Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Laboratory of Hydraulics,
Hydrology and Glaciology

Scalable solvers for optimal control of a power-law fluid

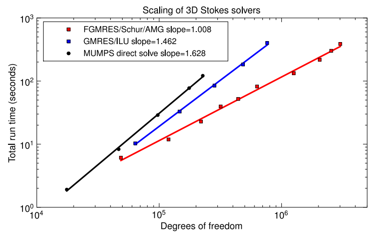
Jed Brown
brown@vaw.baug.ethz.ch
Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie, ETH Zürich, 8092 Zürich, Switzerland

Why do we need scalable solvers?

In the coming years, results of earth system models are expected to have an increasing influence on policy decisions. Qualitative results are not sufficient in this application so high-accuracy analysis of continental and global scale dynamics becomes critical. Such analysis requires extremely large problem sizes and stability constraints necessitate the use of implicit methods. The analysis phase involves a “physics” component which naturally requires $\mathcal{O}(N)$ work where N is the number of unknowns and a “solver” component which is often superlinear (e.g. $\mathcal{O}(N^{3/2})$) and will dominate the runtime for large problems, perhaps even making the target-resolution unattainable. A solver is *algorithmically scalable* when work is (nearly) $\mathcal{O}(N)$ and has optimal parallel scalability when runtime is (nearly) $\mathcal{O}(N/P)$ where P is the number of processors. The present work focuses on algorithmic scalability using components that are known to have good parallel scalability.

Preconditioning matters

The direct solver scales very poorly. Algebraic multigrid fails and ILU is not satisfactory when applied directly to the indefinite matrix. By contrast, the Schur complement preconditioner exhibits optimal scalability. The last point for each solver represents the largest problem size that could be solved in available memory (4GB minus the OS). The Schur complement solver shown only assembles a sparse approximation of the velocity system so it requires much less storage than the stable $Q_2 - Q_1$ scheme used in the ILU solver.



Fast high order finite elements

Standard finite element methods exhibit algebraic convergence with mesh refinement, but p-refinement (increasing the polynomial order) can provide exponential convergence, greatly reducing the number of degrees of freedom required to obtain a given accuracy. Unfortunately, p-refinement normally produces much denser matrices which are expensive to store, apply, and precondition. An alternative is to p-refine but apply the Jacobian matrix-free. A preconditioner will need an assembled matrix at some stage, but it is sufficient to assemble a much sparser matrix based on Q_1 (piecewise trilinear) subelements defined on the nodes of the tensor-product basis. This idea was introduced in [Deville and Mund, 1985] and spectral equivalence of the Q_1 matrix was recently proved in [Kim, 2007]. Since the number of Krylov iterations is essentially independent of the spectral order, the resulting scheme allows for arbitrary order approximations at comparable cost to the lowest-order finite element methods.

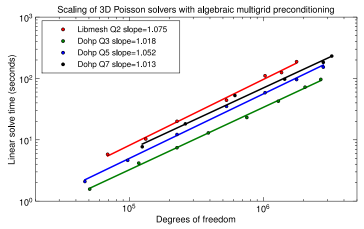
Unassembled matrix application on modern CPU architectures

Sparse matrix-vector products are an essential kernel of iterative linear algebra, but the floating point performance is very poor due to memory bandwidth limitations and irregular access. Current generation Intel and AMD processors achieve an upper limit between 4 and 20% of peak FPU throughput after extensive architecture-specific tuning [Oliker et al., 2008]. Significant improvements in floating point performance are not possible without changing the memory access pattern. High order tensor-product methods circumvent the memory bottleneck by changing the computational kernel from irregular access with significant metadata to dense tensor product operations on contiguous blocks of memory which fit in Level 1 cache. The amount of memory required to store the tensor-product representation is independent of the spectral order and less than a sparse matrix for first-order elements. In addition, it can be computed at little or no extra cost during residual evaluations. A naive implementation obtains several times the floating point performance of highly tuned sparse matrix-vector kernels.

Cost compared to Q_2 low-order finite elements

Event	Dohp Q_5	Libmesh Q_2	Q_1
Assembly	16.8	26.4	17.1
MatMult	26.7	28.3	5.08
PCSetUp	8.3	14.7	8.47
PCApply	25.8	88.4	13.8
KSPSolve	58.3	110.7	22.3
Memory (MB)	1194	2300	1044
GMRES its	29	21	12

Timing for $N = 101^3$



Assembling the actual Jacobian is only cost-effective for linear (Q_1) elements. When a stable basis or higher order accuracy is desired, the dual order method (*Dohp*) requires much less storage and provides significantly faster runtimes. An adaptive analysis using this method may choose the polynomial order based solely on local smoothness properties since there is negligible CPU or memory penalty for high order.

