Algebraic Solvers in FASTMath

Argonne Training Program on Extreme-Scale Computing
August 2015

Algebraic Solvers in FASTMath

- Hypre – see detailed presentation
- PARPACK
- PETSc – see detailed presentation
- SUNDIALS – see detailed presentation
- SuperLU – see detailed presesntation
- Trilinos-ML,NOX

PARPACK

- **Capabilities:**
  - Compute a few eigenpairs of a Hermitian and non-Hermitian matrix
  - Both standard and generalized eigenvalues
  - Extremal and interior eigenvalues
  - Reverse communication allows easy integration with application
  - MPI/BLACS communication
- **Download:**

  http://www.caam.rice.edu/software/ARPACK/

- **Further information:** beyond PARPACK
  - EIGPEN (based on penalty trace minimization for computing many eigenpairs)
  - Parallel multiple shift-invert interface for computing many eigenpairs

Other large-scale eigensolvers

- PPCG (Projected Preconditioned Conjugate Gradient) method for symmetric eigenvalue problems
  - For computing a relatively large number of smallest eigenpairs
  - Reduce Rayleigh-Ritz cost
- GPLHR (Generalized Preconditioned Local Harmonic Ritz) method for interior eigenvalues of a non-Hermitian sparse matrix
- Special solver for linear response eigenvalue problems (TDDFT linear response and Bethe-Salpeter equation)

PARPACK usage

- A list of drivers provided in $ARPACKTOPDIR/ PARPACK/EXAMPLES
- Reverse communication interface

```
10 continue
   call pdsaupd(comm, ido,….)
   if (ido .eq. 1 .or. -1) then
      matvec(…,workd(ipntr(1)), workd(ipntr(2)….
   endif
   goto 10
```

- Hybrid MPI/OpenMP implementation
- MATLAB interface available (eigs)

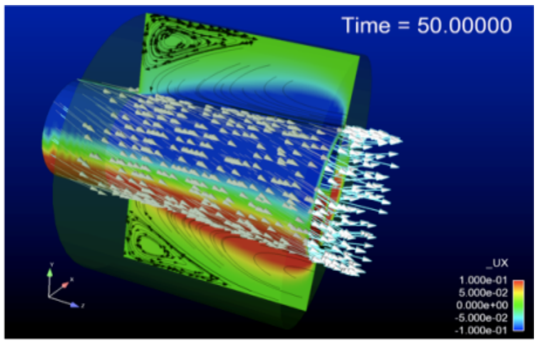
ML and MueLu: Multigrid libraries in Trilinos

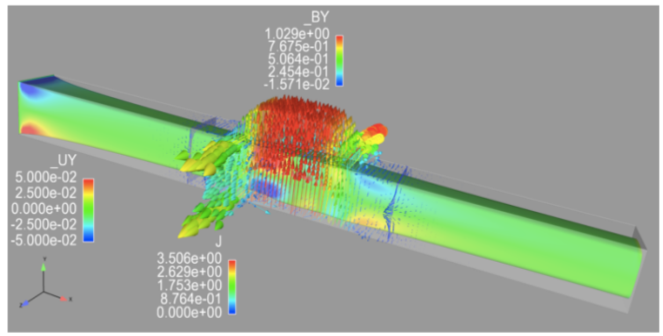
- **ML: aggregation-based algebraic multigrid algorithms**
  - Support for scalar problems (diffusion, convection-diffusion), PDE systems (elasticity), electromagnetic problems (eddy current)
  - Various coarsening and data rebalancing options
  - Smoothers (SOR, polynomial, ILU, block variants, line, user-provided)
  - Written in C
- **MueLu: templated multigrid framework**
  - Support for energy minimizing multigrid algorithms in addition to many algorithms from ML
  - Leverages Trilinos templated sparse linear algebra stack
    - Optimized kernels for multiple architectures (GPU, OpenMP, Xeon Phi)
    - Templated scalar type allowing mixed precision, UQ, …
  - Advanced data reuse possibilities, extensible by design
  - Written in C++
- **Download/further information:** www.trilinos.org

