

ACHIEVING TEXTBOOK MULTIGRID EFFICIENCY FOR HYDROSTATIC ICE SHEET FLOW

JED BROWN*, BARRY SMITH†, AND ARON AHMADIA‡

Abstract. The hydrostatic equations for ice sheet flow offer improved fidelity compared with the shallow ice approximation and shallow stream approximation popular in today’s ice sheet models. Nevertheless, they present a serious bottleneck because they require the solution of a 3D nonlinear system, as opposed to the 2D system present in the shallow stream approximation. This 3D system is posed on high-aspect domains with strong anisotropy and variation in coefficients, making it expensive to solve with current methods. This paper presents a Newton-Krylov multigrid solver for the hydrostatic equations that demonstrates textbook multigrid efficiency (an order of magnitude reduction in residual per iteration and solution of the fine-level system at a small multiple of the cost of a residual evaluation). Scalability on Blue Gene/P is demonstrated, and the method is compared to various algebraic methods that are in use or have been proposed as viable approaches.

Key words. hydrostatic, ice sheet, Newton-Krylov, multigrid, preconditioning

1. Introduction. The dynamic response of ice streams and outlet glaciers is poorly represented by the shallowness assumptions inherent in the present generation of ice sheet models. Accurate simulation of this response is crucial for predicting sea level rise; indeed, the inability of available models, based on the shallow ice approximation (SIA) [21] and shallow stream approximation (SSA) [26, 38], to simulate these processes was cited as a major deficiency in the Fourth Assessment Report of the Intergovernmental Panel on Climate Change [3].

The hydrostatic equations were introduced by [4] as a model of intermediate complexity between the full non-Newtonian Stokes system and the vertically-integrated SIA and SSA models. A more precise analysis, including the limiting cases of fast and slow sliding, was given in [33]. Well-posedness was proven in [8], and approximation properties of finite-element methods were analyzed in [14, 7]. Hindmarsh [20] conducted a perturbation analysis in one horizontal dimension to study the validity of different continuum models including SIA, hydrostatic, and Stokes. In [30], a model intercomparison was presented using several implementations of the hydrostatic equations for periodic benchmark problems with large amplitude basal topography and stickiness variability. The hydrostatic equations were used for transient simulation in [28] and in 3D models in [29], as well as subsequent work. These equations have been known by various names in the literature, including “Blatter”, “Pattyn”, “Blatter-Pattyn”, “LMLa” [20], and “first order” [18].

The use of hydrostatic equations in current models has been limited, however, by the cost of solving the 3D nonlinear system for velocity, typically accounting for more than 95% of the run time [24]. This cost comes from both slow convergence on the nonlinearities (rheology and slip) and expensive linear solves using standard preconditioners such as incomplete factorization and one-level domain decomposition. The poor linear solve performance is attributable to the strong anisotropy and heterogeneity imposed by the rheology and geometry.

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA (jedbrown@mcs.anl.gov)

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA (bsmith@mcs.anl.gov)

‡Supercomputing Laboratory, King Abdullah University of Science and Technology, Thuwal, Makkah, Saudi Arabia (aron.ahmadia@kaust.edu.sa)

In the present work, we introduce a Newton-Krylov multigrid solver that demonstrates textbook multigrid efficiency, characterized by convergence at a small multiple of the cost of a single, fine-level residual evaluation, and typically involving an order of magnitude reduction in residual per multigrid (V or F) cycle. The scheme converges quadratically on the nonlinearities, is rapidly globalized by using grid sequencing, is robust to parameters and geometry, coarsens rapidly, and exhibits excellent parallel scalability. Our code is freely available as part of the Portable Extensible Toolkit for Scientific computing (PETSc) [2, 1].

Section 2 presents the equations and discretization, Section 3 describes the solver, and Section 4 demonstrates performance and scalability with numerical examples. Section 5 summarizes our conclusions.

2. Equations and Discretization. The hydrostatic equations are obtained from the non-Newtonian Stokes equations in the limit where horizontal derivatives of vertical velocity are small. Neglecting these terms allows incompressibility to be enforced implicitly by eliminating pressure and vertical velocity, leaving a system involving only horizontal components of velocity. See [33] for a rigorous derivation and asymptotic analysis.

Consider an ice domain $\Omega \subset \mathbb{R}^3$ lying between a Lipschitz continuous bed $b(x, y)$ and surface $s(x, y)$ with thickness $h = s - b$ bounded below by a positive constant. Complete ice sheet models generally avoid the singular limit $h \rightarrow 0$ by either extending a fictitious domain over the entire computational domain or truncating the domain at a small positive thickness [5, 24]. The velocity $\mathbf{u} = (u, v) \in \mathbf{V} = \mathbf{W}^{1,1+1/n}(\Omega)$ satisfies conservation of momentum, which, omitting inertial and convective terms as is standard for ice sheets, is given by

$$-\nabla \left[\eta \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \end{pmatrix} \right] + \rho g \nabla s = 0, \quad (2.1)$$

where

$$\eta(\gamma) = \frac{B}{2} (\epsilon^2/2 + \gamma)^{\frac{1-n}{2n}} \quad (2.2)$$

is the nonlinear effective viscosity with regularizing strain rate ϵ and

$$\gamma = u_x^2 + v_y^2 + u_x v_y + \frac{1}{4}(u_y + v_x)^2 + \frac{1}{4}u_z^2 + \frac{1}{4}v_z^2$$

is the second invariant. Ice sheet models typically take $n = 3$ as the power-law exponent, though experimental values range from less than 2 to about 4 [27], as well as more general functional forms [17] with similar properties. Large values of n correspond to strong shear-thinning nonlinearity while $n = 1$ is a linear Newtonian rheology. In the solution of our test examples, the nonlinear effective viscosity η will range over three to four orders of magnitude, smallest in shear margins near the bed and largest in smooth regions near the surface. Equation (2.1) is subject to natural boundary conditions at the free surface and either no-slip $\mathbf{u} = 0$ or power-law slip with friction parameter

$$\beta^2(\gamma_b) = \beta_0^2 (\epsilon_b^2/2 + \gamma_b)^{\frac{m-1}{2}},$$

where $\gamma_b = \frac{1}{2}(u^2 + v^2)$, ϵ_b is regularizing velocity, β_0^2 is a “low-speed” reference friction, and $m \in (0, 1]$ is the exponent that produces Navier slip for $m = 1$, Weertman [37]

sliding for $m = 1/n$, and Coulomb slip as $m \rightarrow 0$. In the present work, we define ϵ and ϵ_b using a strain rate of 10^{-5} a^{-1} and a slip velocity of 1 m a^{-1} , respectively.

To discretize this system with a finite-element method, we introduce the test functions $\phi \in \mathbf{V}$ and integrate by parts to produce the weak form: Find $\mathbf{u} \in \mathbf{V}$ such that

$$\int_{\Omega} \nabla \phi : \eta \mathbf{1} : \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \\ & & v_z \end{pmatrix} + \phi \cdot \rho g \nabla s + \int_{\Gamma_{\text{bed}}} \phi \cdot \beta^2 (|\mathbf{u}|^2/2) \mathbf{u} = 0 \quad (2.3)$$

for all $\phi \in \mathbf{V}$, where Γ_{bed} is the slip portion of the bed.

