# SEACISM: A Scalable, Efficient and Accurate Community Ice Sheet Model


Project Members:

Kate Evans, ORNL

Dana Knoll, Los Alamos

J.-F. Lemieux, New York U

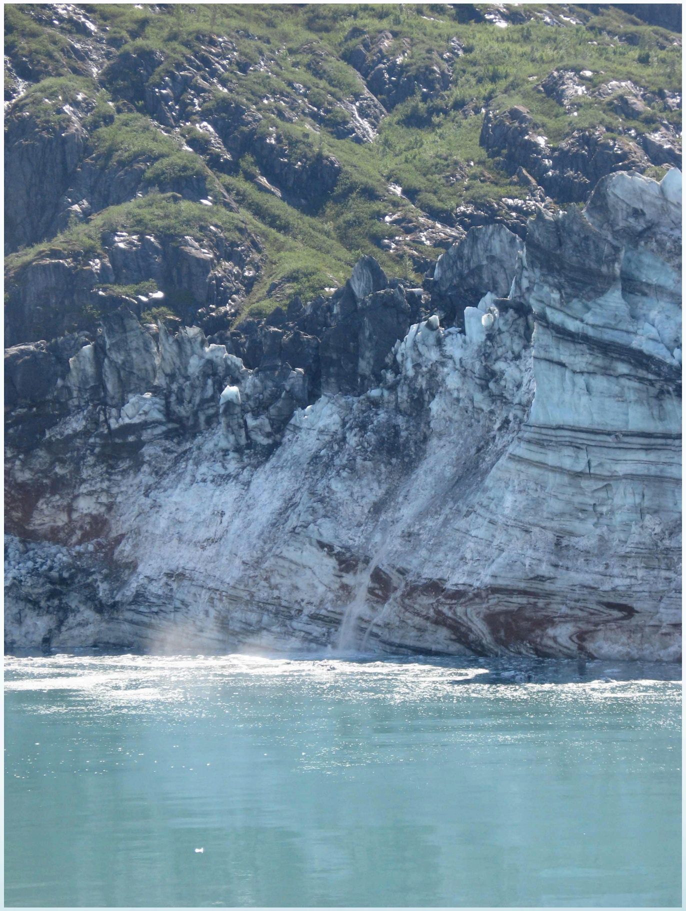
Jeff Nichols, ORNL

Andy Salinger, Sandia

Pat Worley, ORNL

Trey White*, NCAR

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

*SEACISM alumni

Oak Ridge National Laboratory

Managed by UT-Battelle for the Department of Energy

New York University

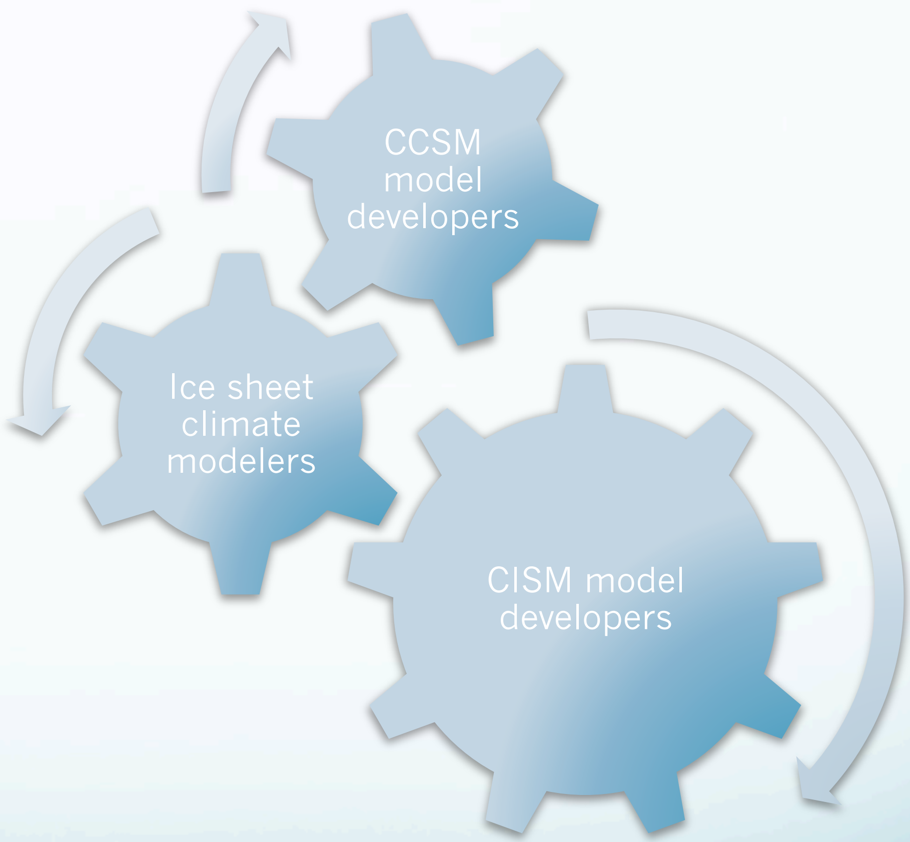
Los Alamos National Laboratory EST. 1943