Power-law fluids

The steady state Stokes system for power-law rheology is: find $(\boldsymbol{u},p) \in \boldsymbol{V}_D \times P = \boldsymbol{H}^1_D(\Omega) \times L_2(\Omega)$ such that

$$\int_\Omega \left[\eta D\boldsymbol{v} : D\boldsymbol{u} - p\nabla\cdot\boldsymbol{v} - q\nabla\cdot\boldsymbol{u} - \boldsymbol{v}\cdot\boldsymbol{f}\right] + \int_\Gamma \boldsymbol{v}\cdot(p\mathbf{1} - \eta D\boldsymbol{u})\cdot\boldsymbol{n} = 0 \tag{1}$$

for all $(\boldsymbol{v},q) \in \boldsymbol{V}_0 \times P$ where $D\boldsymbol{u} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)$ is the symmetric gradient, $\gamma(D\boldsymbol{u}) = \frac{1}{2}D\boldsymbol{u} : D\boldsymbol{u}$ is the second invariant and

$$\eta(\gamma) = \left((\epsilon^2 + \gamma)/\gamma_0\right)^{\frac{1-n}{2n}} \tag{2}$$

is effective viscosity with regularization ϵ and Glen exponent $n \approx 3$. The boundary integral is normally modified to enforce a boundary condition, but certain outflow conditions [Papanastasiou et al., 1992] integrate the term as it appears. The space $\boldsymbol{V}_D$ has inhomogeneous trace on Dirichlet boundaries and V_0 is the corresponding homogeneous space. On slip boundaries, only the tangent component is constrained, while the boundary integral enforces the slip relation. The regularization ϵ has units of strain rate and prevents viscosity from becoming infinite in the zero strain limit. Such regularization is physical [Goldsby and Kohlstedt, 2001] but there is no consensus value. The performance of the nonlinear solver as $\epsilon \to 0$ is investigated in the *Globalization* section.

When solving (1) with a Newton method, we solve discrete forms of: find $(u,p) \in \boldsymbol{V}_0 \times P$ such that

$$\int_\Omega \left[\eta D\boldsymbol{v} : D\boldsymbol{u} + \eta'(D\boldsymbol{v} : D\tilde{\boldsymbol{u}})(D\tilde{\boldsymbol{u}} : D\boldsymbol{u}) - p\nabla\cdot\boldsymbol{v} - q\nabla\cdot\boldsymbol{u}\right] = -\int_\Omega vF(\tilde{\boldsymbol{u}})$$

for all $(v,q) \in \boldsymbol{V}_0 \times P$. In matrix notation,

$$J(\tilde{u})x = \begin{bmatrix} A(\tilde{u}) & B^T \\ B & \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} -F(\tilde{u}) \\ 0 \end{pmatrix}$$

where the viscous contribution A is symmetric positive definite and B is negative divergence. Since J is indefinite, standard preconditioners such as multigrid and domain decomposition methods are ineffective. While ILU fairs slightly better, the convergence rates are disappointing. Effective preconditioning of J uses a block factorization of the form

$$J = \begin{bmatrix} A & B^T \\ B & \end{bmatrix} = \begin{bmatrix} 1 & \\ BA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & B^T \\ & S \end{bmatrix} = \begin{bmatrix} A & \\ B & S \end{bmatrix} \begin{bmatrix} 1 & A^{-1}B^T \\ & 1 \end{bmatrix} \tag{3}$$

where $S = -BA^{-1}B^T$ is the Schur complement which is dense, hence prohibitively expensive to form. A general class of indefinite preconditioners is obtained by replacing each occurence of A^{-1} with a choice of standard preconditioner, possibly dropping either or both off-diagonal terms, and choosing a preconditioner for S. Classical solvers for incompressible flow such as the Uzawa and SIMPLE families, as well as recent block preconditioners correspond to particular choices [Elman et al., 2008]. The heuristic argument

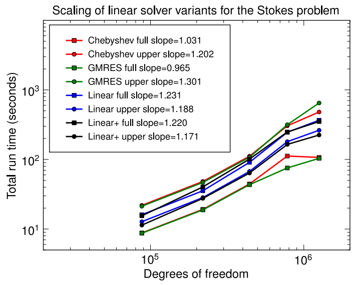
$$S \sim \nabla\cdot\boldsymbol{\Delta}^{-1}\nabla \approx \Delta^{-1}(\nabla\cdot\nabla) = \Delta^{-1}\Delta = \mathbf{1}$$

suggests that S is similar to the identity, hence a mass matrix scaled by η^{-1} should be an effective preconditioner for S.

