This space is reserved for the Procedia header, do not use it

On the scalability of the *Albany/FELIX* first-order Stokes approximation ice sheet solver for large-scale simulations of the Greenland and Antarctic ice sheets

Irina K. Tezaur[1], Raymond S. Tuminaro[2], Mauro Perego[2], Andrew G. Salinger[2], and Stephen F. Price[3]

[1] Quantitative Modeling & Analysis Dept., Sandia National Laboratories, Livermore, CA, USA
Phone: (925)294-2474, E-mail: ikalash@sandia.gov
[2] Computational Mathematics Dept., Sandia National Laboratories, Albuquerque, NM, USA
[3] Fluid Dynamics & Solid Mechanics Group, Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

We examine the scalability of the recently developed *Albany/FELIX* finite-element based code for the first-order Stokes momentum balance equations for ice flow. We focus our analysis on the performance of two possible preconditioners for the iterative solution of the sparse linear systems that arise from the discretization of the governing equations: (1) a preconditioner based on the incomplete LU (ILU) factorization, and (2) a recently-developed algebraic multigrid (AMG) preconditioner, constructed using the idea of semi-coarsening. A strong scalability study on a realistic, high resolution Greenland ice sheet problem reveals that, for a given number of processor cores, the AMG preconditioner results in faster linear solve times but the ILU preconditioner exhibits better scalability. A weak scalability study is performed on a realistic, moderate resolution Antarctic ice sheet problem, a substantial fraction of which contains floating ice shelves, making it fundamentally different from the Greenland ice sheet problem. Here, we show that as the problem size increases, the performance of the ILU preconditioner deteriorates whereas the AMG preconditioner maintains scalability. This is because the linear systems are extremely ill-conditioned in the presence of floating ice shelves, and the ill-conditioning has a greater negative effect on the ILU preconditioner than on the AMG preconditioner.

Keywords: Ice sheet model, first-order Stokes approximation, finite element method, scalability, ILU preconditioner, algebraic multigrid (AMG) preconditioner, semi-coarsening, Greenland, Antarctica.

1 Introduction

In its fourth assessment report (AR4), the Intergovernmental Panel on Climate Change (IPCC) declined to include estimates of future sea-level rise resulting from ice sheet dynamics [17], due to the fact that existing models lacked the capability to reproduce or explain the observed dynamic

behavior of ice sheets. This omission motivated a worldwide effort to develop "next generation", community-supported ice sheet models better able to perform realistic, high-resolution, continental scale simulations of ice sheet evolution. Improved models should be based on partial differential equations (PDEs) that accurately represent the relevant momentum balance by accounting for both vertical and horizontal stress gradients. Care should be taken to select numerical methods that are well-suited for the equations of ice sheet flow, and the resulting discretized models should be robust, parallel, and scalable for use on high performance computing (HPC) platforms. Additional advanced analysis capabilities that are highly desirable include access to model and parameter derivatives (for use in model optimization), data assimilation, and uncertainty quantification.

This paper focuses on the relevant computational aspects of a new finite element momentum balance solver for land ice simulations, based on the first-order approximation of the nonlinear Stokes flow model for glaciers and ice sheets (Section 2). The solver, introduced and verified in [6] and referred to as *Albany/FELIX* (Finite Elements for Land Ice eXperiments), was designed to include many of the capabilities required of a "next-generation" ice sheet code through the use of sophisticated, robust algorithms, a template-based generic programming model, and a collection of computational mathematics libraries from the *Trilinos* suite [5] (Section 3). In [6], model accuracy, convergence, scalability and robustness of *Albany/FELIX* were investigated on problems involving a realistic Greenland ice sheet geometry discretized using both structured and unstructured meshes.

