A Parallel Incomplete Block LU Factorization

Jos van Kan

Department of Technical Applied Analysis, Delft University of Technology, Delft, Netherlands

Abstract. A parallel algorithm is presented to construct an incomplete block factorization for an $N \times N$ block tridiagonal matrix in which the blocks themselves are also tridiagonal matrices of size $N_x \times N_x$. $N = N_x \times N_y$. The algorithm uses $O({}^2\log N_x \times {}^2\log N_y)$ recursive cycles, whereas a fully recursive algorithm would require $O(N)$ recursive cycles.
The factorization may be used as a preconditioner in a Krylov space method or as smoother in a multigrid method.

1 Introduction

Let A be an $N \times N$ tridiagonal block matrix consisting of blocks of size $N_x \times N_x$. The blocks themselves are also tridiagonal and $N = N_x \times N_y$. Such matrices occur for instance in the context of Laplace like equations in general coordinates.

Incomplete block LU factorization (IBLU) is known to be a powerful preconditioner for Krylov space methods like CG, BiCGStab [7] and GMRes [6]. It also is a powerful smoother for multigrid methods [3]. The conventional IBLU algorithm is strongly recursive on two levels: at the block level to make the LU decomposition and within a block to construct the seven diagonals of the inverse of a tridiagonal matrix. We shall show, that both these strong recursions can be replaced by algorithms that take typically $O({}^2\log M)$ recursive cycles on both levels. The outer recursion then will take $O({}^2\log N_y)$ recursive cycles, and on the blocklevel $O({}^2\log N_x)$ recursive cycles.

2 Full Cyclic Reduction for Tridiagonal Matrices

We shall start out by taking a different view on cyclic reduction. Consider an $M \times M$ tridiagonal matrix $A^0 = D^0 - L^0 - U^0$ in which D^0 is a a diagonal, L^0 a subdiagonal and U^0 a superdiagonal matrix. Now consider the following transformation:

$$\begin{aligned} A^1 &= (D^0 + L^0 + U^0)(D^0)^{-1}A \\ &= D^0 - L^0(D^0)^{-1}U^0 - U^0(D^0)^{-1}L^0 - L^0(D^0)^{-1}L^0 - U^0(D^0)^{-1}U^0 \\ &= D^1 - L^1 - U^1 \end{aligned} \tag{1}$$

where

$$D^1 = D^0 - L^0(D^0)^{-1}U^0 - U^0(D^0)^{-1}L^0 \tag{2}$$

$$L^1 = L^0(D^0)^{-1}L^0 \tag{3}$$

$$U^1 = U^0(D^0)^{-1}U^0 \tag{4}$$

In these formulae D^1 is again a diagonal matrix, and L^1 and U^1 are sub and superdiagonals respectively, *once removed*; that is, of the form $l^1_{i,i-2}$ and $u^1_{i,i+2}$ respectively. We can continue in the same vein and define recursively:

$$\begin{aligned} A^{k+1} &= (D^k + L^k + U^k)(D^k)^{-1}A \\ &= D^k - L^k(D^k)^{-1}U^k - U^k(D^k)^{-1}L^k - L^k(D^k)^{-1}L^k - U^k(D^k)^{-1}U^k \\ &= D^{k+1} - L^{k+1} - U^{k+1} \end{aligned} \tag{5}$$

where

$$D^{k+1} = D^k - L^k(D^k)^{-1}U^k - U^k(D^k)^{-1}L^k \tag{6}$$
$$L^{k+1} = L^k(D^k)^{-1}L^k \tag{7}$$
$$U^{k+1} = U^k(D^k)^{-1}U^k \tag{8}$$

The reader will have no trouble to verify, that L^k and U^k are of the form $l^k_{i,i-2^k}$ and $u^k_{i,i+2^k}$ respectively. As soon as 2^k exceeds M, the size of the matrix, L^k and U^k will be empty and hence it will take $^2\log M$ steps to reduce A^0 in this way to a diagonal matrix. This would make little sense for a serial computer, or a parallel computer with few processors. But all matrix multiplications are in principle parallel operations without recursion, so on a computer with M or more processors the number of cycles would effectively reduce to $O(^2\log M)$.