Sandia National Laboratories

U.S. Department of Energy

# SEACISM Goal: Provide a state-of-the-art ice sheet model to the climate community

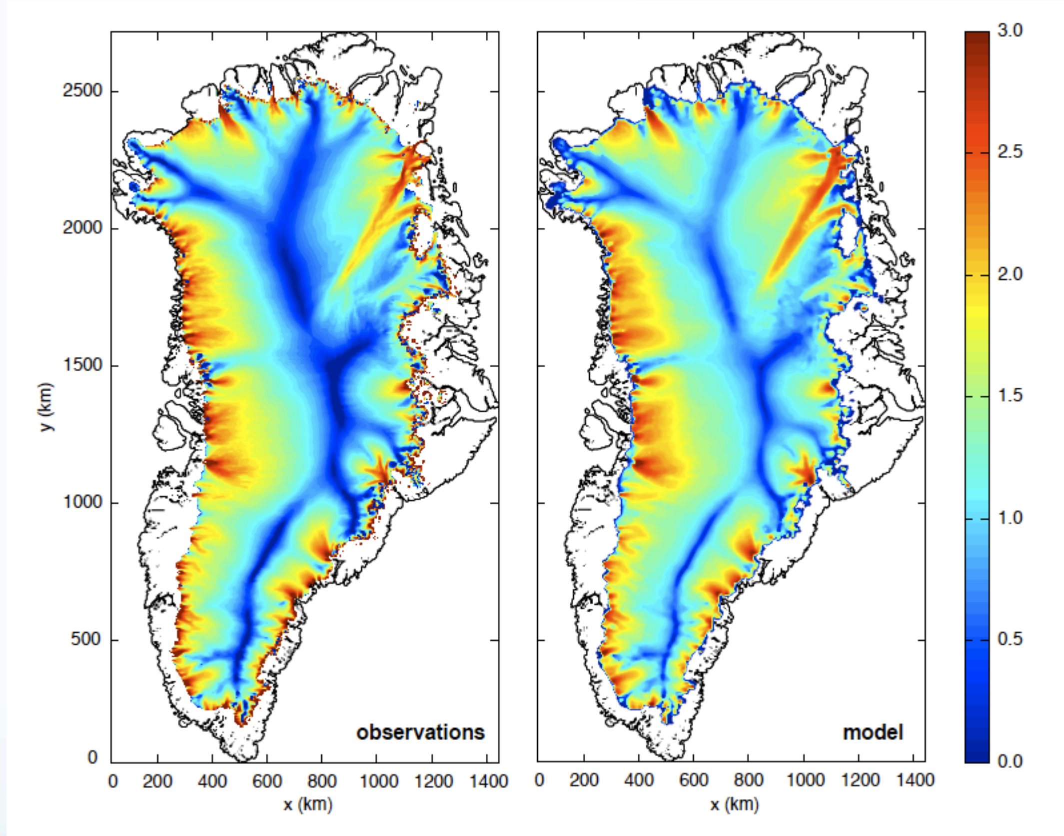
- Implement parallel, scalable capability as soon as possible to allow high-resolution simulations with code extensions with reasonable throughput and accuracy
- Maintain consistency and interaction with the production-level CCSM.
- Enable seamless inclusion of incremental developments such as new parameterizations and higher-order flow equations




