LOW-RANK QUASI-NEWTON UPDATES FOR ROBUST JACOBIAN LAGGING IN NEWTON METHODS

Jed Brown and Peter Brune
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
jedbrown@mcs.anl.gov; brune@mcs.anl.gov

Abstract

Newton-Krylov methods are standard tools for solving nonlinear problems. A common approach is to "lag" the Jacobian when assembly or preconditioner setup is computationally expensive, in exchange for some degradation in the convergence rate and robustness. We show that this degradation may be partially mitigated by using the lagged Jacobian as an initial operator in a quasi-Newton method, which applies unassembled low-rank updates to the Jacobian until the next full reassembly. We demonstrate the effectiveness of this technique on problems in glaciology and elasticity.

Key Words: quasi-Newton, lagged, Newton, preconditioning

1. Introduction

Inexact Newton methods are often used for solving nonlinear partial differential equations (PDEs) and optimization problems [1], and require the repeated construction and solution of the Jacobian linearization of the equations, typically using preconditioned Krylov methods. This "Newton-Krylov" approach has been nearly universally adopted for solving nonlinear PDEs on large-scale problems in parallel computing environments. Therefore, improvements in the efficiency of Newton-type methods are important for modern computational science.

Quasi-Newton methods were developed in the context of optimization in order to approximate the Jacobian* linearization by low rank updates to an (often crude) approximate Jacobian. The most popular variants of these methods are limited memory, storing a number of previous iterations and using the Sherman-Morrison-Woodbury formula [2] to apply the inverse of the updated $A + \sum_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ without explicitly storing a dense matrix, where A is the initial approximation for which A^{-1} can be applied inexpensively. The convergence of quasi-Newton methods is generally not nearly as swift as the full Newton's method and can be related to nonlinear Krylov methods [3].

1.1. Amortizing Setup Costs

Much of the expense associated with the inexact Newton-Krylov solution of nonlinear problems is in the repeated assembly of the Jacobian matrix and preconditioner setup. In response, practitioners often lag the

* We use nonlinear equations terminology "residual" and "Jacobian" throughout. These should be understood to mean "gradient" and "Hessian" when considering an optimization problem, for which an objective functional would also be available.

Jacobian by several iterations or adopt a Jacobian-free Newton-Krylov approach with a lagged preconditioner [4]. If the system is only weakly nonlinear, this lagging does not affect the Newton's method convergence too much, and overall robustness can be maintained. For stronger nonlinearity, however, the convergence rate will be significantly worse than the non-lagged case. Even so, the savings due to less frequent Jacobian assembly and preconditioner setup may outweigh the cost of the increase in the number of iterations.

Quasi-Newton methods attack the problem from the other direction: in typical usage, they start with a rough notion of the Jacobian and improve it progressively through low-rank updates. Many variants of quasi-Newton have a closed-form inverse that may be applied much more quickly than the inversion required in the full Newton method. The initial inverse Jacobian used in the quasi-Newton method can be improved in several ways, including some approximate or diagonal inverse of the full Jacobian. As the lagged Newton and quasi-Newton methods have opposite behavior with respect to performance when compared by per-iteration convergence, we combine the two in a way that amplifies the advantages of both methods.

In this paper, we explore the effectiveness of quasi-Newton methods that start with relatively high-accuracy representations of the Jacobian. We seek only to maintain non-lagged convergence rates despite less frequent Jacobian assembly. We evaluate a suite of restarted quasi-Newton methods, as compared with Jacobian lagging (modified Newton) and preconditioner lagging in "Jacobian-free" Newton-Krylov, on the basis of the number of Jacobian evaluations/preconditioner setup, number of residual evaluations, and number of preconditioner applications. A critical component of our discussion is the use of more sophisticated line searches that, for elliptic problems, are justified via analogy to an associated energy functional (but do not use the energy explicitly).

We have implemented these methods in PETSc, the Portable Extensible Toolkit for Scientific computing [5]. All variants can be used in existing codes purely through run-time options. We test the methods using a nonlinear viscous flow problem in glaciology and large-deformation elasticity.

