Achieving textbook multigrid efficiency for hydrostatic ice sheet flow

Jed Brown[a], Barry Smith[b], Aron Ahmadia[c]

[a] *Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW), ETH Zürich, 8092 Zürich, Switzerland*
[b] *Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439*
[c] *KAUST-IBM Center for Deep Computing Research, Bldg #1, Office 01-123, 4700 King Abdullah University of Science and Technology, Thuwal Makkah 23955-6900, Kingdom of Saudi Arabia*

Abstract

The hydrostatic equations for ice sheet flow offer improved fidelity compared to the shallow ice approximation and shallow stream approximation (SSA) 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 SSA. This 3D system is posed on high-aspect domains with strong anisotropy and variation in coefficients, making it expensive to solve by using 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.

Keywords: hydrostatic, ice sheet, Newton-Krylov, multigrid, preconditioning

1. Introduction

The dynamic response of ice streams and outlet glaciers is poorly represented using the shallowness assumptions inherent in the present generation of ice sheet models. Accurate simulation of this response is crucial for prediction of sea level rise; indeed, the inability of available models, based on the shallow ice approximation (SIA) [1] and shallow stream approximation (SSA) [2, 3], to simulate these processes was cited as a major deficiency in the Fourth Assesment Report of the Intergovernmental Panel on Climate Change [4].

The hydrostatic equations were introduced in [5] as a model of intermediate complexity between the full non-Newtonian Stokes system and the integrated SIA and SSA models. A more precise analysis, including the limiting cases of fast and slow sliding, was given in [6]. Well-posedness was proven in [7], and approximation properties of finite-element methods were analyzed in [8, 9]. The hydrostatic equations were used for transient simulation in [10] and in 3D models in [11], as well as subsequent work. Several models of this form were compared in [12].

The use of hydrostatic equations in current models has been limited, however by the cost of solving the 3D nonlinear system for velocity. 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.

In the present work, we introduce a Newton-Krylov multigrid solver that demonstrates textbook multigrid efficiency, characterized by convergence in 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

Email addresses: brown@vaw.baug.ethz.ch (Jed Brown), bsmith@mcs.anl.gov (Barry Smith), aron.ahmadia@kaust.edu.sa (Aron Ahmadia)

parameters and geometry, coarsens rapidly in almost all cases, and exhibits excellent parallel scalability. Our code is freely available as part of the Portable Extensible Toolkit for Scientific computing (PETSc) [13].

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 [6] for a rigorous derivation and asymptotic analysis.

Consider an ice domain $\Omega \subset \mathbb{R}^3$ lying between a Lipschitz continuous bed $b \in C^0(\mathbb{R}^2)$ and surface $s \in C^0(\mathbb{R}^2)$, with thickness $H = s - b$ bounded below by a positive constant.[1] The velocity $\boldsymbol{u} = (u, v) \in \boldsymbol{V} = \boldsymbol{H}^1(\Omega)$ satisfies conservation of momentum, which, omitting inertial and convective terms as is standard for ice sheets, is given by

$$-\nabla \cdot \left[\eta \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \end{pmatrix}\right] + \rho g \nabla s = 0, \tag{1}$$

where

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

is the nonlinear effective viscosity with regularization 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. Equation (1) is subject to natural boundary conditions at the free surface and either no-slip $\boldsymbol{u} = 0$ or power-law slip with friction parameter

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

where $\gamma_b = \frac{1}{2}(u^2 + v^2)$, ϵ_b is regularization, and $m \in (0, 1]$ is the exponent that produces Navier slip for $m = 1$, Weertman [14] sliding for $m = 1/n$, and Coulomb slip as $m \to 0$. In the present work, we define ϵ and ϵ_b using a strain rate of $10^{-5}\,\mathrm{a}^{-1}$ and a slip velocity of $1\,\mathrm{m\,a}^{-1}$, respectively.

