Present and Future Computing Requirements for AMR for Ice Sheet Modeling

Dan Martin

Applied Numerical Algorithms Group,

Lawrence Berkeley National Laboratory

NERSC ASCR Requirements for 2017
January 15, 2014
LBNL

Predicting **I**ce **S**heet and **C**limate Evolution at **E**xtreme **S**cales (PISCEES)

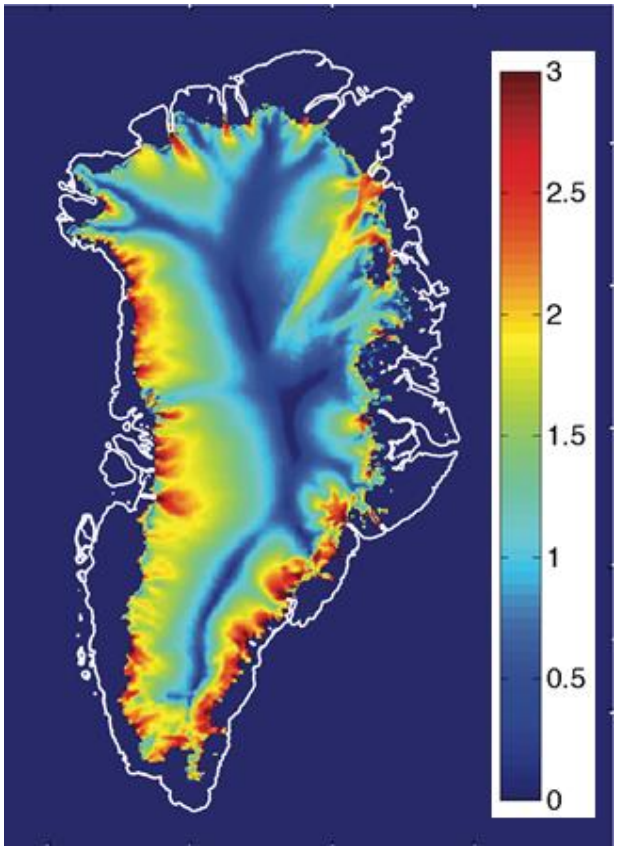
PIs: Stephen Price (LANL), Esmond Ng (LBNL)
Institutions: LANL (PI), LBNL, ORNL, SNL, NCAR, UT, MIT, FSU, USC

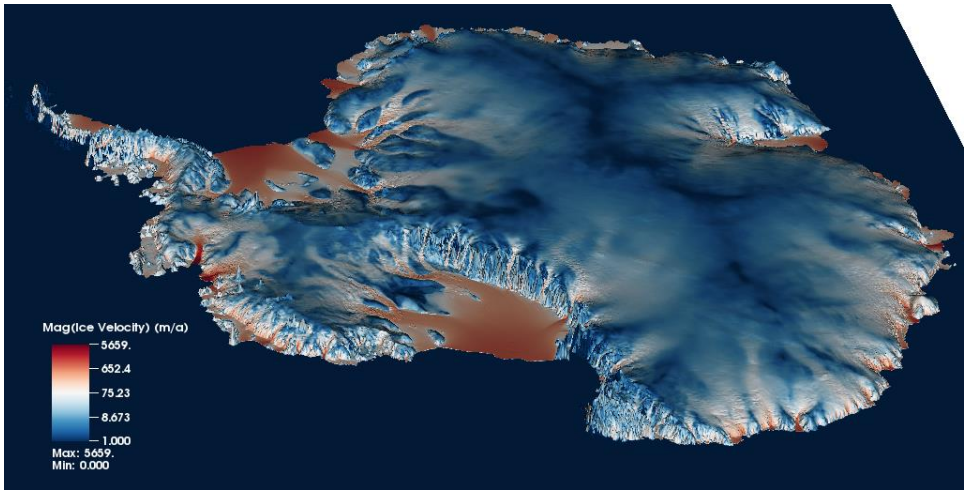
Background:

- Understanding the dynamics of ice sheets (Antarctica, Greenland) essential for projections of sea level rise.

- IPCC AR4: Then-current ice sheet models inadequate to the task of understanding ice sheet response to climate forcings (AR5 – some improvement)

- ASCR-BER SciDAC partnership to develop improved ice sheet models and their application in climate modeling





Project Goals

- Develop and apply robust, accurate, and scalable dynamical cores (“dycores”) for ice sheet modeling on structured and unstructured meshes with local/adaptive refinement.
- Evaluate ice sheet models using new tools and data sets for Verification and Validation (V&V) and Uncertainty Quantification (UQ)
- Integrate models & tools into the Community Ice Sheet Model (CISM) and couple to Earth System Models (ESMs) like CESM, ACME.