1.2. Other Globalization Strategies

Numerous approaches have been developed for strongly nonlinear problems, many of which are based on some form of continuation. In many cases, grid sequencing (full multigrid) [6] is an efficient method for both local and global convergence, obviating the need for any global linearization. Pseudotransient continuation [7,8] is another effective method that is still superlinearly (or quadratically) convergent in the terminal phase. Other continuations, such as arc-length [9] and parameter [10], may also be used for globalization. To restrict the design space of our discussion, we consider problems requiring around 10 Newton iterations. This is the realm in which Newton methods with only line search globalization may be considered appropriate, but is not so benign that nonlinearities are not exercised by the solver. For each

choice of method, the practitioner must also decide how to approximate the inverse of the Jacobian. This step can be done by using approximations in matrix assembly [11] or by defining the inverse through an inexact linear solve, such as one V-cycle of a multigrid preconditioner. Again, we restrict the design space by considering only specific preconditioners that we believe to be appropriate for the problems at hand.

2. Newton and Quasi-Newton Methods

We describe the computational characteristics and convergence properties of inexact Newton and quasi-Newton methods. We include details about line searches, which will be particularly important in our discussion.

2.1. Inexact and Lagged Newton Methods

The inexact Newton-Krylov scheme for solving a nonlinear equation [12]

$$\mathbf{F}(\mathbf{x}) = 0 \tag{1}$$

involves assembling the Jacobian $\mathbf{J}(\mathbf{x})$, (iteratively) solving for $\mathbf{y}_i$ such that

$$|\mathbf{J}(\mathbf{x}_i)\mathbf{y}_i + \mathbf{F}(\mathbf{x}_i)| < \epsilon\,|\mathbf{F}(\mathbf{x}_i)| \tag{2}$$

and updating

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \lambda\mathbf{y}_i \tag{3}$$

until convergence in $|\mathbf{F}(\mathbf{x}_i)|$, with λ determined by line search.

Assembly of the Jacobian requires computing derivatives of $\mathbf{F}(\mathbf{x})$,

$$\mathbf{J}(\mathbf{x})[i,j] = \frac{\partial\mathbf{F}(\mathbf{x})[i]}{\partial\mathbf{x}[j]}, \tag{4}$$

which involves sparse matrix manipulation and local operations (e.g., quadrature over each cell/face in finite element methods) that, despite sparsity because of compactly supported basis functions, may be computationally expensive. Assembly costs are especially significant for problems with complex material models requiring evaluation of many transcendental functions, lookup tables, or implicitly defined constitutive relations (requiring an implicit solve at each quadrature point), and when high-order or exotic spatial discretizations are used.

A standard strategy to reduce assembly costs is to lag the Jacobian such that it is recalculated only every m steps. This lagged Newton (Shamanskii) method [13] finds $\mathbf{y}_i$ such that

$$|\mathbf{J}(\mathbf{x}_{i-k})\mathbf{y}_i + \mathbf{F}(\mathbf{x}_i)| < \epsilon\,|\mathbf{F}(\mathbf{x}_i)|\,, \tag{5}$$

where $k < m$ is the number of steps since the last assembly. The lagged Newton method converges q-superlinearly (for limit solution $\mathbf{x}^*$, $|\mathbf{x}_i - \mathbf{x}^*| \le c\,|\mathbf{x}_{i-1} - \mathbf{x}^*|^q$ with $q > 1$ for some $c > 0$) in the

terminal phase [14,12], as opposed to q-quadratically for Newton, but is typically much less robust when not started close to a solution. As we will see in Section 4, this lack of robustness can be partially compensated for by additional work in a line search.

An alternative lagging scheme is to apply the action of the Jacobian $\mathbf{J}\mathbf{x}_i$ using matrix-free finite differencing but to define the preconditioner by using the lagged assembled Jacobian $\mathbf{J}\mathbf{x}_{i-k}$ (or an approximation thereof). In this approach, which is a common variant of the Jacobian-free Newton-Krylov [4] (JFNK), the linear solver is responsible for compensating for any ill effects of lagging, leaving the nonlinear convergence (including q-quadratic local convergence) intact. Algorithmically, this requires many more residual evaluations (one per inner linear iteration) but potentially many fewer matrix assemblies and preconditioner setups. The downside is that increased lagging reduces the effectiveness of the preconditioner, leading to many more linear iterations.