EVENTUAL GOAL: coupled simulations with other climate components

# Glimmer-CISM: The Community Ice Sheet Model

- New version from which we will perform work
- 2 new tests using new physics of ice sheets now available
- Tuned, steady-state simulation using HO velocity solve takes on 5 km grid takes ~2 wks on 1 processor




1.5 million nodes.
Each iteration: 1-5 minutes
Iteration count: 100's

Left panel: Steady-state surface velocity (log of m/yr) based on modern-day observations.

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

[1]Bamber et al. (*J.Glac.*, **46**, 2000)

# Incorporate SEA-Solvers

Solve higher-order ice sheet momentum equations

- Currently Picard, within which GMRES is called to solve velocity components sequentially
- Use Inexact Newton to solve $F(u) = A(u)u - b = 0$ system of nonlinear equations

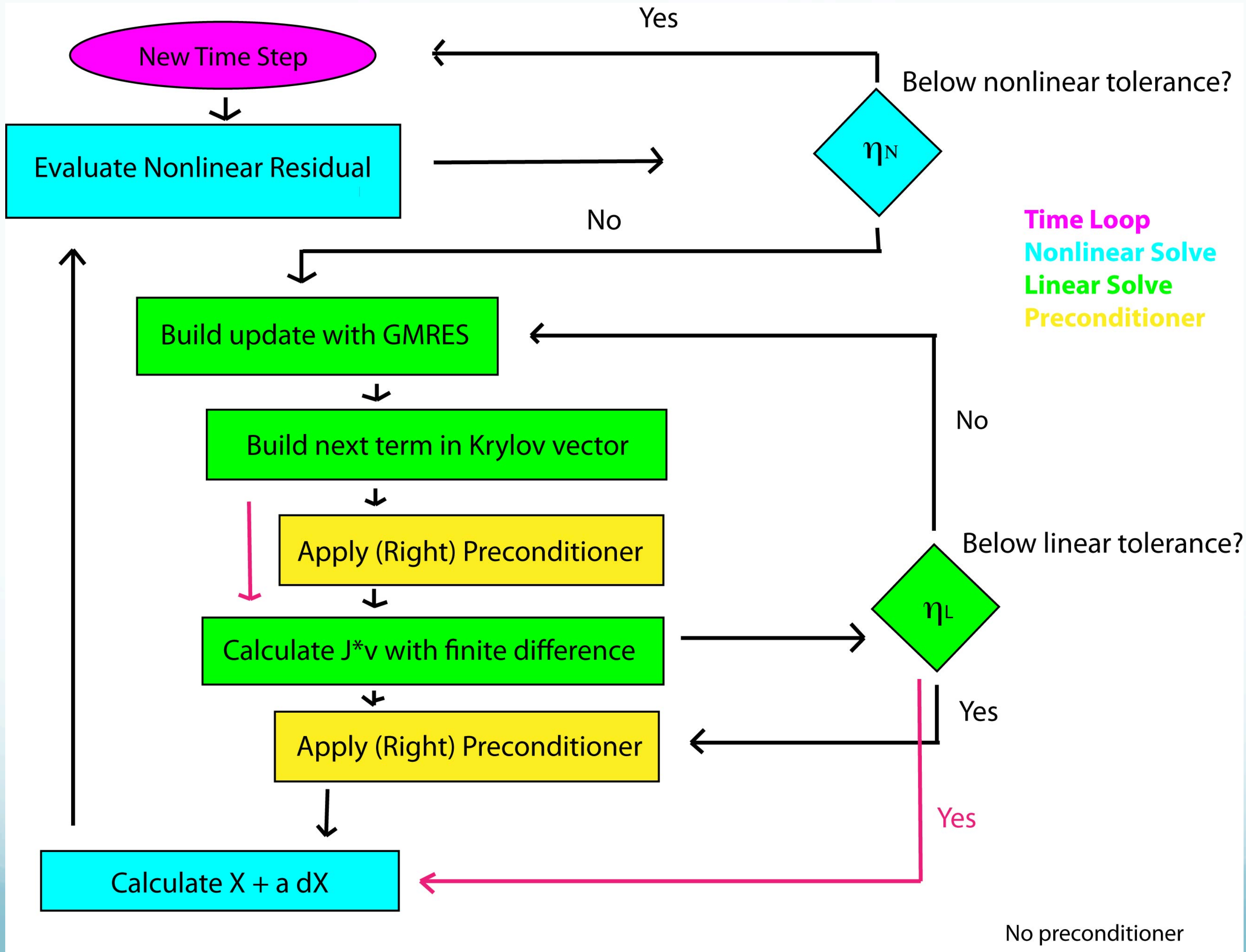
$u^0$: initial iterate
do $k = 1, k_{max}$
  solve $J(u^{k-1})\delta u^k = -F(u^{k-1})$ with preconditioned GMRES method
  $u^k = u^{k-1} + \delta u^k$
  if $||F(u^k)|| < \gamma_{nl}||F(u^0)||$ stop
end do

