An improved solution framework for Greenland Ice Sheet simulation with Glimmer-CISM

Project Members:
John Burkardt, FSU*
Kate Evans, ORNL
Dana Knoll, Los Alamos
J.-F. Lemieux, New York U
Jeff Nichols, ORNL
Mauro Perego, FSU*
Andy Salinger, Sandia
Pat Worley, ORNL

*NEW

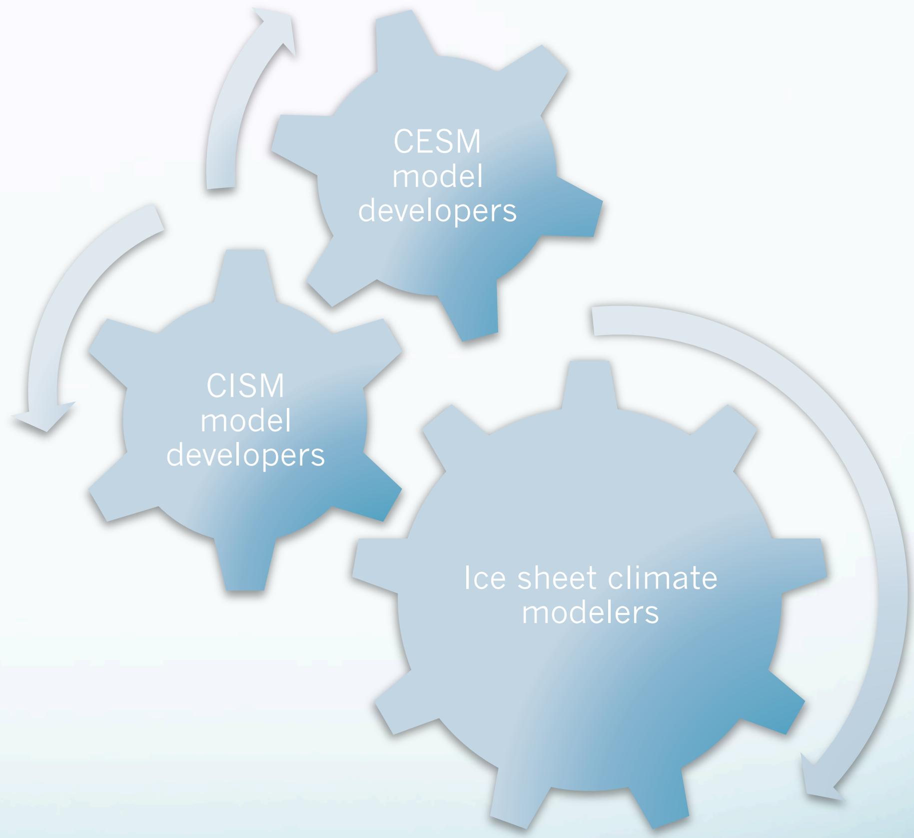
Consultation/Assistance from:
Max Gunzburger, FSU
David Holland, NYU
Bill Lipscomb, Los Alamos
Steve Price, Los Alamos
GLIMMER Steering committee

Thanks to DOE ASCR for supporting ice sheet model development!

Oak Ridge National Laboratory
Managed by UT-Battelle for the Department of Energy
The Florida State University
New York University
Los Alamos National Laboratory EST. 1943
Sandia National Laboratories
ASCR
U.S. Department of Energy

SEACISM Goal: Provide a state-of-the-art ice sheet model with continuous access by the climate community

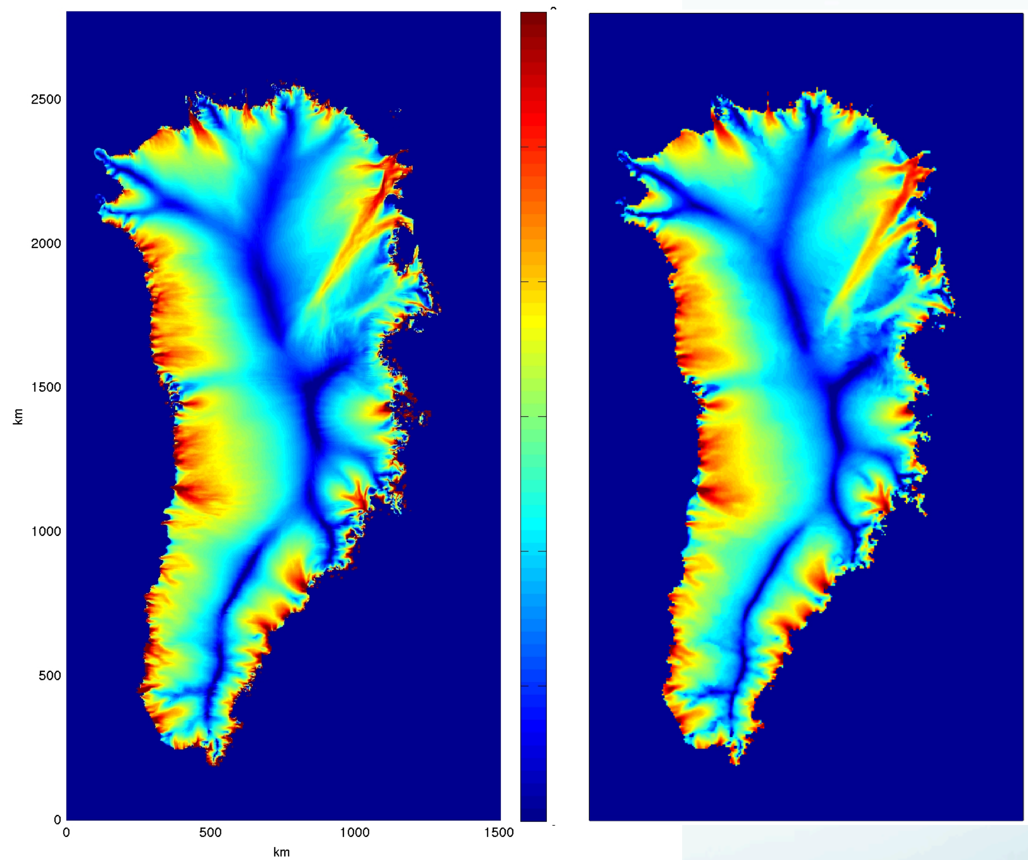
- Implement parallel, scalable capability as soon as possible to allow efficient high-resolution simulations
- Design to allow ongoing code extensions by ice sheet modelers, e.g. new parameterizations and equations
- Maintain consistency and interaction with the production-level CESM