2.2. Quasi-Newton

The quasi-Newton update is constructed as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \lambda \tilde{\mathbf{J}}_i^{-1} \mathbf{F}(\mathbf{x}_i), \tag{6}$$

where $\tilde{\mathbf{J}}$ is the sum or product of some set of low-rank updates that approximates $\mathbf{J}$ from the Newton method. For small problems, the Jacobian is sometimes updated by adding the dense low-rank update directly; but in large-scale problems where sparsity must be preserved, the rank-one updates are stored as a vector pairs. Limited memory variants allow for the inverse of the approximate Jacobian to be applied by using the stored previous iterates. Define $\mathbf{s}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ and $\mathbf{z}_i = \mathbf{F}(\mathbf{x}_i) - \mathbf{F}(\mathbf{x}_{i-1})$. The rank-one Broyden update [15], has a compact recursive application as [16,3]

$$\tilde{\mathbf{J}}_i^{-1} = (\mathbf{I} + \frac{(\mathbf{s}_i - \tilde{\mathbf{J}}_{i-1}^{-1}\mathbf{z}_i\mathbf{s}_i^\top)}{\mathbf{s}_i^\top \tilde{\mathbf{J}}_{i-1}^{-1}\mathbf{z}_i})\tilde{\mathbf{J}}_{i-1}^{-1}. \tag{7}$$

A popular symmetric rank-two update procedure, Broyden-Fletcher-Goldfarb-Shanno (BFGS), has a multiplicative limited memory form (L-BFGS) [17],

$$\tilde{\mathbf{J}}_i^{-1} = (\mathbf{I} - \frac{\mathbf{s}_i\mathbf{z}_i^\top}{\mathbf{s}_i^\top\mathbf{z}_i})\tilde{\mathbf{J}}_{i-1}^{-1}(\mathbf{I} - \frac{\mathbf{z}_i\mathbf{s}_i^\top}{\mathbf{s}_i^\top\mathbf{z}_i}) + \frac{\mathbf{s}_i\mathbf{s}_i^\top}{\mathbf{s}_i^\top\mathbf{z}_i}, \tag{8}$$

which can be applied quickly in a recursive fashion given storage of $\mathbf{s}_j$ and $\mathbf{z}_j$ for $j = i, \dots, i-k$. For L-BFGS with periodic restart, we take $k < m$ as the number of steps since the last restart. We must then choose a method for computing the approximate starting Jacobian inverse $\tilde{\mathbf{J}}_{i-k}^{-1}$. In optimization, this is mostly frequently some diagonal scaling, since the inverse of the true (dense) Jacobian is inaccessible.

2.3. Line Search

When used in conjunction with inexact Newton, a simple *backtracking* (bt) line search featuring cubic interpolation [18] guards against divergence. In the case of nonlinear PDEs, the line search approximately

minimizes the norm of the residual, defaulting to the full step when it is acceptable. However, minimizing the norm of the residual may be far less effective than minimizing some PDE-specific functional, which may be unavailable because of implementation or formulation details.

Suppose there exists an energy minimization formulation of the nonlinear PDE. Define energy $\mathbf{E}(\mathbf{x})$ and residual $\mathbf{F}(\mathbf{x})[i] = \frac{\partial \mathbf{E}(\mathbf{x})}{\partial \mathbf{x}[i]}$. In this case, we have an effective strategy for the line search that does not require the (potentially repeated) explicit evaluation of $\mathbf{E}$. Suppose that we are given a descent direction $\mathbf{y}$ (implying that $\mathbf{y}^\top \mathbf{F}(\mathbf{x}) < 0$) and that we discover $\lambda > 0$ such that

$$\mathbf{y}^\top \mathbf{F}(\mathbf{x} + \lambda \mathbf{y}) = 0.$$

This implies

$$\mathbf{y}^\top \mathbf{F}(\mathbf{x} + \lambda \mathbf{y}) = \frac{d\mathbf{E}(\mathbf{x} + \lambda \mathbf{y})}{d\lambda} = 0,$$