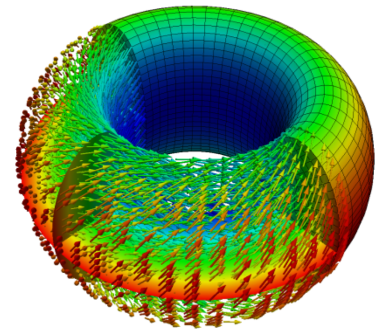
ML and MueLu: Application highlights

- **Magnetohydrodynamics (Drekar)**

  ML scales to 512K cores on BG/Q and to 128K cores on Titan

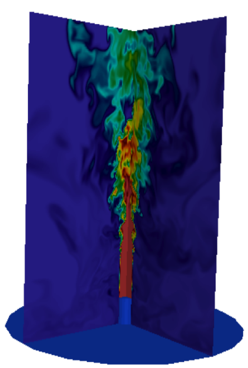


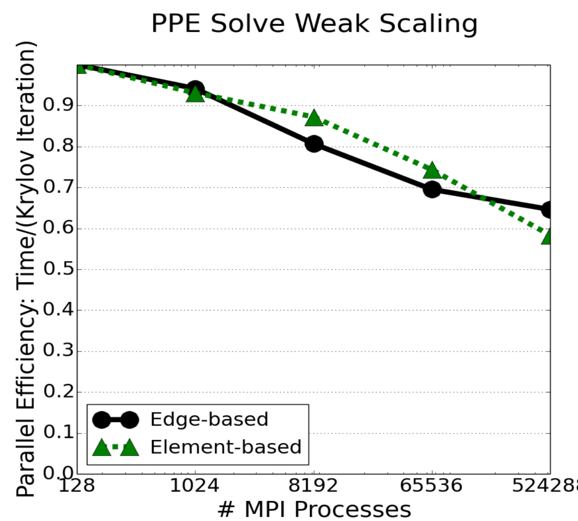




- **Fluid dynamics (Nalu)**

  MueLu scales to 524K cores of BG/Q





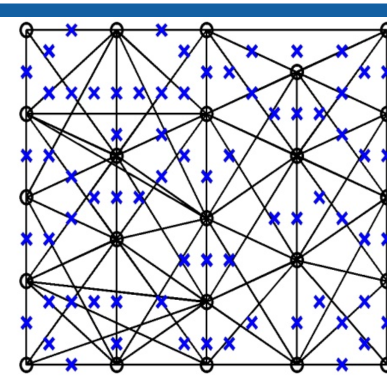
Argonne National Laboratory · Berkeley Lab · Sandia National Laboratories · Berkeley University of California · Columbia University · Rensselaer · SMU · The University of British Columbia

MueLu: Research framework

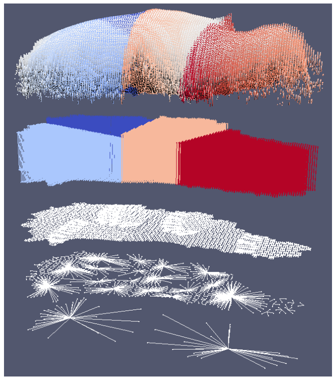
- **Component reuse in multigrid can be effective in reducing setup costs while maintaining solver convergence. We have demonstrated that reuse can yield 2.5x speedup on 25K cores of Cray XE6.**

- **Block systems arise naturally in mixed discretizations. Our new multigrid algorithm preserves such block structure on coarse levels for Stokes and Navier-Stokes systems.**

- **MueLu/ML provide a specialized AMG for PISCEES project through semi-coarsening and line smoothers that exploit partial structure in meshes arising in ice sheet modeling.**



Automatically generated coarse mesh for Q2-Q1 discretization of a Stokes system.



