- Build System
- Verification Tests
- Bug Tracking
- Web Documentation
- Release Process

Sandia National Laboratories

A Domain Model

<u>A Theory Manual for Multiphysics Code Coupling in LIME</u>, R. Pawlowski, R. Bartlett, R. Schmidt, R. Hooper, and N. Belcourt, SAND2011-2195

$$f(\dot{x}, x, \{p_l\}, t) = 0$$

Residual — f

$$\dot{x} = \frac{\partial x}{\partial t}$$

State (DOF) — x

Set of parameters — $\{p_l\}$

Time — t

$x \in \mathbb{R}^{n_x}$ is the vector of state variables (unknowns being solved for),
$\dot{x} = \partial x / \partial t \in \mathbb{R}^{n_x}$ is the vector of derivatives of the state variables with respect to time,
$\{p_l\} = \{p_0, p_1, \ldots, p_{N_p-1}\}$ is the set of N_p independent parameter sub-vectors,
$t \in [t_0, t_f] \in \mathbb{R}^1$ is the time ranging from initial time t_0 to final time t_f,

$$f(\dot{x}, x, \{p_l\}, t) : \mathbb{R}^{\left(2n_x + \left(\sum_{l=0}^{N_p-1} n_{p_l}\right) + 1\right)} \rightarrow \mathbb{R}^{n_x}$$

$$g_j(\dot{x}, x, \{p_l\}, t) = 0, \text{ for } j = 0, \ldots, N_g - 1$$

Response Function — g_j

$g_j(\dot{x}, x, \{p_l\}, t) : \mathbb{R}^{\left(2n_x + \left(\sum_{l=0}^{N_p-1} n_{p_l}\right) + 1\right)} \rightarrow \mathbb{R}^{n_{g_j}}$ is the j^{th} response function.

- Input Arguments: state time derivative, state, parameters, time
- Output Arguments: Residual, Jacobian, response functions, etc…

Sandia National Laboratories

Extension to Multiphysics

Split parameters into “coupling” and truly independent.

$$f_i(\dot{x}_i, x_i, \{z_{i,k}\}, \{p_{i,l}\}, t) = 0$$

Set of *coupling* parameters

Set of *independent* parameters

Require transfer functions:

- Can be complex nonlinear functions themselves

$$z_{i,k} = r_{i,k}(\{x_m\}, \{p_{m,n}\})$$

Transfer Function

Response functions now dependent on z

- Can be used as coupling parameters (z) for other codes

$$g_{i,j}(\dot{x}_i, x_i, \{z_{i,k}\}, \{p_{i,l}\}, t)$$

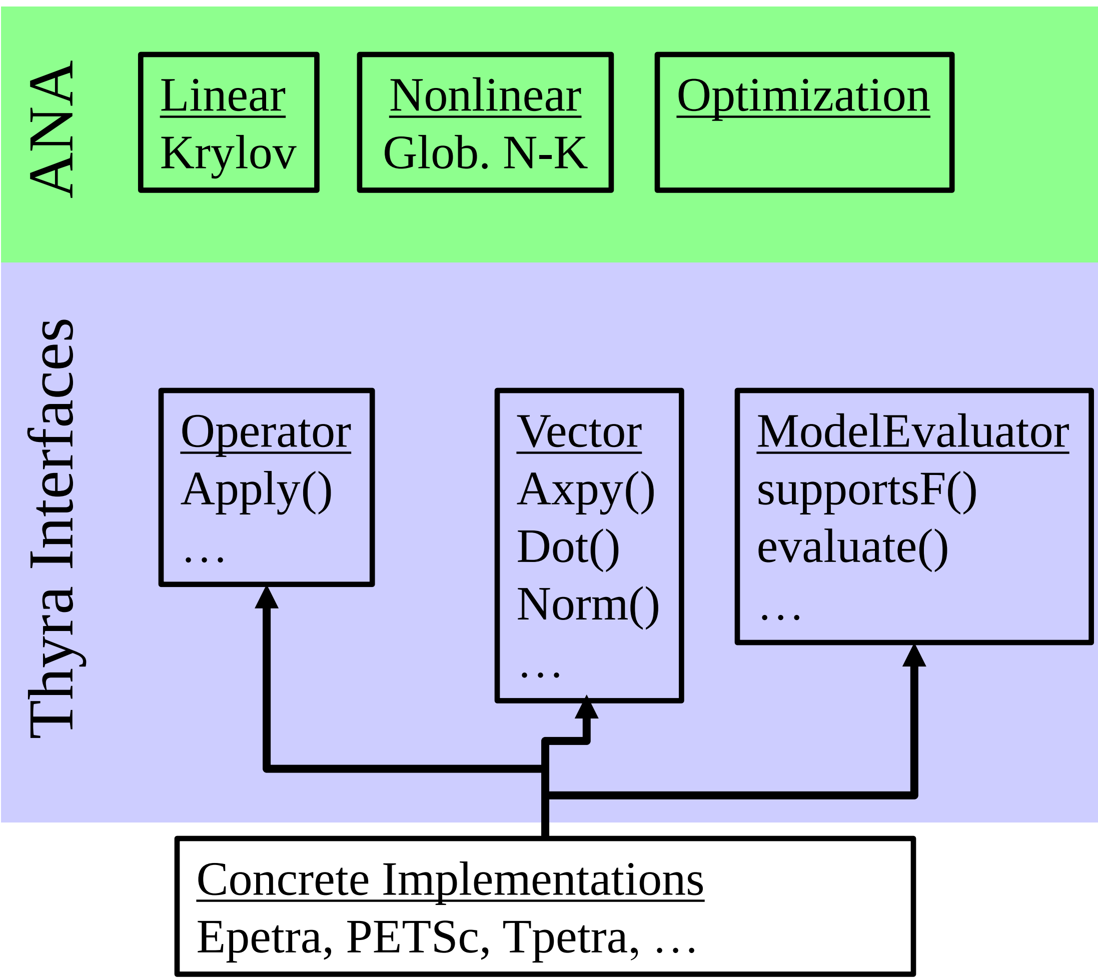
Response Function

Abstract Interfaces

Introducing Abstract Numerical Algorithms

What is an abstract numerical algorithm (ANA)?

An ANA is a numerical algorithm that can be expressed abstractly solely in terms of vectors, vector spaces, linear operators, and other abstractions built on top of these ***without general direct data access or any general assumptions about data locality***



Block composition operators and vectors:

$$\begin{bmatrix} \mathbf{J}_{TT} & \mathbf{J}_{T\mathbf{W}} \\ \mathbf{J}_{\mathbf{W}T} & \mathbf{J}_{\mathbf{WW}} \end{bmatrix} \begin{bmatrix} \Delta T \\ \Delta \mathbf{W} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_T \\ \mathbf{F}_{\mathbf{W}} \end{bmatrix}$$

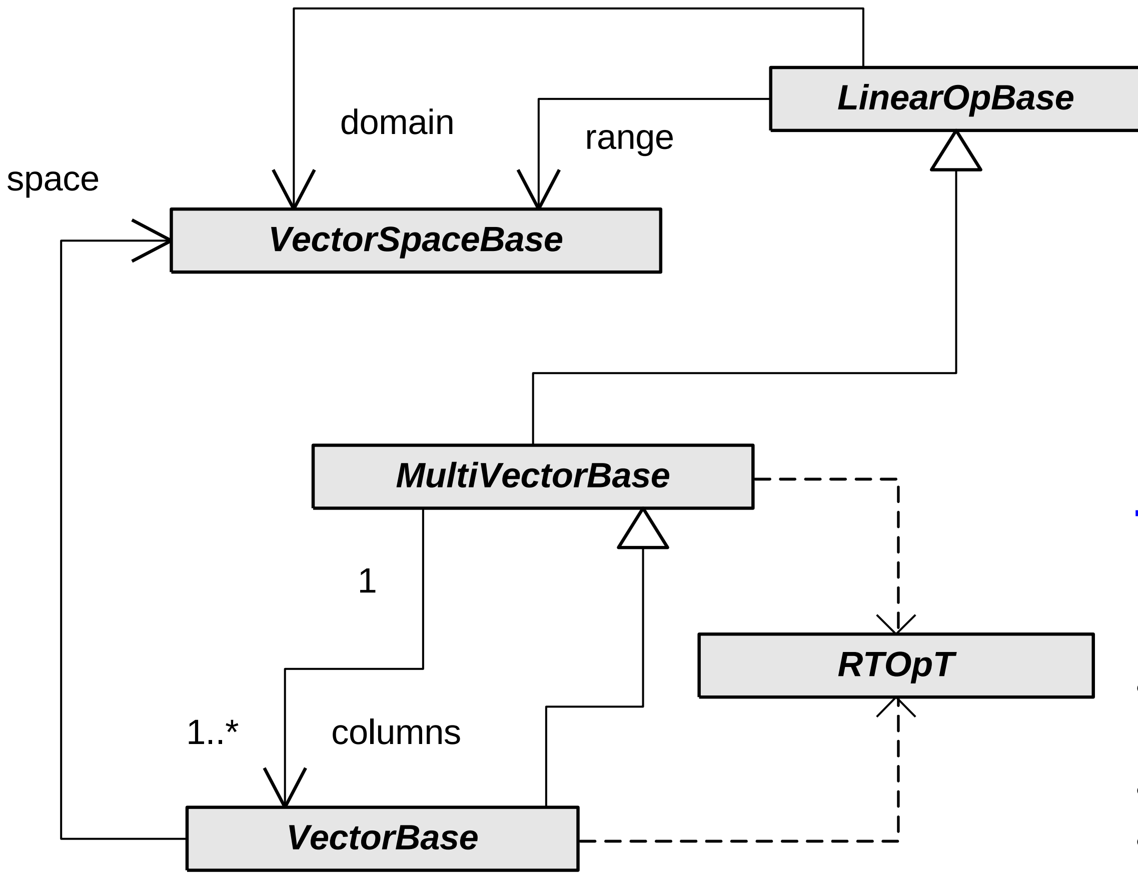
Block Factorization Preconditioners:

$$\begin{bmatrix} I & 0 \\ \hat{B}H_1 & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & -\hat{S} \end{bmatrix} \begin{bmatrix} I & H_2B^T \\ 0 & I \end{bmatrix}$$

$$S = c + \hat{B}F^{-1}B^T$$

Sandia National Laboratories

Fundamental Thyra ANA Operator/Vector Interfaces



A Few Quick Facts about Thyra Interfaces

- All interfaces are expressed as abstract C++ base classes (i.e. object-oriented)
- All interfaces are templated on a Scalar data type (i.e. generic)

The Key to success!

Reduction/Transformation Operators

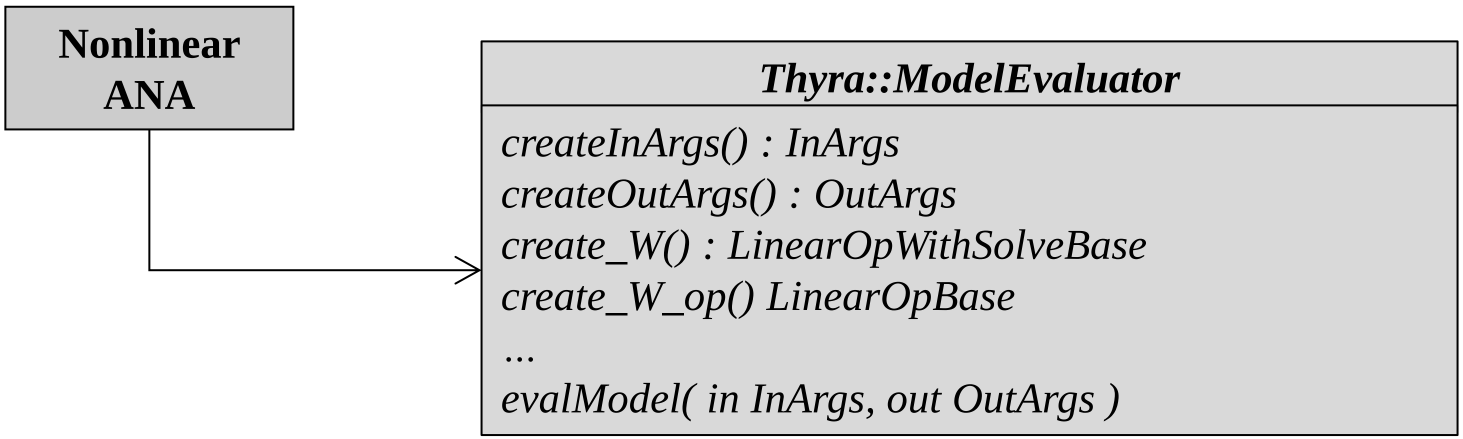
- Supports all needed element-wise vector operations
- Data/parallel independence
- Optimal performance

Matrix/Vector operations are handled in app's native data structures!