which is equivalent to locally minimizing $\mathbf{E}(\mathbf{x} + \lambda \mathbf{y})$ in λ. This can also be related to the second Wolfe condition [19]. Such a point may be approximately discovered iteratively by a secant method,

$$\lambda_{i+1} = \lambda_i - \frac{\mathbf{y}^\top \mathbf{F}(\mathbf{x} + \lambda_i \mathbf{y})(\lambda_i - \lambda_{i-1})}{\mathbf{y}^\top \mathbf{F}(\mathbf{x} + \lambda_i \mathbf{y}) - \mathbf{y}^\top \mathbf{F}(\mathbf{x} + \lambda_{i-1} \mathbf{y})}, \tag{9}$$

defined as a procedure starting from some acceptable guess λ_0. In the case of $\mathbf{y}$ generated by an inexact Newton's method, an acceptable initial guess is $\lambda_0 = 1$. This type of procedure has been recommended [20] for nonlinear conjugate gradient methods. We refer to this as the *critical point* (cp) line search and use it in most of our experiments. It requires a minimum of two function evaluations per nonlinear iteration, compared with the single evaluation needed for a standard backtracking line search (when accepting full steps).

3. Lagged-Jacobian Newton with Rank-One Updates

If we take $\tilde{\mathbf{J}}_{i-k}^{-1} = \mathbf{J}^{-1}(\mathbf{x}_{i-k})$, the L-BFGS application becomes

$$\tilde{\mathbf{J}}_i^{-1} = (\mathbf{I} - \frac{\mathbf{s}_i \mathbf{z}_i^\top}{\mathbf{s}_i^\top \mathbf{z}_i}) \ldots (\mathbf{I} - \frac{\mathbf{s}_{i-k+1} \mathbf{z}_{i-k+1}^\top}{\mathbf{s}_{i-k+1}^\top \mathbf{z}_{i-k+1}}) (\mathbf{J}^{-1}(\mathbf{x}_{i-k}) (\mathbf{I} - \frac{\mathbf{z}_{i-k+1} \mathbf{s}_{i-k}^\top}{\mathbf{s}_{i-k+1}^\top \mathbf{z}_{i-k+1}}) + \frac{\mathbf{s}_{i-k+1} \mathbf{s}_{i-k+1}^\top}{\mathbf{s}_{i-k+1}^\top \mathbf{z}_{i-k+1}}) \ldots (\mathbf{I} - \frac{\mathbf{z}_i \mathbf{s}_i^\top}{\mathbf{s}_i^\top \mathbf{z}_i})) + \frac{\mathbf{s}_i \mathbf{s}_i^\top}{\mathbf{s}_i^\top \mathbf{z}_i}. \tag{10}$$

Note that k steps ago, we assembled $\mathbf{J}(\mathbf{x}_{i-k})$ and set up a preconditioner so as to be able to apply $\mathbf{J}(\mathbf{x}_{i-k})^{-1}$ efficiently using a Krylov method, so the computational cost is effectively like the lagged Jacobian from Section 2.1 plus some vector work proportional to the restart length m (similar to GMRES). For the PDE problems considered herein, we either will apply $\mathbf{J}(\mathbf{x}_{i-k})^{-1}$ using a preconditioned Krylov method or will apply an approximation $\tilde{\mathbf{J}}(\mathbf{x}_{i-k})^{-1}$ such as one V-cycle of linear multigrid constructed from $\mathbf{J}(\mathbf{x}_{i-k})$.

4. Numerical Results

We apply quasi-Newton, lagged Newton, and lagged JFNK to a nonlinear elliptic system in glaciology and large-deformation elasticity. Our implementation has been developed in the PETSc library and will be available in version 3.4 (a preliminary implementation is available in version 3.3). All methods reuse the same algorithmic components and require only run-time options, with no modifications to user code. For example, the L-BFGS algorithm is obtained by using `-snes_type qn -snes_qn_restart_type periodic -snes_qn_scale_type jacobian`, with the linear solve to approximate the inverse of the lagged Jacobian using KSP, as with the standard Newton-Krylov methods.