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

$$\int_\Omega \nabla\boldsymbol{\phi} : \eta \mathbf{1} : \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \end{pmatrix} + \boldsymbol{\phi} \cdot \rho g \nabla s + \int_{\Gamma_{\text{bed}}} \boldsymbol{\phi} \cdot \beta^2(|\boldsymbol{u}|^2/2)\boldsymbol{u} = 0 \tag{3}$$

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

For our numerical studies, equation (3) was discretized on a topologically structured hexahedral grid using Q_1 finite elements and standard 2^3-point Gauss quadrature. Length, time, and mass units were chosen so that thickness, velocity, and driving stresses are $\mathcal{O}(1)$. 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 [13] versions 3.1 and later.[2] Several generalizations of the tests from ISMIP-HOM [12] are implemented, run with the option `-help` for a complete list of options. Incidentally, our results for "5 km test C" in that paper (a slipperiness perturbation on a flat bed) agree to three significant figures with the "Stokes" results therein. This is consistent with the asympotic analysis of [6], which shows that slipperiness perturbations are not present in the leading-order error terms for the hydrostatic equations (which are purely geometric); cf. [12, Table 4 and Figure 8], where the ensemble range is nearly as large as the mean.

[1] This singular limit is important in the case of grounded margins, but the present work does not pursue it.

[2] Upon unpacking the source, which can be downloaded from mcs.anl.gov/petsc, see `src/snes/examples/tutorials/ex48.c`.

3. Solver and implementation

We begin by writing the discretization of (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

$$J(U)\delta U = -F(U). \tag{4}$$

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 also a critical 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 (4) 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, hwoever, 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 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.

3.1. Anisotropy

The ratio of width to thickness for outlet glaciers (the regions of ice sheets with greatest physical interest) ranges from $\mathcal{O}(1)$ to over 100. The nonlinear constitutive relation (2) produces three to four orders of magnitude variation in viscosity (usually with fastest variation in the vertical) and the Newton linearization of (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 like 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 only by 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, 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 since the number of vertical levels used in simulations is typically between 10 and 100; it also is convenient since 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 (frozen or sticky bed) and elements that are wide compared to the ice thickness, a lubrication approximation known as the "shallow ice approximation" is valid, allowing the velocity field to be determined locally from the surface slope and a column integral. Since incomplete factorization with column ordering provides nearly exact coupling in the vertical, it is an effective preconditioner in such

scenarios with no 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 degredation when subdomain size approaches a single column). However, none of these favorable performance characteristics remains when the elements become small relative to the ice thickness or when the bed becomes slippery, since the usual $\mathcal{O}\big((L/H)^2\big)$ condition number for second-order elliptic problems preconditioned by one-level Schwarz methods with subdomains of size H (see [15]) 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, so 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

$$f_u = 2\eta(4h_yh_z/h_x + h_xh_z/h_y + h_xh_y/h_z)u \tag{5}$$

$$f_v = 2\eta(h_yh_z/h_x + 4h_xh_z/h_y + h_xh_y/h_z)u, \tag{6}$$

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. The former involves evaluation of transcendental functions and quadrature loops. While transcendental functions 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 Core 2 and Opteron architectures using both GNU and Intel compilers. There was no manual vectorization for the Blue Gene/P results quoted in Section 4.2.

Assembly costs could be further mitigated by recomputing it 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 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, theyhave 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, thus saving the memory for the finest level(s). Additionally, a truly 2D coarse problem can be defined by using the shallow stream equations [2, 3, 16] and restriction operators defined by integrating the entire column. These possibilities are also runtime options, but they have not exhibited a level of robustness comparable to that of the more conventional methods pursued here.

The Jacobian is always symmetric positive definite and has a natural block size of 2, so we use a symmetric block format (PETSc's SBAIJ). 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

Table 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).

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

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 1.

There are two ways to construct matrices on the coarse levels of multigrid methods. 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 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 $P : \boldsymbol{V}_{\text{coarse}} \to \boldsymbol{V}_{\text{fine}}$ and assembled fine-level matrix A_{fine}, the Galerkin coarse operator is $A_{\text{coarse}} = P^T A_{\text{fine}} P$. These operators work well for some problems, but computing the sparse matrix product in parallel involves significant communication and irregular memory access, so it is relatively expensive. Additionally, second-order-accurate interpolation operators cause a loss of sparsity in the coarse-level operators, an effect known as stencil growth. Stencil growth tends to blur regions where the solution has local structure and usually reduces the effectiveness of inexpensive smoothers.

4. Numerical examples

We present several numerical examples that demonstrate the algorithmic and parallel scalability of the Newton-Krylov multigrid approach.

4.1. Algorithmic Scalability

We consider three model problems inspired by the periodic domain ISMIP-HOM [12] 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. 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 $\alpha = 0.05°$[3], which produce velocities from $0.9\,\text{km a}^{-1}$ to $47\,\text{km a}^{-1}$. A visual representation of the nonlinear solve process is shown in Figure 1. This 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. In this example and the next one, Luis Chacón's variant of the Eisenstat-Walker [17] method was used to automatically adjust linear solve tolerances as the nonlinear

[3]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.

