Multi-scale linear solvers for very large systems derived from PDEs

KLAUS S. LACKNER and RALPH MENIKOFF

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

We present a novel linear solver which works well for large systems obtained from discretizing a PDE. In particular, it is robust and the computational effort scales no worse than $N\log(N)$, where N is the number of equations. The algorithm is based on a wavelength decomposition which combines conjugate gradient, multiscaling and iterative splitting methods into a single approach. On the surface, the algorithm is a simple preconditioned conjugate gradient with all the sophistication of the algorithm in the novel choice of the preconditioning matrix. The preconditioner is an approximate inverse of the linear operator constructed from the inverse of the coarse grained linear operator and a smoothing operator based on an iterative expansion of the operator on the fine grid. To transform the coarse grained operator to the fine grid it is pre- and post-multiplied by reduction and prolongation operators. The coarse graining captures the long wavelength behavior of the inverse operator while the smoothing operator captures the short wavelength behavior. The conjugate gradient iteration accounts for the coupling between long and short wavelengths.

The coarse grained operator corresponds to a lower resolution approximation to the PDEs. While its inverse is not known explicitly, the conjugate gradient algorithm only requires that the preconditioner can be a applied to a vector. The coarse inverse applied to a vector can be obtained as the solution of another preconditioned conjugate gradient solver that applies the same algorithm to the smaller problem. Thus, the method is naturally recursive. The recursion is terminated when the matrix is sufficiently small for a solution to be obtained efficiently with a standard solver.

Basing a preconditioner on the inverse of a simpler problem is similar in some respects to using an incomplete factorization. As with an incomplete factorization, our preconditioner only requires knowledge of sparse matrices. But rather than a simple forward and backward substitution used to evaluate the inverse of the product of a lower and upper triangular matrix, our preconditioner requires an iterative method. Despite the extra effort required for the coarse inverse, the algorithm is efficient because the increased quality of the approximate inverse greatly reduces the number of times the preconditioner needs to be evaluated. In the cases we studied the computational effort is still dominated by the operations performed on the finest scale.

As with the multigrid method, by utilizing several scales we obtain rapid convergence of long wavelengths. In contrast to the way multigrid algorithms are typically applied, the recursive use of the conjugate gradient algorithm at each stage enforces an accurate transition from one scale to the next. In particular, the algorithm naturally accounts for the coupling between the long and short wavelengths introduced at every level of refinement. Thus, it avoids the accumulation of errors that stem from the interaction terms between the different levels. In addition, the transition between levels is determined dynamically rather than preprogrammed with a V-cycle or W-cycle. The algorithm is robust and works well without fine tuning even when the PDE is far from being diagonal in Fourier space.

Let the linear system be given by

$$Ax = b \,. \tag{1}$$

With a preconditioning matrix M^{-1} one solves the equivalent system

$$M^{-1}Ax = M^{-1}b \,. \tag{2}$$

If M^{-1} is a good approximation to A^{-1}, then the effective condition number of the system is greatly reduced and convergence is significantly enhanced. An important property of the conjugate gradient algorithm is that convergence can still be enhanced if the condition number of $M^{-1}A$ remains large provided that M^{-1} fails to account for only a small number of modes. This reduces the quality of approximation required of the coarse grained operator.

To develop an approximate inverse for use as a preconditioner, we start with a standard operator splitting of the form, see for example [1],

$$A = P - Q \,. \tag{3}$$

We assume that P is positive and symmetric. For our algorithm, it is necessary that the inverse of P is easy to evaluate and that $P^{-1}Q$ damps short wave lengths.

With such an operator splitting, the inverse is given by

$$A^{-1} = (\mathbf{I} - P^{-1}Q)P^{-1} = \sum_{n=0}^{\infty} (P^{-1}Q)^n P^{-1} \,. \tag{4}$$

The series converges when $||P^{-1}Q|| < 1$, where $|| \cdot ||$ denotes the L^2 matrix norm. Frequently, the matrix arising from discretizing a PDE is diagonally dominated and it can be shown that the standard splittings, such as the Jacobi or Gauss-Seidel, do indeed satisfy the necessary condition for the series to converge. We observe the following identity

$$A^{-1} = (P^{-1}Q)^m A^{-1} (QP^{-1})^m + \sum_{k=0}^{2m-1} (P^{-1}Q)^k P^{-1} \tag{5}$$

for any integer $m \geq 1$. This identity is easily proved by substituting in the series in Eq. (4) for A^{-1} on the right hand side of the equation.