For each class of methods, the user chooses what matrix to assemble and what algorithm to use for an approximate linear solve. For consistency, our examples assemble an exact Jacobian and use V-cycles of a multigrid algorithm to implement the approximate solve, combined with GMRES when an accurate solve is desired.

For each problem and solution method, we count the number of residual evaluations, Jacobian evaluations (equal to preconditioner setups), and preconditioner applications. The user chooses a restart parameter m based on the relative cost of Jacobian assembly and preconditioner setup to residual evaluation and preconditioner application. The combined choice of nonlinear solver class, restart parameter, and preconditioner allows work to be shifted between these computational stages.

4.1. Hydrostatic Ice-Sheet Flow

We consider the hydrostatic model for ice-sheet flow using the discretization and geometric multigrid preconditioners presented in [21]; see src/snes/examples/tutorials/ex48.c in PETSc. This model for viscous flow with shear-thinning power-law rheology is posed in 3D but has been partially vertically integrated by using incompressibility to eliminate pressure and vertical velocity, leaving only the horizontal components of velocity (u, v) as state variables. The hydrostatic model solves conservation of momentum

$$-\nabla \left[\eta \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \end{pmatrix} \right] + \rho g \nabla s = 0, \tag{11}$$

where

$$\eta(\gamma) = \frac{B}{2} \left(\epsilon^2/2 + \gamma \right)^{\frac{1-n}{2n}} \tag{12}$$

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 of the strain rate tensor. Finite element discretization of this system yields a symmetric positive definite Newton linearization. The regularizing strain rate ϵ is required in order to

prevent the equations from becoming singular at zero-stress points in the domain; and although a corresponding convex energy functional exists, it is no longer explicitly available in the regularized equations (see [22,23] for other approaches).

We consider the ISMIP-HOM [24] test C at $L = 10\,\text{km}$ with an added nonlinear slip boundary condition, in which the nonlinearity is globally activated and is notoriously difficult to converge by using Picard iteration [25], which is the most common approach used by the glaciology community. Two geometric coarse levels are used in the multigrid V-cycle, with rediscretized coarse grid operators and standard bilinear interpolation and restriction. Zero-fill incomplete Cholesky is used as a smoother to overcome anisotropy caused by the grid.

Table I shows the cost measures for each method on a $32 \times 32 \times 21$ element domain. We observe that the critical point line search is much more effective than the backtracking line search and that quasi-Newton methods are slightly more effective than lagged Newton. For this model, a multigrid V-cycle costs about the same as function evaluation, but Jacobian assembly is approximately four times more expensive. Although [21] demonstrates that grid sequencing is effective for globalizing this problem, it requires rediscretization of the nonlinear problem on each level. In comparison, the methods shown in Table I are applicable with a single-level discretization.

4.2. Large-Deformation Elasticity

We consider large-deformation elasticity discretized using a continuous finite element discretization based on high-order Q_3 elements and a St. Venant-Kirchoff stress model. The solution is chosen to be

$$\mathbf{u} = \begin{pmatrix} \cos(ax)\exp(by)\cdot z + \sin(cz) \\ \sin(ax)\tanh(by) + x\cosh(cz) \\ \exp(ax)\sinh(by) + y\log\left(1+(cz)^2\right) \end{pmatrix} \tag{13}$$

on the domain $(-1,1)^3$, with a forcing term manufactured symbolically so that this is an exact solution. This deformation is large enough that the system is nonconvex at many of the iterates in the nonlinear solve. Dirichlet displacement boundary conditions are used on the exterior of the cube. We use one V-cycle of BoomerAMG [26] as a preconditioner. Cost measures are shown in Table II. The high-order elements used for this problem lead to relatively expensive Jacobian assembly and preconditioner setup, combining to make evaluating the Jacobian more than 30 times as expensive as either residual evaluation or preconditioner application. L-BFGS is found to be most effective at reducing Jacobian work and is relatively tolerant of inexact linear solves. Lagging the preconditioner in JFNK causes the number of linear iterations to increase uncontrollably.

5. Discussion

Quasi-Newton methods combined with quality line searches can be used to shift the computational burden from Jacobian assembly and preconditioner setup to residual evaluation and preconditioner application.