In the present work, the performance of *Albany/FELIX* is evaluated on ice sheet problems that are larger and more complex than those considered in [6] and attention is focused on the scalability (both strong and weak) of the code's iterative linear solver (Section 4). Because a linear solver may be called upon thousands of times during prognostic model solutions, a robust and efficient linear solve is essential for overall model efficiency, and has a large effect on the overall code scalability. Two preconditioned Krylov-based methods are evaluated for the iterative solution of the sparse linear systems that arise in our discretized model: Conjugate Gradient (CG), and GMRES. We focus our analysis on the performance of two preconditioners: an incomplete LU (ILU) preconditioner, and an algebraic multigrid (AMG) preconditioner based on the idea of semi-coarsening that was recently developed [18] for this application. Strong scalability is studied in the context of a fine resolution problem posed on a Greenland ice sheet (GIS) geometry in Section 4.1. Following the strong scaling study is a weak scalability study on an Antarctic ice sheet (AIS) problem, fundamentally different from the GIS due to the presence of floating ice shelves (Section 4.2). The numerical studies in Section 4 reveal the following: (a) GMRES is slightly more effective than CG for AIS problems, despite problem symmetry; (b) while an ILU-preconditioned solver is scalable for the GIS problem, it is significantly slower and less scalable than the AMG-preconditioned solver for the AIS problem; (c) both partitioning and ordering are extremely important for the (processor-based) ILU preconditioner. Observations (a) and (b) suggest that GMRES and AMG are less sensitive than CG and ILU (respectively) to the rounding errors associated with the severe ill-conditioning introduced by the presence of ice shelves in AIS problems.

2 First-order Stokes mathematical model for ice flow

Ice sheets and glaciers are typically modeled as an incompressible fluid in a low Reynolds number regime with a power-law viscous rheology. It is widely accepted that the governing PDEs are the Stokes flow equations for glaciers and ice sheets. The model considered here is a first-order approximation to the nonlinear Stokes flow equations [3, 15], also referred to as the "Blatter-

Pattyn" model [9, 1], or simply the "first-order (FO) Stokes model". The FO approximation is derived by assuming the ice has a small geometric aspect ratio $\delta = H/L$ (where H and L are characteristic length scales for the vertical and horizontal dimensions, respectively, and $H \ll L$). It is also assumed that the normal vectors to the ice sheet's upper and lower surfaces, $\mathbf{n} \in \mathbb{R}^3$, are nearly vertical. Effectively, the FO approximation is derived by neglecting $\mathcal{O}(\delta^2)$ terms in the Stokes equations and the respective boundary conditions (Appendix A of [6]). The result is the following elliptic coercive system of PDEs:

$$\begin{cases} -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_1) + \rho g \frac{\partial s}{\partial x} &= 0, \\ -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_2) + \rho g \frac{\partial s}{\partial y} &= 0, \end{cases} \tag{1}$$

where g denotes the gravitational acceleration, ρ denotes the ice density, and $s \equiv s(x, y)$ denotes the upper surface boundary: $\Gamma_s \equiv \{(x, y, z) \in \mathbb{R}^3 | z = s(x, y)\}$ The $\boldsymbol{\epsilon}_i$ terms in (1) are the first order approximations of the effective strain rate tensors:

$$\dot{\boldsymbol{\epsilon}}_1^T = \begin{pmatrix} 2\dot{\epsilon}_{xx} + \dot{\epsilon}_{yy}, & \dot{\epsilon}_{xy}, & \dot{\epsilon}_{xz} \end{pmatrix}, \qquad \dot{\boldsymbol{\epsilon}}_2^T = \begin{pmatrix} \dot{\epsilon}_{xy}, & \dot{\epsilon}_{xx} + 2\dot{\epsilon}_{yy}, & \dot{\epsilon}_{yz} \end{pmatrix}, \tag{2}$$

where

$$\dot{\epsilon}_{xx} = \frac{\partial u}{\partial x}, \quad \dot{\epsilon}_{yy} = \frac{\partial v}{\partial y}, \quad \dot{\epsilon}_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \dot{\epsilon}_{xz} = \frac{1}{2}\frac{\partial u}{\partial z}, \quad \dot{\epsilon}_{yz} = \frac{1}{2}\frac{\partial v}{\partial z}, \tag{3}$$

and where u and v denote the x and y components of the ice velocity vector $\mathbf{u} \equiv \begin{pmatrix} u, & v \end{pmatrix}^T \in \mathbb{R}^2$ (respectively). The effective viscosity μ can be derived using Glen's flow law [2, 8] as:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \dot{\epsilon}_e^{\frac{1}{n}-1}, \tag{4}$$

where $\dot{\epsilon}_e$ is the effective strain rate, given by:

$$\dot{\epsilon}_e^2 \equiv \dot{\epsilon}_{xx}^2 + \dot{\epsilon}_{yy}^2 + \dot{\epsilon}_{xx}\dot{\epsilon}_{yy} + \dot{\epsilon}_{xy}^2 + \dot{\epsilon}_{xz}^2 + \dot{\epsilon}_{yz}^2. \tag{5}$$

