Ice Sheet Modeling with MALI

Presenters: Mauro Perego and Jerry Watkins

Main collaborators:
L. Bertagna, M. Carlson, K. Liegeois, I. Tezaur, R. Tuminaro (Sandia)
T. Hillebrand, M. Hoffman, S. Price (LANL)

Trilinos User Group meeting, Albuquerque, Oct 2022
Talk Outline

• Brief motivation and introduction to ice sheet equations
• MALI ice sheet model description
• Recent developments
• Newly funded project FanSSIE and plan for next five years
• Deep dive into work for improving performance and portability

Supported by US DOE Office of Science projects:

- **ProSPect**: Probabilistic Sea-Level Projections from ice sheets and Earth System Models
- **FASTMath**: Frameworks, Algorithms and Scalable Technologies for Mathematics
- **E3SM**: Energy Exascale Earth System Model
- **FAnSSIE**: Framework for Antarctic System Science In E3SM
Why is it important to model ice sheets? (sea level rise in the last three decades)

Contributors to global sea level rise (1993-2018):

Simulation of Antarctic ice sheet as a consequence of extreme (unrealistic) climate forcing that induces collapse of ice shelves
Goal: probabilistic projections of sea level rise
(projections of future sea level rise)

Possible pathways for future sea-level rise

Note: regional sea-level rise can significantly exceed the global mean sea level rise in some areas (e.g. in the Gulf of Mexico).

Simulation of Antarctic ice sheet as a consequence of extreme (unrealistic) climate forcing that induces collapse of ice shelves.
Brief Motivation on basic physics

- Modeling ice sheets (Greenland and Antarctica) dynamics is essential to provide estimates for sea-level rise in next decades to centuries.

- Ice behaves like a very viscous shear-thinning fluid (similar to lava flow) driven by gravity.

- An essential step in ice sheet modeling is to estimate the unknown or poorly known parameters (e.g. basal friction, bed topography) and the initial thermomechanical state of the ice – *Initialization*

- It is critical to determine how uncertainties in the data and the model affect projections of sea-level rise.
Model: Ice velocity equations

Stokes equations:
\[
\begin{align*}
-\nabla \cdot \sigma &= \rho g \\
\nabla \cdot \mathbf{u} &= 0
\end{align*}
\]

Stress tensor:
\[
\sigma = 2\mu \mathbf{D} - p \mathbf{I}, \quad \mathbf{D}_{ij}(\mathbf{u}) = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)
\]

Ice viscosity (dependent on temperature):
\[
\mu = \frac{1}{2} A(T) \left| \mathbf{D}(\mathbf{u}) \right|^{\frac{1}{n}-1}, \quad n \geq 1
\]

Sliding boundary condition at ice bed:
\[
\begin{align*}
\mathbf{u} \cdot \mathbf{n} &= 0, \quad \text{(impenetrability)} \\
(\sigma \mathbf{n})_\parallel &= \beta \mathbf{u}
\end{align*}
\]

Free slip: \( \beta = 0 \)

No slip: \( \beta = \infty \)

Note: We use a simplification of Stokes equations, called First-Order (FO) model, derived exploiting the shallow nature of the ice sheets and using hydrostatic pressure.
Model: Temperature equation

Heat equation (for cold ice):

\[ \rho c \partial_t T + \nabla \cdot (k \nabla T) + \rho c u \cdot \nabla T = 4 \mu |D(u)|^2 \]

Boundary condition at the ice bed (includes melting and refreezing):

\[ m = G + \beta |u|^2 - k \nabla T \cdot n \]

In this work we use an enthalpy formulation* that accounts for temperate ice as well.

*A. Aschwanden, E. Bueler, C. Khroulev, and H. Blatter, Journal of Glaciology, 2012
**Software: MPAS-Albany Land Ice model (MALI)**

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>SOFTWARE TOOLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Finite Elements on tets/prisms</td>
<td>Albany Land Ice</td>
</tr>
<tr>
<td>Optimization</td>
<td>ROL</td>
</tr>
<tr>
<td>Nonlinear solver (Newton method)</td>
<td>NOX</td>
</tr>
<tr>
<td>Krylov linear solvers/Prec</td>
<td>Belos/MueLu, Belos/FROSch</td>
</tr>
<tr>
<td>Automatic differentiation</td>
<td>Sacado</td>
</tr>
<tr>
<td>Mesh structure</td>
<td>SEACAS /STK</td>
</tr>
</tbody>
</table>

**MPAS** (Model for Prediction Across Scales): *Fortran, finite volumes* library, conservative Lagrangian schemes for advecting tracers (evolution of ice thickness), coupling to E3SM

**Albany Land Ice**: *C++* finite element library built on top of Trilinos achieving performance portability through Kokkos programming model. Provides large scale PDE constrained optimization capabilities

**Note**: MPAS land-ice layer to be removed in the future. Albany will be coupled directly to E3SM

**References**:
- Hoffman, et al. GMD, 2018
- Perego, Price, Stadler, JGR, 2014
Recent development: (code development)

- **Improvements to solvers portability and performance monitoring** (see Jerry’s deep dive)

- **Hessian AD** Piro/ROL/Sacado
  - Matrix-free second-order derivatives of residual and responses w.r.t. solution and distributed parameters
  - Assembly of Hessian matrix of scalar response using coloring and seeding (**Zoltan2**)
  - Enables second-order (e.g. Trust Region) optimization methods

