Berkeley-ISICLES (BISICLES): High Performance Adaptive Algorithms For Ice Sheet Modeling

Dan Martin

Lawrence Berkeley National Laboratory

January 12, 2011

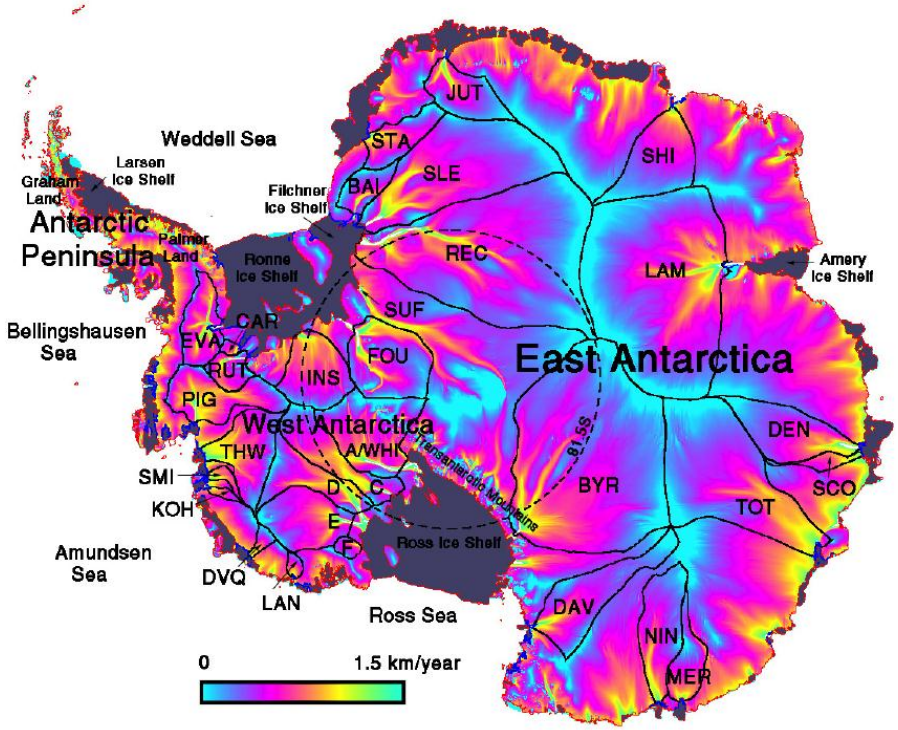
BISICLES - Goal

Goal: Build a parallel, adaptive ice-sheet model

- Localized regions where high resolution needed to accurately resolve ice-sheet dynamics (500 m or better at grounding lines)
- Large regions where such high resolution is unnecessary (e.g. East Antarctica)
- Problem is well-suited for adaptive mesh refinement (AMR)
- Want good parallel efficiency
- Need good solver performance

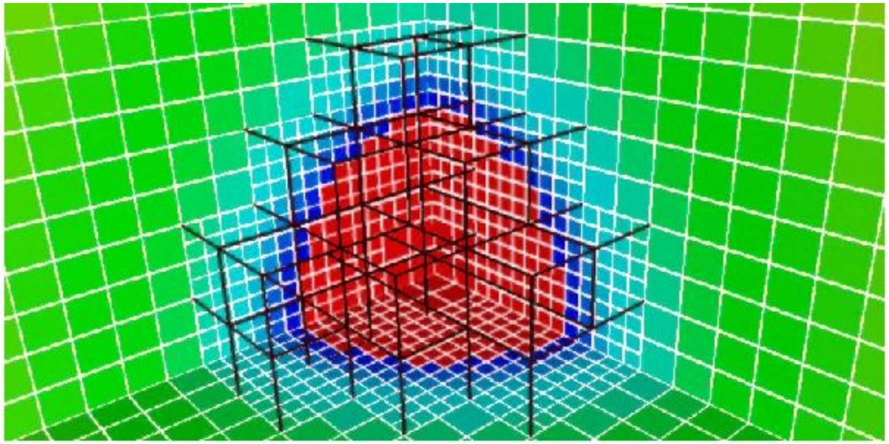
Much higher resolution (1 km versus 5 km) required in regions of high velocity (yellow → green).

[Rignot & Thomas, 2002]



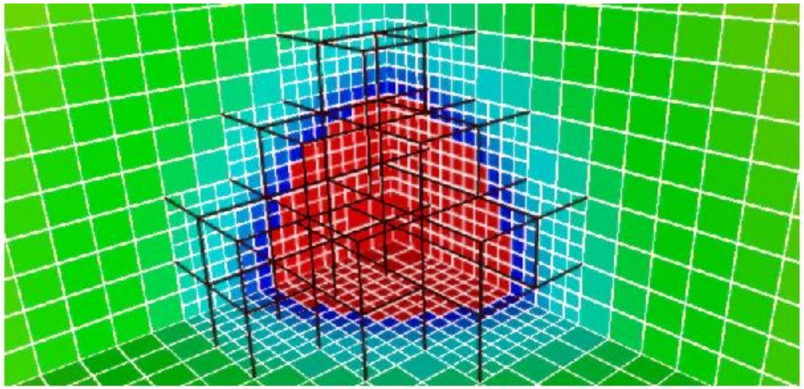
BISICLES – Approaches

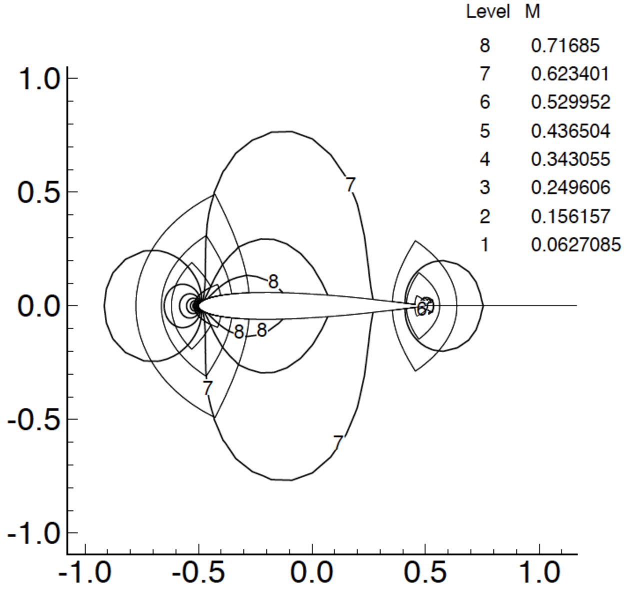
- Develop an efficient parallel implementation of Glimmer-CISM by
  - Incorporating structured-grid AMR using the Chombo framework to increase resolution in regions where changes are more rapid,
  - Exploring new discretizations and formulations where appropriate (L1L2)
  - Improving performance and convergence of linear and nonlinear solvers, and
  - Deploying auto-tuning techniques to improve performance of key computational kernels.