For our numerical studies, equation (2.3) was discretized on a topologically structured hexahedral grid using Q_1 finite elements and standard 2^3 -point (tensor product) Gauss quadrature. Length, time, and mass units were chosen so that thickness, velocity, and driving stresses are of order 1. See Section 3.2 for details on the enforcement of Dirichlet boundary conditions.

The source code for our implementation is distributed as an example in PETSc [2] versions 3.1 and later.¹ Several generalizations of the tests from ISMIP-HOM [30] are implemented, run with the option `-help` for a complete list of options. We note that our results for “5 km test C” in that paper (variable slipperiness on a flat bed) agree to three significant figures with the “Stokes” results therein. This is consistent with the asymptotic analysis of [33], which shows that slipperiness perturbations are not present in the leading-order error terms for the hydrostatic equations (which are purely geometric); c.f. [30, Table 4 and Figure 8], where the ensemble range is nearly as large as the mean.

3. Solver and Implementation. We begin by writing the discretization of (2.3) as an algebraic system $F(U) = 0$ with Jacobian $J(U)$. This nonlinear system is solved with a Newton iteration that requires an approximate solution δU of

$$J(U)\delta U = -F(U). \quad (3.1)$$

Newton methods are quadratically convergent in the terminal phase but may converge slowly or not at all in early phases. Many applications of the present solver are in a time-stepping code where the initial iterates start within the region of quadratic convergence; thus globalization would rarely be a concern. But since a good initial iterate is not available in the present tests, we use grid sequencing (solving the problem on a sequence of coarser grids) to produce an initial iterate on the fine grid. Globalization is a central and unavoidable issue when solving steady-state problems. Grid sequencing requires a geometric hierarchy of meshes with interpolation operators to move the solution to the next finer level. Managing this hierarchy is often seen as a programming burden, but it exposes more robust algorithms than are available otherwise.

The Newton step (3.1) is solved by a Krylov method such as GMRES, for which the iteration count is highly dependent on the quality of the preconditioner. Since $J(U)$ is symmetric positive definite (SPD), methods such as conjugate gradients could be used. This work always uses GMRES, however, because it allows the use of nonsymmetric preconditioners, and the iteration counts are always kept low so that storage and restarts are not an issue. As an SPD system, it has a wide variety of

¹ Upon unpacking the source, which can be downloaded from <http://mcs.anl.gov/petsc>, see `src/snes/examples/tutorials/ex48.c`.

preconditioners to choose from; however, viscosity contrasts and strong anisotropy cause most preconditioners to perform poorly. The rest of this section describes the methods used to produce a scalable algorithm in spite of these difficulties. Specifically, we examine aspects of geometric multigrid methods with domain decomposition smoothers, which turn out to be the most efficient and robust choices.

3.1. Anisotropic Meshes. The ratio of width to thickness for outlet glaciers (the regions of ice sheets with greatest physical interest) ranges from order 1 to over 100. The nonlinear constitutive relation (2.2) produces three to four orders of magnitude variation in viscosity (usually with fastest variation in the vertical) and the Newton linearization of (2.3) produces additional anisotropy, effectively collapsing the conductivity tensor in the direction of the velocity gradient.

For systems with a priori known anisotropy, semi-coarsening has been successful for attaining satisfactory multigrid performance even with weak smoothers such as SOR, but semi-coarsening is unattractive for two reasons. First, semi-coarsening in the vertical direction would necessitate many levels because it reduces the problem size by only a factor of 2 on each coarsening (instead of 8 for isotropic coarsening) and the fully coarsened problem would still be far too large for a direct solver, necessitating further coarsening in the horizontal direction. The presence of many levels leads to more synchronization in parallel, which is detrimental to scalability and makes performance more sensitive to network latency. Second, viscosity contrasts and anisotropy unaligned with the grid arise when the friction parameter β^2 is not smooth (e.g., test X in Section 4.1), as is the standard case when studying the migration of ice stream margins as the bed transitions from frozen (no-slip or very high friction) to temperate and very slippery depending on subglacial hydrology.

To coarsen the system isotropically even on high-aspect domains, we order the unknowns so that columns are contiguous with a natural block size of 2 (i.e., $\{u_{i,j,k}, v_{i,j,k}, u_{i,j,k+1}, v_{i,j,k+1}, \dots\}$, where k is the index that is increasing in the vertical direction) and not decomposed in parallel. This decomposition is reasonable because the number of vertical levels used in simulations is typically between 10 and 100; it also is convenient because it is compatible with decompositions used by other climate model components.

With this ordering, zero-fill incomplete factorization effectively performs an exact solve of the column since all the neglected fill (relative to an exact factorization) is caused by the coupling with adjacent columns. Pure line smoothers were also tried as a smoother on the finest level, but robustness was significantly impacted, and the memory benefits were deemed insufficient to pursue further.

In scenarios with little sliding and gentle geometry compared with the ice thickness, the “shallow ice approximation” (SIA) is valid, allowing the velocity field to be determined locally from the surface slope and a column integral. If horizontal derivatives were discarded from the linearized hydrostatic equations, the corresponding matrix would have only uncoupled tridiagonal systems for each column, and the solution of these systems would be equivalent to linearized SIA. These tridiagonal systems become singular as basal friction $\beta^2 \rightarrow 0$. Furthermore, SIA is physically inconsistent if β^2 is discontinuous even when it is bounded below by a sufficiently large constant, because of discontinuous velocities [12]. The contribution to the hydrostatic matrix from discrete horizontal derivatives is small (order h_z^2/h_x^2) relative to that from vertical derivatives when the element aspect ratio (h_z/h_x) is small; therefore, high-energy modes of the hydrostatic matrix, corresponding to velocity variation primarily in the vertical, become well approximated by the block tridiagonal system representing SIA.

When the bed is very slippery, there are medium- to long-wavelength modes with vanishing energy in the norm induced by the (nearly singular) SIA equations, but nonvanishing energy in the norm induced by the hydrostatic equations. These modes are problematic if the tridiagonal SIA system is used to precondition hydrostatic because it will take many iterations for the Krylov solver to locate all the modes not controlled by the preconditioner.

Incomplete factorization with column ordering provides nearly exact coupling in the vertical, effectively containing the tridiagonal SIA plus some coupling with nearby columns, and is an effective stand-alone preconditioner when the bed is sticky and horizontal grid spacing is large, despite lack of a coarse level. Indeed, with a typical ice thickness of 1 km resting on a frozen bed and elements 5 km on a side, block Jacobi with zero-fill incomplete Cholesky converges to relative tolerance of 10^{-5} in about 10 Krylov iterations independent of the horizontal extent of the domain (number of elements in the horizontal), independent of the number of elements in the vertical, and independent of the number of subdomains (provided they do not get too small: there is some degradation when subdomain size approaches a single column). However, none of these favorable performance characteristics remain when the elements become small relative to the ice thickness or when the bed becomes slippery, since the usual $\mathcal{O}((L/H)^2)$ condition number for second-order elliptic problems preconditioned by one-level Schwarz methods with subdomains of size H (see [34]) becomes apparent. Indeed, we have found low-fill incomplete factorization to be nearly unusable as part of a one-level additive Schwarz method for problems with slippery beds or steep geometry, even at low resolution, as investigated in Section 4.3.