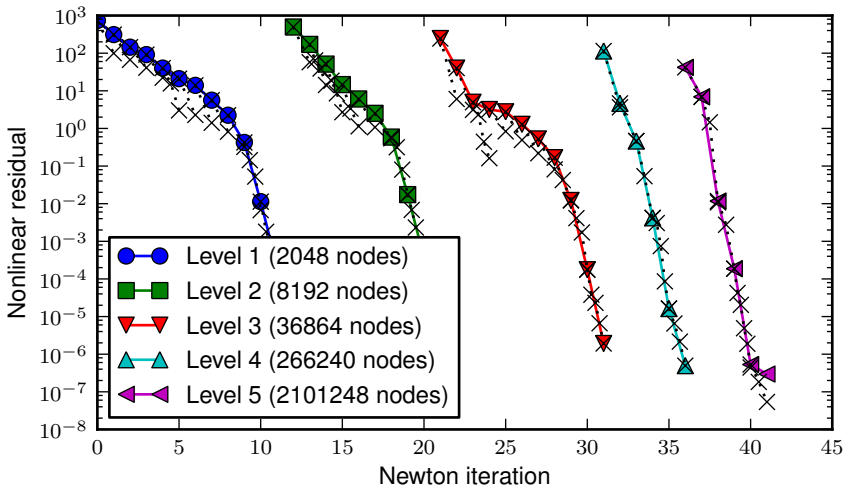

Figure 1: Grid-sequenced Newton-Krylov solution of test X. The solid lines denote nonlinear iterations, and the dotted lines with $\times$ denote linear residuals.

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. We remark that Picard iteration takes at least 50 iterations to reach this tolerance (sometimes many more; cf. [18], in which hundreds or thousands of iterations were needed for an easier problem). 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.

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

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

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

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 since the topography is too abrupt. Such topography is present in reality, however, so we may still desire an efficient solver. Figure 2 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 entirely acceptable up to level 3, however, where the elements are approximately 12 m thick and stretch to reach over a 200 m cliff in 125 m horizontal. On the finest level, they are 6 m thick and stretch over the cliff in 62 m horizontal, a slope of 73°. The approximation properties of such elements is poor and, considering that the continuum equations are invalid here, we believe this resolved topography is significantly rougher than will needed in applications.

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 3.

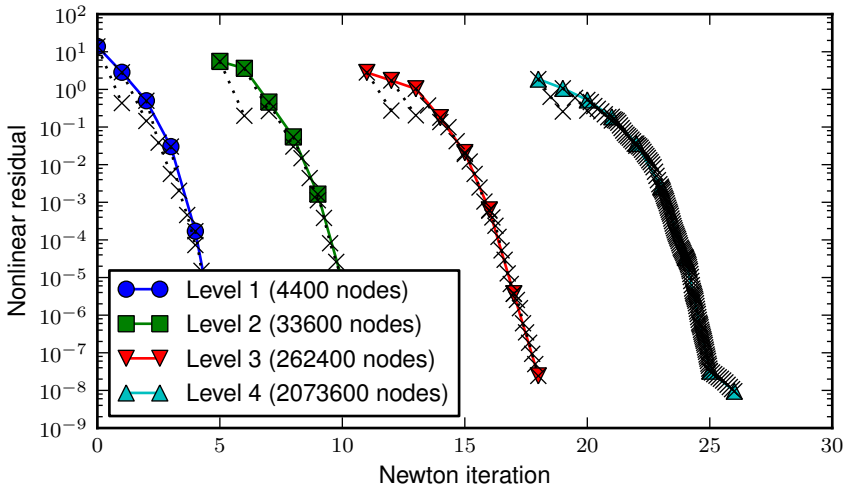

Figure 2: Grid sequenced Newton-Krylov convergence for test Y.