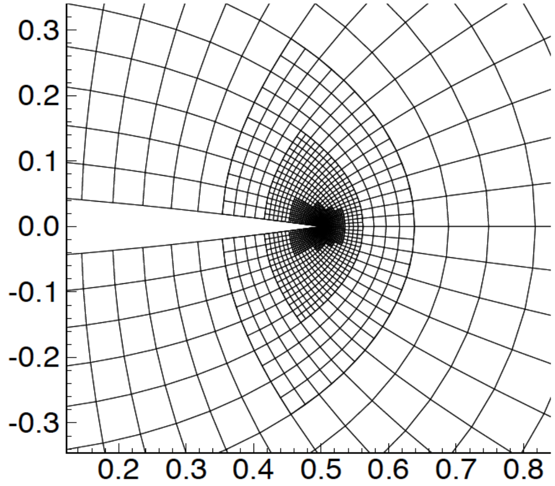
Block-Structured Local Refinement

- Refined regions are organized into rectangular patches.





- Refinement in time as well as in space for time-dependent problems.
- *Algorithmic advantages:*
  - *Build on mature structured-grid discretization methods.*
  - *Low overhead due to irregular data structures, relative to single structured-grid algorithm.*



BISICLES Project Outline

- Joint work involving LBNL and LANL
  - LBNL: Esmond Ng (PI), Dan Martin (AMR), Woo-Sun Yang (Performance Optimization), Sam Williams (Autotuning)
  - LANL: Bill Lipscomb (co-PI), Doug Ranken (software support)
- Collaboration with Tony Payne (one of the original authors of Glimmer) and Stephen Cornford (Univ of Bristol, UK)

- Build AMR implementation of Glimmer-CISM
- Extensions to existing Chombo infrastructure added as needed
- Autotuning techniques deployed as components are developed
- Linear/nonlinear solver improvements

- Coupling with CESM using existing Glimmer-CISM interface and by developing new interfaces as needed

Models and Approximations

❑ Full-Stokes

- Best fidelity to ice sheet dynamics
- Computationally expensive (full 3D coupled nonlinear elliptic equations)

❑ Approximate Stokes

- Use scaling arguments to produce simpler set of equations
- Common expansion is in ratio of vertical to horizontal length scales ($\varepsilon = \frac{[h]}{[l]}$)
- E.g. Blatter-Pattyn (most common “higher-order” model), accurate to O(ε^2)
- Still 3D, but solve simplified elliptic system (e.g. 2 coupled equations)

❑ Depth-integrated

- Special case of approximate Stokes with 2D equation set (“Shelfy-stream”)
- Easiest to work with computationally
- Generally less accurate

"L1L2" Model (Schoof and Hindmarsh, 2010)

- Asymptotic expansion in 2 flow parameters:
  - ε -- ratio of length scales $\frac{[h]}{[x]}$
  - λ – ratio of shear to normal stresses $\frac{[\tau_{shear}]}{[\tau_{normal}]}$
    - Large λ: shear-dominated flow
    - Small λ: sliding-dominated flow
- Blatter-Pattyn approximates full-Stokes to $O(\varepsilon^2)$ for all λ regimes

- Asymptotic expansion: (e.g. $u(x,z) = u_0 + \varepsilon u_1 + O(\varepsilon^2)$)
  - Leading order velocity term: $u_0 = u_0(x)$ **(no vertical dependence)**
  - Don't need shear stresses to $O(\varepsilon^2)$ to compute velocity to $O(\varepsilon^2)$
  - Provides basis for depth-integrated approach

"L1L2" Model (Schoof and Hindmarsh, 2010), cont.

- Use this result to construct a computationally efficient scheme:

1. Approximate constitutive relation relating $grad(u)$ and stress field τ with one relating $grad(u|_{z=b})$, vertical shear stresses τ_{xz} and τ_{xz} given by the SIA / lubrication approximation and other components $\tau_{xx}(x,y,z)$, $\tau_{xy}(x,y,z)$, etc

2. leads to an effective viscosity $\mu(x,y,z)$ which depends only on $grad(u|_{z=b})$ and $grad(z_s)$, ice thickness, etc

3. Momentum equation can then be integrated vertically, giving a nonlinear, 2D, elliptic equation for $u|_{z=b}(x,y)$

4. $u(x,y,z)$ can be reconstructed from $u|_{z=b}(x,y)$

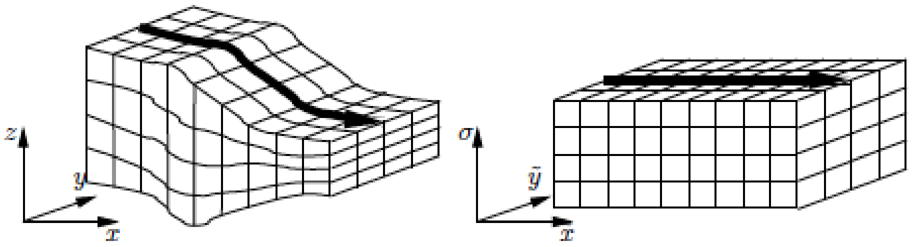
"L1L2" Model (Schoof and Hindmarsh, 2010), cont.

- Similar formal accuracy to Blatter-Pattyn $O(\varepsilon^2)$
  - Recovers proper fast- and slow-sliding limits:
    - SIA $(1 \ll \lambda \leq \varepsilon^{-1/n})$ -- accurate to $O(\varepsilon^2 \lambda^{n-2})$
    - SSA $(\varepsilon \leq \lambda \leq 1)$ - accurate to $O(\varepsilon^2)$

- Computationally **much** less expensive -- enables fully 2D vertically integrated discretizations.