- Use JFNK approach: $J(u^{k-1})v \sim ( F(u^{k-1}+\varepsilon v) - F(u^{k-1}) ) / \varepsilon$
- Develop a physics-based preconditioner and combine with multilevel options available through Trilinos

# Preconditioner: the key to solution efficiency

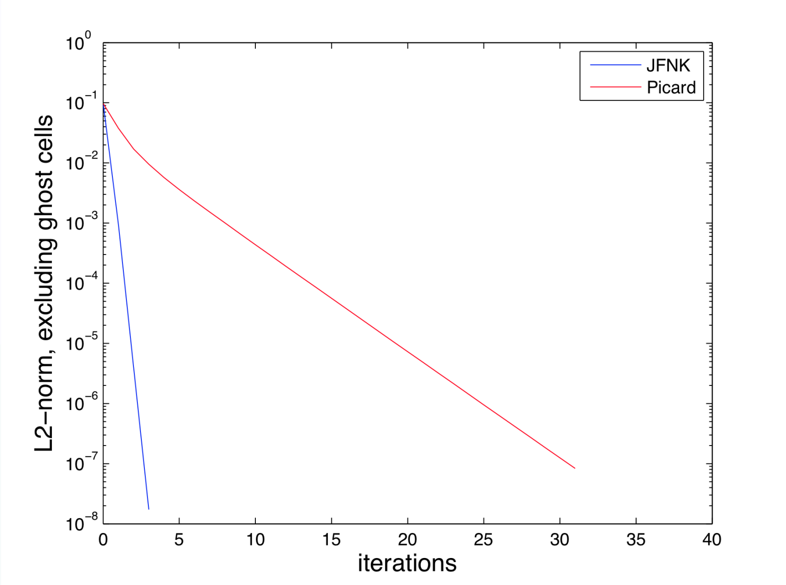
- Physics Based Preconditioning to JFNK produces robust and efficient solution updates for a number of multiphysics applications (fluids, phase transition, chemical transport)
- Combination of physics-based preconditioning with multilevel methods (multigrid, Schwarz) enhances efficiency
  - Enhanced efficiency for a given problem
  - More linear scaling than physics-based preconditioning alone
- Reduce, reuse, recycle
  - Existing Picard solution method as preconditioner within new JFNK solver
  - As Glimmer plans to extend equation set further in the future, existing balanced flow solution can a good physics based preconditioner
- Algebraic multigrid available through Trilinos's ML package to maximize scaling