In (4), A is the flow rate factor and n is the Glen's (power) law exponent, typically taken equal to 3 for ice sheets (hence μ, given by (4), is a nonlinear expression). The flow law rate factor A is strongly temperature-dependent, and can be described through an Arrhenius relation (see, e.g., [2]). Here, we take temperatures as given and thus A is taken as known *a priori*.

To complete the formulation of a problem for the PDEs (1), boundary conditions are required. Suppose the system (1) is posed on a bounded domain Ω with boundary $\Gamma \equiv \Gamma_s \cup \Gamma_b \cup \Gamma_l$, where Γ_s, Γ_b and Γ_l denote the upper, lower and lateral (vertical) boundaries, respectively.

On the upper boundary, Γ_s, a stress-free (homogeneous Neumann) boundary condition is prescribed: $\dot{\epsilon}_i \cdot \mathbf{n} = 0$ for $i = 1, 2$, where $\mathbf{n}$ denotes the outward facing normal vector to Γ_s.

On the lower boundary, Γ_b, either a no-slip or a sliding boundary condition is considered. Here, as is common in other work, we assume sliding can be expressed through a Robin-type boundary condition of the form:

$$2\mu \dot{\boldsymbol{\epsilon}}_1 \cdot \mathbf{n} + \beta u = 0, \;\; 2\mu \dot{\boldsymbol{\epsilon}}_2 \cdot \mathbf{n} + \beta v = 0, \;\; \text{on } \Gamma_b. \tag{6}$$

In (6), $\beta \equiv \beta(x, y) \geq 0$ is the basal sliding (or friction) coefficient. For realistic problems involving geometries such as the GIS and AIS problems considered here, the basal friction coefficient field is calculated by solving a deterministic inversion problem that minimizes the discrepancy between modeled and observed surface velocities (see, e.g., [12] and references

therein). The case when β is large (e.g., $\beta = 10^7$ kPa a m^{-1}) corresponds to a quasi-no-slip boundary condition.

On the lateral boundary, Γ_l, the physically relevant boundary conditions are either a kinematic (Dirichlet) boundary condition in which the values of the ice velocities are prescribed, or a dynamic (Neumann) boundary condition. In the examples presented in this paper, only the latter boundary condition is considered, also referred to as an "open-ocean", or "floating ice", boundary condition. The condition is derived by assuming that the ice shelf is in hydrostatic equilibrium with the air and/or water that surrounds it and takes the form:

$$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \mathbf{n} - \rho g(s-z)\mathbf{n} = \rho_w g \max(z,0)\mathbf{n}, \quad \text{on } \Gamma_l, \tag{7}$$

for $i = 1, 2$, where ρ_w denotes the density of water. In (7), it has been assumed that the coordinate system has been oriented such that z is strictly elevation (that is, $z = 0$ at sea level and values of z increase for higher elevations) [7].

3 Numerical discretization and software implementation: the *Albany/FELIX* solver

During the past several years, an effort has been made at Sandia National Laboratories to develop a new unstructured grid, parallel, scalable and robust finite element solver for the first-order Stokes equations (1). This code, known as *Albany/FELIX*, employs a collection of algorithms and software libraries selected for accuracy, flexibility, robustness, and scalability. The *Albany/FELIX* ice flow solver is implemented in a C++, open-source[1], parallel, unstructured grid, implicit, finite element, multi-physics analysis code base known as *Albany* [14]. *Albany* "glues" together numerous computational mathematics packages from the *Trilinos* suite [5] through the use of Template-Based Generic Programming (TBGP) [11].

For a detailed description of *Albany/FELIX*, the reader is referred to [6]. The key methods implemented in *Albany/FELIX* are summarized below.