EVENTUAL GOAL: coupled simulations with other climate components

Glimmer-CISM: The Community Ice Sheet Model

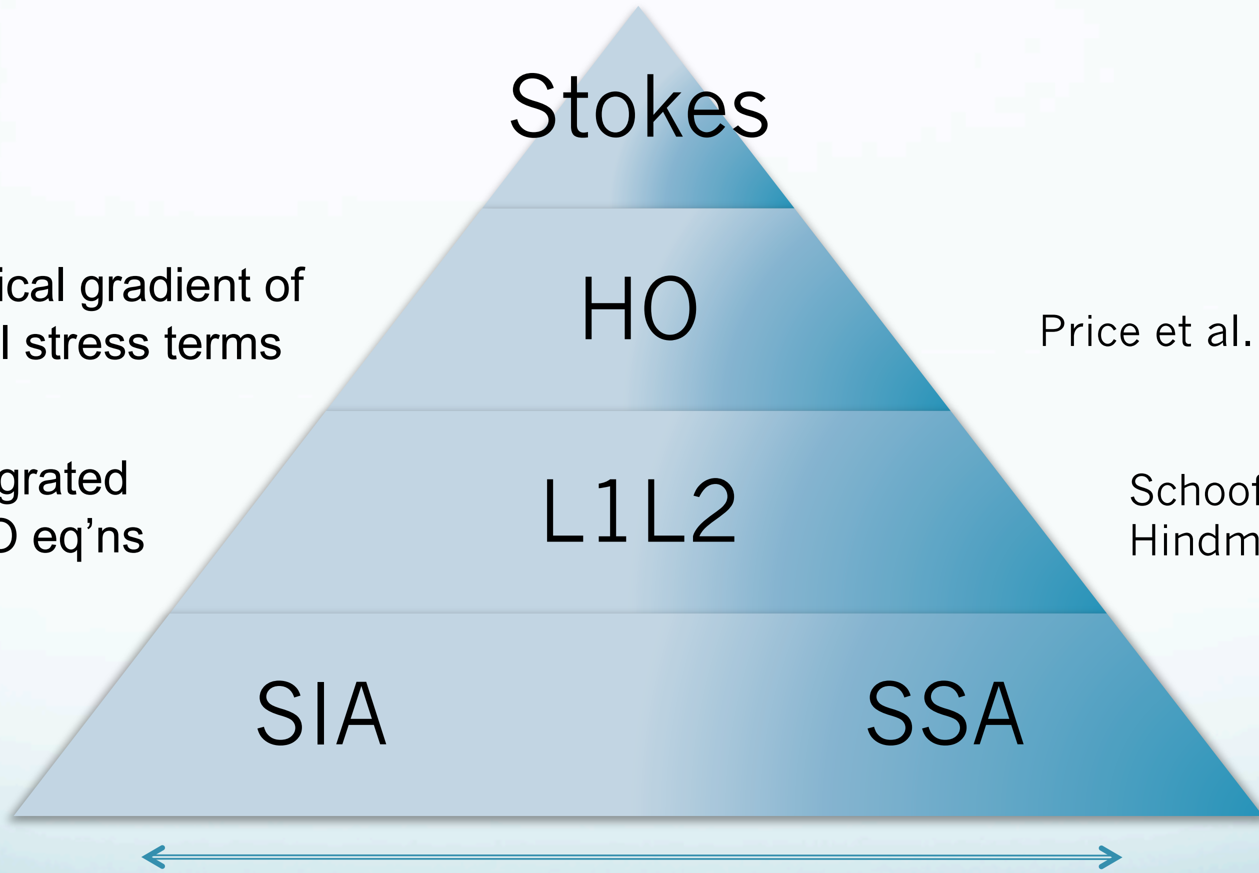
- Range of test cases and boundary conditions to span problem space
- Tuned, steady-state simulation using higher-order velocity solve
- Suites of high resolution (<5km) steady state runs desired to explore parameter space
- Eventually, time evolving ice sheet runs coupled with the ocean to provide predictive capability



Left panel: Balanced Velocity from higher-order flow model with tuned basal parameters.

Right panel: Glimmer-CISM 5km resolution tuned for steady state match of balance velocities. 10m/yr RMS difference in regions of interest, 35 m/yr overall.