Indefinite preconditioning choices

Effective preconditioning of S requires assembly of the scaled mass matrix M_p, but it may either be used directly or used to precondition iterations directly on S. Iterating directly on S moves work into the inner loop, but it is not cost-effective to fully converge an iteration on S. Early termination of a Krylov iteration does not provide a linear operator, but a fixed number of stationary iterations does. For the constant viscosity linear problem, we find self-preconditioning of S with Chebychev or Krylov iterations can reduce solve time, *but* it requires that the full factorization (3) is used and that the outer Krylov method is FGMRES. When the lower block is dropped, it appears that inner iterations do not help.

The self-preconditioned variants, *GMRES* and *Chebyshev*, have outer iteration counts of 2 or 3 for all problem sizes when the full factorization is used. The variants with no inner iteration, *Linear* and *Linear+*, were faster when only the upper block was used. *Linear+* provides a stronger preconditioner for A by using 3 Richardson iterations preconditioned by a V-cycle of AMG applied to the Q_1 preconditioning matrix A_p. There appears to be no cost-effective middle ground, either everything is left to the outer Krylov solver or a fairly strong inner solve must be done.



Funding: Swiss National Science Foundation

Software: PETSc, ML, Hypre, ITAPS/MOAB, Libmesh

Inverse problems and optimization

It is typically not feasible to directly measure properties within, or at the base of, an ice sheet. These properties affect the flow to such an extent that forward modeling with poorly known boundary and interior data is dubious at best. Determining these properties from available data is the object of inverse methods. Traditional methods such as Bayesian inversion [Tarantola, 2005, Raymond, 2007] rely on assembling and factoring a dense $N \times N$ matrix where N is the size of the parameter space. This requires $\mathcal{O}(N^3)$ flops and $\mathcal{O}(N^2)$ memory which is not feasible for moderate problem sizes $N \approx 10^6$ on today’s clusters. To circumvent dense matrices, we pose inverse problems as PDE-constrained optimization, for which scalable methods have been developed [Biros and Ghattas, 2005]. The key to these methods is scalable preconditioning of the resulting indefinite linear systems.

PDE-constrained optimization: Lagrange-Newton-Krylov-Schur

We consider problems of the form

$$\min_{x_s,x_d} f(x_s,x_d) \quad \text{subject to} \quad \boldsymbol{c}(x_s,x_d) = \mathbf{0}$$

where f is the objective, $\boldsymbol{c}$ is a partial differential equation, $x_s \in V$ is a function in the state space, and $x_d \in V_d$ is a function in the design space. An example in glaciology would have x_s as combined velocity and pressure while x_d is a distributed basal slipperiness or flow-law parameter like temperature or anisotropy which we would like to estimate. For an inverse problem, the objective has the form

$$f = \frac{1}{2}\left\|x_d\right\|^2 + \frac{1}{2}\int (x_s - \hat{x}_s)^2\,\mathrm{d}\sigma$$

where $\|\cdot\|$ is a seminorm on the design space, $\hat{x}_s$ are observations and σ is an (often discrete) measure on the misfit. In many cases, the objective and seminorm can be chosen so as to be equivalent to a given Bayesian formulation.

We enforce constraints by introducing adjoint variables $x_a \in V_0$ as Lagrange multipliers. After integration by parts, the PDE is equivalent to a weak form $b(x_d;x_a,x_s) = 0$ for all x_a. We define the Lagrangian

$$L(x_s,x_a,x_d) = \frac{1}{2}\left\|x_d\right\|^2 + \frac{1}{2}\int (x_s - \hat{x}_s)^2\,\mathrm{d}\sigma + b(x_d;x_a,x_s)$$

which must be stationary for optimality, leading to a nonlinear system of equations

$$F(\boldsymbol{x}) = \nabla L(x_s,x_a,x_d) = 0 \tag{4}$$

which is discretized and solved using Newton methods. At each Newton iteration, the linear system

$$J(\tilde{\boldsymbol{x}})\boldsymbol{x} = \begin{bmatrix} W_{ss} & A_s^T & W_{sd} \\ A_s & 0 & A_d \\ W_{ds} & A_d^T & W_{dd} \end{bmatrix} \begin{pmatrix} x_s \\ x_a \\ x_d \end{pmatrix} = -\begin{pmatrix} F_s \\ F_a \\ F_d \end{pmatrix} = -\boldsymbol{F}(\tilde{\boldsymbol{x}})$$