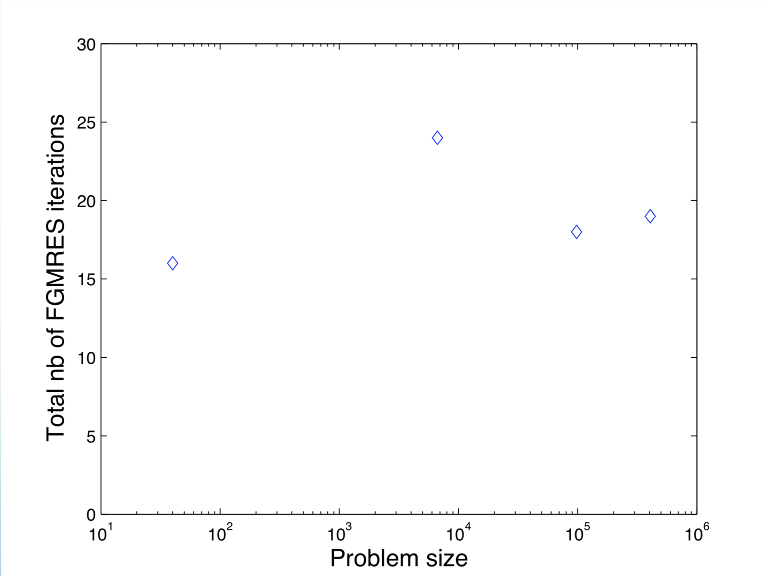
# JFNK solution method



# JFNK progress in Glimmer-CISM*

- Improved convergence with the GIS test case using JFNK with Picard as a preconditioner versus Picard as a solver
- # GMRES iterations are reduced by using JFNK, amount is tolerance dependent and will be explored to max performance
- Picard preconditioner produces rather flat growth of iterations with problem size for initial test cases
- JFNK used here will be replaced with Trilinos NOX JFNK, which will link to parallel code




* Hot off the presses, still validating

# Trilinos Interface in CISM

## Incorporating Trilinos

- Generalization of matrix type and solver calls to be changed in the code
  - Expansion of implementation by Jesse Johnson
  - Generic matrix derived type
  - Generic functions to access solvers

    e.g. sparse_preprocess
  - Current solver options:
    - SLAP, UMFPACK, Pardiso, Trilinos
  - Working on direct incorporation of Epetra matrix type
- Implemented C++ interface layer to expose Trilinos functions
- Configure options added