‘Higher Order/First-order’ set of 3D momentum and mass balance equations; intermediate complexity



Includes vertical gradient of the horizontal stress terms

Price et al. (2011)

Vertically integrated alternative HO eq’ns

Schoof and Hindmarsh (2010)

No slip

Free slip

Local Tri-diag solve

Elliptic solve in 1 layer

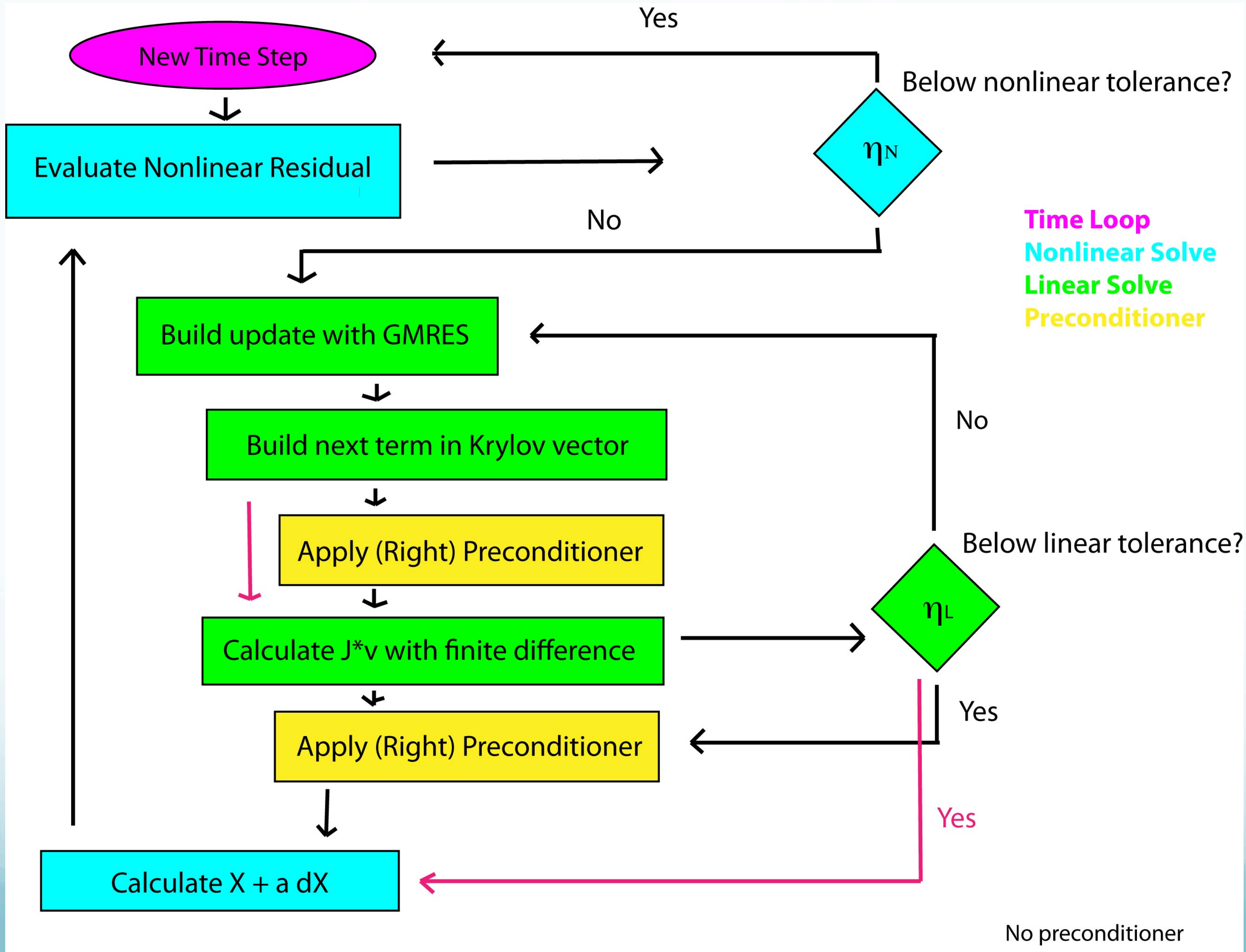
Solving the HO ice sheet momentum equations coherently to a set nonlinear tolerance

- Current G-CISM uses Picard, within which GMRES is called to solve velocity components sequentially
- Instead: Use Inexact Newton to solve F(u) = A(u)u – b = 0 system of nonlinear equations
- Picard: slow/cheap, Newton: fast/expensive (per iteration)

$$F'(x^{t+1})\delta x = -f(x^{t+1})$$

- Both can fail to converge, but Picard tends to stall whereas Newton can blow up spectacularly.
- Thus, we use Newton to solve, and several iterations of Picard as a preconditioner

Newton-Krylov solution method



Preconditioner: the key to solution efficiency

- Physics Based Preconditioning to JFNK produces robust and efficient solution updates for a number of multiphysics applications (fluids, phase transition, chemical transport)
- Reduce, reuse, recycle
  - Existing Picard solution method as preconditioner within new JFNK solver
  - As approximate update, use 1 step of Picard and FGMRES with a loose tolerance and Ifpack (Jacobi) preconditioner
  - Current solver can become preconditioner to more complete models coming down the road, e.g. full Stokes
- Next steps: Combination of physics-based preconditioning with multilevel methods (multigrid, Schwarz)
  - Enhanced efficiency for a given problem
  - More linear scaling than physics-based preconditioning alone