R. A. Bartlett, B. G. van Bloemen Waanders and M. A. Heroux. *Vector Reduction/Transformation Operators*, ACM TOMS, March 2004

Sandia National Laboratories

Application Interface: Model Evaluator



- **Set your inputs in an InArgs container:** $\dot{x}, x, p, t$
- **Set your outputs in an OutArgs container:** $f,\ W,\ M,\ g,\ \frac{\partial f}{\partial p},\ \frac{\partial g}{\partial x},\ \frac{\partial g}{\partial p}$
- **model_evaluator->evalModel(in_args, out_args)**

- **Common interface for ANAs: Nonlinear, Optimization, Bifurcation, …**
- **Inputs and outputs are extensible without requiring changes to apps**
- **Efficient shared calculations (e.g. automatic differentiation)**
- **Self describing: query what inputs and outputs it supports**

Application Classification

Inputs and outputs are ***optionally*** supported by physics model → restricts allowed solution procedures

Name	Definition	Required Inputs	Required Outputs	Optional Outputs	Time Integration Control
Response Only Model **(Coupling Elimination)**	$p \to g(p)$	p	g		**Internal**
State Elimination Model	$p \to x(p)$	p	x	g	**Internal**
Fully Implicit Time Step Model	$f(x,p) = 0$	x, p	f	$W,\ M,\ g$	**Internal**
Transient Explicitly Defined ODE Model	$\dot{x} = f(x,p,t)$	x, p, t	f	$W,\ M,\ g$	**External**
Transient Fully Implicit DAE Model	$f(\dot{x},x,p,t) = 0$	$\dot{x}, x, p, t$	f	$W,\ M,\ g$	**External or Internal**

$$W = \alpha \frac{\partial f}{\partial \dot{x}} + \beta \frac{\partial f}{\partial x} \qquad M = \text{preconditioner}$$

An Assortment of Coupling Algorithms

- **Picard-based (Black-Box)**
  - Block Nonlinear Jacobi
  - Block Nonlinear Gauss-Seidel
  - Anderson Acceleration
- **Newton Based (Block Implicit)**
  - Jacobian-free Newton-Krylov
  - Newton-Krylov (Explicit Jacobian)
  - Nonlinear Elimination (Schur complement formulation)

Example: Two Component system

$$f_0(x_0, z_{0,0}) = 0$$
$$f_1(x_1, z_{1,0}) = 0$$
$$z_{0,0} = r_{0,0}(x_1)$$
$$z_{1,0} = r_{1,0}(x_0)$$

Picard Iteration: Nonlinear Block Gauss-Seidel

Require: Initial guesses $x_0^{(0)}$ and $x_1^{(0)}$ for x_0 and x_1:
k = 0
while not converged **do**
  k = k+1
  Solve $f_0(x_0^{(k)}, r_{0,0}(x_1^{(k-1)})) = 0$ for $x_0^{(k)}$
  Solve $f_1(x_1^{(k)}, r_{1,0}(x_0^{(k)})) = 0$ for $x_1^{(k)}$
end while

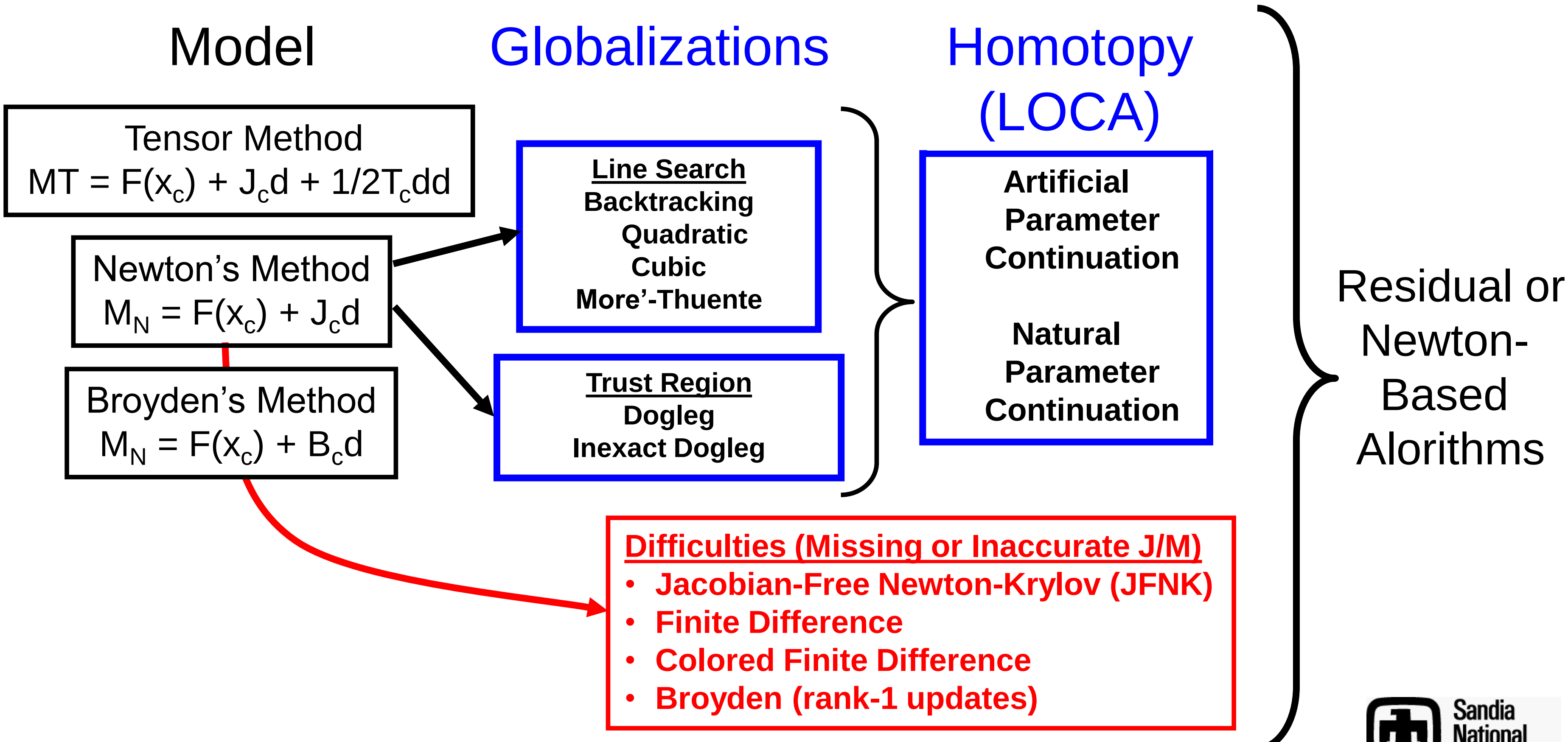
Newton-based

$$\begin{bmatrix} \frac{\partial f_0}{\partial x_0} & \frac{\partial f_0}{\partial z_{0,0}} \frac{\partial r_{0,0}}{\partial x_1} \\ \frac{\partial f_1}{\partial z_{1,0}} \frac{\partial r_{1,0}}{\partial x_0} & \frac{\partial f_1}{\partial x_1} \end{bmatrix} \begin{bmatrix} \Delta x_0^{(k)} \\ \Delta x_1^{(k)} \end{bmatrix} = - \begin{bmatrix} f_0(x_0^{(k-1)}, r_{0,0}(x_1^{(k-1)})) \\ f_1(x_1^{(k-1)}, r_{1,0}(x_0^{(k-1)})) \end{bmatrix}$$

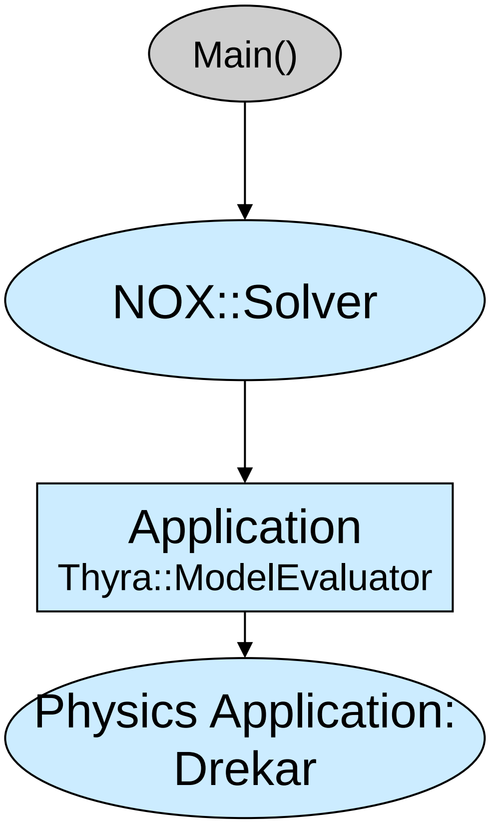
- Off-block diagonals may be hard to compute
- Can avoid computing Jacobian by using JFNK,
- BUT you still need to precondition ($M \approx W^{-1}$)

Implicit Solvers
NOX and LOCA: Nonlinear Solution and Homotopy

- Efficient: Quadratic convergence rates, no CFL limit
- Robust: globalization techniques



Simple Nonlinear Solve



Globalized
Newton-Krylov

$$x^k = x^{k-1} - \alpha J^{-1} f$$

$$f_{Drekar}$$

Block Composite Model

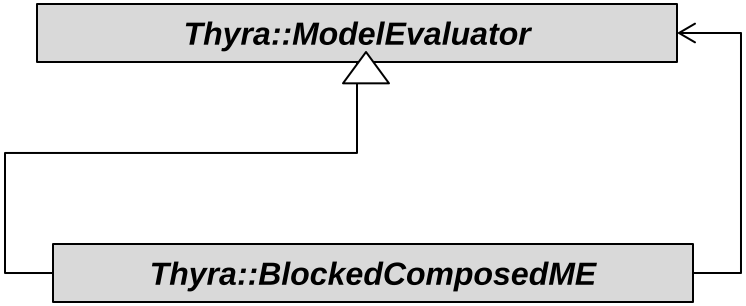
- The entire coupled system can be cast as a monolithic system:

$$\hat{f}(\dot{\hat{x}}, \hat{x}, \hat{p}, t) = 0$$

$$\begin{aligned}
\dot{\hat{x}} &= \left[\dot{x}_0, \ldots, \dot{x}_i, \ldots, \dot{x}_{N_f-1}\right], \\
\hat{x} &= \left[x_0, \ldots, x_i, \ldots, x_{N_f-1}\right], \\
\hat{p} &= \left[p_{0,0}, \ldots, p_{0,N_{p0}-1}, \ldots, p_{i,0}, \ldots, p_{i,N_{p_i}-1}, \ldots, p_{N_f-1,0}, \ldots, p_{N_f-1,N_{p_{N_f-1}}-1}\right], \\
\hat{f} &= \begin{bmatrix} f_0(\dot{x}_0, x_0, \{r_{0,k}(\{x_m\},\{p_{m,n}\})\}, \{p_{0,l}\}, t)) \\ \vdots \\ f_i(\dot{x}_i, x_i, \{r_{i,k}(\{x_m\},\{p_{m,n}\})\}, \{p_{i,l}\}, t)) \\ \vdots \\ f_{N_f-1}(\dot{x}_{N_f-1}, x_{N_f-1}, \{r_{N_f-1,k}(\{x_m\},\{p_{m,n}\})\}, \{p_{N_f-1,l}\}, t)) \end{bmatrix}.
\end{aligned}$$