PISCEES now vs. 2017

Now:

- New dycore development (BISICLES, FELIX)
- Model coupling (dycores into CISM & CISM into ESMs)
- Initial V&V efforts (test suite automation, dataset gathering/massaging for use in model-obs comparison)
- Initial UQ (low param. space / sampling-based approaches & exploratory use of adjoint-capable codes)
- Baseline performance evaluations for analysis by SUPER

2017:

- Stand-alone and ESM fully-coupled (ocean-atmosphere-ice) runs with optimized initial conditions
- Ensembles of forward model runs for UQ on model outputs (e.g., sea-level rise)

2. Computational Strategies

- High-level computational approach: Solve systems of continuum PDEs which describe ice sheet evolution using finite volume or finite element approaches.

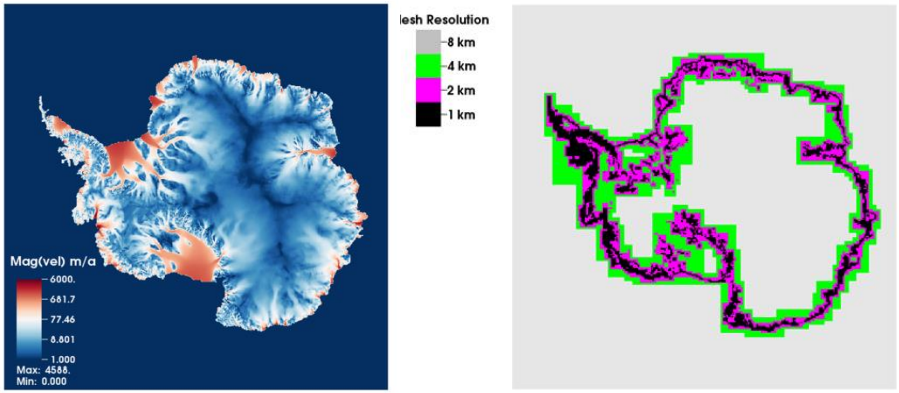
- Ice: shear-thinning non-Newtonian viscous fluid.

- Computation time dominated by repeated solution of large, nonlinear sparse elliptic systems of equations (solving momentum balance for the velocity field).

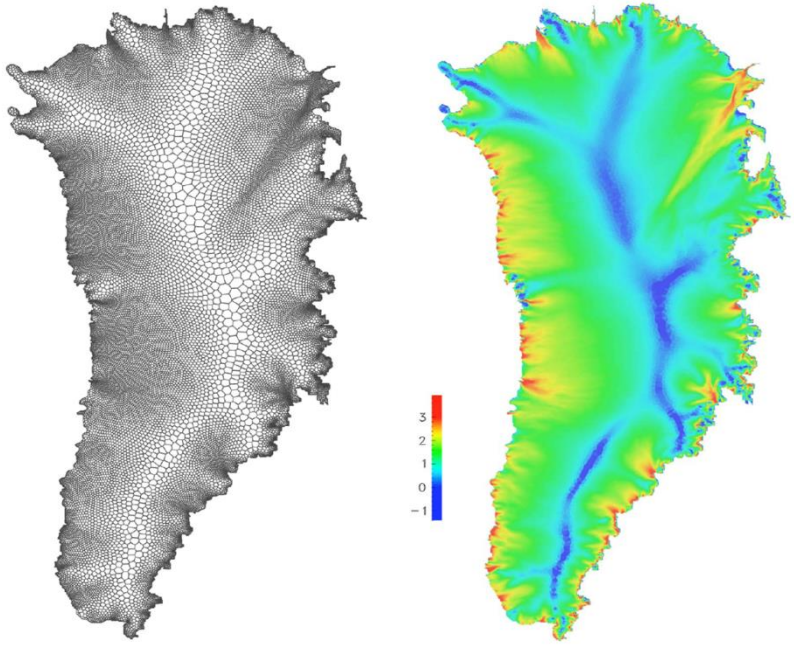
- Codes we use: BISICLES (Chombo) and FELIX (Trilinos).

- **These codes are characterized by these algorithms:**

  - **BISICLES**: finite-volume block-structured adaptive mesh refinement (AMR) dynamical core (dycore) based on the Chombo framework, using PETSc linear solvers.



  - **FELIX**: finite-element dycore on an unstructured mesh, using the Model for Prediction Across Scales (MPAS) framework and Trilinos software packages



- **Algorithms:**
  - FEM & FVM discretizations using Trilinos and Chombo libraries; MPAS unstructured meshing framework
  - Newton-based (JFNK) methods for nonlinear systems
  - AMG and Krylov-based methods for linear systems (PETSc and Trilinos)

- **Computational challenges:**
  - robustness of nonlinear solver over range of input datasets/resolutions
  - problem-specific solver convergence;
  - performance variability related to preconditioning

- Scaling limited by:
  - problem size
  - communication overhead
  - hardware issues on *LCF (e.g. cpu layout)
- Most work done within **Trilinos**, **Chombo,** & **PETSc**
  - leverage improvements made there.
  - (At least for Chombo) feed improvements back to the development trunk for eventual (upcoming) release.

3. Current HPC Usage

- Machines currently using (NERSC or elsewhere)
  - Hopper, initial ports to titan, local development machines.

