Updating of Preconditioners for Large, Sparse, Nonsymmetric Linear Systems

Jurjen Duintjer Tebbens and Miroslav Tůma

Institute of Computer Science,
Academy of Sciences of the Czech Republic
{tebbens,tuma}@cs.cas.cz

Abstract

THIS poster considers a technique to update preconditioners for the solution of sequences of large, sparse, nonsymmetric linear systems which was introduced by the authors in [2] and extended in [3, 1]. In addition to a basic description of the technique, the poster addresses some new relevant issues that can not be found in [2, 3, 1].

1. The Basic Preconditioner Update Technique

CONSIDER a system

$$Ax = b$$

with a factorized preconditioner

$$M = LDU \approx A.$$

We search for an updated preconditioner M^+ for a system

$$A^+x^+ = b^+$$

arising later in the sequence. With

$$B \equiv A - A^+$$

we have $\|A - M\| = \|A^+ - (M - B)\|$, hence $M - B$ is a preconditioner for A^+ of the same accuracy as M is for A. Clearly, for general B the preconditioner $M - B$ cannot be used in practice since the systems are too expensive to solve. Instead, we will consider cheap approximations of $M - B$, namely

$$M - B = L(DU - L^{-1}B) \approx L(DU - B) \approx L(DU - triu(B)), \quad (1)$$

where $triu$ denotes the upper triangular part, or

$$M - B = (LD - BU^{-1})U \approx (LD - B)U \approx (LD - tril(B))U, \quad (2)$$

where $tril$ denotes the lower triangular part. The choice between (1) and (2) can be based on the distance of L and U to identity. If $\|I-L\| < \|I-U\|$ then we base our updates on (1), otherwise we use (2). Alternatively, the choice between (1) and (2) can be based on comparison of the triangular parts of B; i.e. if $triu(B)$ dominates over $tril(B)$ (in some norm) we use (1), else we use (2).
Note the updated preconditioners (1) and (2)

- are very simple
- are per definition given as a product of triangular factors
- can be obtained with very few costs (only subtraction of a triangular matrix and some extra storage)
- are inexpensive to apply; the additional costs of back- and forward solves depend on the sparsity of $tril(B)$ and $triu(B)$.

As a typical example of the behavior of these updates, consider the two-dimensional nonlinear convection-diffusion model problem

$$-\Delta u + Ru\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) = 2000x(1-x)y(1-y),$$

on the unit square, discretized by 5-point finite differences on a uniform 70×70 grid. The initial approximation is the discretization of $u_0(x,y) = 0$ and $R = 50$. We obtain a small sequence of 7 matrices with 24220 nonzeros each. Table 1 displays the number of BiCGSTAB iterations necessary to solve the systems and the total time to solve the whole sequence.

Table 1: *Nonlinear convection-diffusion model problem with R=50, n=4900, nnz=24220, ILU(0).*

ILU(0), psize ≈ 24000			
Matrix	Recomp	Freeze	Update
$A^{(0)}$	40	40	40
$A^{(1)}$	29	36	32
$A^{(2)}$	21	39	27
$A^{(3)}$	20	48	26
$A^{(4)}$	17	55	26
$A^{(5)}$	16	58	29
$A^{(6)}$	15	50	22
$A^{(7)}$	15	62	26
$A^{(8)}$	17	68	28
$A^{(9)}$	15	71	27
$A^{(10)}$	15	51	24
overall time	11 s	7.5 s	5 s

Recomputing gives the lowest numbers of BiCGSTAB iterations, but is time-inefficient. Freezing the initial preconditioner and using it for all subsequent systems gives deteriorating numbers of BiCGSTAB iterations. Clearly, updating is most efficient.
More sophisticated adaptations of the simple basic updates (1) and (2) are contained in the three publications [2, 3, 1].

2. Some Theoretical Results

OUR goal is to find sufficient conditions for the updated preconditioner to be more powerful than a frozen or even recomputed preconditioner. In [2] we showed that the update (1) (or (2)) has the potential to be arbitrarily accurate and stable if L (or U) is close enough to identity and $triu(B)$ (or $tril(B)$) contains enough information of the whole difference matrix B. This is in accordance with the two approximation steps in (1) (or (2)). Unfortunately, the bounds used in these results are often poor in numerical experiments. In [1] we proposed a result which corresponds much better to the situation encountered in the experiments of that paper:

Lemma 1 *Denote by E the error $E \equiv A - LDU$ of the ILU(0) preconditioner and let*

$$\rho \equiv \frac{\|(I-L)triu(B)\|_F\,(2\|E - stril(B)\|_F + \|(I-L)triu(B)\|_F)}{\|triu(B)\|_F^2}$$

where $stril$ denotes the strict lower triangular part. If $\rho < 1$, then the accuracy of the updated preconditioner $\|A^+ - L(DU - triu(B))\|_F$ is higher than the accuracy of the frozen preconditioner $\|A^+ - LDU\|_F$ with

$$\|A^+ - L(DU - triu(B))\|_F \leq \sqrt{\|A^+ - LDU\|_F^2 - (1-\rho)\|triu(B)\|_F^2}.$$

P r o o f : See [1].
A new, alternative lemma with a weaker, but more transparent condition is

Lemma 2 *Let*

$$\sqrt{\|E\|_F^2 + \|stril(B)\|_F^2} < \frac{1 - \|I-L\|_F^2}{2\|I-L\|_F}\|triu(B)\|_F.$$

Then the accuracy of the updated preconditioner is higher than the accuracy of the frozen preconditioner.