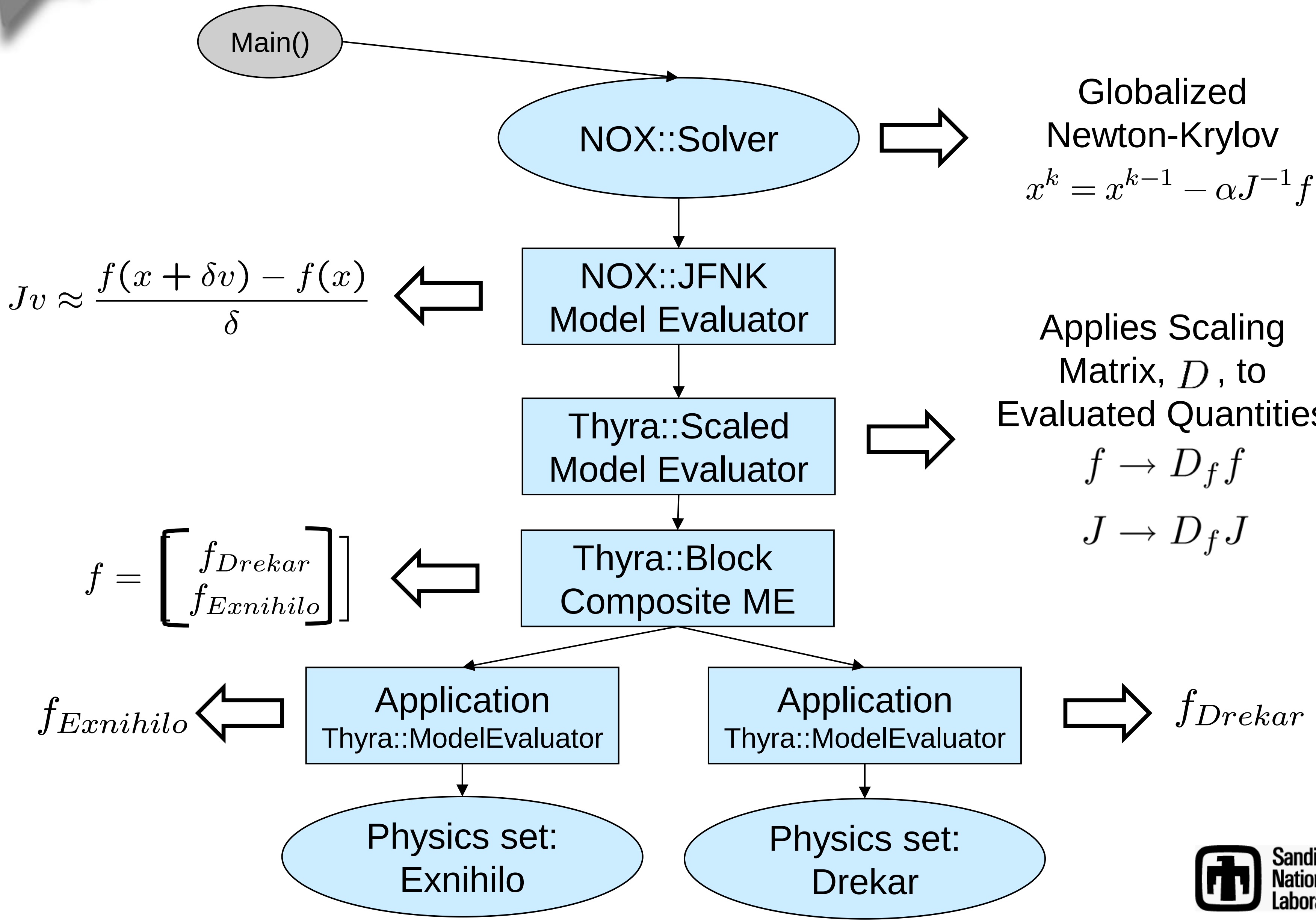
The Power of Decorators

- Use **inheritance** and **composition** to wrap analysis tools as model evaluators to build a hierarchical chain.



- Example ANA decorator subclasses
  - BlockCompositeModelEvaluator: Aggregate physics into blocked objects
  - FiniteDifferenceModelEvaluator: Global finite differences w.r.t. inputs
  - JacobianFreeNewtonKrylovModelEvaluator: Wraps a “residual-only” model evaluator to provide a Jacobian operator
  - StateEliminationModelEvaluator: Eliminates steady state equations/variables using a NonlinearSolverBase object
  - DiagonalScalingModelEvalautor: Apply a user defined diagonal scaling operator for outArgs
  - DefaultEvaluationLoggerModelEvaluator: Log evaluations vs. time and print out summary table

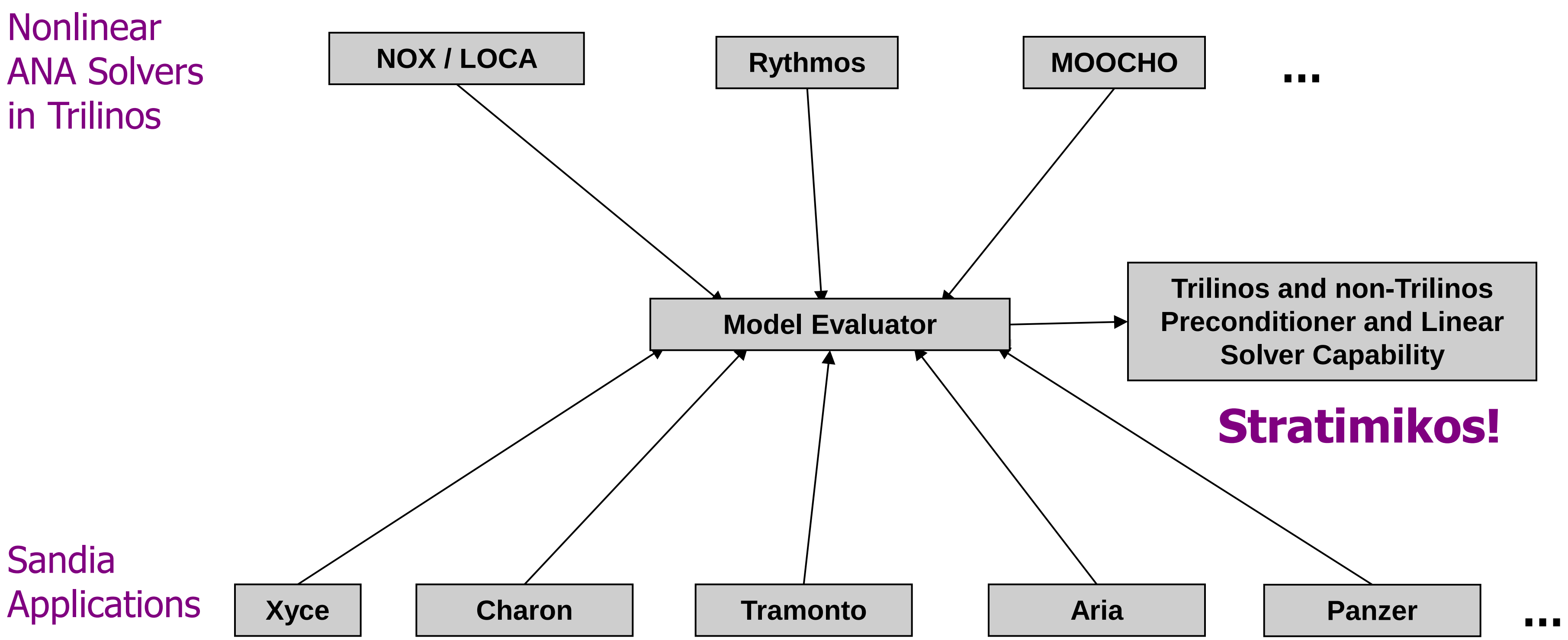
Uses **Decorator** to better condition a poorly scaled system of equations



Sandia National Laboratories

Linear Solvers
and Preconditioners

Nonlinear Algorithms and Applications : Thyra & Model Evaluator!



Key Points

- Provide single interface from nonlinear ANAs to applications
- Provide single interface for applications to implement to access nonlinear ANAs
- Provides shared, uniform access to linear solver capabilities
- Once an application implements support for one ANA, support for other ANAs can quickly follow

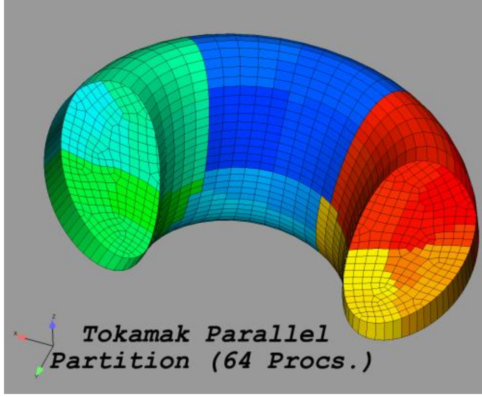
All Linear Solvers in Trilinos can be selected at run time from an XML File

```xml
<ParameterList name=”Stratimikos" >
   <ParameterList name="AztecOO">
      <Parameter name="Aztec Preconditioner" type="string" value="ilu"/>
       <Parameter name="Aztec Solver" type="string" value="GMRES"/>
       <Parameter name="Maximum Iterations" type="int" value="100"/>
       ...
   <ParameterList name="Belos">
      <ParameterList name="Solver Types">
          <ParameterList name="Block GMRES">
            <Parameter name="Convergence Tolerance" type="double" value="1e-5"/>
            <Parameter name="Maximum Iterations" type="int" value="100"/>
            <Parameter name=”Flexible GMRES" type=”bool" value=”false"/>
            <Parameter name="Orthogonalization" type="string" value="DGKS"/>
          <ParameterList name="Block CG”>
         ...
   <ParameterList name="Preconditioner Types">
      <ParameterList name="Ifpack">
         <Parameter name="Prec Type" type="string" value="ILU"/>
         <Parameter name="Overlap" type="int" value="0"/>
         <Parameter name=”Fill Factor" type="int" value=“1"/>
       ...
      <ParameterList name="ML">
        <Parameter name=”nodes per aggregate" type="int" value=“27"/>
        <Parameter name=”coarse: max size" type="int" value=“512"/>
       ...
</ParameterList>
```

Trilinos Linear Solvers: Heroux, Tuminaro, Hu, Bartlett, Thornquist, Hoemmen, Cyr,…

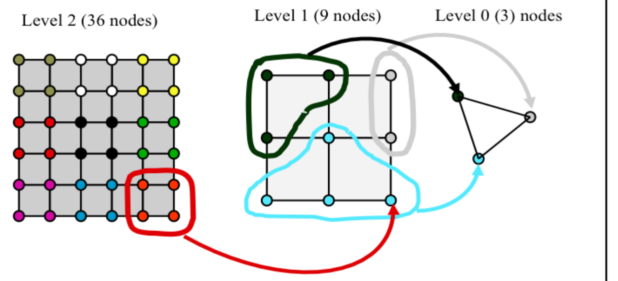
Sandia National Laboratories

Three Types of Preconditioning



1. Domain Decomposition (Trilinos/IFPack)
   - 1 –level Additive Schwarz DD
   - ILU(k) Factorization on each processor (variable levels of overlap)
   - High parallel efficiency, non-optimal algorithmic scalability

2. Multilevel Methods for Systems: (Trilinos/ML/MueLu)
   - Fully-coupled Algebraic Multilevel methods
   - Consistent set of DOF at each node (e.g. stabilized FE)
   - Uses block non-zero structure of Jacobian
   - Aggregation techniques and coarsening rates can be set
     - Smoothed aggregation (SA)
     - Aggressive Coarsening (AggC)
   - Jacobi, GS, ILU(k) as smoothers
   - Can provide optimal algorithmic scalability



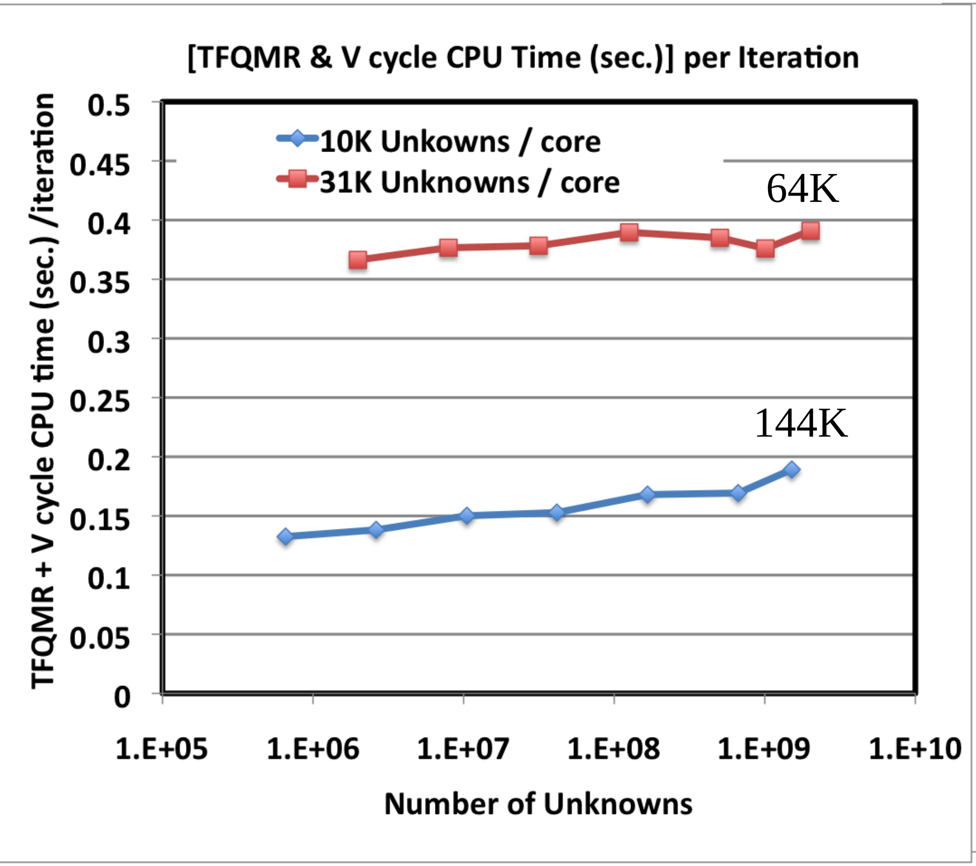
Visualization of effect of partition of matrix graph on mesh