- ***Classical Galerkin finite element method (FEM) discretization***: In *Albany/FELIX*, the classical Galerkin FEM was selected to discretize the FO Stokes model (1) for its flexibility in using unstructured meshes (e.g., grids with increased resolution in areas of large velocity gradients such as in the vicinity of outlet glaciers) and straightforward implementation of the basal sliding boundary condition (6). The *STK* package of *Trilinos* was used for mesh database structures and mesh I/O. The *Intrepid* package of *Trilinos* was used as a finite element shape function library with general integration kernels.

- ***Newton's method with automatic differentiation (AD) Jacobians, and homotopy continuation***: Once the large, sparse system of nonlinear algebraic equations for the ice velocities is created following discretization by the FEM, the fully-coupled nonlinear system is solved using Newton's method. An analytic Jacobian matrix is computed at each iteration of Newton's method using AD, available through the *Sacado* package of *Trilinos*. Because the Glen's law effective viscosity (4) is not well-defined for a constant $\mathbf{u}$, a Newton iteration may not reliably converge in the case a constant solution is taken as the initial guess for Newton's method (common in the scenario where a "good" initial guess is lacking). A common practice to circumvent this difficulty is through the addition

[1]The *Albany* code can be obtained from its public `github` repository by the interested reader: `https://github.com/gahansen/Albany`.

of a regularization parameter γ ($\gamma > 0, \gamma \ll 1$) to the sum of the strain rates in the effective strain rate (5). In [6], it was shown that a robust nonlinear solution procedure can be obtained by performing a homotopy continuation with respect to the regularization parameter γ to step to the final solution by solving a series of nonlinear problems that converge reliably. For details on this algorithm, the reader is referred to Section 3.1.1 of [6]. The Newton-based nonlinear system solver and homotopy continuation algorithm are implemented in the *NOX* and *LOCA* packages of *Trilinos*, respectively.

- ***Iterative linear solver with ILU or AMG preconditioning***: Within each Newton iteration detailed above, a number of linear systems arise. These systems are solved using a preconditioned iterative method (CG or GMRES, both available through the *Belos* package of *Trilinos*). Although the model is symmetric, and hence amenable to the CG iterative linear solver, it was found that faster convergence can be obtained with the GMRES iterative linear solver for some problems (e.g., simulations of the AIS). Two options for the preconditioner are considered: an ILU additive Schwarz preconditioner with 0 overlap and 0 level-of-fill, and a recently proposed (introduced in [6]; detailed in [18]) AMG preconditioner, constructed based on the idea of semi-coarsening (i.e., coarsening only in the structured dimension, in this case z–dimension). These preconditioners are available through the *Ifpack* and *ML* packages of *Trilinos*, respectively.
- ***Adjoint-based optimization for ice sheet initialization:*** To calculate the ice sheet initial conditions, namely the basal sliding and basal topography fields, we formulate and solve a PDE-constrained optimization problem that minimizes the mismatch between model output and observations (detailed in [12]). The optimization is performed using the LBFGS method, as implemented in the *ROL Rapid Optimization Library* of *Trilinos*. The cost function gradients with respect to the parameter fields are computed using adjoints.

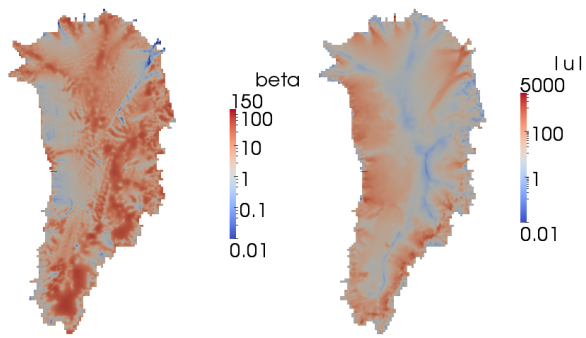
4 GIS and AIS simulations using *Albany/FELIX*

Having described the mathematical model for ice sheet flow and its numerical implementation in the *Albany/FELIX* code, we now turn our attention to evaluating the performance of the solver on realistic simulations of the GIS and AIS, with a focus is on scalability. In HPC, the term scalability (or scaling) refers to the efficiency of an application when the number of parallel processing elements (e.g., cores, processors, threads) is increased. There are two common notion of scalability: strong and weak scaling. Strong scaling measures speedups when a fixed-size problem is run on increasing number of processing element. To achieve ideal strong scaling, the problem should scale linearly, i.e., the speedup should be equal to the number of processing elements used. In contrast, weak scaling measures speedups assuming a fixed problem size per processing element. Since each processing element has the same amount to do, in the ideal case, the execution time should remain constant across the runs in the weak scaling study.