Table I. Cost measures for quasi-Newton and lagged Newton methods applied to the hydrostatic ice sheet flow problem. Divergence is indicated with "—", demonstrating the limitations of the backtracking line search (bt) compared to the critical point line search (cp). For each lagging scheme, we compare a single V-cycle (preonly) to a linear solve with relative tolerance 10^{-5}. The number of iterations is for comparison, but the work is better quantified by the number of residual evaluations, Jacobian evaluations (coincides with preconditioner setup), and preconditioner applications (V-cycles).

Method	Lag	LS	Linear Solve	Iterations	Residual	Jacobian	Preconditioner
LBFGS	3	cp	`preonly`	15	31	4	15
LBFGS	3	cp	10^{-5}	10	21	3	68
LBFGS	6	cp	`preonly`	16	33	3	16
LBFGS	6	cp	10^{-5}	15	31	3	100
Broyden	3	cp	`preonly`	14	29	4	14
Broyden	3	cp	10^{-5}	12	25	3	76
Broyden	6	cp	`preonly`	18	37	3	18
Broyden	6	cp	10^{-5}	15	31	3	88
Newton	0	bt	`preonly`	23	31	23	23
Newton	0	bt	10^{-5}	12	21	12	66
Newton	0	cp	`preonly`	14	29	14	14
Newton	0	cp	10^{-5}	6	13	6	38
Newton	1	bt	`preonly`	—	—	—	—
Newton	1	bt	10^{-5}	—	—	—	—
Newton	1	cp	`preonly`	14	29	7	14
Newton	1	cp	10^{-5}	9	19	5	59
Newton	3	cp	`preonly`	15	31	4	15
Newton	3	cp	10^{-5}	12	25	3	74
Newton	6	cp	`preonly`	18	37	3	18
Newton	6	cp	10^{-5}	15	31	3	87
JFNK	0	cp	`preonly`	14	43	14	14
JFNK	0	cp	10^{-5}	6	83	6	38
JFNK	1	cp	`preonly`	15	46	8	15
JFNK	1	cp	10^{-5}	6	101	3	47
JFNK	3	cp	`preonly`	16	49	4	16
JFNK	3	cp	10^{-5}	6	155	2	74

Table II. Cost measures for quasi-Newton and lagged Newton methods applied to large-deformation elasticity. For each lagging scheme, we compare a single V-cycle (preonly) to a linear solve with relative tolerance 10^{-5}. The number of iterations is for comparison, but the work is better quantified by the number of residual evaluations, Jacobian evaluations (coincides with preconditioner setup), and preconditioner applications (V-cycles).

Method	Lag	LS	Linear Solve	Iterations	Residual	Jacobian	Preconditioner
LBFGS	3	cp	`preonly`	18	37	5	18
LBFGS	3	cp	10^{-5}	21	43	6	173
LBFGS	6	cp	`preonly`	24	49	4	24
LBFGS	6	cp	10^{-5}	30	61	5	266
Newton	0	bt	`preonly`	13	14	13	13
Newton	0	bt	10^{-5}	10	11	10	77
Newton	0	cp	`preonly`	11	23	11	11
Newton	0	cp	10^{-5}	8	17	8	60
Newton	1	bt	`preonly`	16	21	8	16
Newton	1	bt	10^{-5}	17	23	9	128
Newton	1	cp	`preonly`	15	31	8	15
Newton	1	cp	10^{-5}	13	27	7	103
Newton	3	cp	`preonly`	23	47	6	23
Newton	3	cp	10^{-5}	22	45	6	179
Newton	6	cp	`preonly`	36	73	6	36
Newton	6	cp	10^{-5}	35	71	5	294
JFNK	0	cp	`preonly`	11	23	11	11
JFNK	0	cp	10^{-5}	8	69	8	60
JFNK	1	cp	`preonly`	15	31	8	15
JFNK	1	cp	10^{-5}	7	2835	4	2827
JFNK	3	cp	`preonly`	23	47	6	23
JFNK	3	cp	10^{-5}	7	3143	2	3135

These methods were shown to be more robust than conventional lagged Newton methods and significantly reduce the required number of required iterations. This improved performance is obtained with only a small amount of vector work to store and apply the low-rank updates. These methods, available within PETSc using only run-time options, are likely to benefit applications solving nonlinear elliptic equations in cases where either Jacobian assembly or preconditioner setup is expensive. This situation arises with high-order and exotic spatial discretizations, as well as with sparse direct and adaptive multilevel [27] solvers. As with Newton methods, given a scalable preconditioner such as multigrid, the nonlinear convergence rate is independent of grid resolution.