Discretizations

- Baseline model is the one used in Glimmer-CISM:
  - Logically-rectangular grid, obtained from a time-dependent uniform mapping.
  - 2D equation for ice thickness, coupled with 2D steady elliptic equation for the horizontal velocity components. The vertical velocity is obtained from the assumption of incompressibility.
  - Advection-diffusion equation for temperature.



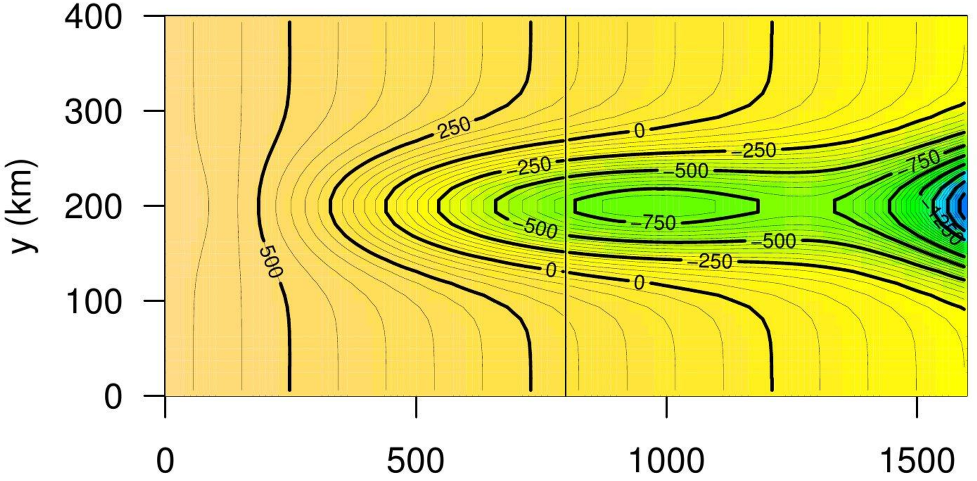
$$\frac{\partial H}{\partial t} = b - \nabla \cdot H\bar{\mathbf{u}}$$

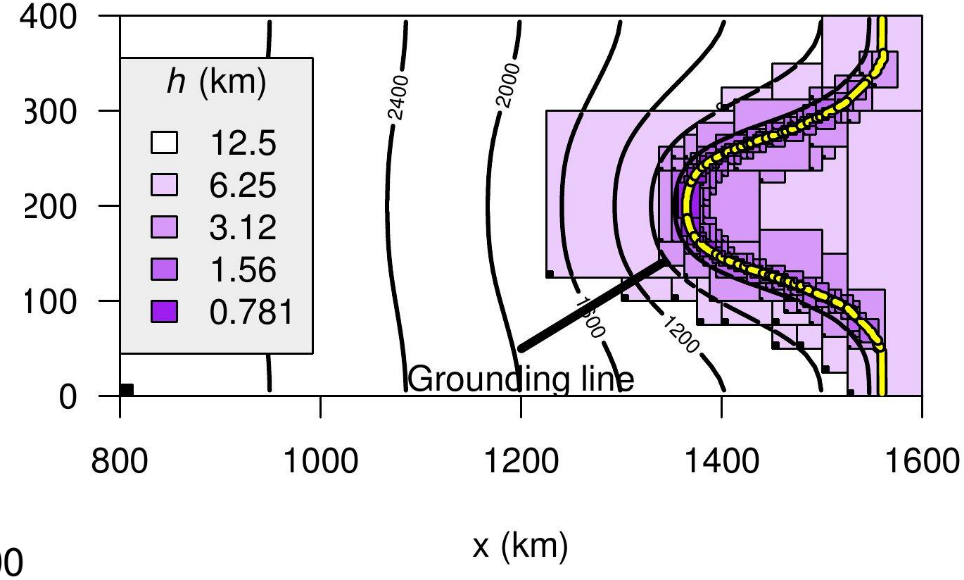
$$\frac{\partial T}{\partial t} = \frac{k}{\rho c}\nabla^2 T - \mathbf{u}\cdot\nabla T + \frac{\Phi}{\rho c} - w\frac{\partial T}{\partial z}$$

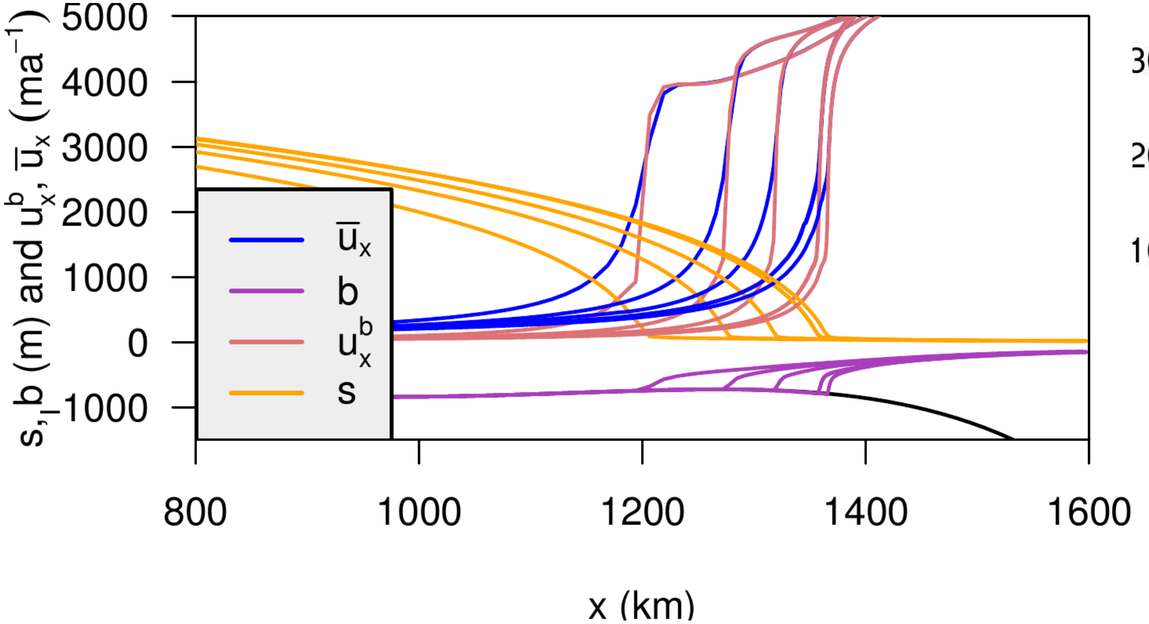
- Use of Finite-volume discretizations (vs. Finite-difference discretizations) simplifies implementation of local refinement.
- Software implementation based on constructing and extending existing solvers using the Chombo libraries.