3.2. Dirichlet Boundary Conditions. Multigrid is often sensitive to the enforcement of boundary conditions. Ideally, Dirichlet conditions would be completely removed from the solution space, but doing so complicates grid management on structured grids. Instead, we leave these degrees of freedom in the system but decouple them from the other equations. During residual evaluation in the finite-element context, this strategy corresponds to evaluating integrals with the Dirichlet condition satisfied exactly and setting the residual on the Dirichlet nodes to be equal to a multiple of the current solution. With this scheme, all rows and columns of the Jacobian corresponding to Dirichlet nodes are zero except for a single diagonal entry. Thus the system retains symmetry, and satisfaction of the Dirichlet conditions does not interfere with solving the other equations. For good multigrid performance, the diagonal entry should be similar to the diagonal entry of the Jacobian for nearby nodes. To ensure this, we set the residual at Dirichlet nodes to

$$\begin{aligned} f_u &= 2\eta h_x h_y h_z (4/h_x^2 + 1/h_y^2 + 1/h_z^2)u \\ f_v &= 2\eta h_x h_y h_z (1/h_x^2 + 4/h_y^2 + 1/h_z^2)v, \end{aligned} \tag{3.2}$$

where h_x, h_y, h_z are the local element dimensions. This scaling produces the same diagonal entries that would appear if the domain was extended so that constant viscosity momentum equations appeared at the formerly Dirichlet nodes.

3.3. Matrices. The most expensive operations are Jacobian assembly and sparse matrix kernels such as matrix-vector multiplication and solves with incomplete triangular factors. The former involve evaluation of fractional powers and finite element quadrature loops. While fractional powers take most of the time for residual evaluation, they are less significant than quadrature loops for assembly. The quadrature loops were explicitly vectorized by using SSE2 intrinsics, which led to a 30% speedup on

TABLE 3.1

Throughput (Mflop/s) for different matrix formats on Core 2 Duo (P8700) and Opteron 2356 (two sockets). MatSolve is a forward- and back-solve with incomplete Cholesky factors. The AIJ format is using “inodes,” which unrolls across consecutive rows with identical nonzero pattern (pairs in this case). BAIJ is statically blocked and SBAIJ is symmetric and statically blocked.

Kernel	Core 2, 1 process			Opteron, 4 processes		
	AIJ	BAIJ	SBAIJ	AIJ	BAIJ	SBAIJ
MatMult	812	985	1507	2226	2918	3119
MatSolve	718	957	955	1573	2869	2858

Core 2 and Opteron architectures using both GNU and Intel compilers. No manual vectorization was used for the Blue Gene/P results quoted in Section 4.2.

Assembly costs could be further mitigated by recomputing the matrix less frequently, either by using a modified Newton method (degrades nonlinear convergence rate) or by applying the current operator matrix-free by finite differencing the residual or by using automatic differentiation, in which case only the preconditioner is lagged. These are runtime options in the present code; but since they do not offer a clear benefit, they have not been pursued in the present work. If matrix-free application of the true Jacobian is used, several other preconditioning options become available without impacting the nonlinear convergence rate. One could assemble only the block-tridiagonal column coupling, ignoring horizontal coupling and thus saving the memory for the finest level(s). Additionally, a truly 2D coarse problem can be defined by using the shallow stream equations [26, 38, 32] and restriction operators defined by integrating the entire column. These possibilities are also runtime options but have not exhibited a level of robustness comparable to that of the more conventional methods pursued here. Specifically, with this 2D coarse level, the required number of iterations was often more than 10 and required problem-specific tuning.

The Jacobian is always symmetric positive definite and has a natural block size of 2, so we use a symmetric block format (PETSc’s SBALJ). This format stores one column index per 2×2 block in the upper triangular part of the matrix and therefore uses about half the storage of the nonsymmetric BAIJ format, which in turn uses 25% less memory than a scalar format (AIJ). Multiplication for symmetric storage requires twice as much parallel communication as nonsymmetric storage, albeit with the same number of messages. In return, the diagonal part of a parallel decomposition does twice as much work per matrix entry and thus achieves higher throughput, as shown in Table 3.1.

Matrices on the coarse levels of multigrid methods can be constructed in two ways. The first, which we use in almost all our numerical examples, is to rediscretize the system on the coarse mesh. In our implementation, this involves re-evaluating nonlinearities on each level of the hierarchy, although restricting fine-level coefficients of the linearized problem would also be possible. This procedure produces coarse operators that are inexpensive to construct and as sparse as possible on each of the levels. The Galerkin procedure is an alternative that is mandatory for algebraic multigrid. Given an interpolation operator $P : \mathbf{V}_{\text{coarse}} \rightarrow \mathbf{V}_{\text{fine}}$ and assembled fine-level matrix A_{fine} , the Galerkin coarse operator is $A_{\text{coarse}} = P^T A_{\text{fine}} P$. Computing the sparse matrix product in parallel involves significant communication and irregular memory access, accounting for a large fraction of the setup cost in most algebraic multigrid methods.

4. Numerical Examples. We present several numerical examples that demonstrate the algorithmic and parallel scalability of the Newton-Krylov multigrid approach. Standard second-order accurate interpolation and restriction operators are used for the geometric methods, though comparisons are made to algebraic multigrid methods that construct operator-dependent interpolation. The chosen smoothers are relatively inexpensive yet allow more rapid coarsening than would be possible with pointwise smoothers. All configurations discussed use only one smoothing iteration; multiple steps were not found to be faster.

4.1. Algorithmic Scalability. We consider three model problems inspired by the periodic domain ISMIP-HOM [30] tests. All use surface $s(x, y) = -x \sin \alpha$, where α is the surface slope (the coordinate system is not rotated) and a bed similar to $b_A(x, y) = s(x, y) - 1000 \text{ m} + 500 \text{ m} \cdot \sin \hat{x} \sin \hat{y}$ for $(x, y) \in [0, L]^2$ with $\hat{x} = 2\pi x/L, \hat{y} = 2\pi y/L$. Test X uses bed $b_X = b_A$ and stickiness parameter

$$\beta_X^2(x, y) = \begin{cases} 2000 \text{ Pa a m}^{-1}, & \text{if } r = |(\hat{x}, \hat{y}) - (\pi, \pi)| < 1 \\ 0, & \text{otherwise} \end{cases}$$

which is free slip except for a sticky circle at the center of the domain, which is not aligned with the grid. A solution cutout for $L = 10 \text{ km}$ is shown in Figure 4.1. This problem exhibits shear localization at the edges of the sticky region and is most extreme at high aspect ratio. We choose $L = 80 \text{ km}$ and $\alpha = 0.05^\circ$, which produce velocities from 0.9 km a^{-1} to 47 km a^{-1} .² In order to demonstrate algorithmic scalability as the problem size is increased, this test was run on 8 processors starting from a coarse grid of $16 \times 16 \times 1$, refining twice in the horizontal by factors of 2 in both x and y , then three times in the vertical by factors of 8 each to reach a fine mesh of $64 \times 64 \times 513$, which has elements of nominal dimension $1250 \times 1250 \times 1.95$ meters. Block Jacobi with zero fill incomplete Cholesky in subdomains is used as a smoother.