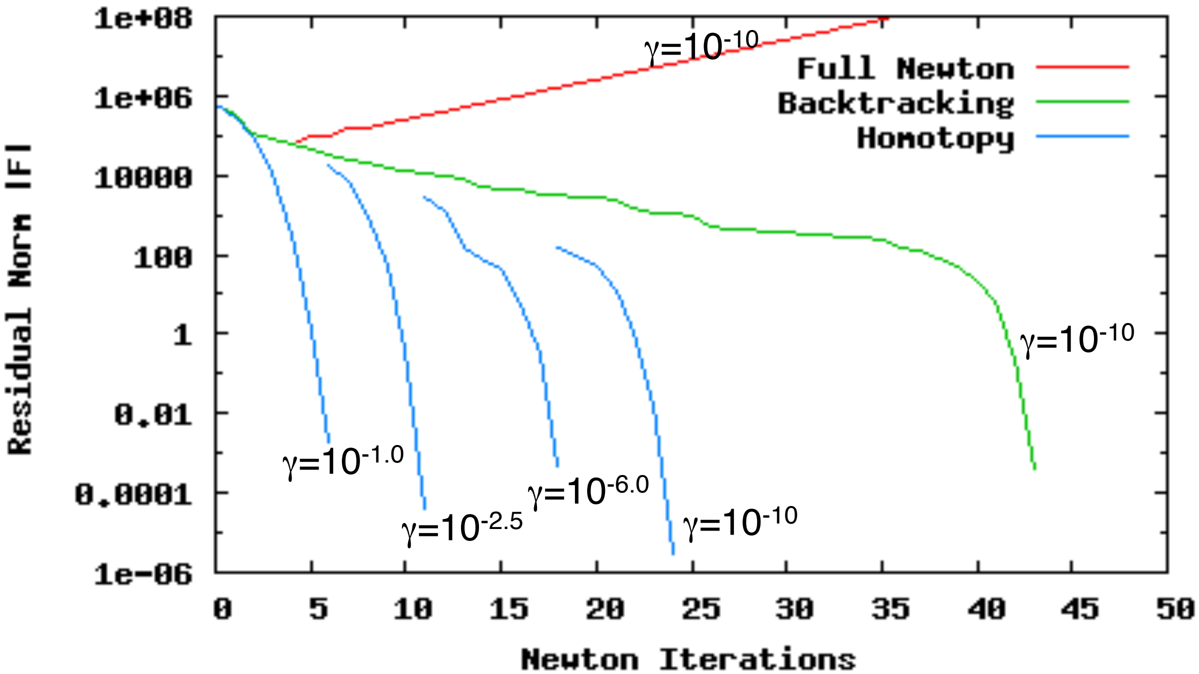
Semicoarsening followed by regular 2D coarsening for Greenland model.