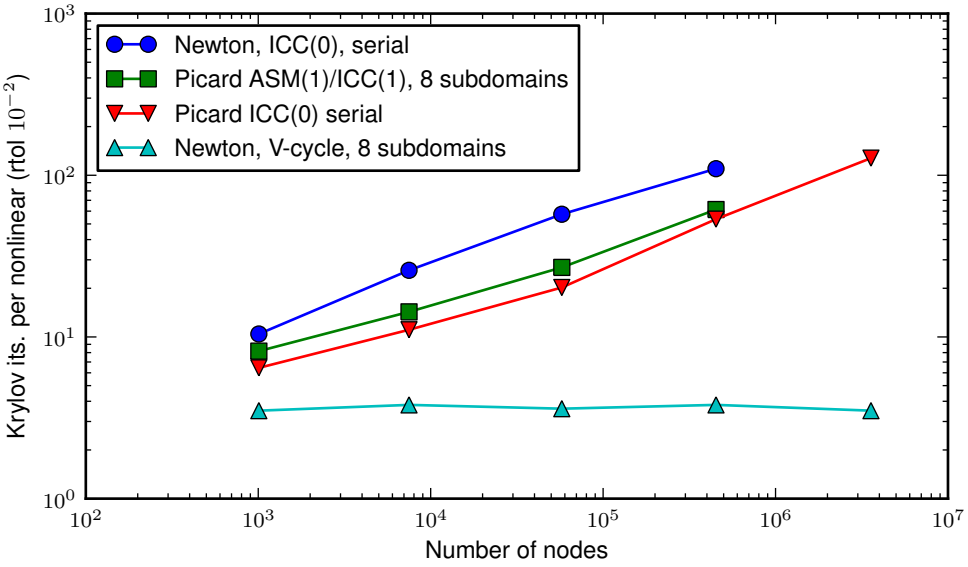

Figure 3: Average number of Krylov iterations per nonlinear iteration. Each nonlinear system was solved to a relative tolerance of 10^{-2}.

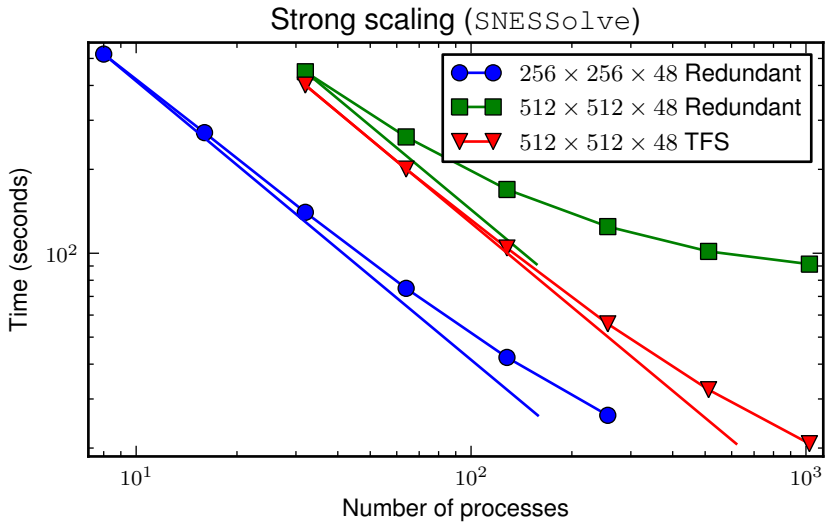

Figure 4: Strong scaling on Shaheen. The straight lines on the strong scaling plot have slope -1, which is optimal. Grid sequencing is used, but only the nonlinear solve on the finest level is shown since strong scalability is most important when many time steps are needed.

4.2. Parallel Scalability on Blue Gene/P

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. Two problem sizes are solved, with coarse meshes of $16 \times 16 \times 3$ and $32 \times 32 \times 3$, respectively. Both use five levels of isotropic refinement to reach target meshes of $256 \times 256 \times 48$ and $512 \times 512 \times 48$, the latter with nominal element sizes of $156 \times 156 \times 21$ meters. The coarse problems are solved redundantly in each case and also using the XX^T direct solver of [19] (TFS), which exhibits significantly better scalability, as shown in Figure 4 on a fixed global problem size (strong scalability).

Figure 5 shows weak scalability. 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.

4.3. Algebraic methods

Building a geometric hierarchy with rediscretization 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, $L = 80\,\text{km}, \alpha = 0.03$, 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 multigrids, 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 number of Krylov iterations to solve with 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 [20, 21] 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 [22] is denoted RASM(k). The average number of GMRES iterations per Newton is shown in Table 2. Note that increasing overlap has no benefit when incomplete subdomain solvers are used.