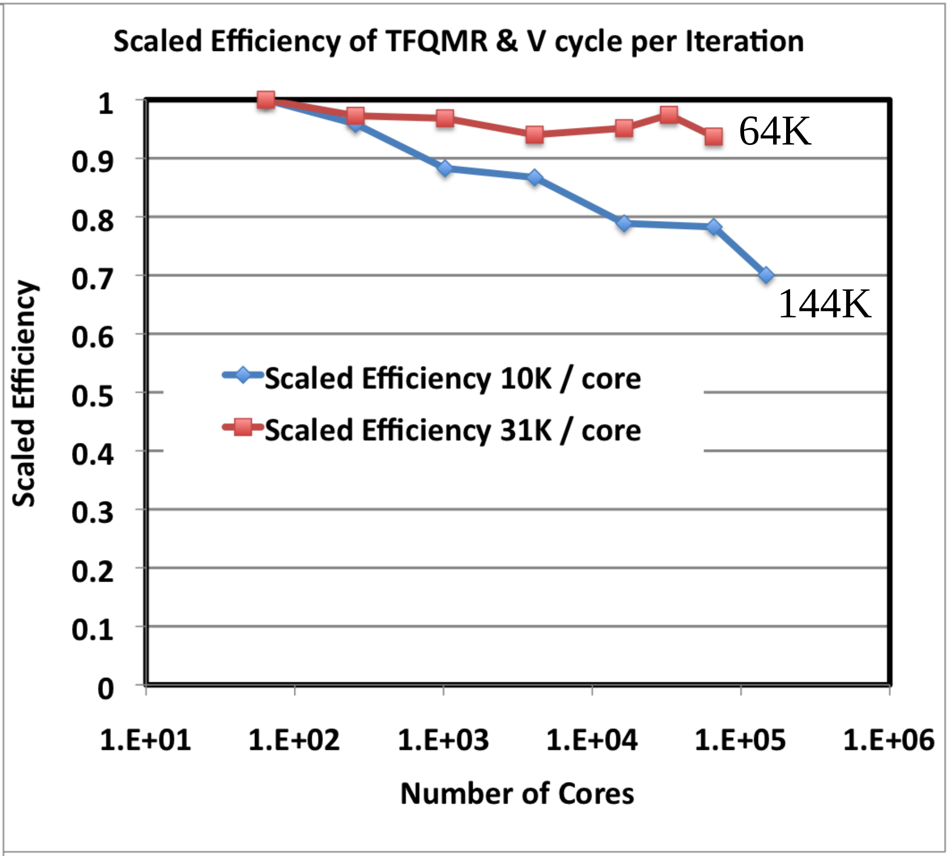
3. Approximate Block Factorization / Physics-based (Trilinos/Teko)
   - Applies to mixed interpolation (FE), staggered (FV), using segregated unknown blocking
   - Applied to systems where coupled AMG is difficult or might fail
   - Can provide optimal algorithmic scalability

Aggregation based Multigrid: Vanek, Mandel, Brezina, 1996; Vanek, Brezina, Mandel, 2001; Sala, Formaggia, 2001

Sandia National Laboratories

Weak Scaling Uncoupled Aggregation Scheme: Time/iteration on BlueGene/P

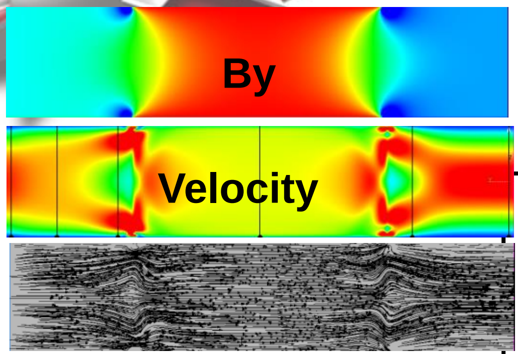


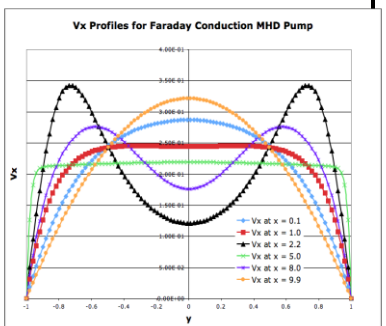


- TFQMR: used to look at time/iteration of multilevel preconditioners.
- W-cyc time/iteration not doing well due to significant increase in work on coarse levels (not shown)
- Good scaled efficiency for large-scale problems on larger core counts for 31K Unknowns / core

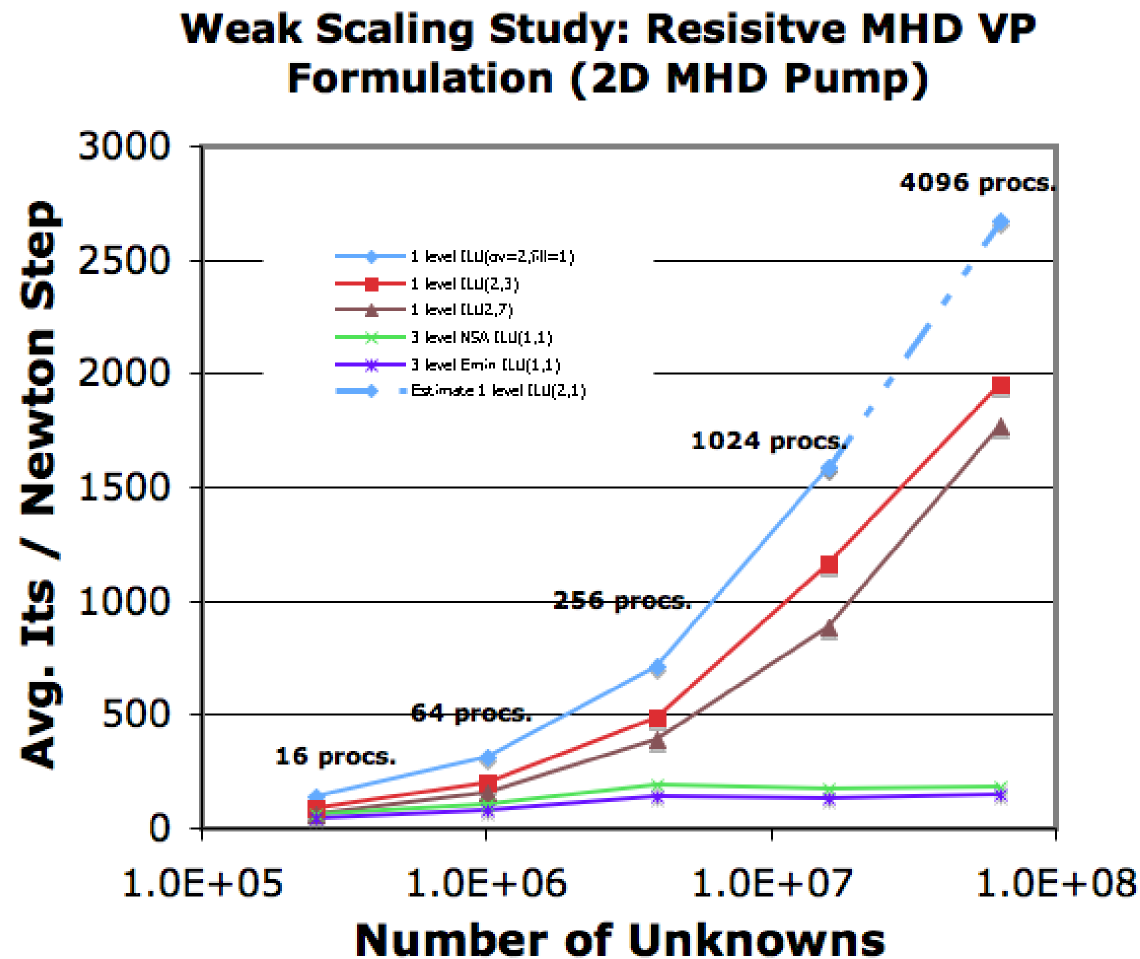
Sandia National Laboratories

Scalability
(MHD Pump, Cray XT3)





MHD Pump



Preconditioners

- 1-level ILU(2,1)
- 1-level ILU(2,3)
- 1-level ILU(2,7)
- 3-level ML(NSA,Gal)
- 3-level ML(EMIN, PG)

ML: Tuminaro, Hu
Ifpack: Heroux

Sandia National Laboratories

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
$\begin{pmatrix} F & B^T \\ \hat{B} & -C \end{pmatrix} \begin{pmatrix} \Delta u_k \\ \Delta p_k \end{pmatrix} = \begin{pmatrix} g_u^k \\ g_p^k \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ \hat{B}F^{-1} & I \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -S \end{pmatrix} \begin{pmatrix} I & F^{-1}B^T \\ 0 & I \end{pmatrix}$ $S = C + \hat{B}F^{-1}B^T$	$\begin{bmatrix} I & 0 \\ \hat{B}H_1 & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & -\hat{S} \end{bmatrix} \begin{bmatrix} I & H_2 B^T \\ 0 & I \end{bmatrix}$

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

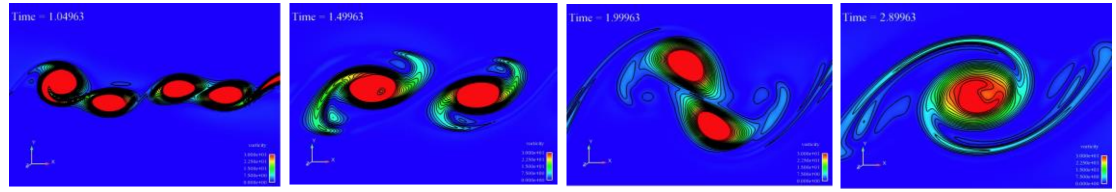
Now use AMG type methods on sub-problems.

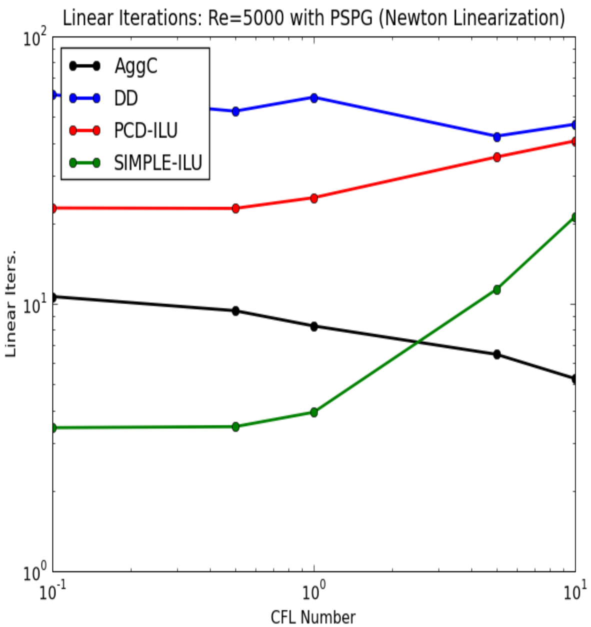
Momentum transient convection-diffusion: $F\Delta \mathbf{u} = \mathbf{r_u}$

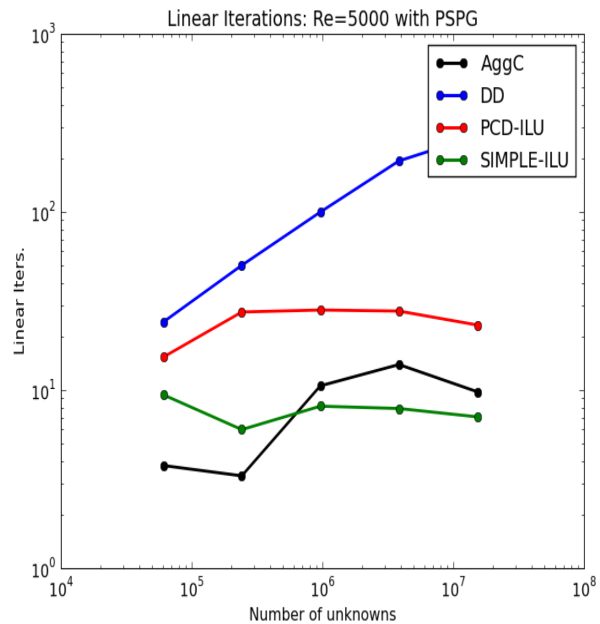
Pressure – Poisson type: $-\hat{S}\Delta p = \mathbf{r}_p$