must be solved. A_s is the discrete form of $\partial_{x_s} b(\tilde{x}_d;\cdot,\tilde{x}_s)$, the Jacobian of the PDE operator, which is independent of the adjoint variable. The matrix $J(\tilde{\boldsymbol{x}})$ is indefinite and very poorly conditioned so standard preconditioners are not effective, hence we use the factorization

$$\begin{bmatrix} W_{ss} & A_s^T & W_{sd} \\ A_s & 0 & A_d \\ W_{ds} & A_d^T & W_{dd} \end{bmatrix} = \begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ W_{ss}A_s^{-1} & 1 & \\ W_{ds}A_s^{-1} & A_d^T A_s^{-T} & 1 \end{bmatrix} \begin{bmatrix} A_s & & A_d \\ & A_s^T & R \\ & & S \end{bmatrix}$$

where

$$R = W_{sd} - W_{ss}A_s^{-1}A_d \qquad \text{and} \qquad S = W_{dd} - A_d^T A_s^{-T} R - W_{ds}A_s^{-1}A_d$$

to construct a preconditioner. S is known as the reduced Hessian and appears in popular optimization methods such as Quasi-Newton Reduced space Sequential Quadratic Programming (QN-RSQP). Such methods require exact PDE solves at each iteration and the number of required iterations scales as the square root of the condition number of S. By applying the powerful Newton-Krylov machinery directly to (4), the LNKS method retains quadratic convergence and replaces the PDE solves at each iteration with preconditioners. A preconditioner for S may be obtained by stationary iteration, BFGS update, or approximate commutators, the last of which uses problem-specific structure and has proven effective for other indefinite problems.

Globalization

The constitutive relation (2) becomes highly nonlinear when $\epsilon \to 0$ and algebraic globalization such as line search and trust region methods stagnate. Effective globalization exploits the problem structure by continuation in the exponent. A continuation with one to three steps was effective even with exponents larger than $n = 5$ and with viscosity variation of up to 10^7. The most effective solver combination used the [Eisenstat and Walker, 1996] method to adjust linear solve tolerances in a line-search guarded Newton method where the linear system was solved using LGMRES(30) with block triangular preconditioning, no inner iterations, and algebraic multigrid preconditioning for A. Assembly of the Q_1 preconditioning matrix for A is quite cheap so lagging the preconditioner was not helpful.

References

[Biros and Ghattas, 2005] Biros, G. and Ghattas, O. (2005). Parallel Lagrange-Newton-Krylov-Schur methods for PDE constrained optimization. Part I: The Krylov-Schur solver. *SIAM Journal on Scientific Computing*, 27(2):687–713.

[Deville and Mund, 1985] Deville, M. and Mund, E. (1985). Chebyshev Pseudospectral Solution of Second-Order Elliptic Equations with Finite Element Preconditioning. *Journal of Computational Physics*, 60:517.

[Eisenstat and Walker, 1996] Eisenstat, S. C. and Walker, H. F. (1996). Choosing the forcing terms in an inexact newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32.

[Elman et al., 2008] Elman, H., Howle, V., Shadid, J., Shuttleworth, R., and Tuminaro, R. (2008). A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 227(1):1790–1808.

[Goldsby and Kohlstedt, 2001] Goldsby, D. and Kohlstedt, D. (2001). Superplastic deformation of ice: Experimental observations. *Journal of Geophysical Research*, 106(B6):11017–11030.

[Kim, 2007] Kim, S. (2007). Piecewise bilinear preconditioning of high-order finite element methods. *Electronic Transactions on Numerical Analysis*, 26:228–242.

[Oliker et al., 2008] Oliker, L., Williams, S., Vudac, R., Shalf, J., Yelick, K., and Demmel, J. (2008). Multicore optimization of sparse matrix vector multiplication. In *SIAM conference on parallel processing for scientific computing*.

[Papanastasiou et al., 1992] Papanastasiou, T., Malamataris, N., and Ellwood, K. (1992). A new outflow boundary condition. *International Journal for Numerical Methods in Fluids*, 14(5):587–608.

[Raymond, 2007] Raymond, M. (2007). *Estimating basal properties of glaciers and ice streams from surface measurements*. PhD thesis, Swiss Federal Institute of Technology Zurich.

[Tarantola, 2005] Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial Mathematics.