A visual representation of the nonlinear solve process is shown in Figure 4.2. In this example and the next one, Luis Chacón’s variant of the Eisenstat-Walker [10] method was used to automatically adjust linear solve tolerances as the nonlinear solve converges. Solving the linear systems to higher tolerance would have little impact on the number of nonlinear iterations and would be visible in the form of more \times marks below the solid lines. That most \times marks lie on the solid line for nonlinear residual is an indication that effort is well balanced between linear and nonlinear solves. Note that approximately 10 linear V-cycles on the fine level are required to reduce the residual by 10 orders of magnitude. Across these 10 orders of magnitude, the convergence is fastest before discretization error is reached, but it is still steady and close to an order of magnitude per V-cycle all the way to machine precision. We remark that Picard iteration takes at least 50 iterations to reach this tolerance (cf. [9], in which hundreds or thousands of iterations were needed for an easier problem). When used in a time-stepping code, the solution at the last time step is available; hence, typically only the terminal convergence rate is important, which further favors Newton methods. For example, an initial guess might have a residual 10^3 higher than the required tolerance, which takes 15 Picard iterations to solve, but only one Newton iteration. Additionally, each linear solve for this fine-level problem requires hundreds or thousands of iterations with a one-level additive Schwarz method; see Section 4.3, which considers a smaller problem.

² This problem may be run with the options `-thi_hom X -thi_L 80e3 -thi_alpha 0.05`; the other cases can be selected with similar options.

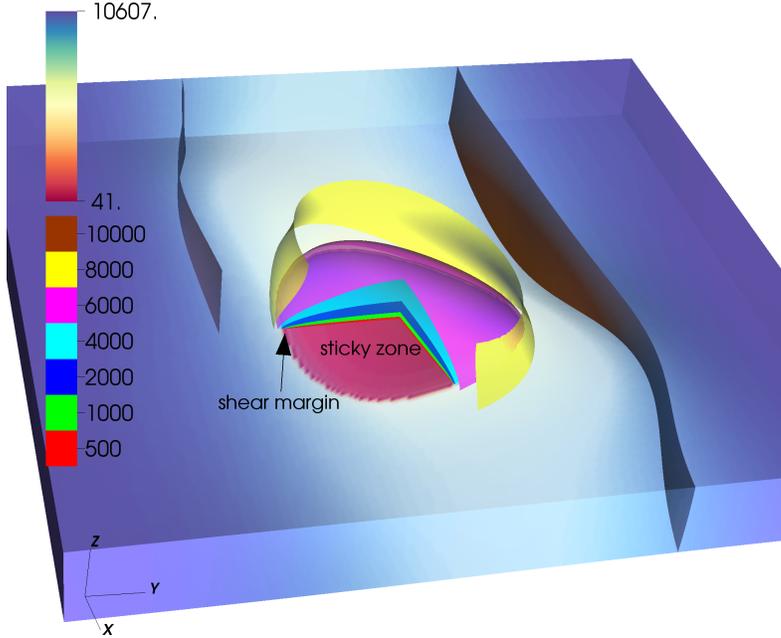


FIG. 4.1. Contours of velocity magnitude for test X at $L = 10$ km with $m = 1/10$ nearly plastic basal yield model. The surface slopes downward in the positive x -direction. The velocity ranges from 41 m a^{-1} at the bed in the sticky region to 10 km a^{-1} in the slippery region away from the sticky zone.

Test Y places a 200 m tower with vertical walls on the top of each bedrock bump and uses an uncorrelated but smoothly varying stickiness resembling a dimpled sombrero:

$$b_Y(x, y) = \begin{cases} b_A(x, y), & \text{if } b_A(x, y) < -700 \text{ m} \\ b_A(x, y) + 200 \text{ m}, & \text{otherwise} \end{cases}$$

$$\beta_Y^2(x, y) = 1000 \text{ Pa a m}^{-1} \cdot (1 + \sin(\sqrt{16r})/\sqrt{10^{-2} + 16r}) \cos \frac{3\hat{x}}{2} \cos \frac{3\hat{y}}{2}.$$

This tests the quality of the coarse grids even when large geometric errors are committed. Note that the hydrostatic equations cannot be considered valid in this regime because the topography is too abrupt. Such topography is present in reality, however, so we may still desire an efficient solver. Figure 4.3 depicts the solve for this problem in a 10 km square domain. Because of the successively better resolution of the “cliff,” performance deteriorates on each level, as can be seen by the closer spacing of linear solve marks (\times). It is acceptable up to level 3, however, where the elements are approximately 12 m thick and stretch to reach over a 200 m vertical cliff in 125 m horizontal, a slope of 58° . On the finest level, they are 6 m thick and stretch over the 200 m vertical cliff in 62 m horizontal, a slope of 73° . The approximation properties of such deformed elements are poor, the continuum equations are invalid for such steep topography, and the resolution necessary to resolve such steep features (which are rare and localized in nature) is well beyond anything that has been proposed for ice sheet modeling. We note that substituting Galerkin coarse operators in place of the rediscritized operators used in Figure 4.3 improves the linear convergence rate by nearly four times in the most extreme case, but the convergence behavior is almost

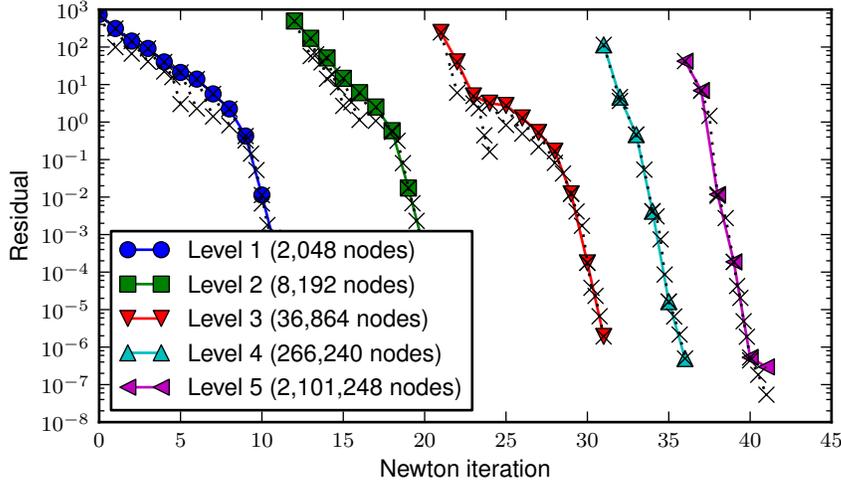


FIG. 4.2. Grid-sequenced Newton-Krylov solution of test X. The solid lines denote nonlinear iterations, and the dotted lines with \times denote linear residuals.