The results reported in this paper were computed on the *Hopper* Cray XE6 supercomputer at the National Energy Research Scientific Computing (NERSC) Center[2]. All meshes considered were structured, uniform hexahedral meshes, obtained by first generating a uniform quadrilateral mesh of a two-dimensional (2D) cross-section of the ice geometry, then extruding this mesh uniformly in the third (vertical) direction using a specified number of layers. Structured meshes were chosen for the ability to perform controlled scaling studies.

[2]More information on the *Hopper* machine can be found here: `http://www.nersc.gov/users/computational-systems/hopper`.

4.1 Strong scaling study for a fine-resolution GIS problem



(a) Basal friction (b) Solution Magnitude

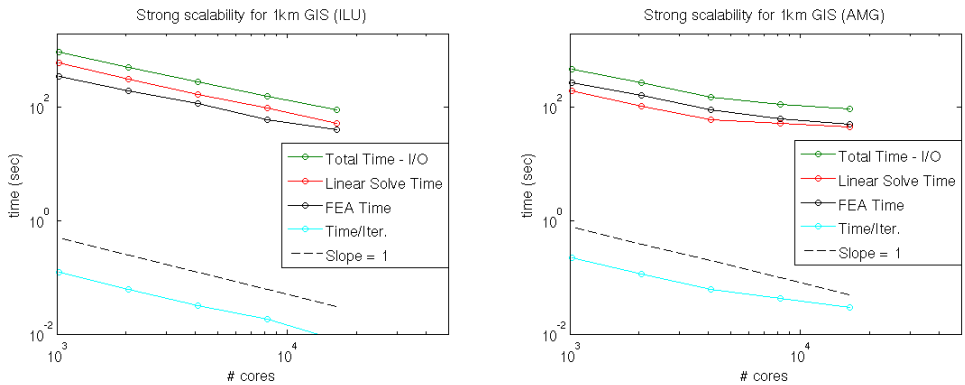
Figure 1: Optimized basal sliding coefficient field from [12] and modeled surface velocity for a 1 km horizontal resolution mesh with 40 vertical layers.

First reported are the results of a strong scaling study performed for a fine-resolution GIS problem with realistic initial conditions. Realistic basal friction coefficient (Figure 1(a)) and bed topography fields were calculated by solving a deterministic inversion problem that minimizes simultaneously the discrepancy between modeled and observed surface velocities (see [12] for more details). A realistic, 3D temperature field, originally calculated using the *Community Ice Sheet Model* (*CISM*) for the study in [16] provided realistic values for the flow-law rate factor (4). These data were processed as explained in Section 6.1 of [6]. A uniform quadrilateral mesh having a horizontal resolution of 1 km was generated, then extruded into a 3D mesh having 40 vertical layers. This mesh consisted of 69.8 million hexahedral elements, giving rise to 143 million unknowns.

For the GIS problem, the iterative solver was a preconditioned CG method (appropriate, as the problem is symmetric). The AMG scheme uses damped line Jacobi smoothing on the finest level (where lines are defined in the mesh extruded direction) and Chebyshev smoothing on all coarser levels. Both partitioning and ordering of equations is extremely important for the processor-based (or domain decomposition) ILU preconditioner. In particular, it is essential that the incomplete factorization accurately capture vertical coupling, which is dominant due to the highly anisotropic mesh. This is accomplished by ensuring that all points along a vertically extruded grid line reside within a single processor and by ordering the equations such that all unknowns associated with grid layer k's nodes are ordered before all unknowns associated with the grid layer $k+1$ (known as "row-wise ordering"). If either ordering or partitioning is done improperly, convergence when using ILU deteriorates severely.