Trilinos Interface in CISM for velocity solve

- Using new Piro package as a user-friendly wrapper
  - calls nonlinear solvers, time integrators, continuation etc.
  - U/Q and optimization around your simulation, e.g. Dakota
- Implemented C++ interface layer to expose Trilinos functions
- Allows transfer to new finite element version of code, LIFE-V
- Configure options added
  `e.g. --with-trilinos` link to Trilinos libraries

Current Packages Being Used within Piro

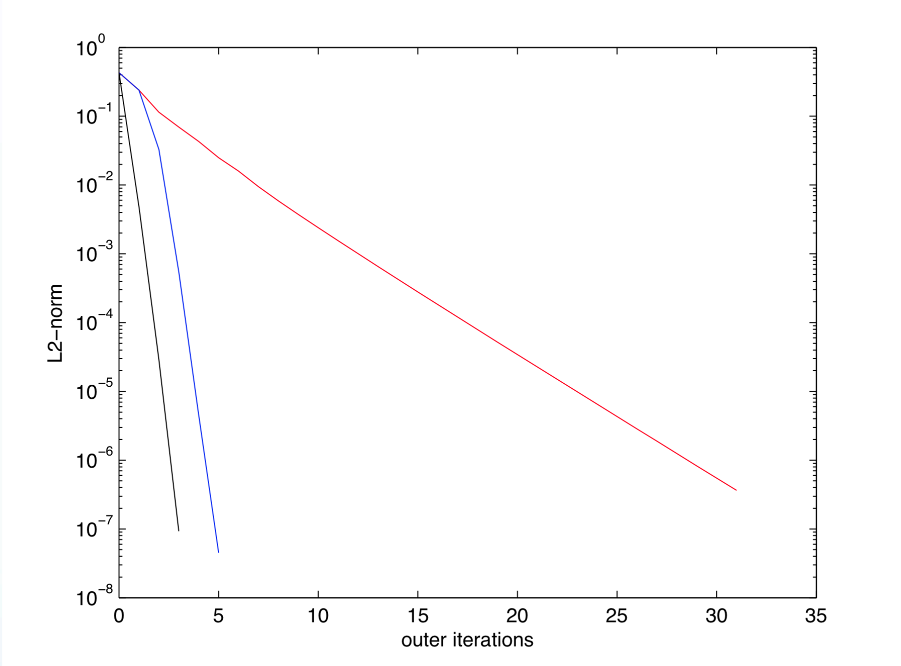
NOX: nonlinear solvers

Stratimikos: allows user to specify solver options at runtime in an XML file

Belos: linear solvers – FGMRES, can use GPU through tpetra

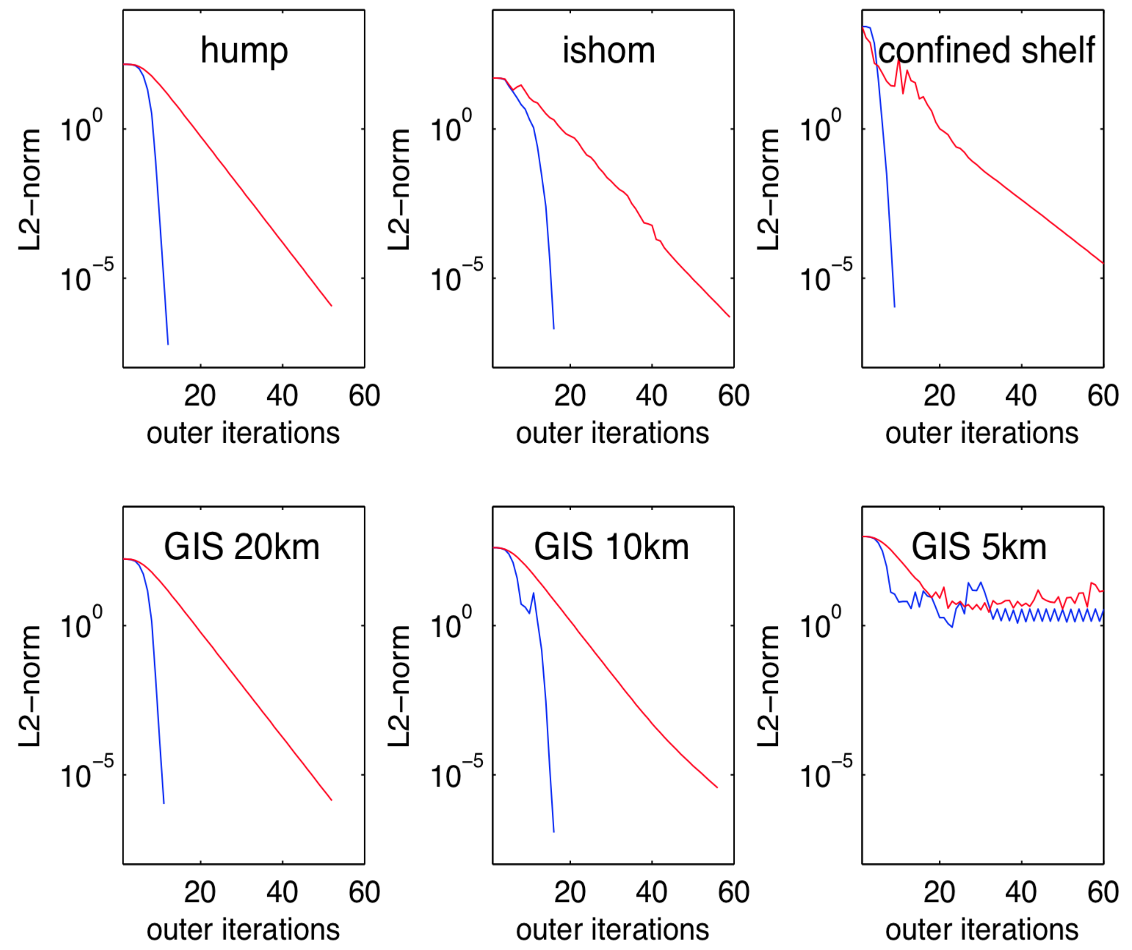
Jacobian-Free Newton-Krylov as a solver in Glimmer-CISM