- **PyAlbany** (see related PyTrilinos talk tomorrow by K. Liegeois)
  - Python interface to Albany based on pybind11.
  - Drives Albany forward solves and analysis solves and allows fast and easy prototyping and postprocessing
  - Used for parameter tuning of solvers, running inverse problems and ensembles for UQ analysis
Showcase: Thermo-mechanical initialization of Greenland ice sheet

modeled ice speed | observed ice speed | modeled basal friction | modeled temperature

300K parameters, 14M unknowns. Initialization: ~10 hours on 2k nodes on NERSC Cori (Haswell), The optimization is constrained by the coupled velocity-temperature solvers. Most large scale-ice sheets codes constrain the optimization only with the velocity solver, which results in a temperature field that is not consistent with the velocity
Framework for Antarctic System Science in E3SM
(Recently funded Office of Science BER SciDAC 5-years project)

Project main goals:
• improve modeling of physical processes (ice calving, fracture, ocean coupling, surface processes) that can significantly impact the contribution of Antarctic ice sheet to sea-level change
• provide uncertainty quantification of sea level projections
• Improvements to model to support science and run efficiently on DOE machines

Project team:
LANL (lead institute, PI: Matt Hoffman)
Sandia (M. Perego, L. Bertagna, M. Carlson, J. Hu, K. Liegeois, I. Tezaur, J. Watkins)
LBNL (POC: S. Williams)
BNL (POC: N. Urban)
UM (J. Bassis)
RPI (M. Shephard)
UCI (C. Zehnder)
Plan for next 5 years: (code development)

- **Improvements to performance and portability** (see Jerry’s deep dive)

- **Mesh Adaptation** (collaboration w/ RPI)
  - Use Omega_h mesh
  - Align mesh with important ice features (calving front, grounding line, rifts)
  - Add level-set equations for capturing ice front and rifts (possibly use Krino)

- **High-order discretization**
  - Exploit tensor-product structure of extruded mesh – PanzerDofMgr/Intrepid2

- **Transient PDE-Constrained Optimization** – Piro/ROL/Tempus/Sacado
Performance Highlights

• **Major improvements to finite element assembly time**
  - **Memoization** to avoid unnecessary data movement and computation
  - **Tpetra::FECrsMatrix** refactor to reduce memory footprint and data movement
  - **Boundary condition** refactor to reduce memory footprint and data movement

• **Solver portability on Cori and Summit**
  - **MueLu SemiCoarsen** refactor using Kokkos
  - **Ifpack2 portable smoothers** tuned to GPU hardware

• **Automated performance testing**
  - **Changepoint detection** for performance monitoring

Watkins, J., Carlson, M., Shan, K., Tezaur, I., Perego, M., Bertagna, L., Kao, C. *et al.*
“Performance portable ice-sheet modeling with MALI.” (Submitted to IJHPCA, 2022) [https://arxiv.org/abs/2204.04321](https://arxiv.org/abs/2204.04321)
Performance on Cori and Summit

• Setup
  ➢ Weak scalability – First-order Stokes, Antarctica 1km resolution, up to 256 nodes
  ➢ Different smoothers – CPU block smoother, GPU point smoother

• Results
  ➢ CPU scales better than GPU – 65% (CPU) vs. 42% (GPU)
  ➢ Speedup on GPU – 1.9-1.2x speedup over CPU

• Areas to improve
  ➢ Assembly on CPU – 40-60% of total time
  ➢ Solver on GPU – 80-90% of total time
Detecting performance regressions/improvements

Example: Transition to Kokkos 3.5.0 caused a performance regression but was soon fixed

Regression

Improvement

Total Fill time for a 1-to-7 km resolution Greenland mesh, executed nightly in Albany Land Ice
Monitoring performance comparisons

Example: Memoization comparison (w. & w.o.) shows that relative performance has increased

Speedup of Total Fill time from memoization for a 1-to-7 km resolution Greenland mesh, executed nightly in Albany Land Ice
Automatic performance tuning

Example: Autotuning used to improve performance of multigrid smoothers on GPU

Best runtimes for a range of function evaluation budgets – 3-to-20 km resolution Greenland mesh

Best runtimes using GPTune Bayesian optimization, Latin Hypercube Sampling and Monte Carlo Sampling – 3-to-20 km resolution Greenland mesh

Future Performance Overview

Applications
- E3SM + MALI
- MALI-standalone

DOE Exascale
- NERSC Perlmutter
- OLCF Frontier
- ALCF Aurora

Performance
- Algorithmic improvements
- Performance Optimization

- FAST MATH
- RAPIDS
Algorithmic improvements and performance optimization

• **Team members**
  - SNL: Luca Bertagna, Max Carlson, Jonathan Hu, Kim Liegeois, Mauro Perego, Irina Tezaur, Jerry Watkins
  - LBNL: Oscar Antepara, Sam Williams

• **Algorithmic improvements**
  - Improved convergence of linear solver on GPUs (FY24)
    - Block fine grid smoother
  - Physics-based block preconditioning for coupled problems (FY25)

• **Performance optimization**
  - Unified Virtual Memory (UVM) optional (FY23)
  - Optimize multigrid solvers for 2D advection-diffusion (FY25)
  - Evaluate/improve performance of initialization (optimization problem) (FY25)
  - Investigate load balancing and frequency of mesh adaptation (FY26)
  - Optimize data movement in MALI & Omega_h coupling (FY26)
  - Evaluate portability to DOE Exascale (Frontier/Aurora) (FY27)