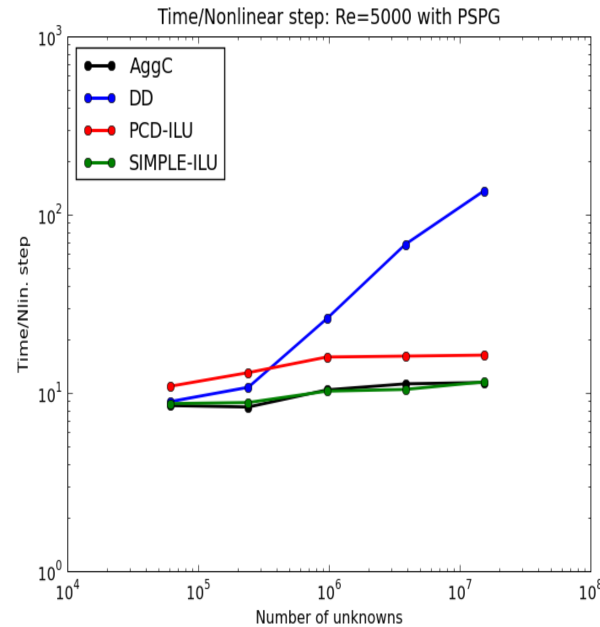
Sandia National Laboratories

Transient Kelvin-Helmholtz









Kelvin Helmholtz: Re=5000, Weak scaling at CFL=2.5

- **Run on 1 to 256cores**
- **Pressure - PSPG, Velocity - SUPG (residual and Jacobian)**

1. **SIMPLEC strongly dependent on CFL**
2. **Block methods scale as well as AggC and do not require non-zero C matrix**

Sandia National Laboratories

Incompressible MHD
2D Vector Potential Formulation

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

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

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

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

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

Discretized using a stabilized finite element formulation

Structure of discretized Incompressible MHD system is

$$\mathcal{J}\mathbf{x} = \begin{bmatrix} F & B^T & Z \\ B & C & 0 \\ Y & 0 & D \end{bmatrix}\begin{bmatrix} u \\ p \\ A \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ e \end{bmatrix}$$

Matrices F and D are transient convection operators, C is stabilization matrix

Teko Block Preconditioners

Nested Schur Complements:

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

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

$$\mathcal{M} = \begin{bmatrix} F & B^T & Z \\ & S_{Neu} & -BF^{-1}Z \\ & & P_{Neu} \end{bmatrix}$$

Physics Based: Operator Splitting:

$\hat{x} = \textbf{SplitPrec-NS}(\mathcal{J}, b)$:

$$x^* = \begin{bmatrix} F & & Z \\ & I & \\ Y & & D \end{bmatrix}^{-1} b,$$

$$r^* = b - \mathcal{J}x^*,$$

$$e = \begin{bmatrix} F & B^T & \\ B & C & \\ & & I \end{bmatrix}^{-1} r^*,$$

$$\hat{x} = x^* + e$$

$$\mathcal{M}_{Split} = \begin{bmatrix} F & B^T & Z \\ B & C & \\ Y & \boxed{YF^{-1}B^T} & D \end{bmatrix} = \begin{bmatrix} F & & Z \\ & I & \\ Y & & D \end{bmatrix} \begin{bmatrix} F^{-1} & & \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} F & B^T & \\ B & C & \\ & & I \end{bmatrix}$$

- Eliminates nested Schur Complements
- Requires two 2x2 solves
- Navier-Stokes operator well studied
- Magnetics-Velocity operator is difficult

Sandia National Laboratories

Physics-based/ABF Preconditioning

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbb{I} + \mathbf{\Pi}) - \frac{1}{\mu_0}\nabla \times \mathbf{B} \times \mathbf{B} = 0$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) + \nabla \times (\frac{\eta}{\mu_0}\nabla \times \mathbf{B}) = 0$$

⇩

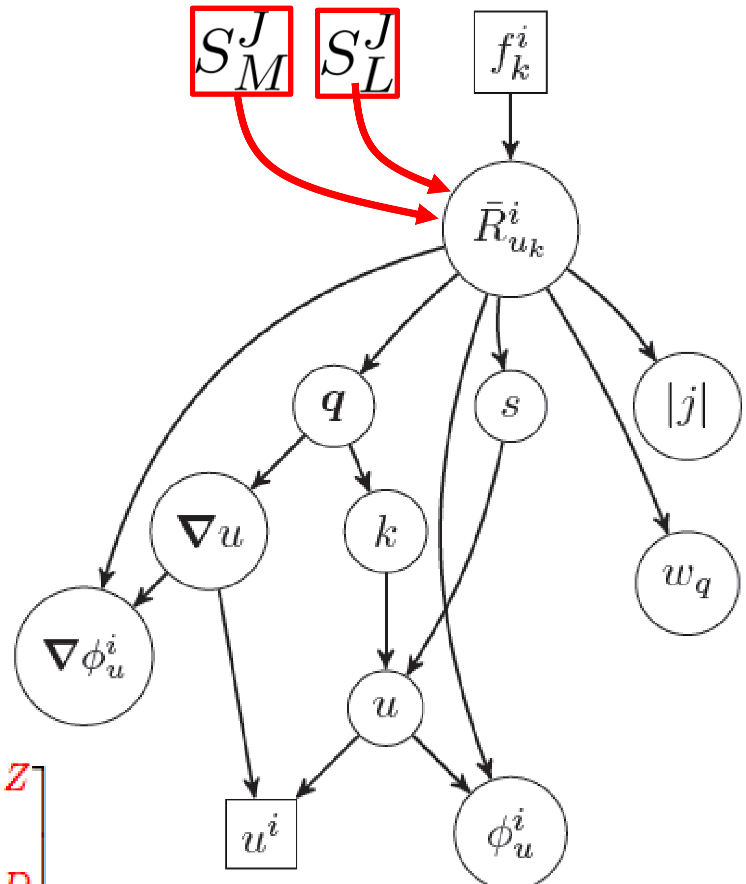
$$\begin{bmatrix} \begin{bmatrix} F & B^T \\ B & C \end{bmatrix} & \begin{bmatrix} Z \\ 0 \end{bmatrix} \\ \begin{bmatrix} Y & 0 \end{bmatrix} & D \end{bmatrix} \begin{bmatrix} u \\ p \\ b \end{bmatrix} = \begin{bmatrix} f \\ g \\ h \end{bmatrix}$$

⇩

$$\begin{bmatrix} F & B^T & Z \\ B & C & 0 \\ Y & 0 & D \end{bmatrix}$$

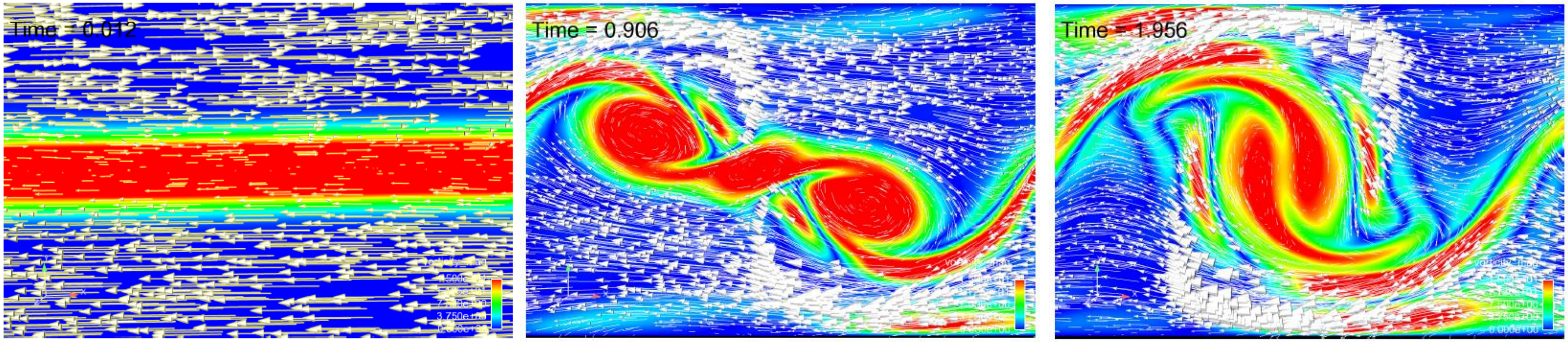
$$\approx \begin{bmatrix} F & & Z \\ & I & \\ Y & & D \end{bmatrix} \begin{bmatrix} F^{-1} & & \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} F & B^T & \\ B & C & \\ & & I \end{bmatrix} = \begin{bmatrix} F & B^T & Z \\ B & C & \\ Y & \boxed{YF^{-1}B^T} & D \end{bmatrix}$$

JFNK + Block Scattering for Preconditioning



National Laboratories

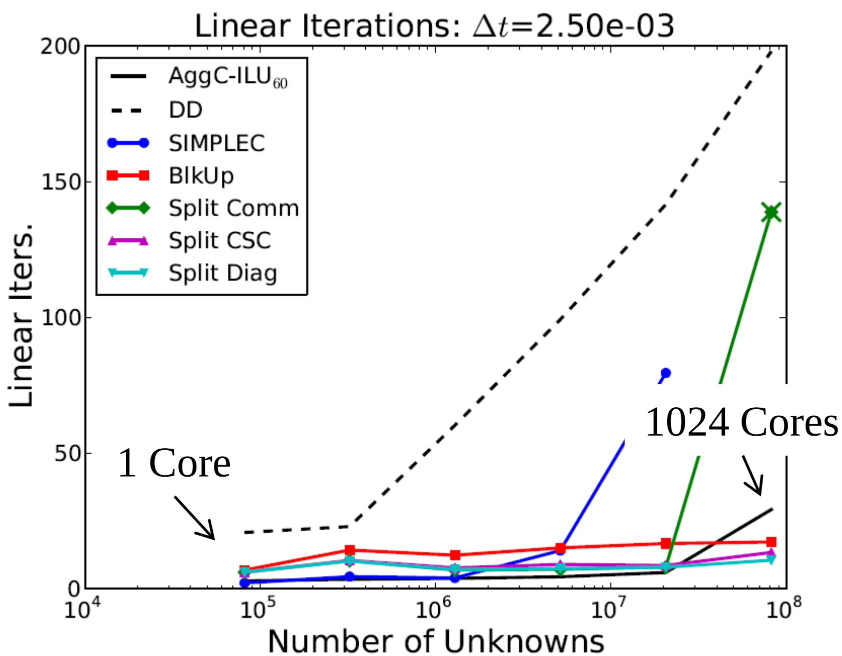
Hydromagnetic Kelvin-Helmholtz

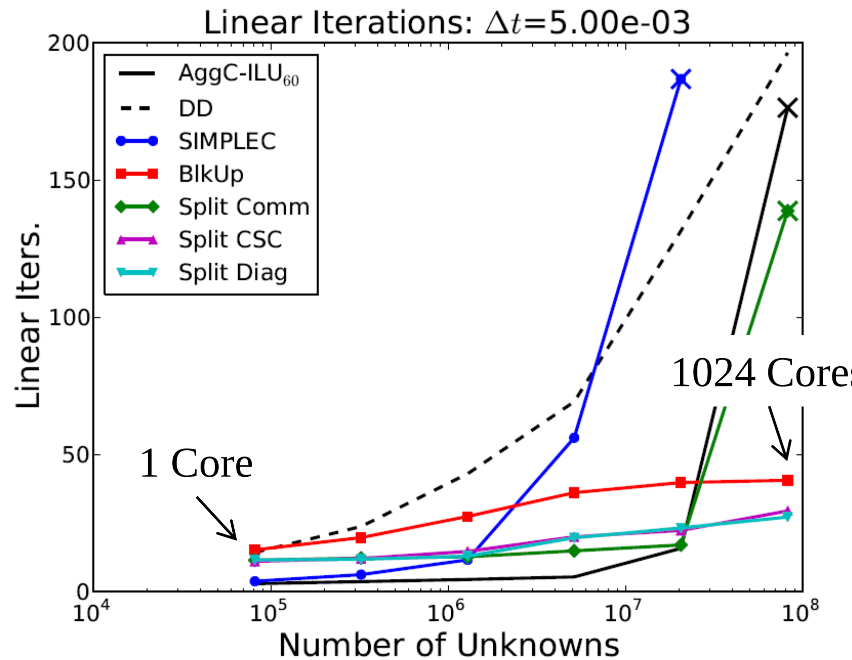


t = 0.012 t = 0.906 t = 1.956

- Velocity shear flow
- Magnetic field in x-direction
- Reynolds number = 10^3
- Lundquist number = 10^4

MHD Weak Scaling: Hydromangetic Kelvin-Helmholtz (Fixed time step)





Fully coupled Algebraic

AggC: Aggressive Coarsening Multigrid

DD: Additive Schwarz Domain Decomposition

Block Preconditioners

Split: New Operator split preconditioner

SIMPLEC: Extreme diagonal approximations

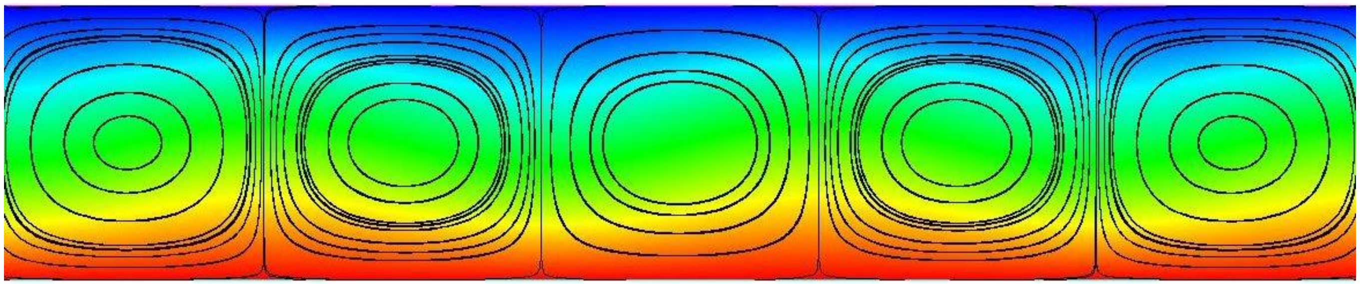
Take home: Split preconditioner scales algorithmically, more relevant for mixed discretizations, multiphysics