- GMRES iterations are reduced with JFNK for a given level of tolerance
- For test cases evaluated, JFNK is ~2-3.5 times faster than Picard for a given # processors
- We use same tolerance for both solvers for testing, so accuracy is comparable. Results similar to original solver settings for test cases
- Picard as preconditioner works as designed, with more gains possible

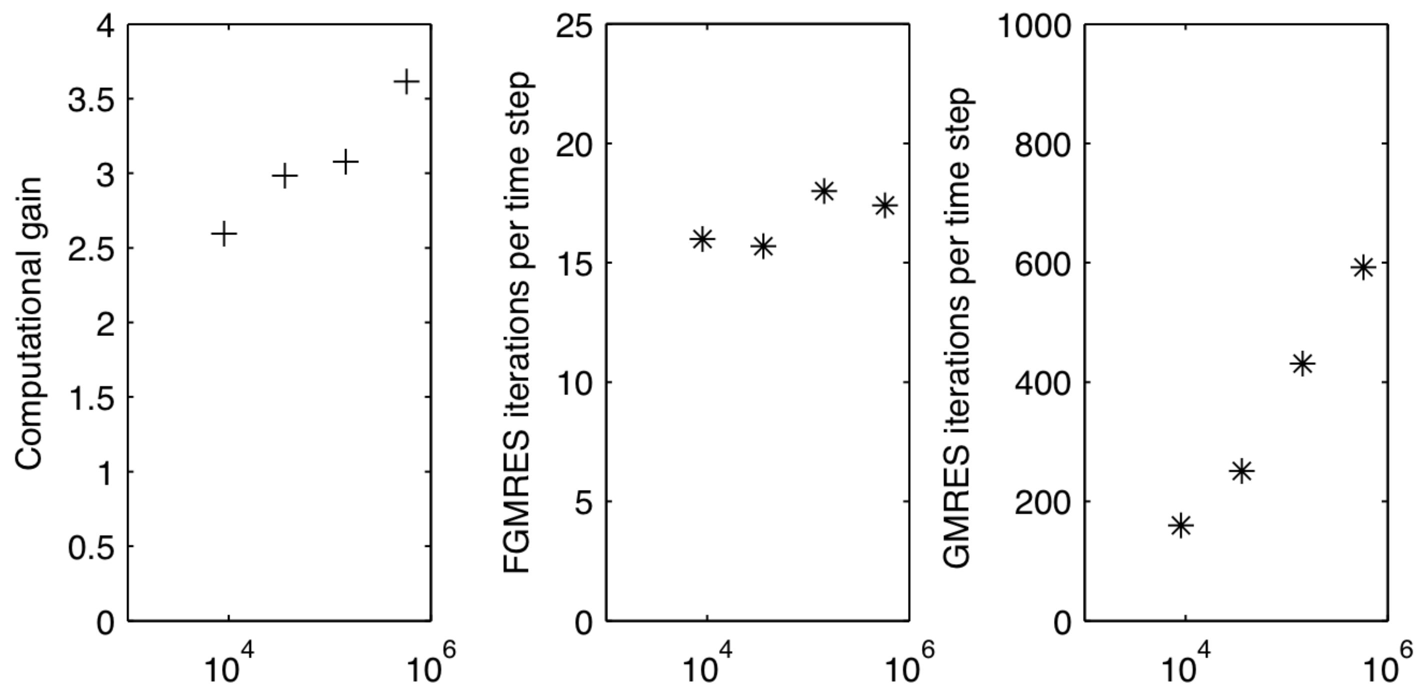


Greenland 10km test case, one time step

JFNK behavior for suite of test cases



Picard preconditioner: proof of principle

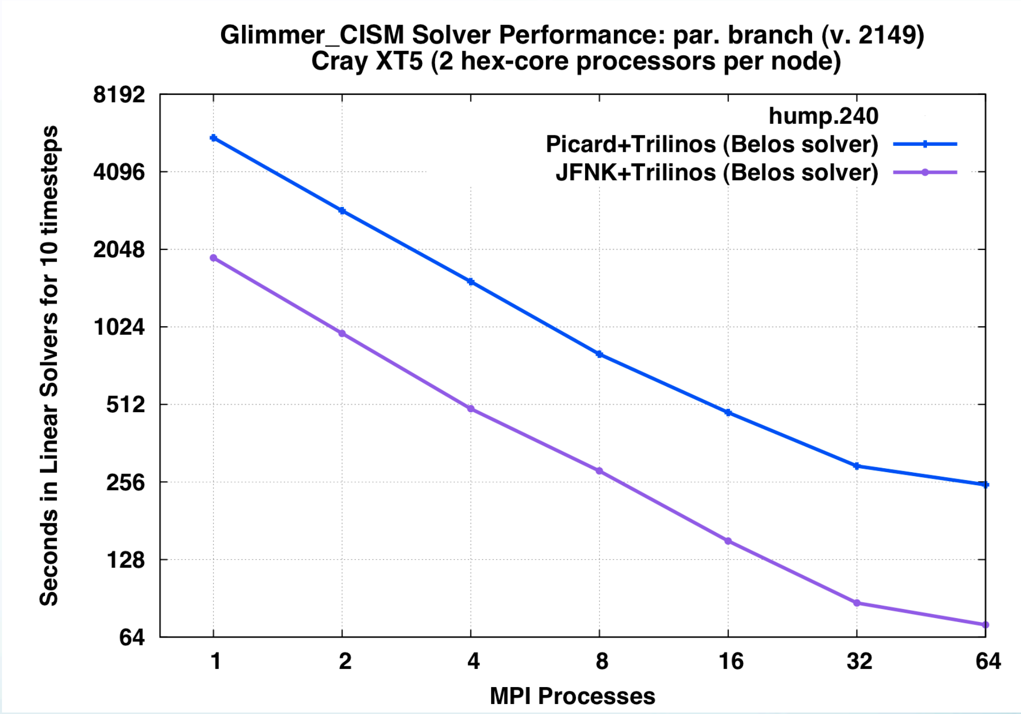


Relative efficiency improves with problem size, but is currently limited (speed and memory) by use of GMRES iterations in the preconditioner.

Lemieux, J.F., S. Price, K. Evans, D. Knoll, A. Salinger, D. Holland, T. Payne, submitted J. Comp. Phys. 12/10.