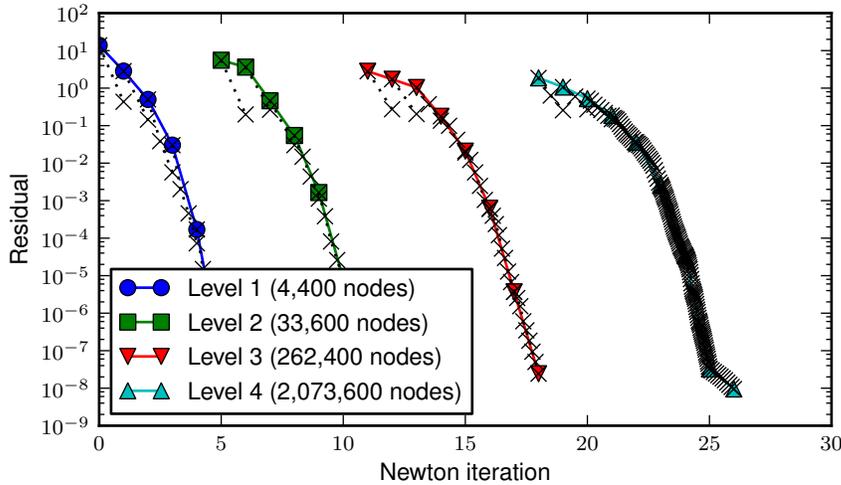


FIG. 4.3. Grid sequenced Newton-Krylov convergence for test Y at $L = 10$ km. Each successively finer grid better resolves the 200 m vertical cliffs in this problem. From left to right, the resolved cliff angles are 22° , 39° , 58° , and 73° . The slower linear convergence on steeper grids demonstrates the “breaking point” of rediscritized coarse level operators.

identical for resolved cliff angles less than 45° .

Test Z sets $b_Z = b_A$, $\beta_Z^2 = \beta_X^2$, and nonlinear sliding with exponent $m = 0.3$. It is a regime where the hydrostatic equations are valid, provided the wavelength L is not too small. We use this case to explore linear solve performance in Figure 4.4.

4.2. Parallel Scalability on Blue Gene/P. Two types of parallel scalability are considered. Strong scalability is the ability to solve a fixed problem size faster by

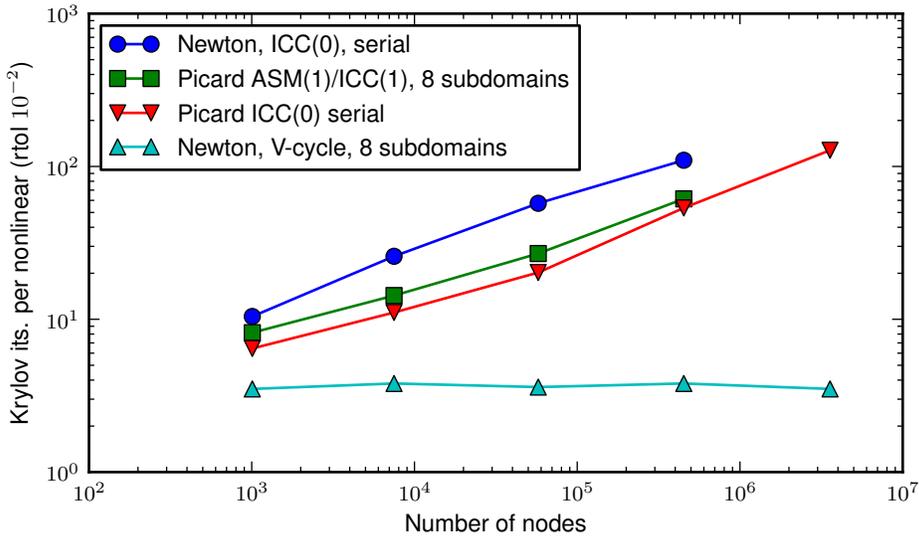


FIG. 4.4. Average number of Krylov iterations per nonlinear iteration for test Z . Each linear system is solved to a relative tolerance of 10^{-2} ; the nonlinear problem is solved to a relative tolerance of 10^{-8} . Note that many more nonlinear iterations are required for Picard than Newton.

using more processors. It is represented by a log-log plot of time versus number of processors, where a slope of -1 is optimal. Weak scalability is the ability to solve larger problems in constant time by using more processors with the problem size per processor remaining fixed. It is represented by a plot of time versus number of processors, where a slope of 0 is optimal. Strong scalability is important in transient simulations with many time steps; weak scalability is more important for attaining high resolution in steady-state simulations or transient simulations with fewer time steps. Optimal strong and weak scalability together would imply that a problem with N elements could be solved in $\mathcal{O}(N/P)$ time on P processors.

We investigate strong scaling using test Z at 80 km with basal friction exponent $m = 0.3$ on Shaheen, a Blue Gene/P at the KAUST Supercomputing Laboratory. Shaheen is configured with 4 GB of memory per node, and all tests were run with four MPI processes per node (VN mode). Figure 4.5 shows strong scalability for four fixed global problem sizes with coarse meshes of $16 \times 16 \times 3$, $32 \times 32 \times 3$, $64 \times 64 \times 3$, and $64 \times 64 \times 1$. The first three use five levels of isotropic refinement to reach target meshes of $256 \times 256 \times 48$, $512 \times 512 \times 48$, and $1024 \times 1024 \times 48$, the latter with nominal element sizes of $78 \times 78 \times 21$ meters. The last coarse mesh is refined anisotropically and more aggressively to reach $1024 \times 1024 \times 64$ with only four levels. The smallest two coarse problems are solved redundantly, and the second is also solved by using the “ XX^T ” direct solver of [36] (TFS), which exhibits significantly better scalability but for which the coarse-level problem still presents a bottleneck. The largest two coarse problems are solved only approximately by using a V-cycle of the algebraic multigrid package BoomerAMG [19], which removes the severe scalability bottleneck of a direct coarse-level solve. These coarse problems have been designed so that BoomerAMG is effective; see Section 4.3 for BoomerAMG applied to fine-level problems.

Figure 4.6 shows weak scalability for test X at 40 km with basal friction exponent

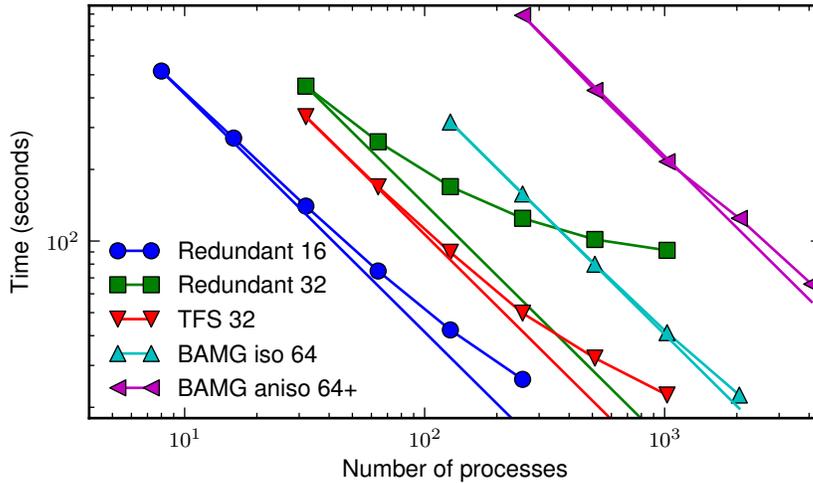


FIG. 4.5. Strong scaling on Shaheen for test Z with different-size coarse-level problems and different coarse-level solvers (see text for details). The straight lines on the strong scaling plot have slope -1 , which is optimal. Grid sequencing is used, but only the nonlinear solve on the finest level is shown because strong scalability is most important when many time steps are needed.