BISICLES results – Grounding line study

- Bedrock topography based on Katz and Worster (2010)
- Evolve initially uniform-thickness ice to steady state
- Repeatedly add refinement and evolve to steady state
- G.L. advances with finer resolution
- Appear to need better than 1 km

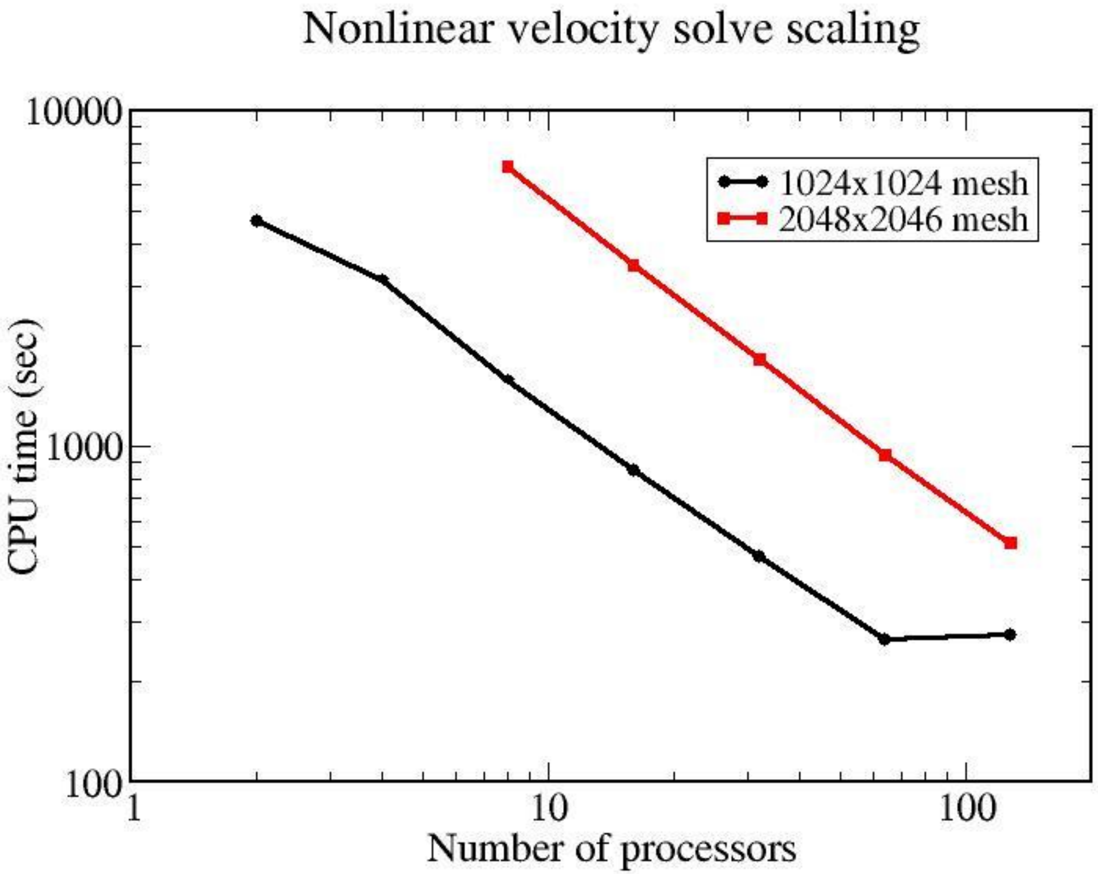






BISICLES -- Scaling

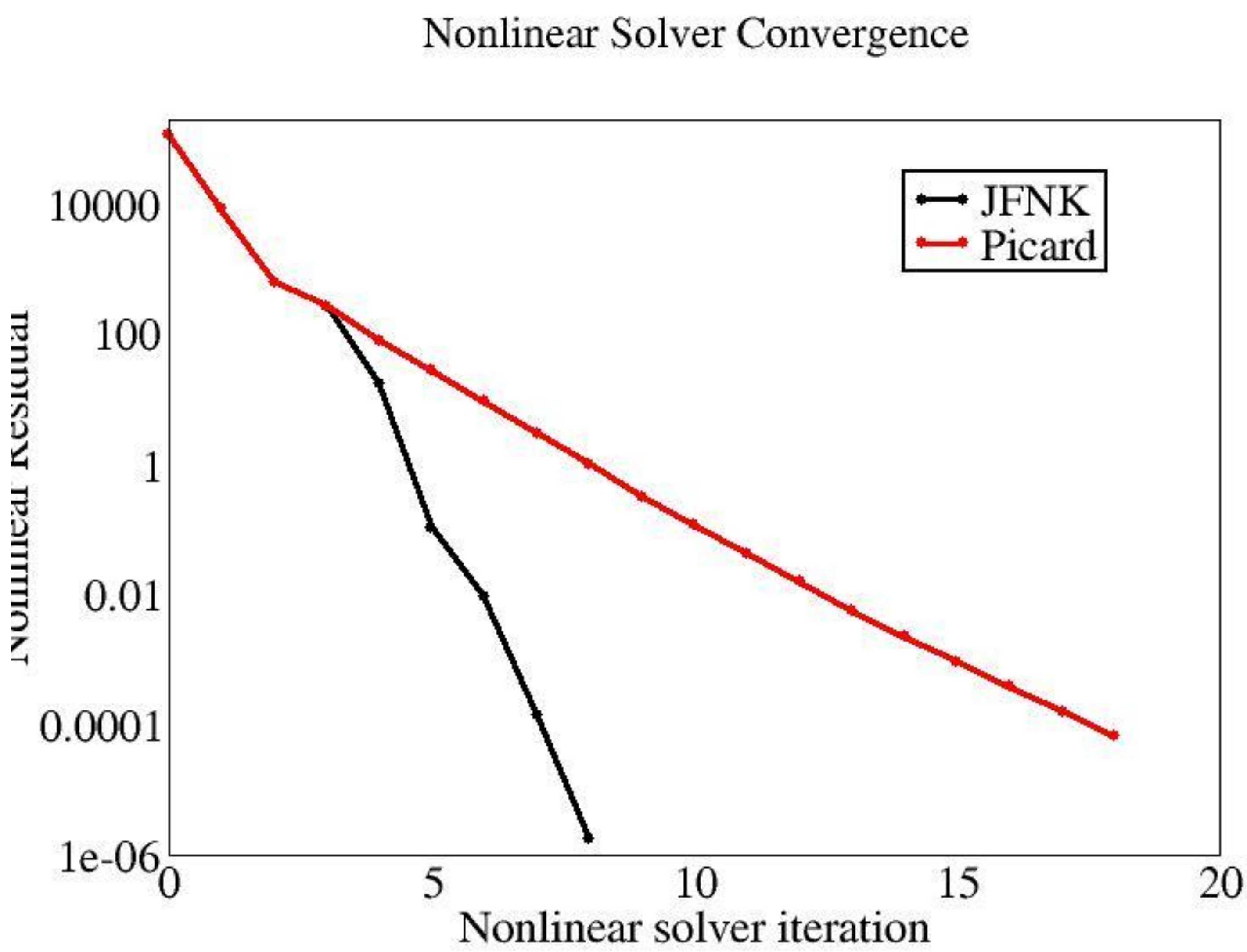
Initial tests show good strong scaling to at least 128 processors for nonlinear velocity solve (L1L2 approximation):