- Hours used in 2012-2013 (list different facilities)
  - NERSC (Hopper): 500,000 on Hopper (BISICLES: 300k, FELIX: 200k)

- The rest of this discussion will primarily center on BISICLES.

3. Current HPC Usage (cont)

- Typical parallel concurrency & run time, no. runs/year
  - Entering "production" phase with BISICLES – Example run: 1 km Antarctica for 100 years:
    768 processors for 77 hours (7 x 11hrs) = **60k CPU-hours**.
  - Next up: 1km->500m->250m (16x); expect 4x concurrency (3k processors), 4x time (300 hrs) = **960k cpu-hours**.

- Data read/written per run
  - 1km run: **75 GB written, 750 MB read (checkpoint)**
  - 250m run: **300 GB written, 3GB read**
- Memory used per (node | core | globally)
  - Memory has never been a constraint for us; the memory available to us on Hopper has been sufficient (parallelism has been driven by execution time, not memory footprint).

- Necessary software, services or infrastructure
  - Compilation: C++ and Fortran compilers, PERL, MPI
  - Libraries: HDF5, NetCDF, LAPack, PETSc, (Chombo)
  - DDD or similar debugger, performance tools

- Data resources used and amount of data stored
  - /scratch: **10 TB**
  - /project: **1 TB**
  - HPSS: **7500 SRU**

4. HPC Requirements for 2017

- Compute hours needed (in units of Hopper hours)
  - **2B** Hopper CPU-hours.

- Changes to parallel concurrency, run time, number of runs per year
  - Increased size (2.5D->3D): **5-10x**
  - Improved physics: **2-3x**
  - Expect roughly 25x unknowns – rough guide to concurrency increase: **25k MPI tasks**
  - Increased number of runs: **100 runs/yr**

4. HPC Requirements for 2017

- Changes to data read/written
  - Expect more-frequent checkpointing, less-frequent regular data output
  - Expect roughly 500GB/run x 100 runs = **50TB/year**

- Changes to memory needed per (core | node | globally)
  - Roughly 25x current unknowns globally. Expect to remain execution-time bound
- Changes to necessary software, services or infrastructure
  - At the very least, add OpenMP threading (already underway)

5. Strategies for New Architectures (1 of 2)

- Does your software have CUDA/OpenCL directives; if yes, are they used, and if not, are there plans for this?
  - Not currently, no plans at the moment
- Does your software run in production now on Titan using the GPUs?
  - No
- Does your software have OpenMP directives now; if yes, are they used, and if not, are there plans for this?
  - Not currently, but implementation in Chombo is underway
- Does your software run in production now on Mira or Sequoia using threading?
  - No.
- Is porting to, and optimizing for, the Intel MIC architecture underway or planned?
  - Not currently on my side of things.

5. Strategies for New Architectures (2 of 2)

- Have there been or are there now other funded groups or researchers engaged to help with these activities?
  - PISCEES is engaged with SUPER (Williams (LBNL) and Worley (ORNL)) to assist with performance improvement on current and impending architectures.

- If you answered "no" for the questions above, please explain your strategy for transitioning your software to energy-efficient, manycore architectures
  - We plan to **heavily** leverage FASTMath/SUPER-sponsored work to transition/adapt Chombo, PETSc, HDF5, etc to the next generation of architectures.

- What role should NERSC play in the transition to these architectures?
  - At the very least, training and support for users making these transitions.
  - Work with tool developers to design the next generation of debuggers and performance tools.

- What role should DOE and ASCR play in the transition to these architectures?
  - Algorithm, library, and tool development to move existing scientific software forward to the new architectures.

5. Special I/O Needs

- Does your code use checkpoint/restart capability now?
  - Yes.

- Do you foresee that a burst buffer architecture would provide significant benefit to you or users of your code?
  - Yes – would result in more-frequent checkpointing resulting in less lost work due to failures (due to code crashes, hardware failures, and simple underestimation of runtimes).
  - Checkpoint size is substantially bigger than regular data plotfiles, so they get written less frequently

Scenarios for possible Burst Buffer use are on http://www.nersc.gov/assets/Trinity--NERSC-8-RFP/Documents/trinity-NERSC8-use-case-v1.2a.pdf

6. Summary

- What new science results might be afforded by improvements in NERSC computing hardware, software and services?
  - More reliable/accurate estimates of ice sheet response to more climate forcing scenarios.

- Recommendations on NERSC architecture, system configuration and the associated service requirements needed for your science
  - We don't expect to need the massive single-job parallelism like others, but we do anticipate a fairly large (by today's standards) need for computing resources.
  - Our needs are more aligned with **capacity class** (i.e. NERSC) vs. leadership class resources.

- NERSC generally refreshes systems to provide on average a 2X performance increase every year. What significant scientific progress could you achieve over the next 5 years with access to 32X your current NERSC allocation?
  - Improved spatial and temporal accuracy
  - improved (more complex/realistic) physics
  - Many runs to explore parameter spaces, begin to evaluate uncertainties (hundreds?)

- What "expanded HPC resources" are important for your project?
  - Capacity to support reasonable queue turnaround time for the problem sizes we need.