$m = 0.2$ on Shaheen. The size of the coarse grid was held constant, and additional levels were added as the number of processes was increased, such that the subdomain sizes remain approximately constant. As the resolution increases, the nonlinearity strength and linear stiffness also changes, explaining the relatively expensive solve with four processes. This effect is also apparent in the Level 3 convergence of Figure 4.2. Four phases of the solution process dominate the run time and are identified in Figure 4.6. Multigrid setup costs account for less time than does nonlinear residual evaluation for the larger problem sizes.

4.3. Algebraic Methods. Building a geometric hierarchy with rediscrization on coarse levels adds software complexity that many developers of numerical models do not want to deal with. In this section, we summarize the performance characteristics of several popular algebraic methods. We consider test X , $L = 80$ km, $\alpha = 0.03^\circ$, with $40 \times 40 \times 12$ elements distributed over four processes. We compare our multigrid method with several one-level domain decomposition methods, two algebraic multigrid methods, and field-split approaches. This problem is challenging for the standard Newton iteration, which requires 37 iterations and should be accompanied by grid sequencing for efficiency. It takes the problem through a range of nonlinearities, however. Thus, the average number of Krylov iterations to solve each Newton step to a relative tolerance of 10^{-5} , presented below, is a good test of the linear solver.

We first consider one-level domain decomposition methods with incomplete factorization, which are currently used to solve the hydrostatic equations by [11, 24] (among others). To keep iteration counts representative, we use full GMRES (no restart) with modified Gram-Schmidt orthogonalization (note that neither is practical for production use). Conventional symmetric additive Schwarz is denoted $ASM(k)$, where k is the overlap, restricted additive Schwarz [6] is denoted $RASM(k)$. The average number of GMRES iterations per Newton iteration is shown in Table 4.1. Note that increasing

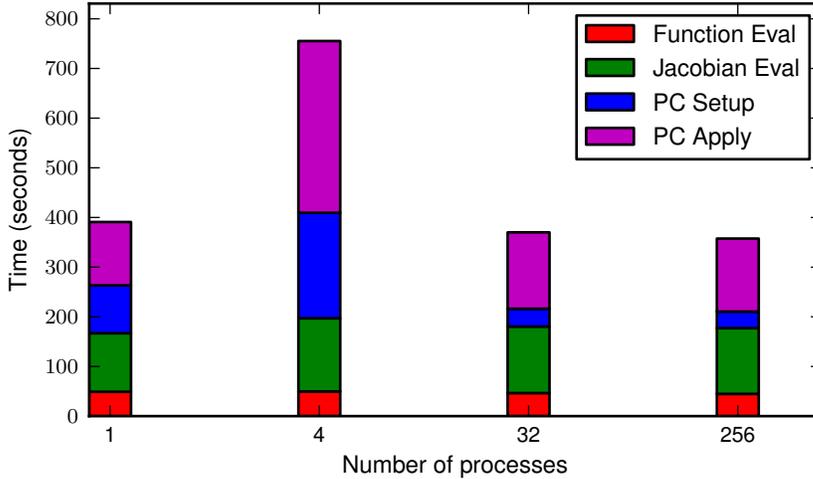


FIG. 4.6. Weak scaling on Shaheen for test X at $L = 40$ km with a breakdown of time spent in different phases of the solution process. Times are for the full grid-sequenced problem instead of just the finest-level solve.

TABLE 4.1

Average number of GMRES iterations per Newton iteration (each solved to $\text{rtol} = 10^{-5}$) for one-level domain decomposition applied to test X with $L = 80$ km with different overlap and fill.

Decomposition	Subdomain Solver			
	ICC(0)	ICC(1)	ICC(4)	Cholesky
Block Jacobi	367	315	220	97
ASM(1)	508	441	296	59
RASM(1)	368	306	190	52
ASM(2)	521	445	316	44
RASM(2)	365	305	189	38

overlap has no benefit when incomplete subdomain solvers are used.

The parallel algebraic multigrid packages ML [13] and BoomerAMG [19] provide potentially scalable alternatives. ML is based on smoothed aggregation, tends to coarsen rapidly, and provides its restriction and coarse-level matrices to PETSc so that that elaborate smoothers can be used. ML does not converge for this problem with standard options; but with FGMRES on the outside of the V-cycle and GMRES(1) with RASM(1) as the smoother, using ICC(0) for the subdomain solve except on level 1, where a direct solve was used, we see 34 V-cycles per Newton iteration. ML needs only three levels to reach a coarse level with 144 degrees of freedom. BoomerAMG is a classical algebraic multigrid, which tends to coarsen slowly on anisotropic problems and does not expose the internal details, so smoother choices are limited. BoomerAMG needs seven levels to reach a coarse grid with 663 degrees of freedom and averages 76 iterations per Newton iteration. In other, still somewhat challenging, problems, BoomerAMG was competitive with geometric multigrid in terms of iteration count, but the setup costs and required number of levels were always large. ML never exhibited

TABLE 4.2

Average number of GMRES iterations per Newton iteration (each solved to $\text{rtol} = 10^{-5}$) for field-split preconditioners applied to problem X at $L = 80$ km with different ways of combining the splits and different solvers within the splits. BoomerAMG used 7 levels and ML had 3 with the same solver parameters as discussed in the text for the coupled approach.

Solver in Splits	Method of Combining Splits		
	Additive	Multiplicative	Sym. Multiplicative
Cholesky	19	9.9	9.3
ML	41	34	30
BoomerAMG	89	83	78
RASM(1)+Cholesky	186	173	84

low iteration counts, even for easy problems.

Another approach to solving multicomponent problems is to split the components and solve scalar problems for each in the hope that the scalar problems can be more readily handled by available software such as algebraic multigrid. The split problems can be combined additively, multiplicatively, or symmetric multiplicatively. Unlike most Schwarz methods, additive methods are not typically implemented to expose concurrency, but they are simpler to implement in a matrix-light way because only the “self-coupling” terms need to be made available. Multiplicative methods need to apply the off-diagonal submatrix, and the most efficient way to do so is usually by assembling it; but the submatrix can also be applied (at higher cost) by finite differencing the full residual and restricting the result. The results, shown in Table 4.2, are uninspiring, especially when considering that field-split creates additional synchronization points and attains lower throughput because it works with scalar matrices instead of block matrices (see Table 3.1).

A good geometric multigrid for this problem uses four levels with a coarse grid of $10 \times 10 \times 2$ elements, which is semi-refined twice by a factor of 2 in each horizontal direction, then by a factor of 6 in the vertical to reach the target $40 \times 40 \times 12$ grid. The smoothers consist of a domain decomposition method and a subdomain solver that may be exact or inexact. Direct solves for the subdomain problems on level 1 are inexpensive and tend to improve robustness, so we always use a direct solve. A different refinement involves a $10 \times 10 \times 1$ coarse grid and semi-refines twice by a factor of 2 in each horizontal direction, then by a factor of 12 in the vertical to reach the same target grid. The coarse levels are smaller in this case, so the refinement is more efficient provided the iteration counts are similar. Table 4.3 explores a variety of multigrid preconditioner configurations with each coarse level. A distinct effect is that inexact subdomain solvers cause minimal performance degradation; cf. Section 4.1, where a factor of 5 to 10 degradation is observed for one-level methods. Note that semi-coarsening with standard smoothers would normally use a factor of 2 or 3; but with a smoother that is strong in the semi-coarsened direction (see Section 3.1), the convergence rate is fast for large factors such as 12. The poor performance of symmetric additive Schwarz as a smoother for this problem is likely due to introducing high-frequency error in the overlapping region, an effect that was observed indirectly in the original paper on restricted additive Schwarz [6]. Block Jacobi and restricted additive Schwarz are robust to changes in resolution, spatial domain, and number of processes. These tests have also been run with Galerkin coarse-level operators, and the convergence rates (not shown) were found to be comparable to rediscretized operators

TABLE 4.3

Average number of GMRES iterations per Newton iteration (each solved to $\text{rtol} = 10^{-5}$) for different multigrid preconditioners applied to problem X , $L = 80$ km.