Performance behavior of Picard and JFNK with parallel linear solver

- Parallelize the solver first to get quick gains for modelers
- Trilinos solver within Picard and JFNK scale similarly
- Picard with parallel GMRES version of code is going into Glimmer-CISM trunk, then CESM this spring
- The full nonlinear JFNK solve is now performed through Trilinos NOX/Belos



* JFNK shown here is initial, hand rolled version

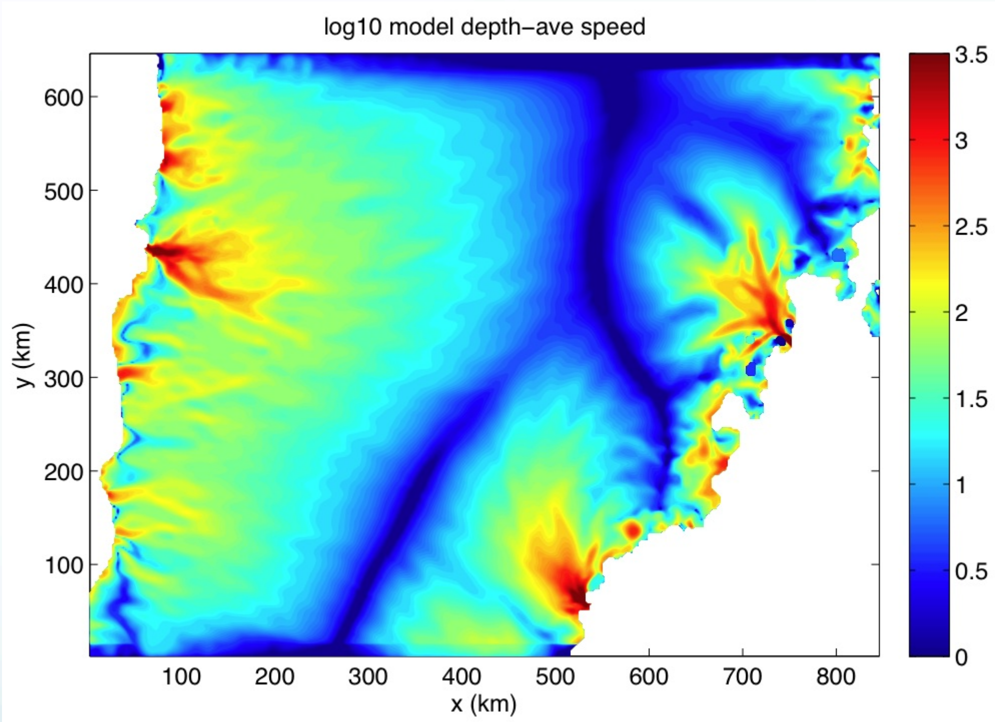
Recent ice sheet model results using parallel Trilinos solver in Glimmer-CISM

When the stress balance at the front of the outlet glaciers is changed in a Greenland configuration:

5km simulations produced ice thinning in a radially propagating direction upstream

2km resolution simulations show ice thinning occurring more along the ice channels as observed.