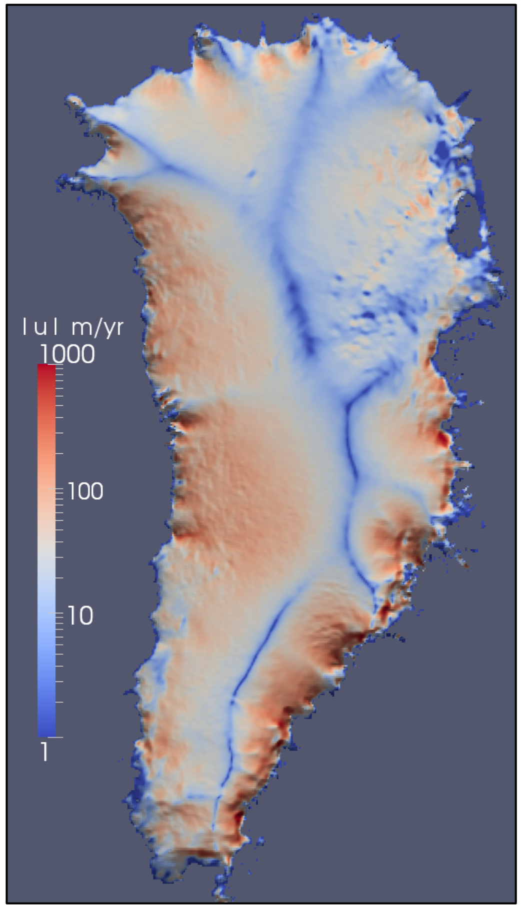
Trilinos/NOX Nonlinear Solver

- **Capabilities:**
  - Newton-Based Nonlinear Solver
    - Linked to Trilinos linear solvers for scalability
    - Matrix-Free option
  - Anderson Acceleration for Fixed-Point iterations
  - Globalizations for improved robustness
    - Line Searches, Trust Region, Homotopy methods
  - Customizable: C++ abstractions at every level
  - Extended by LOCA package
    - Parameter continuation, Stability analysis, Bifurcation tracking
- **Download: Part of Trilinos (trilinos.sandia.gov)**
- **Further information: Andy Salinger [agsalin@sandia.gov]**

Trilinos/NOX: Robustness for Ice Sheet Simulation:
PISCEES SciDAC Application project (BER-ASCR)

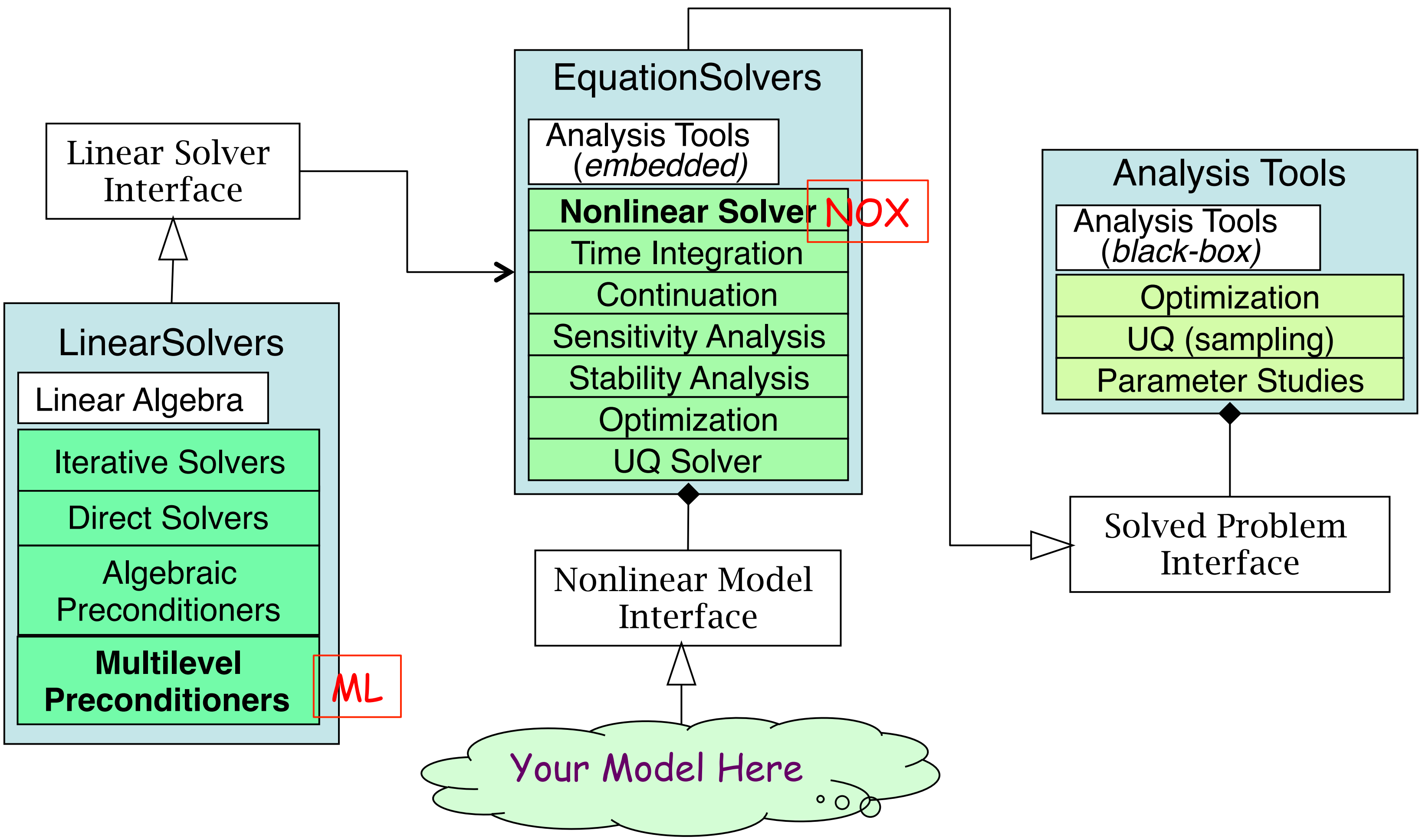
- Ice Sheets modeled by nonlinear Stokes's equation
  - Initial solve is fragile: Full Newton fails
  - Homotopy continuation on regularization parameter "γ" saves the day