If one replaces A^{-1} on the right hand side of the equation with some matrix W corresponding to an approximation of A^{-1}, then one obtains a better approximate inverse

$$M^{-1} = (P^{-1}Q)^m W (QP^{-1})^m + \sum_{k=0}^{2m-1} (P^{-1}Q)^k P^{-1} \ . \tag{6}$$

A measure of the improvement of the inverse can be obtained from the formula

$$AM^{-1} - \mathbf{I} = (QP^{-1})^m \cdot (AW - \mathbf{I}) \cdot (QP^{-1})^m \ . \tag{7}$$

Consequently, $||AM^{-1} - \mathbf{I}|| \leq ||P^{-1}Q||^{2m} ||AW - \mathbf{I}||$. Thus, with respect to this norm, AM^{-1} is closer to the identity than AW provided that $||P^{-1}Q|| < 1$. However, for large systems $||P^{-1}Q||$ is near 1 and for the M^{-1} to be an effective preconditioner W^{-1} and $P^{-1}Q$ need to be complementary in the sense that they approximate different parts of the spectrum of A. In this case, M^{-1} can be a much better inverse of A then either W or the series given in Eq. (4) truncated after $2m$ terms.

Equation (6) forms the basis for a number of approximation schemes. Tatebe [2] pointed out that M^{-1} will be symmetric positive definite if W^{-1}, P, and Q are all symmetric and positive. We note that W need be only positive and not positive definite for M^{-1} to be a valid preconditioner for a conjugate gradient algorithm. Consequently, the coarse grained inverse can be used for W even though it has a non-zero null space.

The simplest choice is $W = P^{-1}$. In this case, M^{-1} corresponds to a conventional polynomial preconditioner; *i.e.*, the first $2m+1$ terms of the series for A^{-1} in Eq. (4). At high resolution when the linear system is very large, $||P^{-1}Q||$ is close to 1 and M^{-1} is a poor preconditioner. The underlying reason is that P connects only neighboring grid points and consequently M^{-1} provides a poor approximation of A^{-1} at long wave lengths. This is born out by experience showing that the number of iterations grows rapidly with the dimension of the system.

A better choice is that advocated by Tatebe which aims to account for the long wavelengths by basing the preconditioner on a single step of a multigrid algorithm. Let

G_k be a sequence of successively coarser grids, $R_{k+1,k}$ and $E_{k,k+1}$ be the reduction and prolongation operators connecting the grids G_k and G_{k+1}, A_k the coarsened operator on the grid G_k, and P_k and Q_k the splitting of A_k. Here $k = 0$ corresponds to the finest mesh and $k_{\max}$ to the coarsest mesh. The preconditioner can be defined recursively as

$$M_k^{-1} = (H_k)^m (E_{k,k+1} M_{k+1}^{-1} R_{k+1,k})(H_k^T)^m + \sum_{j=0}^{2m-1} (H_k)^j P_k^{-1} \ , \tag{8}$$

where $H_k = P_k^{-1} Q_k$. Provided that the reduction and prolongation operators are chosen such that $R_{k+1,k}^T = E_{k,k+1}$, the approximate inverse at every level, $W_k = E_{k,k+1} M_{k+1}^{-1} R_{k+1,k}$ is symmetric. On the coarsest mesh, the problem can be solved exactly. Thus, the recursion ends with $M_{k_{\max}}^{-1} = A_{k_{\max}}^{-1}$. The preconditioner for the initial system is M_0^{-1}. For this preconditioner to be effective, the splitting must be chosen such that H_k smoothes the shorter wavelengths to complement W_k which operates only on the longer wavelengths. Thus, the underlying rational for the multigrid preconditioner to be a good approximate inverse is a wavelength decomposition. When the system is large enough to require many coarsening levels, two problems can arise. The first is that many iterations are needed to account for the coupling between short and long wavelengths. The second is that errors due to the coupling can accumulate at each level to such an extent that the iterative improvement to the solution saturates and the the desired tolerance can not be achieved.

These difficulties can be overcome by defining the preconditioner in term of the exact inverse of the coarsened operator A_c as follows

$$M^{-1} = (P^{-1}Q)^m (E A_c^{-1} R)(Q P^{-1})^m + \sum_{j=0}^{2m-1} (P^{-1}Q)^j P^{-1} \ , \tag{9}$$

In effect, $W = E A_c^{-1} R$ and the evaluation of A_c^{-1} on a vector is computed by applying the same algorithm to the coarse grained operator. Again the recursion ends by solving the problem exactly on the coarsest level. Using the exact inverse A_c^{-1} for W is the best preconditioner based on a single level of scaling. Heuristically one expects a small number of conjugate gradient iterations, since $P^{-1}Q$ lowers the maximum eigenvalue and A_c^{-1} raises the minimum eigenvalue of $M^{-1}A$. Thus, the ratio of the eigenvalues of $M^{-1}A$ is greatly reduced leading to a moderate condition number for the preconditioned problem.

An important feature of this preconditioner is that the long and short wavelengths are coupled at each level by a conjugate gradient solver before proceeding to the next finer grid. This prevents truncation errors from the coarsening of the operator at each level from accumulating. In particular, if the coarsened operator does poorly on a few modes, which typically occurs when the coefficients of the underlying PDE are discontinuous, then these modes will be corrected by the conjugate gradient solver with only a small penalty. The robustness of our algorithm is in large measure due to the local feedback which corrects errors at each level before they can build up. The efficiency of the algorithm is due to the small number of iteration made possible by having a very good approximate inverse.