For this experiment: Faster model throughput allowed the finer resolution runs to occur, and higher resolutions produced more accurate simulations



Courtesy S. Price, LANL, presented at AGU Nat'l meeting, Dec. 2010

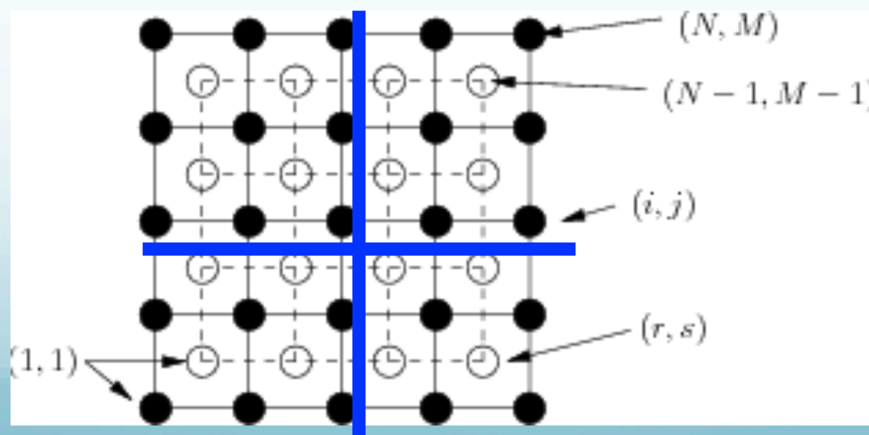
Distributed Parallel CISM

- Initial implementation
  - Ported to Jaguar
  - Distributed-memory parallelism
  - Ice Dome test cases
- Improve performance/memory use
  - Trilinos interface, parameters
  - OpenMP parallelism, Multicore
  - Parallel I/O
- Extend to temperature, vertical remapping (Bill Lipscomb)
- Next: Port to Intrepid at ALCF

*SEACISM has received an ALCC allocation from DOE-ASCR to develop Glimmer CISM at scale

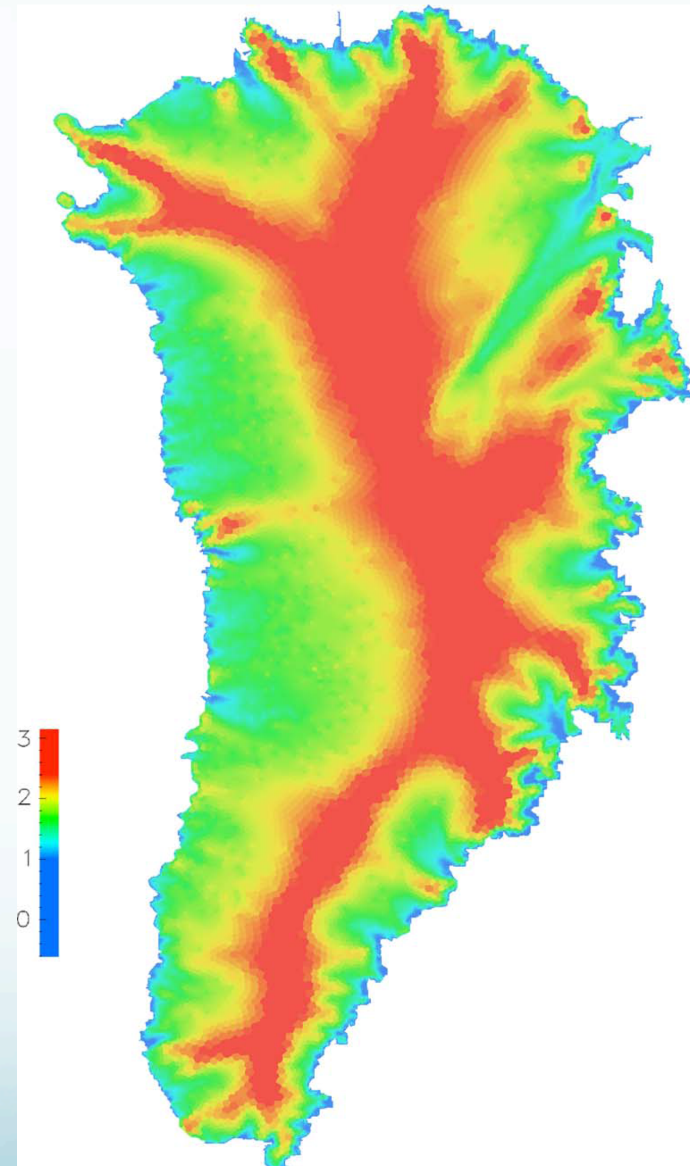
Test case in CISM using parallel code, Picard with Trilinos FGMRES

- 2D decomposition, selected automatically at runtime to minimize communication (i.e. square domains)
- Ice dome test cases and simplified Greenland (diagnostic temperature)
- Exhibits linear speed-up down to 15 grid points/row
- Currently extending to larger ice dome and other more complex test cases



Moving Toward Unstructured Meshes

- New finite element dynamical core with local mesh refinement
  - Collaboration with Florida State
  - MPAS – create mesh based on spherical centroidal Voronoi tessellations (SCVT)
  - Test suite of ice sheet equations to evaluate (SIA -> Stokes)
  - Targeting use on HPC platforms
  - Similar approach for mesh generation already being used ocean & sea ice components in CCSM



Ringler, T., L. Ju and M. Gunzburger, 2008, A multiresolution method for climate system modeling: application of spherical centroidal Voronoi tessellations, Ocean Dynamics, 58 (5-6), 475-498.

HO momentum equations: Finite Element approx.