  `e.g. --with-trilinos` link to Trilinos libraries

## Current Packages Being Used

Epetra: data structures

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

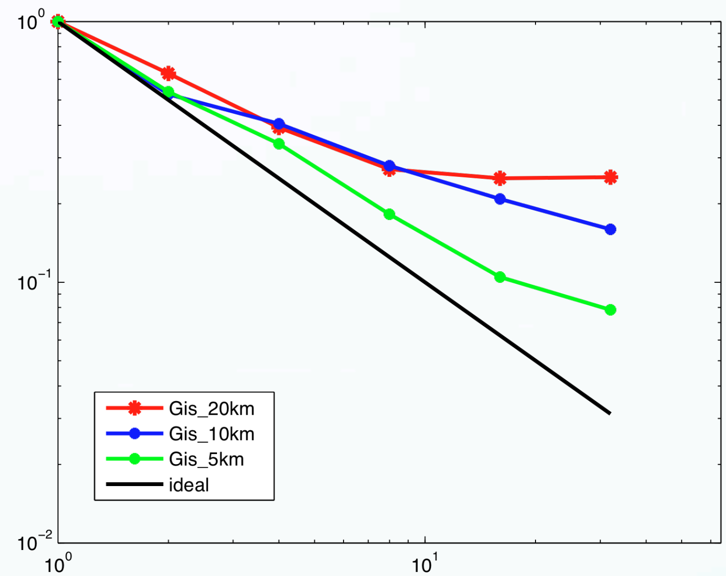
Belos: linear solvers - GMRES

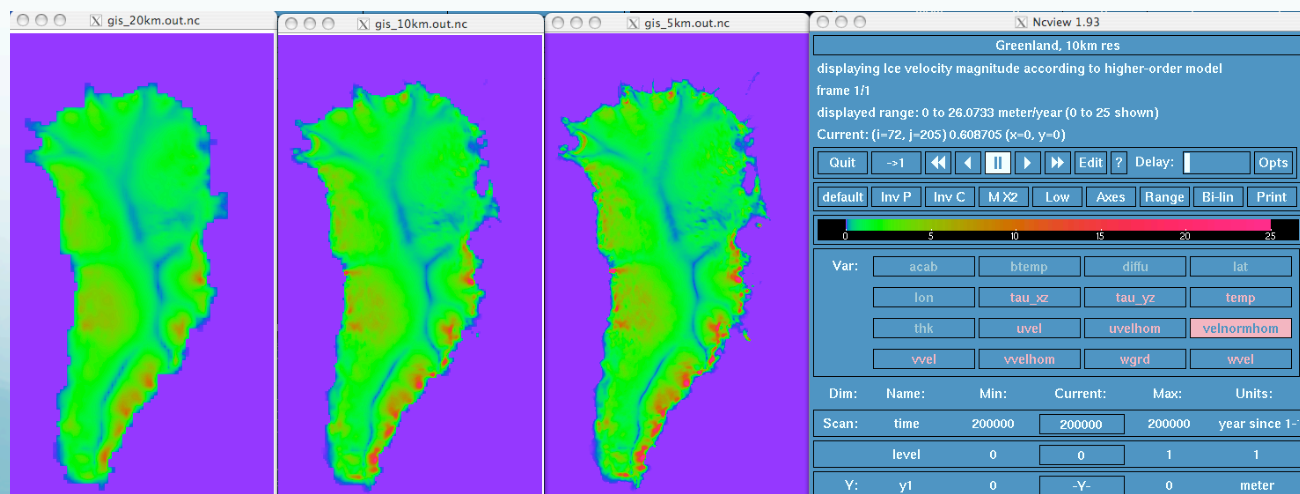
Ifpack: preconditioners – ILU

NOX: nonlinear solvers – implementation in progress

# Performance Analysis of "GIS" test case in CISM using Trilinos

- Test case of Greenland, key for evaluating numerical methods
- 20km, 10km, 5km resolutions run using Trilinos (GMRES for linear solve)
- As the number of grid points increases, total linear solve time decreases
- Scalability is approaching ideal



# Parallel CISM

- Initial implementation
  - Port to Jaguar
  - Distributed-memory parallelism
  - Get "tests/ho-other/hump.config" to work
- Target Greenland Ice Sheet
  - Extend parallel support as necessary
  - Analyze performance
- Tune performance
  - Trilinos interface, parameters
  - OpenMP parallelism
  - Parallel I/O