For the strong scaling study, the problem is run on different numbers of cores on *Hopper*, from 1024 to 16,384, a 16-fold increase. The total solve time minus the mesh import, the total linear solve times and the finite element assembly times for each of the runs are plotted on a log-log scale in Figure 2(a) and (b) as a function of the number of cores for the ILU and AMG preconditioners considered (respectively). The black-dashed line in the Figure 2 represents ideal (linear) strong scaling.

The reader can observe by examining Figure 2 that at lower core counts, the AMG preconditioned solve times are much faster than the corresponding ILU solve times. For example, 194.3



(a) ILU preconditioner (b) AMG preconditioner

Figure 2: Greenland strong scaling study (1 km resolution with 40 vertical layers)*

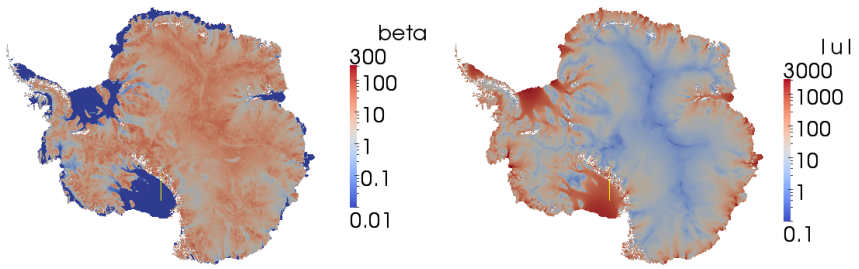
* "FEA" denotes "finite element assembly".

seconds are required for the multigrid solver versus ILU times of 607.9 seconds on 1024 cores. This is due primarily to a significantly better convergence rate obtained with AMG versus ILU. In particular, the average number of linear solves required to reduce the linear residual by six orders of magnitude is 17.3 with AMG preconditioning while it is 122.1 with ILU.

The story is quite different for the 16,384 processor simulation. In this case, the AMG convergence rate is nearly identical to the 1024 core run while ILU requires slightly more iterations per linear solve (145.6 iterations per solve). However, the cost per iteration of the AMG solver is much higher relative to that of the ILU solver. As sub-domain sizes are now smaller, the ILU preconditioner actually requires less computational work in both the setup and per iteration solve phase. On the other hand, the multigrid solver is very inefficient when the number of unknowns per core becomes small. For this problem, there are a little less than 8800 degrees-of-freedom per core on the fine grid and approximately 220 degrees-of-freedom on the next coarsest grid (the multigrid scheme approximately reduces the number of unknowns by a factor of 40). In a traditional serial setting, coarse level processing is nearly insignificant. However, coarse level processing can be as costly or even more costly than fine level processing when communication costs dominates. In fact in the 16,384 processor case, the cost per iteration is nearly identical to the 1024 processor run as communication costs dominate. Thus, ILU preconditioning is fairly effective relative to AMG when the number of unknowns per core is modest (i.e., less than 10,000 degrees-of-freedom).

4.2 Weak scaling study for a moderate-resolution AIS problem

Next, we report on a weak scalability study performed on simulations of the AIS. The ice sheet geometry is based on BEDMAP2 [4] and the three-dimensional (3D) temperature field used in the calculation of the rate factor A is from [10]. The basal friction field, optimized to match observed surface velocities from [13], is obtained using the methods discussed in [12]. Figures 3(a) and (b) show the optimized basal friction coefficient field and the modeled magnitude of the ice surface velocity in the finest spatial resolution (2 km) Antarctica simulation. The AIS is fundamentally different from the GIS (Section 4.1) in that it contains large ice shelves, which are the floating extensions of land ice. Along the fronts of these ice shelves, open-ocean boundary conditions (7) are imposed, and at their base, a zero traction boundary condition is



(a) Basal friction (b) Solution Magnitude

Figure 3: As in Figure 1 but for Antarctica with 2 km horizontal resolution and 20 vertical layers.