Nonlinear Solver Improvements

- Most computational effort spent in nonlinear ice velocity solve.
- Picard iteration:
  - Robust
  - Simple to implement
  - Slow (but steady) convergence
- Jacobian-free Newton-Krylov (JFNK):
  - More complex to implement
  - Works best with decent initial guess
  - Rapid convergence
  - Well-suited for Chombo AMR elliptic solvers
- Approach – use Picard iteration initially, then switch to JFNK when convergence slows

Nonlinear Solver Improvements (cont)

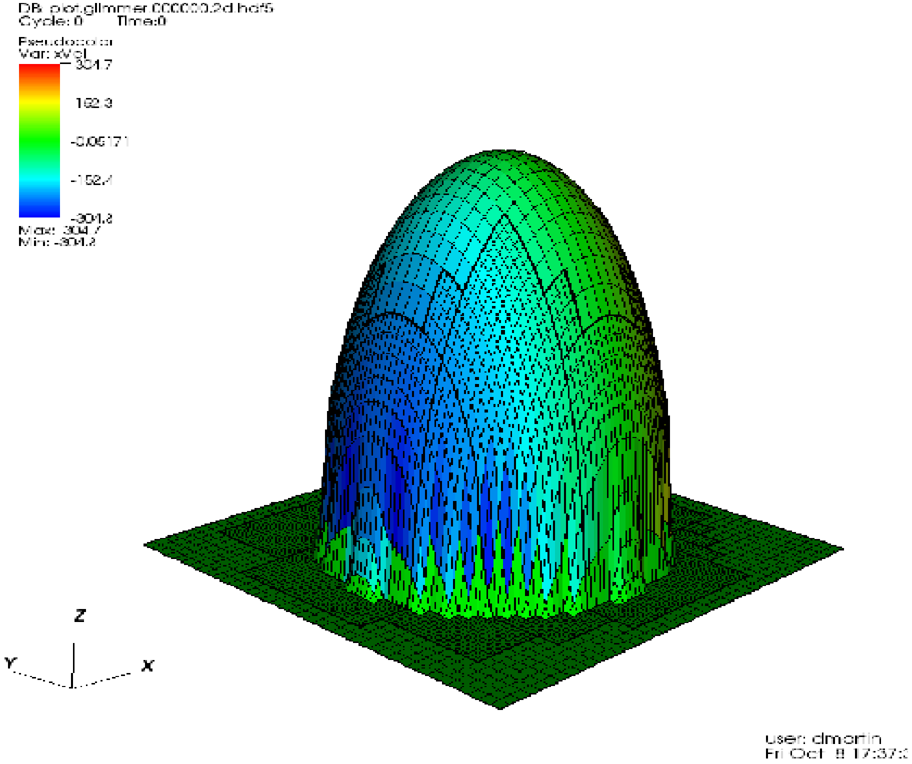


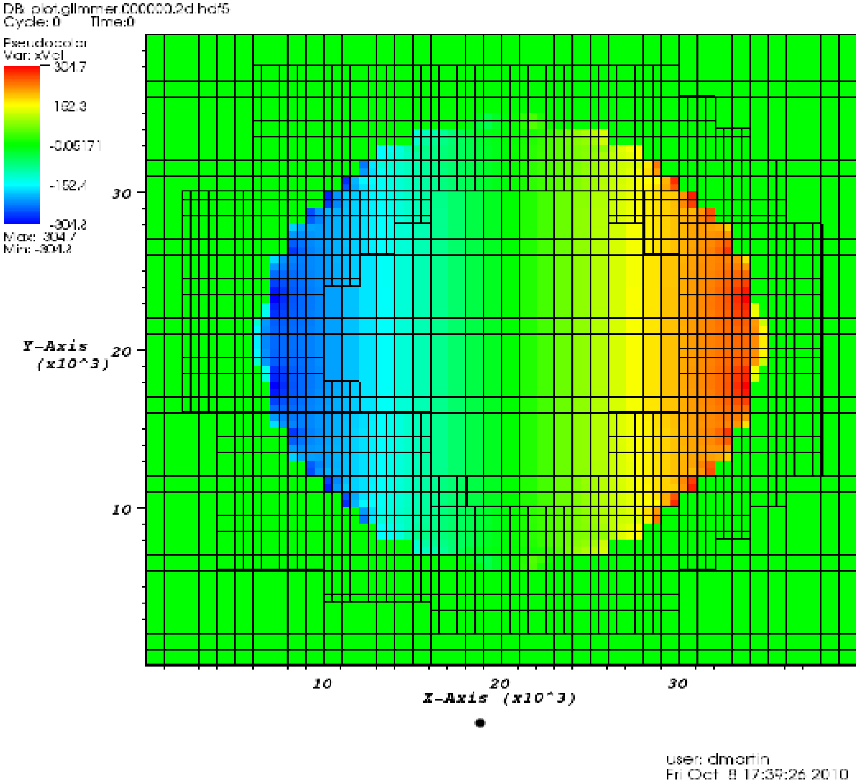
Interface with Glimmer-CISM

- Glimmer-CISM has coupler to CESM, additional physics
  - Well-documented and widely accepted
- Our approach – couple to Glimmer-CISM code as an alternate "dynamical core"
  - Allows leveraging existing capabilities
  - Use the same coupler to CESM
  - BISICLES code sets up within Glimmer-CISM and maintains its own storage, etc.
  - Communicates through defined interface layer
  - Instant access to a wide variety of test problems
  - Interface development almost complete
  - Part of larger alternative "dycore" discussion for Glimmer-CISM