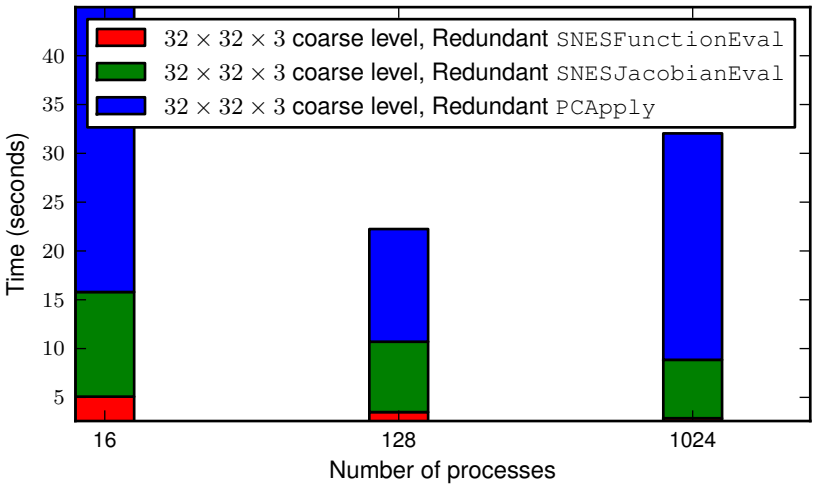

Figure 5: Weak scaling on Shaheen with a breakdown of time spent in different phases of the solution process. Times are for the full grid-sequenced problem instead of just the finest level solve.

Table 2: Average number of GMRES iterations per Newton for one-level domain decomposition with different overlap and fill. Negative pivots appeared frequently in all cases where incomplete factorization was used.

Decomposition \ Subdomain	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

Table 3: Average number of GMRES iterations per Newton for field-split preconditioners 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	Additive	Multiplicative	Sym. Multiplicative
Cholesky	19	9.9	9.3
ML	41	34	30
BoomerAMG	89	83	78
RASM(1)+Cholesky	186	173	84

The parallel algebraic multigrid packages ML [23] and BoomerAMG [24] provide potentially scalable alternatives. ML is based on smoothed aggregation, tends to coarsen very 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. 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. There were other, somewhat challenging problems for which BoomerAMG was competitive in terms of iteration count, but the setup costs and required number of levels were always large.

Another approach to solving multicomponent problems is to split the components and solve scalar problems for each in hopes 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 it is 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 by finite differencing the full residual. The results, shown in Table 3, are uninspiring, especially when considering that field-split creates additional synchronization points and attains lower throughput since it works with scalar matrices instead of block matrices (see Table 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 the horizontal, 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 the horizontal, 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 iteratation counts are similar. Table 4 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 2 where a factor of 5 to 10 degradation is visible. Use of Galerkin coarse operators has a catastrophic effect on the iteration count and may help explain the poor robustness exhibited by the algebraic multigrids. We cannot explain why symmetric additive Schwarz performs so poorly in this problem, but the other numbers are robust to changes in resolution, spatial domain, and number of processes.

We remark that when using grid sequencing and the method in the last row of Table 4, the problem can be solved in seven Newton iterations on the fine level and an average of 5.4 V-cycles per Newton. If the Eisenstat-Walker method is used to avoid oversolving, the problem takes eight Newton iterations with a total of 12 V-cycles.

Table 4: Average GMRES iterations per Newton for different multigrid preconditioners.

Coarse Problem	Level 1		Level 2 and 3		Its
	Decomp.	Subdomain	Decomp.	Subdomain	
$10 \times 10 \times 2$ Redisc	BJacobi	Cholesky	BJacobi	Cholesky	8.9
$10 \times 10 \times 2$ Redisc	BJacobi	Cholesky	BJacobi	ICC(0)	9.6
$10 \times 10 \times 2$ Redisc	BJacobi	Cholesky	ASM(1)	ICC(0)	11.9
$10 \times 10 \times 2$ Redisc	BJacobi	Cholesky	RASM(1)	ICC(0)	6.9
$10 \times 10 \times 2$ Redisc	ASM(1)	Cholesky	ASM(1)	ICC(0)	20.2
$10 \times 10 \times 2$ Redisc	RASM(1)	Cholesky	RASM(1)	ICC(0)	5.9
$10 \times 10 \times 2$ Galerkin	RASM(1)	Cholesky	RASM(1)	ICC(0)	54.
$10 \times 10 \times 1$ Redisc	BJacobi	Cholesky	BJacobi	ICC(0)	10.3
$10 \times 10 \times 1$ Redisc	ASM(1)	Cholesky	BJacobi	ICC(0)	10.1
$10 \times 10 \times 1$ Redisc	RASM(1)	Cholesky	BJacobi	ICC(0)	6.9