Hydromagnetic Rayleigh-Bernard

B_0 ↑ g ↓

$v_x = 0$, $v_y = 0$ $\quad T = -0.5 \quad \frac{\partial \mathcal{A}}{\partial y} = 0$

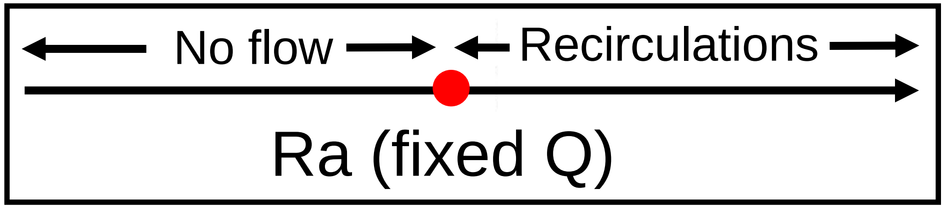
$v_x = 0$, $\mathcal{A} = C\sqrt{(Q)}$



$v_x = 0$, $\mathcal{A} = -C\sqrt{(Q)}$

$v_x = 0$, $v_y = 0$ $\quad T = 0.5 \quad \frac{\partial \mathcal{A}}{\partial y} = 0$

Parameters:
- Q ~ B_0^2 (Chandresekhar number)
- Ra (Rayleigh number)

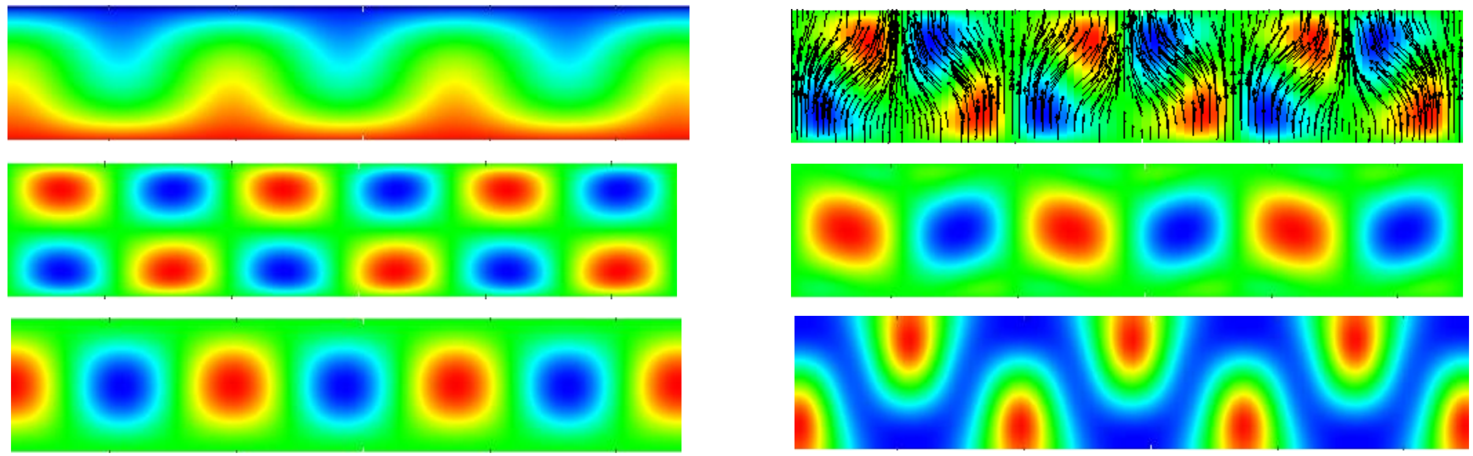


$$Q = \frac{B_0^2 d^2}{\mu_0 \rho \nu \eta} \qquad Ra = \frac{g\beta \Delta T d^3}{\nu \alpha} \qquad Pr = \frac{\nu}{\alpha} \qquad Pr_m = \frac{\nu}{\eta}$$

- Buoyancy driven instability initiates flow at high Ra numbers.
- Increased values of Q delay the onset of flow.
- Domain: 1x20

Sandia National Laboratories

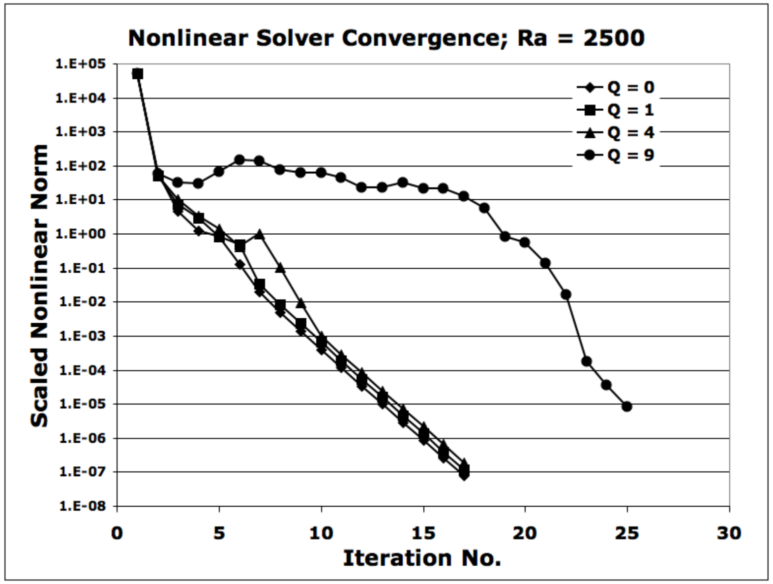
Hydro-Magnetic Rayleigh-Bernard Stability: Direct Determination of Nonlinear Equilibrium Solutions (Steady State Solves, Ra=2500, Q=4)

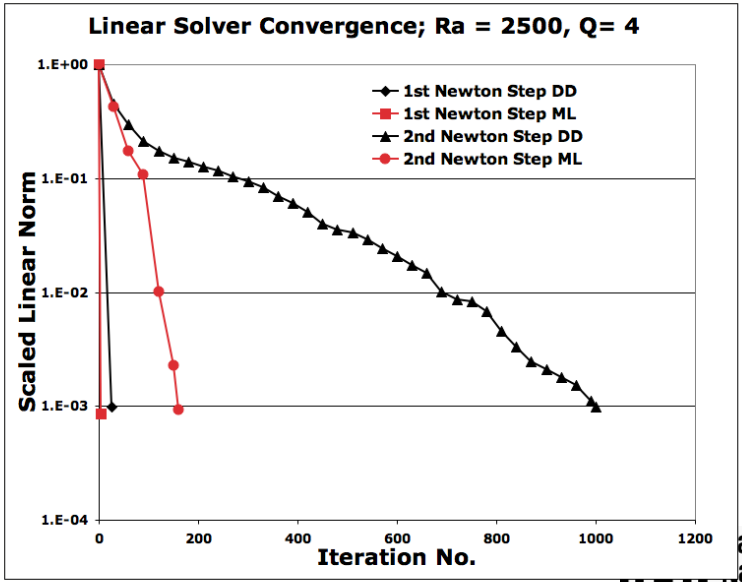


Temp Jz

Vx Bx

Vy By

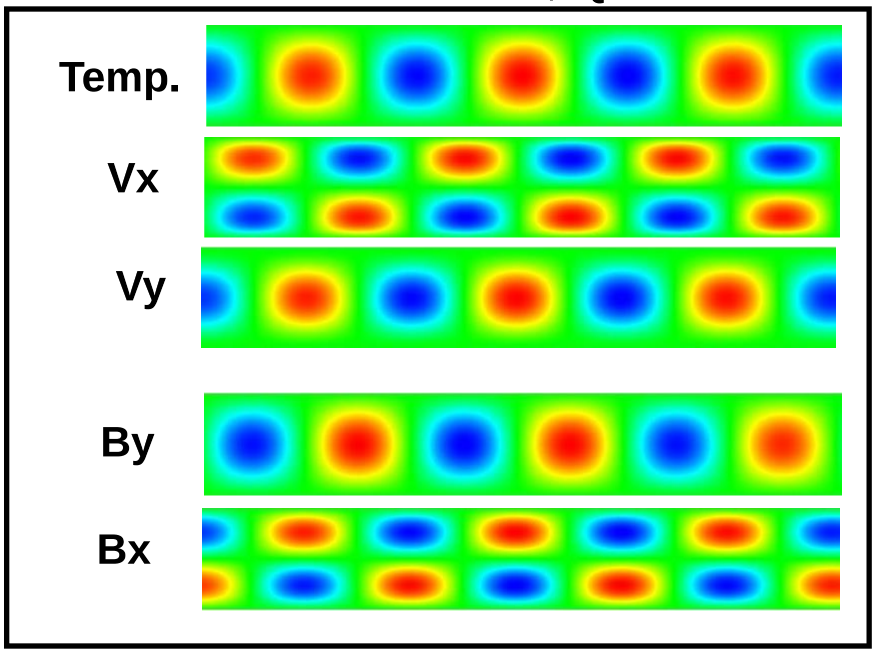




andia
ational
Laboratories

Hydro-Magnetic Rayleigh-Bernard Stability: Direct Determination of Linear Stability and Nonlinear Equilibrium Solutions (Steady State Solves)

Leading Eigenvector at Bifurcation Point,
Ra = 1945.78, Q=10



Q	Ra*	Ra_{cr} [Chandrasekhar[]]	% error
0	1707.77	1707.8	0.002
10^1	1945.78	1945.9	0.006
10^2	3756.68	3757.4	0.02

- 2 Direct-to-steady-state solves at a given Q
- Arnoldi method using Cayley transform to determine approximation to 2 eigenvalues with largest real part
- Simple linear interpolation to estimate Critical Ra*

Arc-length Continuation: Identification of Pitchfork Bifurcation, Q=10

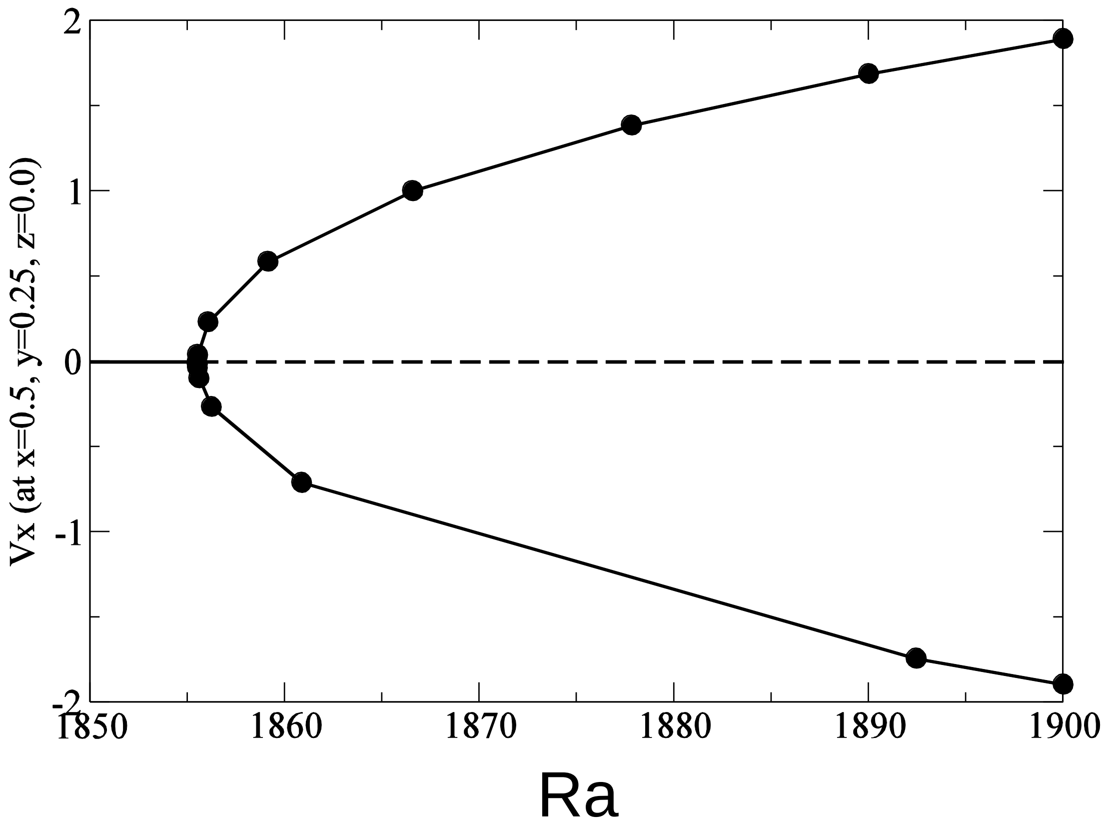
Nonlinear system:

$$f(x(s), p(s)) = 0$$

$$g(x(s), p(s), s) = 0$$

Newton System:

$$\begin{bmatrix} \boldsymbol{J} & \boldsymbol{f_p} \\ \left(\frac{\partial x}{\partial s}\right)^T & \frac{\partial p}{\partial s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta p} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}$$