To validate the algorithm we have implemented the solver in an object oriented C++ based code. The implementation is general enough to allow for an arbitrary choice of solver and preconditioner at every level. We actually tested the code on the three recursive cases described above: polynomial preconditioner, Tatebe's algorithm and the recursive conjugate gradient. We also specialized the implementation to the standard multi-grid algorithm.

First, we applied the algorithm to the simple case of a Poisson equation. In the range we tested the solver, from $N = 10^3$ to 5×10^6 grid points, the computational effort (as measured by the cpu time on a workstation) is proportional to N. Thus, the convergence criteria of the solvers at every level led naturally to an overall rate of convergence that multigrid algorithms can only achieve by tuning the cycling scheme. The algorithm proved to be memory efficient and fast. The problem required storage for just 6 fine grid vectors and the solution was found in the time equivalent to computing 300 scalar products of such vectors. Moreover, the algorithm is insensitive to the scaling factor, degree of smoothing and convergence tolerance at the coarser level. Accurate solutions could be obtained even with singular source terms. By tightening the solvers tolerance, machine accuracy of the solution could be achieved. The number of iterations grew slowly and logarithmically with the desired accuracy.

After the Poisson equation we turned to the more complicated problem of porous flow. A qualitatively similar performance is observed except that the computational effort grows as $N \log(N)$. For a smooth permeability field the computational effort is roughly 5 times as large as for the Poisson equation. This is attributable to the more complex structure of the matrix compared to the Laplacian operator.

The porous flow equation is of the form

$$\nabla \cdot \mathbf{K} \nabla P = S \; .$$

Here, $\mathbf{K}$ is a permeability tensor, P is the pressure and S is a source. Equations of this form also occur in other physical systems such as incompressible fluid flow. The linear system for our examples are derived from the standard 5-point stencil for the finite difference operator on a two-dimensional regular grid. The permeability tensor is diagonal and generated with a random spatial variation constrained to satisfy an autocorrelation with either a Gaussian or a power law behavior. We use a symmetric Gauss-Seidel splitting and a simple coarsening procedure. The prolongation operator is a tensor product based on a linear interpolation in each spatial direction and the reduction is the transpose of the prolongation. For the coarse grained permeability tensor, the K_{xx} component is taken as

$$\frac{1}{K^c_{xx}} = \sum_i \frac{1}{\sum_j K_{xx}(i,j)}$$

where the sum is over fine cells contained within a coarse cell. A similar construction is used for K_{yy}.

In spite of the simple coarse graining procedure we were able to solve problems in which the permeability varied spatially by a factor of up to 10^9 and on grids as large as 1000×1000 on a workstation. Typically, we scaled between successive grid levels by a linear factor of 4 which reduces the dimension of the problem at each level by a factor of 16. It is noteworthy that the scale factor is not limited to a factor of two, nor for that matter to an integer. Consequently, the number of grid points in a linear dimension does not have to be a power of the scale factor. The effective scale factor between adjacent levels may vary in order to obtain an integer grid dimension. The ability of the algorithm to work under these conditions is an indication of its robustness and it shows that fine tuning of the coarsening operation is not required.

The degree m of smoothing is related to the scale factor. Since the smoothing operator $P^{-1}Q$ typically connects only neighboring grid points we have chosen m to be the same as the scale factor. As a result the preconditioning matrix fully couples every fine grid cell. Smaller values of m would require a larger number of iterations for problems in which the short wavelengths dominate the solution.

We base the convergence criterion of the conjugate gradient solvers on the norm of the residual with respect to the preconditioning matrix. Since the preconditioner is an approximate inverse, this convergence criterion tends to weigh the long wavelengths more heavily than the short wavelengths. At each grid level the convergence tolerance is chosen such that the average error per grid point remains constant. For simple problems the algorithm is not sensitive to the tolerance of the coarse solvers; a lower tolerance may increase the number of iteration and decrease the overall efficiency. Harder problems require tighter tolerances for the coarse solvers.

The solver is very effective. We have successfully solved porous flow problems with random permeability fields as well as with discontinuous fields like the checkers problem. Fields with fluctuation on the order of 1000 typically required 3 to 15 iterations to achieve 6 digits of accuracy for each point of the pressure field. Even for extreme permeability ratios, 10^6 or larger, the number of iterations rarely exceeds 40 to 50.

In summary, we have developed a general method for solving linear systems derived from PDEs. The intuition based on the wave length decompositition, which we presented above, is born out by the numerical examples. The solver algorithm is robust, fast and memory efficient. We intend to improve further our implementation by introducing more sophisticated coarse graining or upscaling algorithms, and by generalizing the conjugate gradient to other preconditioned solvers such as generalized Krylov methods. Generalized Krylov methods would benefit even more than the conjugate gradient algorithm from the greatly reduced number of iterations that are made possible by our high quality preconditioner. With such more general methods, high resolution solution would become accessible for a large class of problems that can not easily be cast as symmetric and positive linear systems.

REFERENCES

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge Univ. Press, N.Y., 1994.

[2] O. Tatebe, *The Multigrid Preconditioned Conjugate Gradient Method*, Proceedings of the Sixth Copper Mountain conference on Multigrid Methods, 1993.