ACKNOWLEDGMENTS

We thank Barry Smith and Todd Munson for fruitful discussions. We also thank two anonymous reviewers for valuable feedback. This work was supported by U.S. Department of Energy's Office of Science under Contract DE-AC02-06CH11357.

REFERENCES

[1] C.-J. Lin and J. Moré, "Newton's method for large bound-constrained optimization problems," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1100–1127, 1999.

[2] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.

[3] H. Fang and Y. Saad, "Two classes of multisecant methods for nonlinear acceleration," *Numerical Linear Algebra with Applications*, vol. 16, no. 3, pp. 197–221, 2009.

[4] D. Knoll and D. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357–397, 2004.

[5] 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.

[6] A. Brandt and O. Livne, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, Revised Edition*. SIAM, 2011.

[7] C. Kelley and D. Keyes, "Convergence analysis of pseudo-transient continuation," *SIAM Journal on Numerical Analysis*, vol. 35, p. 508, 1998.

[8] T. S. Coffey, C. T. Kelley, and D. E. Keyes, "Pseudotransient continuation and differential-algebraic equations," *SIAM Journal on Scientific Computing*, vol. 25, no. 2, pp. 553–569, 2003.

[9] E. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.

[10] R. Katz, M. Knepley, B. Smith, M. Spiegelman, and E. Coon, "Numerical simulation of geodynamic processes with the Portable Extensible Toolkit for Scientific Computation," *Physics of the Earth and Planetary Interiors*, vol. 163, no. 1-4, pp. 52–68, 2007.

[11] J. Brown, "Efficient nonlinear solvers for nodal high-order finite elements in 3D," *Journal of Scientific Computing*, vol. 45, pp. 48–63, 2010. 10.1007/s10915-010-9396-8.

[12] C. Kelley, *Iterative methods for linear and nonlinear equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1995.

[13] V. Shamanskii, "A modification of Newton's method," *Ukrainian Mathematical Journal*, vol. 19, no. 1, pp. 118–122, 1967.

[14] J. Traub, *Iterative methods for the solution of equations*. Englewood Cliffs, NJ: Prentice-Hall, 1964.

[15] C. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, vol. 19, no. 92, pp. 577–593, 1965.

[16] P. Brown, A. Hindmarsh, and H. Walker, "Experiments with quasi-Newton methods in solving stiff ODE systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 2, pp. 297–313, 1985.

[17] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[18] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

[19] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.

[20] J. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," tech. rep., Carnegie Mellon University, Pittsburgh, PA, 1994.

[21] J. Brown, B. F. Smith, and A. Ahmadia, "Achieving textbook multigrid efficiency for hydrostatic ice flow," *SIAM Journal on Scientific Computing*, *in press* 2013. Also preprint ANL/MCS-P743-1298.

[22] C. Schoof, "A variational approach to ice stream flow," *Journal of Fluid Mechanics*, vol. 556, pp. 227–251, 2006.

[23] D. Goldsby and D. Kohlstedt, "Superplastic deformation of ice: Experimental observations," *Journal of Geophysical Research*, vol. 106, no. B6, pp. 11017–11030, 2001.

[24] 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*, vol. 2, no. 1, pp. 95–108, 2008.

[25] 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*, vol. 56, no. 196, pp. 257–261, 2010.

[26] V. Henson and U. Yang, "BoomerAMG: A parallel algebraic multigrid solver and preconditioner," *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177, 2002.

[27] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, "Bootstrap AMG," *SIAM Journal on Scientific Computing*, vol. 33, no. 2, pp. 612–632, 2011.