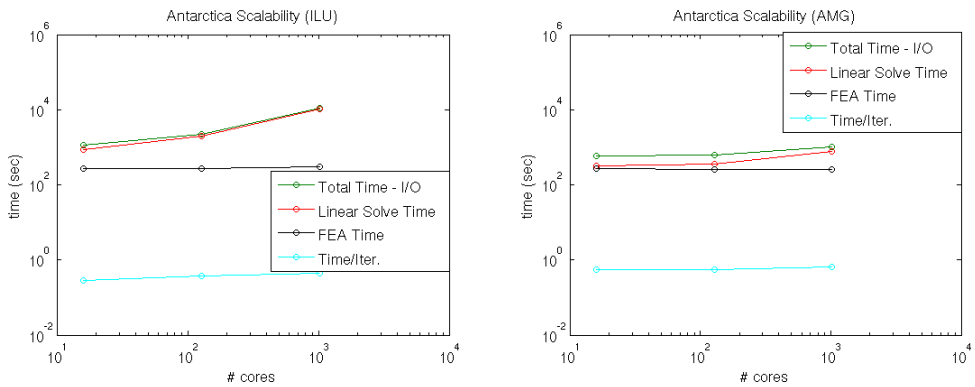
applied.

For the scalability study summarized below, three meshes were considered: an 8 km resolution mesh with 5 vertical layers, a 4 km resolution mesh with 10 vertical layers, and a 2 km resolution mesh with 20 vertical layers. The number of cores for each run was calculated so that for each size problem, each core had approximately the same number of degrees of freedom (dofs): 138–158 K dofs/core. Toward this effect, the 8 km (2.52 million dofs), 4 km (18.5 million dofs) and 2 km (141.5 million dofs) problems were run on 16, 128 and 1024 cores of *Hopper*, respectively.

For the AIS problem, a GMRES iteration is used for the Krylov solver and the multigrid scheme uses line Gauss-Seidel smoothing on the finest level (where lines are defined in the mesh extruded direction) and Chebyshev smoothing on all coarser levels. All other solver parameters remain the same as with the Greenland simulations (Section 4.1) and ordering/partitioning for ILU is even more important in order to maintain acceptable convergence rates. A GMRES method was found to be a bit more effective than CG, even though the problem is symmetric. We believe that GMRES is somewhat less sensitive to rounding errors associated with the severe ill-conditioning induced by the presence of ice shelves, though GMRES and CG also minimize different norms as well.

Figure 4 shows the following timing information: total time minus mesh import, total linear solve time, total finite element assembly time, and the time per linear iteration for (a) the ILU preconditioner, and (b) the AMG preconditioner considered. The Antarctica weak scaling data strongly favors the AMG preconditioning approach. In particular, the ILU solver is more than 10 times slower than the AMG solver on the 1024 core problem. This is due to the extremely poor convergence of the ILU solver (requiring on average over 700 iterations per solve). The large number of iterations is due to the ill-conditioning of the under-lying linear systems. As mentioned earlier, matrix entries corresponding to vertical coupling are much stronger than entries corresponding to horizontal coupling. For vertical grid lines that lie within ice shelves, the top and bottom boundary conditions resemble Neumann conditions and so the sub-matrix associated with one of these vertical lines is nearly singular (as the constant function applied to this sub-matrix is almost identically zero). We believe that this ill-conditioning creates

significant challenges for any solver and in the case of ILU preconditioner lead to very poor convergence rates for the larger problem. While the AMG iterations do grow as we refine the problem (from 14.4 iterations per solve on 16 cores to 35.3 iterations per solve on 1024 cores), it is much better suited to the linear systems associated with large simulations of Antarctica. Again, this difference is due to the presence of large ice shelves, which are not present in the Greenland problem.



(a) ILU preconditioner (b) AMG preconditioner

Figure 4: Weak scaling for the Antarctica problem.

5 Summary

This paper reports on the results of strong and weak scalability studies of the *Albany/FELIX* finite element solver on large-scale simulations of the Greenland and Antarctic ice sheets. Attention is focused on the scalability of the iterative linear solver (either CG or GMRES). The performance of two preconditioners for the iterative linear solves arising in the discretization of this model is evaluated: an ILU preconditioner, and an AMG preconditioner constructed based on the idea of semi-coarsening. Both preconditioners perform reasonably for the GIS strong scaling study. The AMG preconditioner delivers the solution faster than the ILU preconditioner but gives rise to a less scalable linear solve. The weak scaling study on the Antarctic ice sheet problem reveals: that (a) GMRES is more effective than CG as the iterative linear solver, and (b) the AMG preconditioner is significantly more scalable and effective than the ILU preconditioner. We believe this is due to GMRES and AMG being less sensitive to the severe ill-conditioning induced by the presence of ice shelves than CG and ILU, respectively. These findings lead to the practical recommendation of using an AMG preconditioner (over an ILU preconditioner) when solving linear systems arising from the discretization of ice sheet problems with floating ice shelves or for problems where horizontal flow coupling is important over a large fraction of the computational domain.