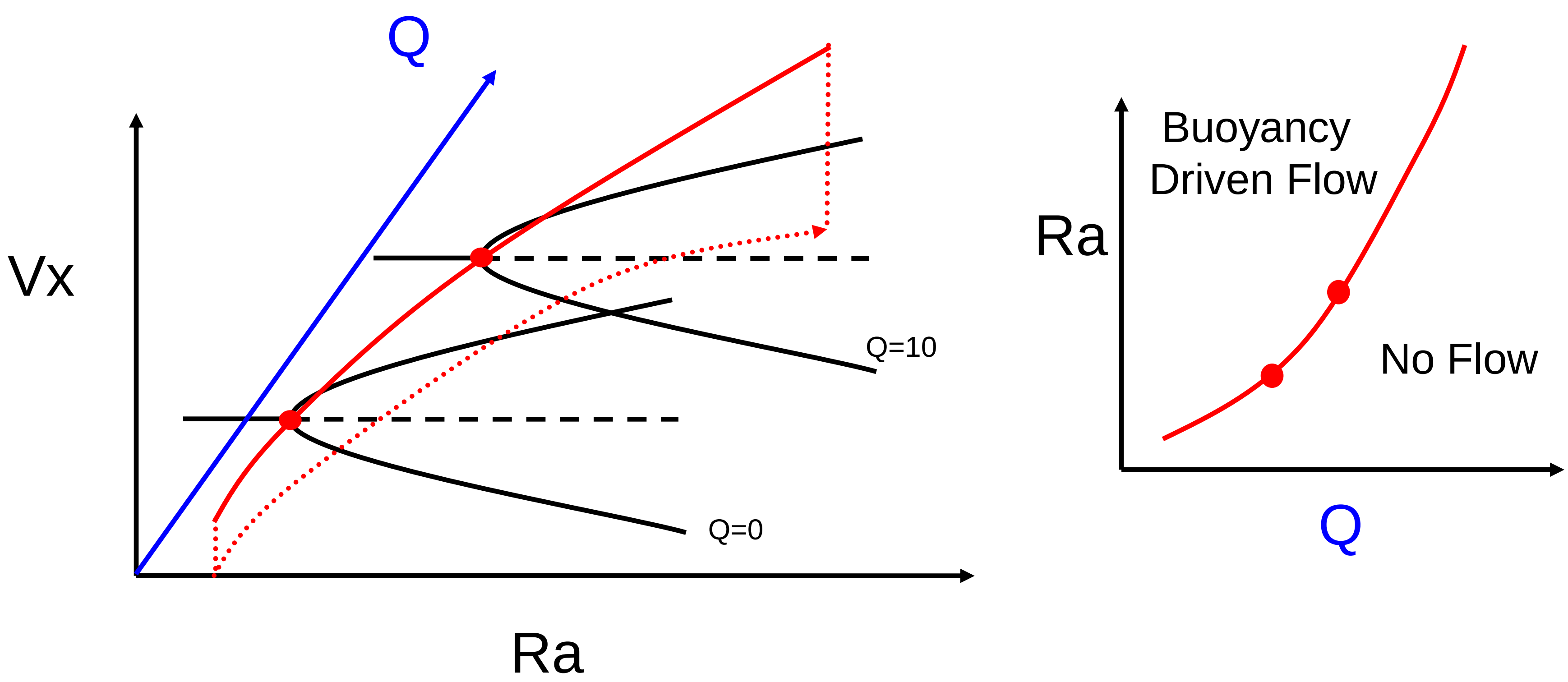
Bordered Solver:

$$Ja = -f \qquad \Delta p = -(g + \frac{\partial x}{\partial s} \cdot a)/(\frac{\partial p}{\partial s} + \frac{\partial x}{\partial s} \cdot b)$$

$$Jb = -f_p \qquad \Delta x = a + \Delta p b$$

Sandia National Laboratories

Design (Two-Parameter) Diagram



- “No flow” does not equal “no-structure” – pressure and magnetic fields must adjust/balance to maintain equilibrium.
- LOCA can perform multi-parameter continuation

Bifurcation Tracking
(Govaerts 2000)

Moore-Spence

- Turning point formulation:

$$f(x,p) = 0$$
$$Jn = 0$$
$$\phi \cdot n - 1 = 0$$

- Newton's method (2N+1):

$$\begin{bmatrix} J & 0 & f_p \\ (Jn)_x & J & J_p n \\ 0 & \phi^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta n \\ \Delta p \end{bmatrix} = \begin{bmatrix} -f \\ -Jn \\ 1 - \phi^T \cdot n \end{bmatrix}$$

- 4 linear solves per Newton iteration:

$$Ja = -f$$
$$Jb = -f_p$$
$$Jc = -(Jn)_x a - Jn$$
$$Jd = -(Jn)_x b - J_p n$$
$$\Delta p = (1 - \phi \cdot n - \phi \cdot c)/(\phi \cdot d)$$
$$\Delta n = c + \Delta p d$$
$$\Delta x = a + \Delta p b$$

Minimally Augmented

- Widely used algorithm for small systems:

$$\begin{bmatrix} \boldsymbol{J} & \boldsymbol{a} \\ \boldsymbol{b}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{J}^T & \boldsymbol{b} \\ \boldsymbol{a}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\Longrightarrow s = t = -\boldsymbol{u}^T \boldsymbol{J} \boldsymbol{v}$$

- J is singular if and only if s = 0

- Turning point formulation (N+1):

$$\boldsymbol{f}(\boldsymbol{x}, p) = 0$$
$$s(\boldsymbol{x}, p) = 0$$

- Newton's method:

$$\begin{bmatrix} \boldsymbol{J} & \boldsymbol{f}_p \\ \boldsymbol{s}_x & s_p \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta p \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f} \\ s \end{bmatrix},$$
$$\boldsymbol{s}_x = -(\boldsymbol{u}^T \boldsymbol{J} \boldsymbol{v})_x = -(\boldsymbol{u}^T \boldsymbol{J})_x \boldsymbol{v}.$$

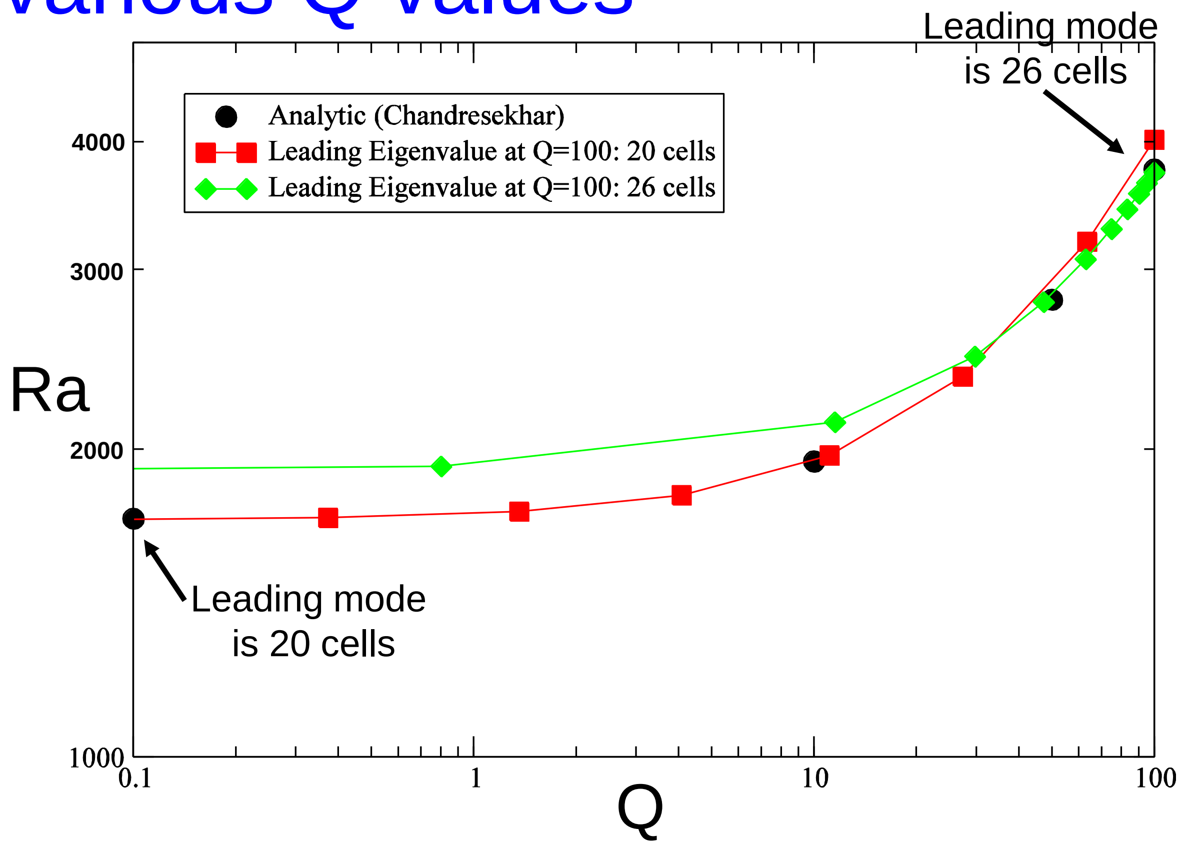
- 3 linear solves per Newton iteration

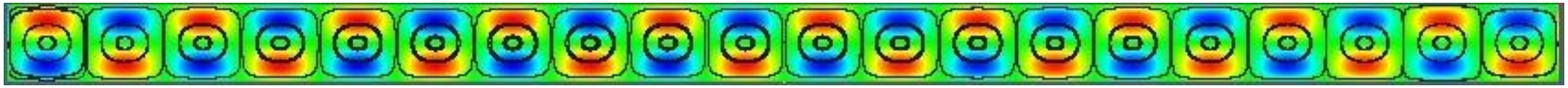
Extension to large-scale iterative solvers

Sandia National Laboratories

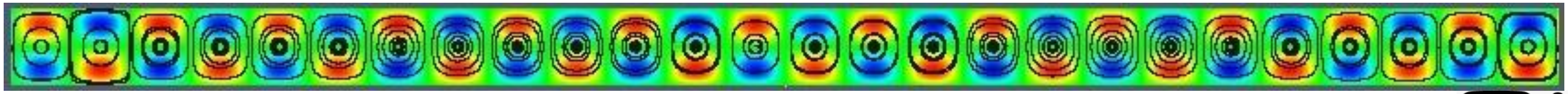
Leading Mode is different for various Q values

- Analytic solution is on an infinite domain with two bounding surfaces (top and bottom)
- Multiple modes exist, mostly differentiated by number of cells/wavelength.
- Therefore tracking the same eigenmode does not give the stability curve!!!
- Periodic BCs will not fix this issue.