"Hump" test problem

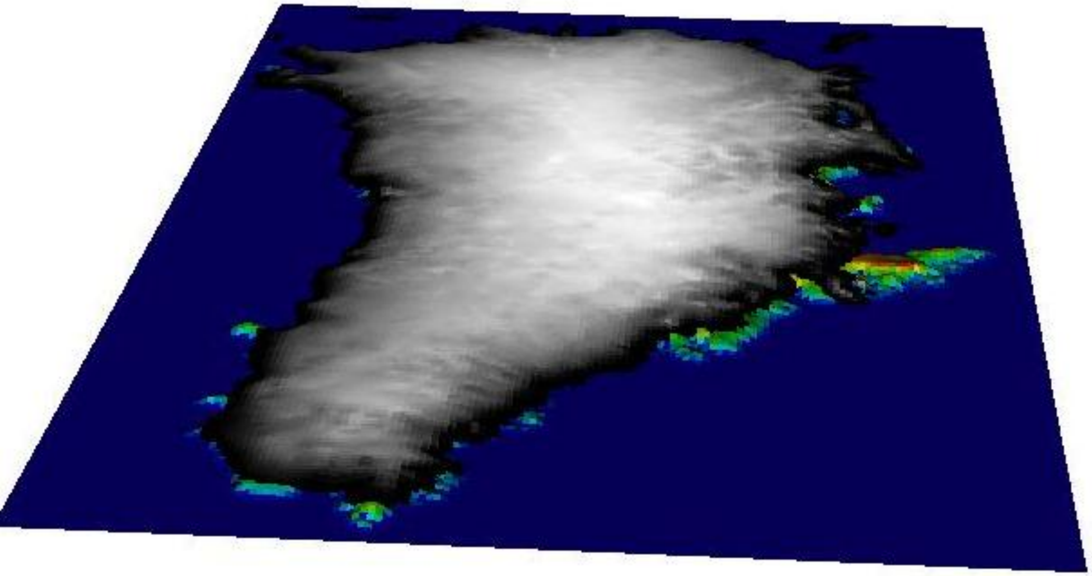
- Standard test problem -- isolated "blob" of ice on level ground