5. Conclusion

We have presented a grid-sequenced Newton-Krylov multigrid algorithm for solving the hydrostatic equations for ice sheet flow. This geometric multigrid method demonstrates textbook multigrid efficiency for extreme topography and basal conditions and offers $\mathcal{O}(1000)$ speedups relative to Picard linearization and one-level domain decomposition with incomplete factorization, the methods currently used to solve these equations [20, 21, 25, 18, 11]. Algebraic multigrid and field-split preconditioners were not found to be competitive in terms of robustness or efficiency.

Acknowledgments

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 Shaheen Supercomputing Laboratory at KAUST.

References

[1] K. Hutter, Theoretical glaciology: material science of ice and the mechanics of glaciers and ice sheets, Springer, 1983.
[2] L. Morland, Unconfined ice-shelf flow, Dynamics of the West Antarctic ice sheet, CJ van der Veen and J. Oerlemans, eds., Kluwer Academic Publishers (1987) 99–116.
[3] M. Weis, R. Greve, K. Hutter, Theory of shallow ice shelves, Continuum Mechanics and Thermodynamics 11 (1999) 15–50.
[4] L. Bernstein, et al., Climate Change 2007: Synthesis report, Intergovernmental Panel on Climate Change, 2008. `www.ipcc.ch/ipccreports/ar4-syr.htm`.
[5] H. Blatter, Velocity and stress fields in grounded glaciers: A simple algorithm for including deviatoric stress gradients, Journal of Glaciology 41 (1995) 333–344.
[6] C. Schoof, 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) 73–114.
[7] J. Colinge, J. Rappaz, A strongly nonlinear problem arising in glaciology, Mathematical Modelling and Numerical Analysis 33 (1999) 395–406.
[8] R. Glowinski, 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) 175–186.
[9] S. Chow, G. Carey, M. Anderson, Finite element approximations of a glaciology problem, Mathematical Modelling and Numerical Analysis 38 (2004) 741–756.
[10] F. Pattyn, Transient glacier response with a higher-order numerical ice-flow model, Journal of Glaciology 48 (2002) 467–477.
[11] 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 (2003) 10–1029.
[12] F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. De Smedt, O. Gagliardini, G. Gudmundsson, R. Hindmarsh, A. Hubbard, J. Johnson, et al., Benchmark experiments for higher-order and full Stokes ice sheet models (ISMIP-HOM), The Cryosphere 2 (2008) 95–108.
[13] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Web page, 2010. `mcs.anl.gov/petsc`.

[14] J. Weertman, On the sliding of glaciers, Journal of Glaciology 3 (1957) 33–38.
[15] B. Smith, P. Bjørstad, W. Gropp, Domain decomposition: parallel multilevel methods for elliptic partial differential equations, Cambridge University Press, New York, 1996.
[16] C. Schoof, A variational approach to ice stream flow, Journal of Fluid Mechanics 556 (2006) 227–251.
[17] S. C. Eisenstat, H. F. Walker, Choosing the forcing terms in an inexact newton method, SIAM Journal on Scientific Computing 17 (1996) 16–32.
[18] B. De Smedt, F. Pattyn, 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) 257–261.
[19] H. Tufo, P. Fischer, Fast Parallel Direct Solvers for Coarse Grid Problems* 1, Journal of Parallel and Distributed Computing 61 (2001) 151–177.
[20] K. Evans, A. Salinger, et al., A Scalable, Efficient, and Accurate Community Ice Sheet Model, 2010. www.csm.ornl.gov/SEACISM/.
[21] E. Larour, M. Morlighem, H. Seroussi, Ice Sheet System Model, 2010. issm.jpl.nasa.gov/.
[22] X. Cai, M. Sarkis, A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems, SIAM Journal on Scientific Computing 21 (1999) 797.
[23] M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala, ML 5.0 Smoothed Aggregation User's Guide, Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
[24] V. Henson, U. Yang, BoomerAMG: A parallel algebraic multigrid solver and preconditioner, Applied Numerical Mathematics 41 (2002) 155–177.
[25] J. Johnson, J. Staiger, Modeling long-term stability of the Ferrar Glacier, East Antarctica: Implications for interpreting cosmogenic nuclide inheritance, Journal of Geophysical Research 112 (2007) F03S30.