Greenland Ice Sheet Surface Velocities
(constant friction model)

NOX and ML are part of larger Trilinos solver stack: Linear solvers, Equations solvers, Analysis tools



TAO

Toolkit for Advanced Optimization

Now available as part of PETSc distribution (as of PETSc 3.5)

Solves Nonlinear Optimization Problems:

$$f : \mathbb{R}^N \mapsto \mathbb{R}$$

$$\min_{x \in \mathbb{R}^N} f(x)$$

With optional variable bounds:

$$\text{subject to} \quad x_l \leq x \leq x_u \quad \text{(bounds)}$$

Or complementarity constraints:

$$\begin{aligned} F_i(x^*) &\geq 0 \quad \text{if } x_i^* = \ell_i \\ F_i(x^*) &= 0 \quad \text{if } \ell_i < x_i^* < u_i \\ F_i(x^*) &\leq 0 \quad \text{if } x_i^* = u_i. \end{aligned}$$

There is also some support for PDE-constrained applications and general contraints

TAO Algorithms

TAO provides a suite of (iterative) nonlinear optimization algorithms. Typically, each iteration involves calculating a *search direction* d_k, then function values and gradients along that direction are calculated until desired conditions are met.

- Newton's Method
  Calculate the direction d_{k+1} by solving the system:
  $$\nabla^2 f(x_k) d_{k+1} = -\nabla f(x_k)$$
- Quasi-Newton Methods
  Use approximate Hessian $B_k \approx \nabla^2 f(x_k)$. Choose a formula for B_k so that B_k relies on first derivative information only, can be easily stored and $B_k d_{k+1} = -\nabla f(x_k)$ can be easily solved.
- Conjugate Gradient
- Derivative Free

TAO Solvers

Solvers available in TAO

	handles constraints	requires gradient	requires Hessian
Quasi-Newton (lmvm)	no	yes	no
Newton Line Search (nls)	no	yes	yes
Newton Trust Region (ntr)	no	yes	yes
Newton Trust with Line Search (ntl)	no	yes	yes
Conjugate Gradient (cg)	no	yes	no
Nelder-Mead (nm)	no	no	no
Quasi-Newton (blmvm)	bounds	yes	no
Newton Trust Region (tron)	bounds	yes	yes
Conjugate Gradient (gpcg) (Quadratic objective only)	bounds	yes	no
Model-based derivative free nonlinear least-squares (pounders)	yes	no	no
Semismooth – Feasibility-enforced (SSFLS)	complementarity	yes	yes
Semismooth – Feasibility not enforced (SSILS)	complementarity	yes	yes
Active-Set Semismooth – Feasibility-enforced (ASFLS)	complementarity	yes	yes
Active-Set Semismooth – Feasibility not enforced (ASILS)	complementarity	yes	yes
Linearly Constrained Lagrangian	pde		
Interior Point Method (ipm)	general	yes	yes