References

[1] Blatter, H. Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients. *J. Glaciol.*, 41(138):333–344, 1995.

[2] Cuffey, K. *et al.* *The physics of glaciers.* Butterworth-Heinemann, Oxford, 4th edition edition, 2010.

[3] Dukowicz, J. *et al.* Consistent approximations and boundary conditions for ice-sheet dynamics from a principle of least action. *J. Glaciol.*, 56(197):480–496, 2010.

[4] Fretwell, P. *et al.* Bedmap2: improved ice bed, surface and thickness datasets for Antarctica. *The Cryosphere*, 7(1):375–393, 2013.

[5] Heroux, M. *et al.* An Overview of the Trilinos Project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[6] Kalashnikova, I. *et al.* Albany/FELIX: A Parallel, Scalable and Robust Finite Element Higher-Order Stokes Ice Sheet Solver Built for Advance Analysis. *Geosci. Model Develop. Discuss.*, 7:8079–8149, 2014.

[7] MacAyeal, D. *et al.* An ice-shelf model test based on the Ross Ice Shelf, Antarctica. *Ann. Glaciol.*, 23:46–51, 1996.

[8] J. Nye. The distribution of stress and velocity in glaciers and ice-sheets. *Proc. R. Soc. London, Ser. A*, 239(1216):113–133, 1957.

[9] F. Pattyn. A new three-dimensional higher-order thermomechanical ice-sheet model: basic sensitivity, ice stream development, and ice flow across subglacial lakes. *J. Geophys. Res.*, 108(B8, 2382):1–15, 2003.

[10] F. Pattyn. Antarctic sublacial conditions inferred from a hybrid ice sheet/ice stream model. *Earth and Plenetary Science Letters*, 295, 2010.

[11] Pawlowski, R. *et al.* Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: Template-based generic programming. *Sci. Program.*, 20:197–219, 2012.

[12] Perego, M. *et al.* Optimal Initial Conditions for Coupling Ice Sheet Models to Earth System Models. *J. Geophys. Res*, 119:1894–1917, 2014.

[13] Rignot, E. *et al.* Ice Flow of the Antarctic Ice Sheet. *Science*, 333(6048):1427–1430, 2011.

[14] Salinger, A. *et al.* Albany: A Component-Based Partial Differential Equation Code Built on Trilinos. *Comput. Sci. Disc. (in preparation)*, 2014.

[15] Schoof, C. *et al.* Thin-film flows with wall slip: An asymptotic analysis of higher order glacier flows. *Q.J. Mech. Appl. Math.*, 63:73–114, 2010.

[16] Shannon, S. *et al.* Enhanced basal lubrication and the contribution of the greenland ice sheet to future sea-level rise. *P. Natl. Acad. Sci.*, 110:14156–14161, 2013.

[17] Solomon, S. *et al.* Climate change 2007: The physical science basis. Technical report, Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, UK, 2007.

[18] Tuminaro, R. *et al.* A hybrid operator dependent multi-grid/algebraic multi-grid approach: Application to ice sheet modeling. *SIAM J. Sci. Comput. (in preparation)*, 2014.

Acknowledgments

Support for all authors was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy (DOE), Office of Science, Advanced Scientific Computing Research and Biological and Environmental Research Programs. This research used resources of the National Energy Research Scientific Computing Center (NERSC; supported by the Office of Science of the U.S. Department of Energy under Contract DE-AC02-05CH11231) and the Oak Ridge Leadership Computing Facility (OLCF; supported by the DOE Office of Science under Contracts DE-AC02-05CH11231 and DE-AC05-00OR22725). The authors thank Daniel Martin for help in obtaining datasets and for the initial guess used in the Antarctic basal friction coefficient optimization.