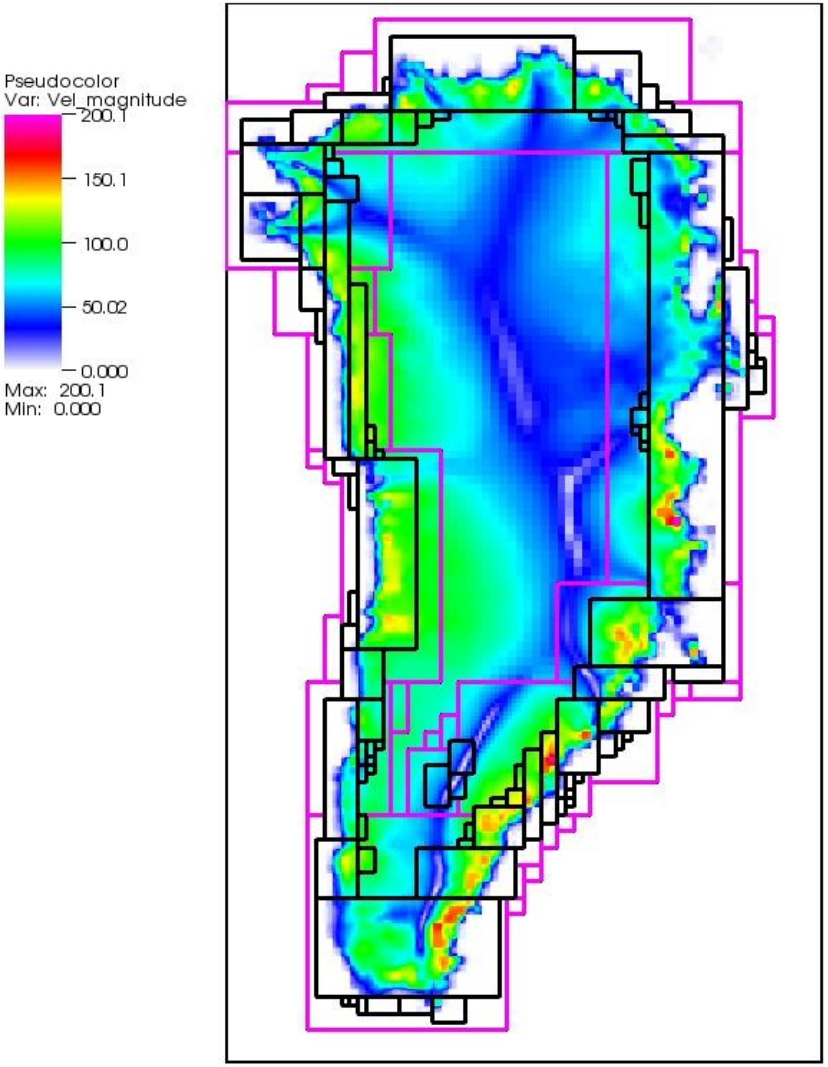




Greenland test problem

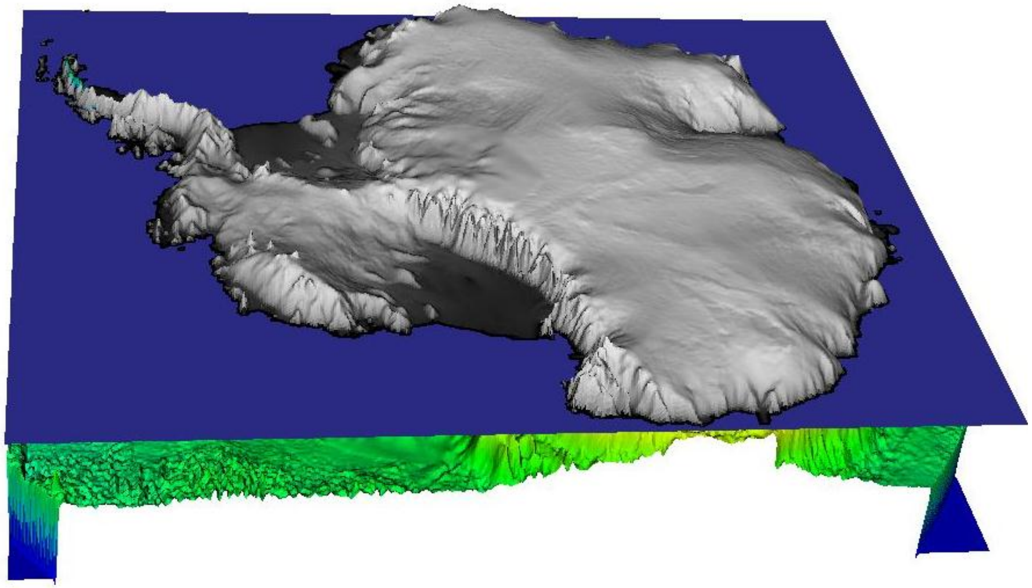
- Problem setup from Steve Price (LANL)
- Need to process initial condition somewhat in order for solver to converge (still work in progress)

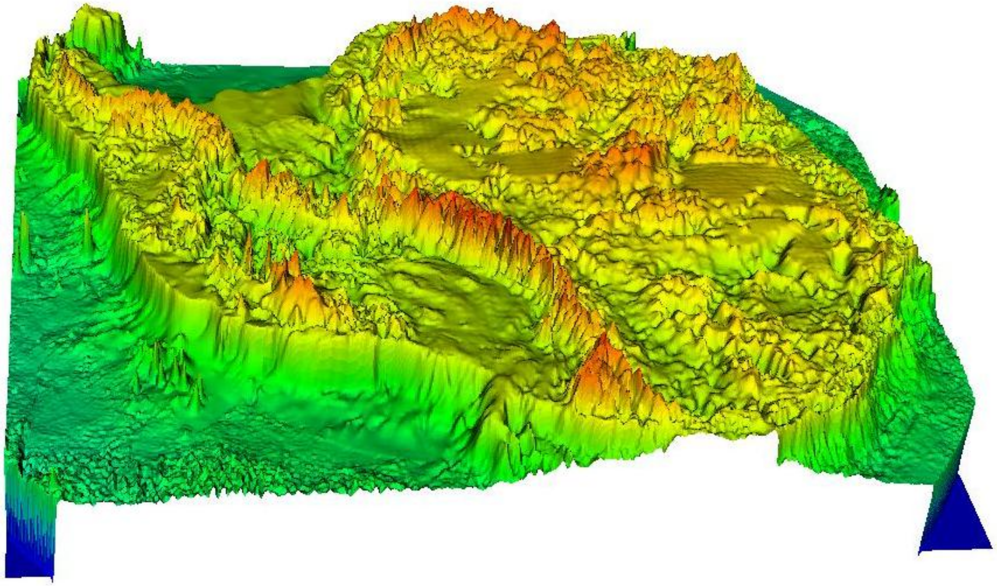




Antarctica

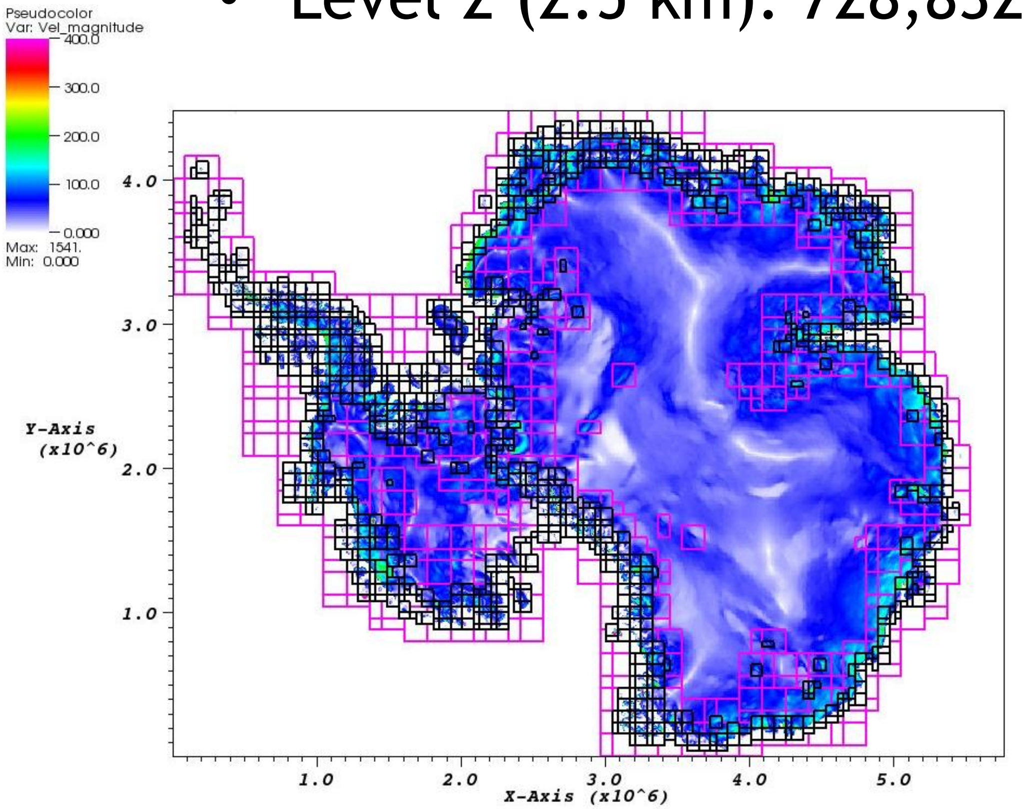
Uses new “model-friendly” problem setup
(Le Brocq, Payne, Vieli (2010))