Coarse Problem	Level 1		Level 2 and 3		Its
	Decomp.	Subdomain	Decomp.	Subdomain	
$10 \times 10 \times 2$	BJacobi	Cholesky	BJacobi	Cholesky	8.9
$10 \times 10 \times 2$	BJacobi	Cholesky	BJacobi	ICC(0)	9.6
$10 \times 10 \times 2$	BJacobi	Cholesky	ASM(1)	ICC(0)	11.9
$10 \times 10 \times 2$	BJacobi	Cholesky	RASM(1)	ICC(0)	6.9
$10 \times 10 \times 2$	ASM(1)	Cholesky	ASM(1)	ICC(0)	20.2
$10 \times 10 \times 2$	RASM(1)	Cholesky	RASM(1)	ICC(0)	5.9
$10 \times 10 \times 1$	BJacobi	Cholesky	BJacobi	ICC(0)	10.3
$10 \times 10 \times 1$	ASM(1)	Cholesky	BJacobi	ICC(0)	10.1
$10 \times 10 \times 1$	RASM(1)	Cholesky	BJacobi	ICC(0)	6.9

in all cases (identical or within one iteration). The Galerkin procedure can thus be considered a robust alternative for producing coarse-level operators if rediscrretization is not convenient, although the setup cost is higher, and there is no practical analogue of grid sequencing when rediscrretization is not available.

We remark that when using grid sequencing and the method in the last row of Table 4.3, the problem can be solved in seven Newton iterations on the fine level and an average of 5.4 V-cycles per Newton iteration. If the Eisenstat-Walker method is used to avoid oversolving, the problem takes eight Newton iterations with a total of 12 V-cycles.

5. Conclusion. Several projects [25, 23, 22, 9, 29] have developed software to simulate ice sheets using the hydrostatic equations. All these projects solve the resulting algebraic equations by using one-level domain decomposition (if parallel) and incomplete factorization, and all but [25] use Picard linearization. Since these methods are not scalable in the strong or weak sense, resolution and simulation turn-around time have been severely limited by the cost of the solver.

We have presented a grid-sequenced Newton-Krylov multigrid algorithm for solving the algebraic equations resulting from a finite-element discretization of the hydrostatic equations. This geometric multigrid method demonstrates textbook multigrid efficiency for extreme topography and basal conditions and offers thousandfold speedups relative to the methods currently in use. Algebraic multigrid and field-split preconditioners were not found to be competitive in terms of robustness or efficiency.

The present velocity solver can be immediately incorporated in a semi-implicit ice sheet evolution model, but the resulting scheme is only conditionally stable (time steps must satisfy a CFL condition). Fully implicit time integration offers an alternative that has no such constraint and is thus suitable for simulations over full ice ages and longer, but it requires the solution of more difficult algebraic equations. A prototype using the present velocity discretization has been developed, but scalable linear solvers for velocity-surface coupled problems is the subject of future investigation.

Several models involving only a vertically-integrated elliptic solve have been proposed as less costly alternatives to solving the hydrostatic equations, (e.g., [5, 16, 15]). A scalable solver has not yet been proposed for these vertically-integrated elliptic problems, however, the present model is faster than existing implementations

at sufficiently high resolution. To make a useful comparison, we estimate relative costs with the assumption that a similarly optimal solver exists for the vertically-integrated system, in which case the cost difference should be a constant factor independent of problem size and number of processors. The “hybrid” model of [15] requires evaluating effective viscosity in 3D and integrating to produce depth-averaged viscosity, which appears in the vertically-integrated equations. Evaluating effective viscosity accounts for most of the cost of residual evaluation for the hydrostatic equations. Therefore the cost of such an evaluation is comparable to the cost of the nonlinear viscosity evaluation required in a Picard iteration for the hybrid model of [15]. Our solver produces high-accuracy solutions in about 30 times the cost of a residual evaluation, less if a good initial guess was available, such as in transient simulations. Goldberg [15] uses a Picard iteration to solve the nonlinear system, requiring 50 iterations per nonlinear solve; therefore the hybrid solver is likely to have similar cost to our hydrostatic solver only when the linear solves require negligible time. If the hybrid equations were instead solved by Newton iteration and the linear solves required time similar to or less than that for evaluating viscosity, then the hybrid equations might be solved as much as 5 times faster than the hydrostatic equations. Bueller and Brown [5] factor depth-dependent velocity and temperature contributions out of the vertically-integrated nonlinear problem. The process applies only if the constitutive relation can be factored, however, the resulting continuum equations have a lower formal order of accuracy with respect to shallowness and stickiness distribution than the model of [15]. In exchange, the nonlinear elliptic solve can be completed without revisiting the 3D velocity field, so the model cost has weaker dependence on the number of nonlinear iterations and thus potential to be a few times faster than the model of [15]. Vertically-integrated models do not involve linear algebra operating on the 3D velocity field, so they require significantly less memory than do the hydrostatic equations when all operators are assembled.

The hydrostatic equations are well suited to investigation of ice streams and outlet glaciers with gentle slope and arbitrary stickiness distribution. For example, 3D ice stream thermodynamics (see [31] for analytic and [35] for numerical investigation with along-flow symmetry) could be reliably investigated by using velocity defined by the hydrostatic equations. The methods in this paper can efficiently solve the discrete hydrostatic equations in regimes with extreme basal topography and stickiness distributions. The solution method is efficient even in regimes that are far in excess of where the continuum hydrostatic approximation is valid as an approximation to the Stokes equations. The latter issue of continuum validity has been partially addressed by using asymptotic analysis [33] and numerical comparison [20, 30]. Despite being more complete than models involving only a vertically-integrated elliptic solve, the hydrostatic equations are not suitable for steep outlet glaciers such as Jakoshavn Isbræ or Pine Island Glacier, where solution of Stokes problems appears to be unavoidable.

Comprehensive log files and the scripts used to produce the figures in this paper are publicly available. They are currently hosted along with the source for this document at <https://github.com/jedbrowm/tme-ice>.

Acknowledgments. We are grateful to Edward L. Bueller for his helpful commentary on an early draft of this paper. We also thank two anonymous reviewers for valuable feedback. This work was supported by Swiss National Science Foundation Grant 200021-113503/1, U.S. Department of Energy’s Office of Science Ice Sheet Initiative for CL-imate ExtremeS program under Contract DE-AC02-06CH11357, and

the Shaheen Supercomputing Laboratory at KAUST.