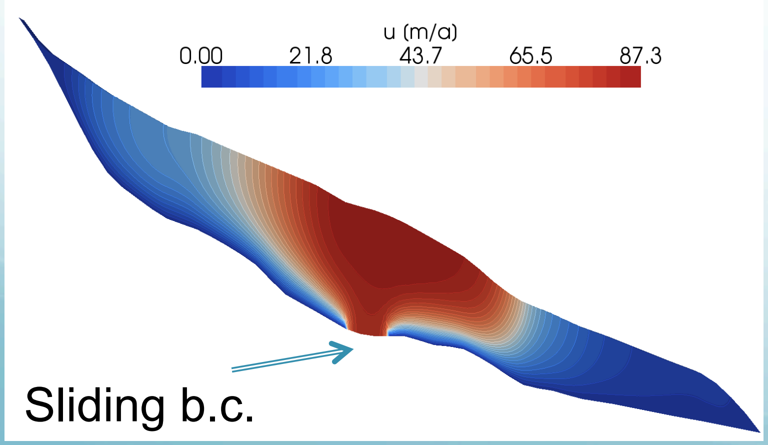
Finite elements are suitable and efficient for solving elliptic problems. Complex geometries and unstructured meshes are easily handled.

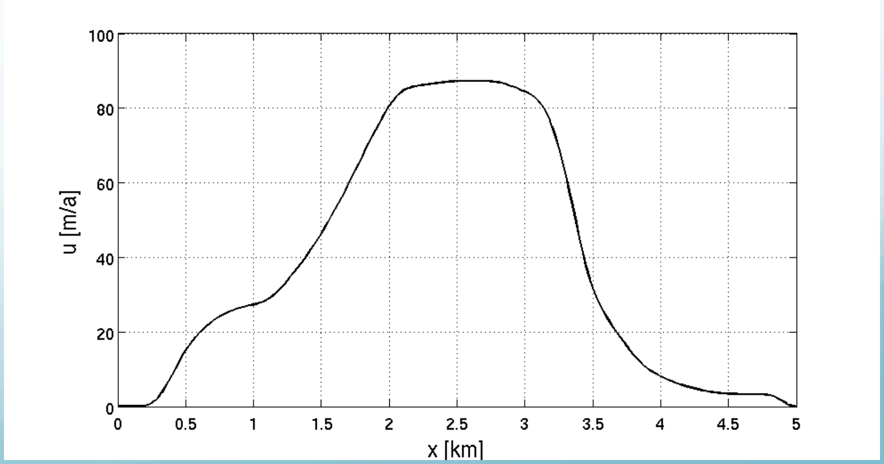
- Finite Elements are implemented using the C++ library LifeV[1]. Linear (P1) or quadratic (P2) finite elements on tetrahedra are used.
- The nonlinear system is solved using the Newton method implemented in NOX (Trilinos). The Jacobian matrix is computed at each iteration (assembly of the Jacobian matrix is expensive but highly parallelizable).

First results:

Glacier D'Arolla, test case E of ISMIP-HOM test suite.

Left: horizontal velocity. Right: horizontal velocity on the surface.





[1] www.lifev.org

Near term efforts within Glimmer-CISM

- **Dec 2011: AGU National Meeting** early production runs using SEA-Glimmer-CISM in CESM
- **June 2011: CESM developers meeting and ICIAM** coupled early version of Glimmer-CISM capability in the CESM
- **April 2011:** Frozen SEACISM code version ported to Glimmer-CISM trunk. Submit paper to IEEE special issue on the software frameworks with Glimmer-CISM
  - Extend parallelization JFNK, refactored temperature and vertical remap portions of code
  - Extend current preconditioner for JFNK solver to improve scalability
  - Optimize and tune existing code changes for robustness

Thank you. Cites mentioned in talk:

Price, S.F., A.J. Payne, I.M. Howat, B.E. Smith (2011) "Committed sea-level rise for the next century from Greenland ice sheet dynamics during the past decade," Proceedings of the National Academy of Sciences of the U.S., submitted.

Bamber et al. (2000) "An analysis of balance velocities over the Greenland ice sheet and comparison with synthetic aperture radar interferometry," Journal of Glaciology 46:67-74.

Lemieux, J.F., S. Price, K. Evans, D. Knoll, A. Salinger, D. Holland, T. Payne (2011) "Implementation of the Jacobian-free Newton-Krylov method for solving the first-order ice sheet momentum balance" J. Comp. Phys, submitted.

Schoof, C. and R. Hindmarsh (2010) "Thin-film flows with wall slip: An asymptotic analysis of high-order glacier from models," Q. J. Appl. Mech. Math. 63:73-114.

Questions?

Incorporate SEA-Solvers: Picard vs Newton

$$F'(x^{t+1})\delta x = -f(x^{t+1})$$

KEY: Picard is a simpler form of Newton

$$F'_p = A + \frac{1}{\delta t}F \qquad \text{Picard}$$

$$F'_{ij} = A_{ij} + \frac{1}{\delta t}F_{ij} + \sum_s \frac{\partial A_{is}}{\partial x_j}x_s + \frac{1}{\delta t}\sum_s \frac{\partial F_{is}}{\partial x_j}(x_s^{t+1} - x_s^t) + \frac{\partial b_i}{\partial x_j} \qquad \text{Newton}$$

Use Inexact Newton to solve for x:

solve top equation with preconditioned GMRES method

$x^{k+1} = x^k + \delta u^k$

if $||F(x^{k+1})|| < \gamma_{nl}||F(x^0)||$ stop

end do

- Use JFNK approach: $J(x^k)v \sim (F(x^k+\varepsilon v) - F(x)) / \varepsilon$
- Develop a physics-based preconditioner and combine with multilevel options available through Trilinos