Mode: 20 Cells: Q=100, Ra=4017



Mode: 26 Cells: Q=100, Ra=3757

Sandia National Laboratories

SEACISM: Parallel Glimmer-CISM2

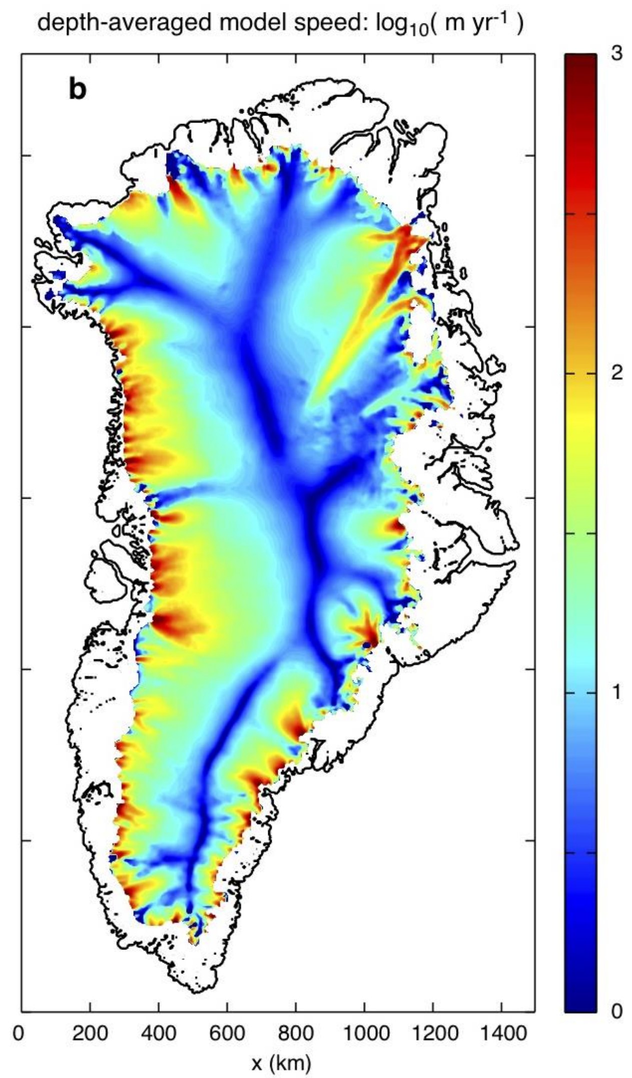
Evans, Worley, Nichols, Norman (ORNL)
Price, Lipscomb, Hoffman (LANL)
Salinger, Kalashnikova, Tuminaro (SNL)
Lemieux (NYU), Sachs (NCAR)

Glimmer Code ~2009:

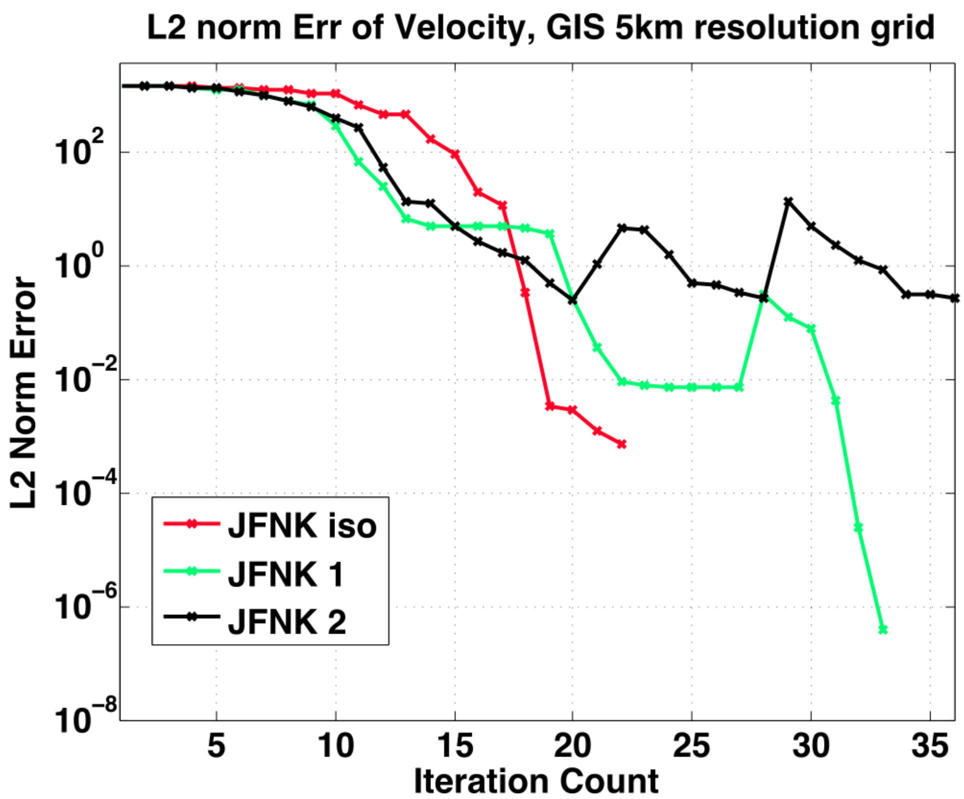
- First-Order Approx to Stokes: 3D for [U,V]
- Structured Grid
- Finite Difference
- Serial
- Picard Solver
- Autoconf

Glimmer Code ~2013:

- Parallel Assembly
- Parallel Solve
- Newton Solver
- Cmake
- Built in CESM development branch



Convergence is not adequately robust reliable for Greenland problems



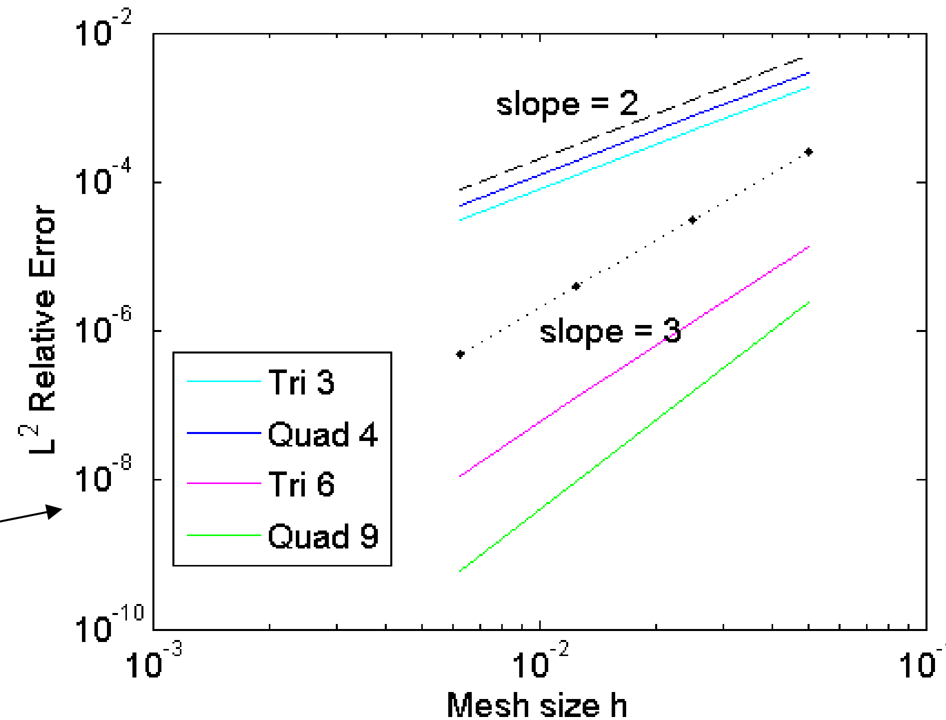
Why the poor robustness?

- Real, noisy data
- Nonlinear viscosity model
- Structured grid Finite Diff
- Finite Diff for Stress BCs
- Jacobian-Free perturbations
- Picard matrices

New FELIX Codes address many of these issues (PISCEES SciDAC-BER)

FELIX Codes[1,2]:

- Real, noisy data
- Nonlinear viscosity model
  - Included in Jacobian
- Unstructured Grid
- Finite Element Stress BCs
- Newton, Analytic Jacobian
- Rigorous Verification
- Hooks to UQ Algorithms
- Numerous Trilinos Libraries
  - Discretization
  - Load Balancing

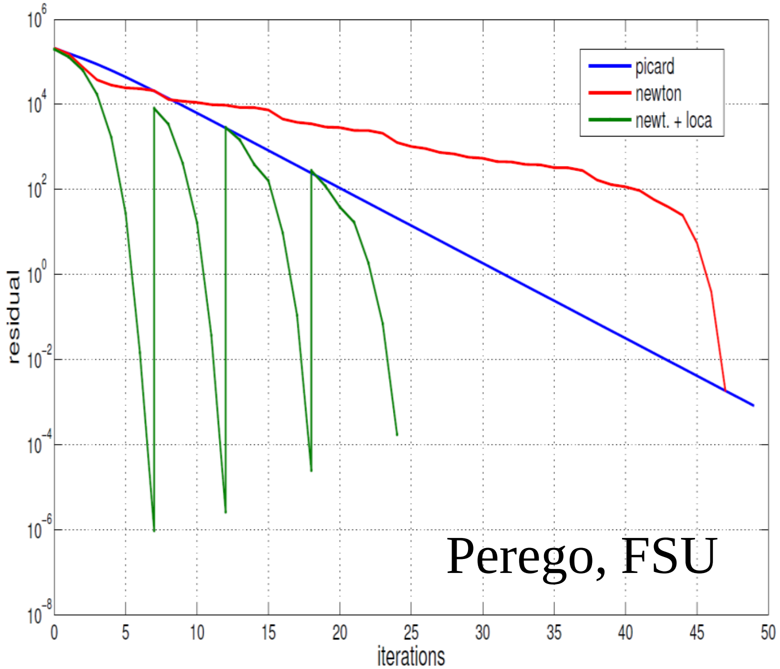


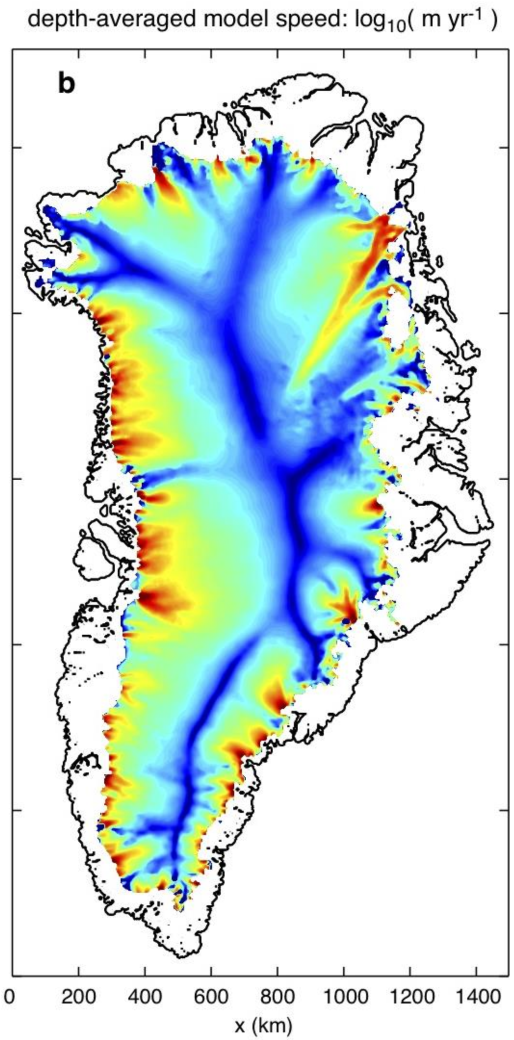
Manufactured Solution

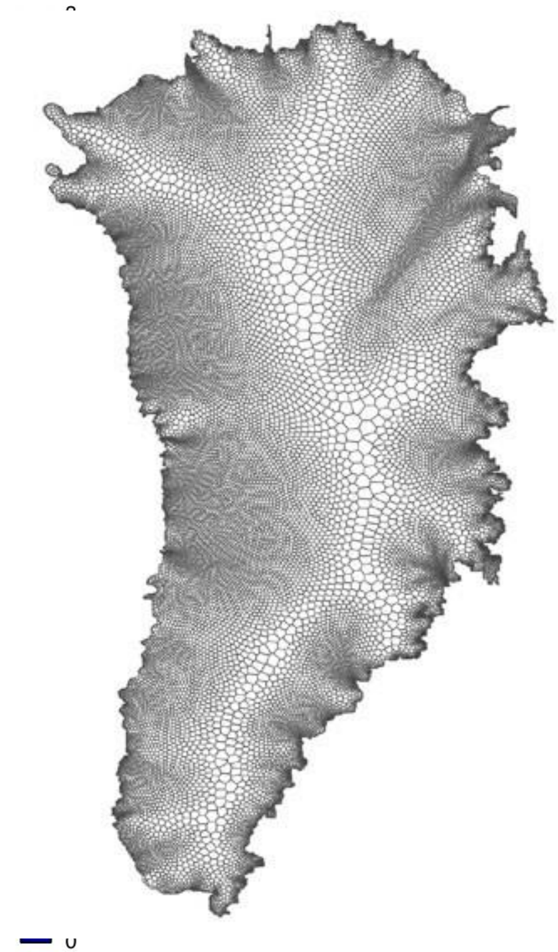
$$
\begin{aligned}
u &= \sin(2\pi x)\cos(2\pi y) + 3\pi x, \\
v &= -\cos(2\pi x)\sin(2\pi y) - 3\pi y
\end{aligned}
$$

1. Perego, Gunzberger, Ju (LifeV, Trilinos, MPAS)
2. Salinger, Kalashnikova, Perego, Tuminaro (Albany, Trilinos, *MPAS*)

Full Newton with Analytic Jacobian fixes some causes of robustness issues







Conclusions

- Trilinos contains a diverse set of algorithms
  - Abstract Interfaces
  - Linear Algebra
  - Linear solvers
  - Preconditioners
  - Nonlinear solvers and Analysis
  - Discretization libraries
- The toolkit approach is critical
  - Flexibility is key
  - Each physics is unique and requires its own strategy
- Coupled codes must leverage large body of knowledge from stand-alone applications
  - Directly use app solver: Picard it
  - Use app solver in a physics-based preconditioner