REFERENCES

- [1] S. BALAY, J. BROWN, K. BUSCHELMAN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.
- [2] ———, *PETSc Web page*. <http://www.mcs.anl.gov/petsc>, 2012.
- [3] L. BERNSTEIN ET AL., *Climate Change 2007: Synthesis report*, Intergovernmental Panel on Climate Change, 2008. www.ipcc.ch/ipccreports/ar4-syr.htm.
- [4] H. BLATTER, *Velocity and stress fields in grounded glaciers: A simple algorithm for including deviatoric stress gradients*, *Journal of Glaciology*, 41 (1995), pp. 333–344.
- [5] E. BUELER AND J. BROWN, *Shallow shelf approximation as a “sliding law” in a thermomechanically coupled ice sheet model*, *Journal of Geophysical Research-Earth Surface*, 114 (2009), p. F03008.
- [6] X. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, *SIAM Journal on Scientific Computing*, 21 (1999), p. 797.
- [7] S. CHOW, G. CAREY, AND M. ANDERSON, *Finite element approximations of a glaciology problem*, *Mathematical Modelling and Numerical Analysis*, 38 (2004), pp. 741–756.
- [8] J. COLINGE AND J. RAPPAZ, *A strongly nonlinear problem arising in glaciology*, *Mathematical Modelling and Numerical Analysis*, 33 (1999), pp. 395–406.
- [9] B. DE SMEDT, F. PATTYN, AND P. DE GROEN, *Using the unstable manifold correction in a Picard iteration to solve the velocity field in higher-order ice-flow models*, *Journal of Glaciology*, 56 (2010), pp. 257–261.
- [10] S. C. EISENSTAT AND H. F. WALKER, *Choosing the forcing terms in an inexact Newton method*, *SIAM Journal on Scientific Computing*, 17 (1996), pp. 16–32.
- [11] K. EVANS, A. SALINGER, E. BARKER, D. HOLLAND, D. KNOLL, J. F. LEMIEUX, B. LIPSCOMB, R. NONG, S. PRICE, K. RODDY, T. WHITE, AND P. WORLEY, *A Scalable, Efficient, and Accurate Community Ice Sheet Model*. www.csm.ornl.gov/SEACISM, 2010.
- [12] A. FOWLER, *Modelling the flow of glaciers and ice sheets*, *Continuum Mechanics and Applications in Geophysics and the Environment*, (2001), pp. 201–221.
- [13] M. GEE, C. SIEFERT, J. HU, R. TUMINARO, AND M. SALA, *ML 5.0 smoothed aggregation user’s guide*, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.
- [14] R. GLOWINSKI AND J. RAPPAZ, *Approximation of a nonlinear elliptic problem arising in a non-Newtonian fluid flow model in glaciology*, *Mathematical Modelling and Numerical Analysis*, 37 (2003), pp. 175–186.
- [15] D. GOLDBERG, *A variationally derived, depth-integrated approximation to a higher-order glaciological flow model*, *Journal of Glaciology*, 57 (2011), pp. 157–170.
- [16] D. GOLDBERG, D. HOLLAND, AND C. SCHOOF, *Grounding line movement and ice shelf buttressing in marine ice sheets*, *Journal of Geophysical Research*, 114 (2009), p. F04026.
- [17] D. GOLDSBY AND D. KOHLSTEDT, *Superplastic deformation of ice: Experimental observations*, *Journal of Geophysical Research*, 106 (2001), pp. 11017–11030.
- [18] R. GREVE AND H. BLATTER, *Dynamics of ice sheets and glaciers*, Springer Verlag, 2009.
- [19] V. HENSON AND U. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, *Applied Numerical Mathematics*, 41 (2002), pp. 155–177.
- [20] R. HINDMARSH, *A numerical comparison of approximations to the Stokes equations used in ice sheet and glacier modeling*, *Journal of Geophysical Research*, 109 (2004), p. F01012.
- [21] K. HUTTER, *Theoretical glaciology: Material science of ice and the mechanics of glaciers and ice sheets*, Springer, 1983.
- [22] J. JOHNSON AND J. STAIGER, *Modeling long-term stability of the Ferrar Glacier, East Antarctica: Implications for interpreting cosmogenic nuclide inheritance*, *Journal of Geophysical Research*, 112 (2007), p. F03S30.
- [23] E. LAROUR, M. MORLIGHEM, AND H. SEROUSSI, *Ice Sheet System Model*, 2010. issm.jpl.nasa.gov.
- [24] E. LAROUR, H. SEROUSSI, M. MORLIGHEM, AND E. RIGNOT, *Modeling the flow of glaciers in steep terrains: The integrated second-order shallow ice approximation (iSOSIA)*, *Journal of Geophysical Research*, 117 (2012).
- [25] J.-F. LEMIEUX, S. F. PRICE, K. J. EVANS, D. KNOLL, A. G. SALINGER, D. M. HOLLAND, AND A. J. PAYNE, *Implementation of the Jacobian-free Newton-Krylov method for solving the first-order ice sheet momentum balance*, *Journal of Computational Physics*, 230 (2011), pp. 6531–6545.

- [26] L. MORLAND, *Unconfined ice-shelf flow*, in Dynamics of the West Antarctic ice sheet, C. van der Veen and J. Oerlemans, eds., Kluwer Academic Publishers, 1987, pp. 99–116.
- [27] W. PATERSON, *Physics of glaciers*, Butterworth-Heinemann, 3rd ed., 1998.
- [28] F. PATTYN, *Transient glacier response with a higher-order numerical ice-flow model*, Journal of Glaciology, 48 (2002), pp. 467–477.
- [29] ———, *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 (2003), pp. 10–1029.
- [30] F. PATTYN, L. PERICHON, A. ASCHWANDEN, B. BREUER, B. DE SMEDT, O. GAGLIARDINI, G. GUDMUNDSSON, R. HINDMARSH, A. HUBBARD, J. JOHNSON, T. KLEINER, Y. KONOVALOV, C. MARTIN, A. PAYNE, D. POLLARD, S. PRICE, M. RÜCKAMP, F. SAITO, O. SOUČEK, S. SUGIYAMA, AND T. ZWINGER, *Benchmark experiments for higher-order and full Stokes ice sheet models (ISMIP-HOM)*, The Cryosphere, 2 (2008), pp. 95–108.
- [31] C. RAYMOND, *Energy balance of ice streams*, Journal of Glaciology, 46 (2000), pp. 665–674.
- [32] C. SCHOOF, *A variational approach to ice stream flow*, Journal of Fluid Mechanics, 556 (2006), pp. 227–251.
- [33] C. SCHOOF AND R. HINDMARSH, *Thin-film flows with wall slip: an asymptotic analysis of higher order glacier flow models*, The Quarterly Journal of Mechanics and Applied Mathematics, 63 (2010), pp. 73–114.
- [34] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
- [35] M. TRUFFER AND K. ECHELMAYER, *Of Isbræ and ice streams*, Annals of Glaciology, 36 (2003), pp. 66–72.
- [36] H. TUFO AND P. FISCHER, *Fast parallel direct solvers for coarse grid problems*, Journal of Parallel and Distributed Computing, 61 (2001), pp. 151–177.
- [37] J. WEERTMAN, *On the sliding of glaciers*, Journal of Glaciology, 3 (1957), pp. 33–38.
- [38] M. WEIS, R. GREVE, AND K. HUTTER, *Theory of shallow ice shelves*, Continuum Mechanics and Thermodynamics, 11 (1999), pp. 15–50.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.