P r o o f : Analogue to Lemma 1.
With Lemma 1 we obtain the following new theorem.

Theorem 3 *With the assumptions of Lemma 1, let*

$$\alpha = \frac{\|E\|_F^2 + \|stril(B)\|_F^2}{\|triu(B)\|_F^2}.$$

If A satisfies the diagonal dominance condition

$$\sum_{i \neq j}^{N} |a_{ij}| \leq \delta |a_{jj}|, \qquad j = 1, \ldots, N, \quad (3)$$

for some $\delta > 0$ with

$$0 < \delta < \sqrt{\alpha + 1} - \sqrt{\alpha}, \quad (4)$$

then the accuracy of the updated preconditioner is higher than the accuracy of the frozen preconditioner.

P r o o f : Based on the fact that with condition (3) we have $\|(I-L)triu(B)\|_F \leq \delta\|triu(B)\|_F$.

Interestingly, the stronger is the *triangular* dominance of $triu(B)$ over $stril(B)$, the weaker is the *diagonal* dominance needed in (3). Note all the results of this section assume an ILU(0)-decomposition; for different types of ILU-decompositions the results must be modified and proofs will be less elegant.

3. Matrix-free Environment

THE updates proposed in [2, 3, 1] were derived for situations where the system matrices of the sequence are explicitly given. Can we also use these updates in matrix-free environment ?
Let the entries of A have been estimated (e.g. by graph coloring) and the upper triangular part be stored explicitly. Estimation is necessary to compute the incomplete factorization LDU (whose factors we also store). Let the current matrix A^+ be given only in the form of its action on vectors, expressed by the function $f^+ : \mathbb{R}^n \to \mathbb{R}^n$. Our goal is to apply the updated preconditioner (1) without the necessity to estimate and store A^+ or its upper triangular part. We propose the following:

- Find the main diagonal of A^+: If f_i^+ denotes the ith component of the function f^+, then the diagonal $\{a_{11}^+, \ldots, a_{nn}^+\}$ can be found by computing

  $$a_{ii}^+ = f_i^+(e_i), \qquad 1 \leq i \leq n.$$

  By restricting the function evaluations to one component, the total costs to find the diagonal are those of one full function evaluation. This needs to be done only once before starting the iterative method.
- The forward solves with L are trivial.
- For the backward solves, use a mixed explicit-implicit strategy: Split $DU - triu(B)$ in the explicitly given part $X \equiv DU - triu(A)$ and the implicit part $triu(A^+)$ corresponding to the current system matrix. We then have to solve triangular systems of the form

  $$\left(X + triu(A^+)\right) z = y,$$

  yielding the standard cycle

  $$z_i = \frac{y_i - \sum_{j>i} x_{ij}z_j - \sum_{j>i} a_{ij}^+ z_j}{x_{ii} + a_{ii}^+}, \qquad i = n, \ldots, 1. \quad (5)$$

  In the nominator of (5) the first sum can be computed explicitly and the second sum can be computed by the function evaluation

  $$\sum_{j>i} a_{ij}^+ z_j = f_i^+\left((0, \ldots, 0, z_{i+1}, \ldots, z_n)^T\right).$$

  Thus the implicit part of the whole cycle (5) asks for one full function evaluation in total.

Assuming computation of individual components of functions can be separated, the cost difference between *defining* a recomputed and an updated preconditioner in matrix-free environment is rather important. Recomputing asks for 'matrix estimation algorithm + a number of matvecs + incomplete factorization' whereas the update asks for 'one full function evaluation' to obtain the main diagonal of A^+. On the other hand the *application* of the update is more expensive in matrix-free environment; in every backward solve we have one additional function evaluation.

References

[1] Ph. Birken, J. Duintjer Tebbens, A. Meister and M. Tůma *Preconditioner updates applied to CFD model problems*. Submitted to Applied Numerical Mathematics.

[2] J. Duintjer Tebbens and M. Tůma *Efficient preconditioning of sequences of nonsymmetric linear systems*. To appear in SIAM J. Sci. Comput. in 2007.

[3] J. Duintjer Tebbens and M. Tůma *Improving triangular preconditioner updates for nonsymmetric linear systems*. To appear in the proceedings of the 6th International Conference on Large-Scale Scientific Computations, Springer, LNCS, in 2007.