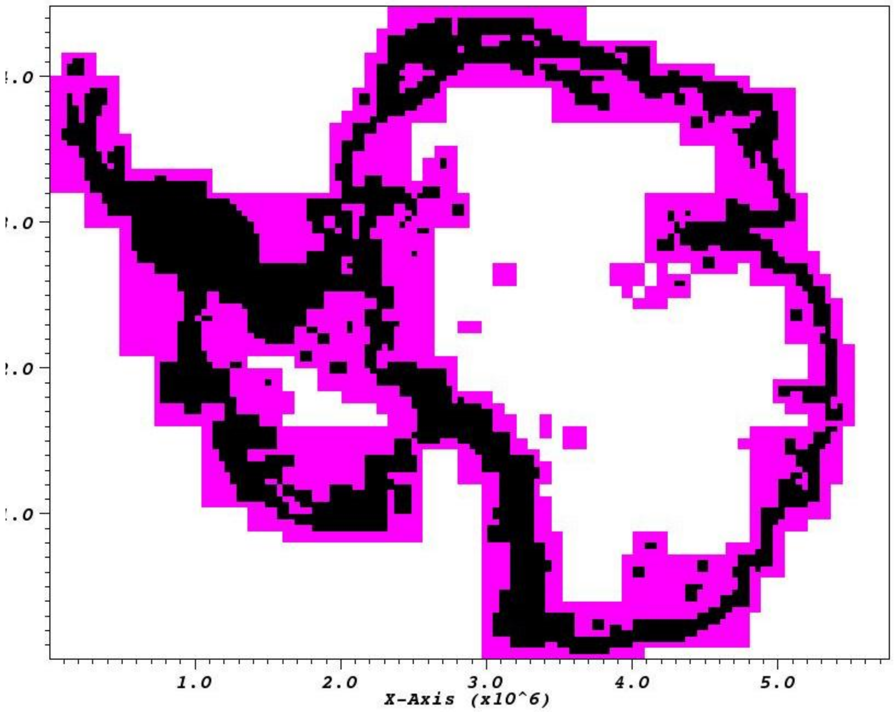




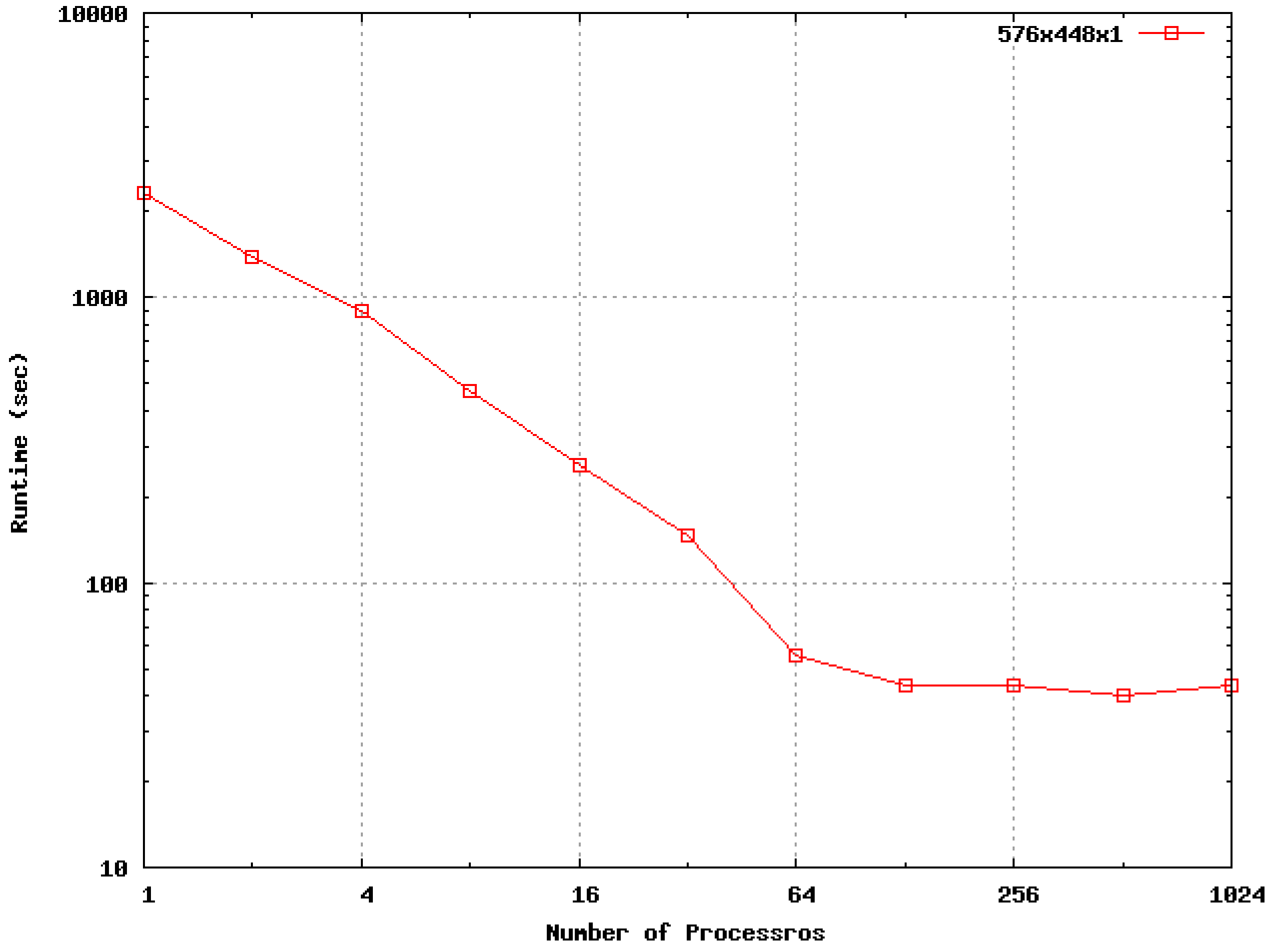
Antarctica, cont

- 10 km base mesh with 2 levels of refinement (5 km, 2.5 km)
  - base level (10 km): 258,048 cells (100% of domain)
  - level 1 (5 km): 431,360 zones (41.8% of domain)
  - Level 2 (2.5 km): 728,832 cells (17.7% of domain)





Parallel scaling, Antarctica benchmark



BISICLES – Next steps

- More work with nonlinear velocity solve.
- Semi-implicit time-discretization for stability, accuracy.
- Non-isothermal capability
- Finish coupling with existing Glimmer-CISM code
- Performance optimization and autotuning.
- Begin work on full 3D velocity solve.
- Refinement in time?