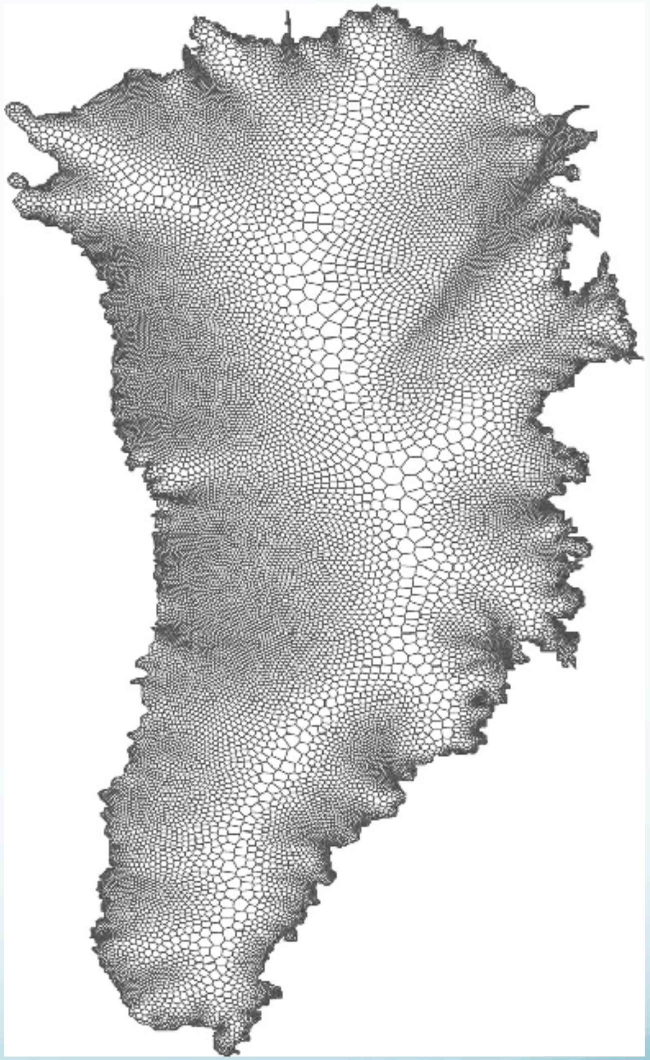

*SEACISM has received an ALCC allocation from DOE-ASCR to develop Glimmer CISM at scale

# Maximizing Performance

- Trilinos implementation
  - Form matrix structures once, avoid heaviest communication
  - Use Trilinos for full nonlinear solve
- OpenMP parallelism
  - Important for scaling on multi-core architectures
  - Soon to be supported in Trilinos
- Parallel I/O
  - Only need to modify the new module
  - Various options: PIO, NetCDF4, Adios, ...

# Moving Toward Unstructured Meshes

- MPAS – modeling processes across scales
  - New dynamical core with local mesh refinement
  - Collaboration with FSU
  - Spherical centroidal voronoi tessellations (SCVT)
  - Utilizing Trilinos & solvers developed in structured grid code
  - Targeting use on HPC platforms (Roadrunner, BlueGene, Jaguar)
  - Already being developed for ocean & sea ice componets in CCSM



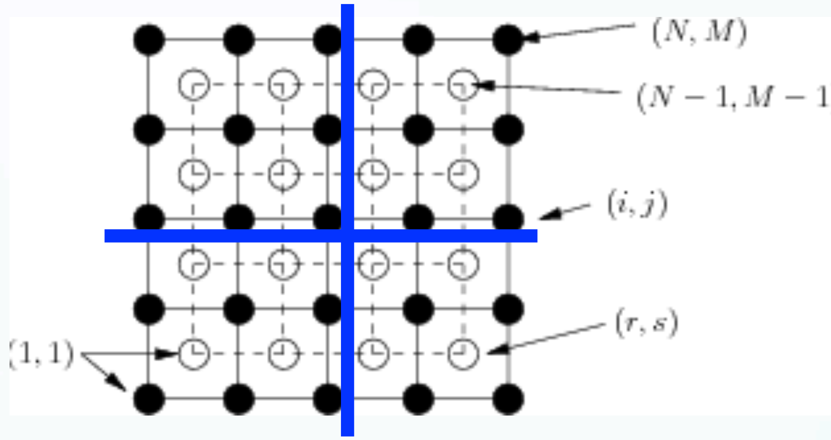
Ringler, T., L. Ju and M. Gunzburger, 2008, A multiresolution method for climate system modeling: application of spherical centroidal Voronoi tessellations, Ocean Dynamics, 58 (5-6), 475-498.

# Questions?

**Best tip for climate model development thanks to NVIDIA:**
**“Your code would run a lot faster if it didn’t have so much I/O”**

# 2D Decomposition

- Selected automatically at runtime based on number of MPI tasks
- Uses all tasks
- Gives each task as square a piece as possible
- Mostly nearest-neighbor "halo" exchanges




http://glimmer-cism.berlios.de/docs/current/manual/num/figs/grid.png

- Carefully redefines "ewn" and "nsn" so most loops work without modification