3 Incomplete Block LU Factorization (recursive)

Various authors [1], [2],[4] and [5] have considered incomplete block LU factorization, called IBLU (incomplete block LU), both as a preconditioner for a Krylov space method or as a smoother for a multi grid method. The principle is as follows. Let $A = D - L - U$ be a *block* tridiagonal matrix. We write (following an idea of Eisenstadt's)

$$A = (\tilde{D} - L)\tilde{D}^{-1}(\tilde{D} - U) \tag{9}$$

and we choose $\tilde{D}$ such, that $D = \tilde{D} + L\tilde{D}^{-1}U$. It will be readily seen, that the latter identity can be obtained by block recursion. The first block of D, D_1 and of $\tilde{D}$, $\tilde{D}_1$ are identical and we have the recursion

$$\tilde{D}_{j+1} = D_{j+1} + L_{j+1,j}\tilde{D}_j^{-1}U_{j,j+1} \tag{10}$$

So far this is an exact relation. If the blocks of D are themselves tridiagonal, the inverses will be full matrices. An incomplete version is obtained, by retaining only *three* diagonals in every $\tilde{D}_j$. Define the operator $\mathcal{E}_m$ as a *diagonal extractor*. $\mathcal{E}_0$ extracts the main diagonal, $\mathcal{E}_1$ three diagonals (sub, main and super), $\mathcal{E}_2$ five, etcetera. With this notation the incomplete version of 10 can be constructed as follows:

$$\tilde{D}_{j+1} = D_{j+1} + \mathcal{E}_1(L_{j+1,j}\tilde{D}_j^{-1}U_{j,j+1}) \tag{11}$$

The reader will have no trouble in verifying, that this is equivalent to

$$\tilde{D}_{j+1} = D_{j+1} + \mathcal{E}_1(L_{j+1,j}\mathcal{E}_3(\tilde{D}_j^{-1})U_{j,j+1})) \tag{12}$$

since $L_{j+1,j}$ and $U_{j,j+1}$ are tridiagonal. So in order to calculate this expression one needs the seven diagonals of $\tilde{D}_j^{-1}$. It is explained a.o. in [4] how to calculate those diagonals. This algorithm, however, is strongly recursive and unsuitable for parallel computation.

4 Full Cyclic Block Reduction

To find a parallel IBLU, let us reinterpret the formulae of section 2 in the following way. Let A be a *block* tridiagonal matrix. Let the diagonal D^0 of A consist of N_y blocks of size $N_x \times N_x$, $D_j^0, j = 1 \ldots N_y$, the superdiagonal U^0 of blocks $U_{j,j+1}^0, j = 1 \ldots N_y - 1$ and the subdiagonal L^0 of blocks $L_{j,j-1}^0, j = 2 \ldots N_y$. Now we may perform the exact same transformations as in section 2 to obtain a full block cyclic reduction. In fact, there is a complete block analogue to equations 1-8, using the same formulae but with the connotation that it concerns block matrices.

However, equations 6-8 may also be written down blockwise. Equation 6 then becomes:

$$\begin{aligned}
D_j^{k+1} &= D^k - U_{j,j+2^k}^k(D_{j+2^k}^k)^{-1}L_{j+2^k,j}^k, & j &= 1,\ldots,2^k & (13)\\
D_j^{k+1} &= D^k - L_{j,j-2^k}^k(D_{j-2^k}^k)^{-1}U_{j-2^k,j}^k + & & & (14)\\
&\quad - U_{j,j+2^k}^k(D_{j+2^k}^k)^{-1}L_{j+2^k,j}^k, & j &= 2^k+1,\ldots,N_y-2^k & \\
D_j^{k+1} &= D^k - L_{j,j-2^k}^k(D_{j-2^k}^k)^{-1}U_{j-2^k,j}^k, & j &= N_y-2^k+1,\ldots,N_y & (15)
\end{aligned}$$

Equation 7 becomes

$$L_{j+2^{k+1},j}^{k+1} = L_{j+2^{k+1},j+2^k}^k(D_{j+2^k}^k)^{-1}L_{j+2^k,j}^k, j = 1,\ldots,N_y - 2^{k+1} \tag{16}$$

and finally equation 8 becomes

$$U_{j,j+2^{k+1}}^{k+1} = U_{j,j+2^k}^k(D_{j+2^k}^k)^{-1}U_{j+2^k,j+2^{k+1}}^k \tag{17}$$

Note, that there is no recursion on j in these expressions. For any given k all calculations can be done in parallel. The only recursion left is that on k and for the same reason as in section 2 this will take ${}^2\log N_y$ cycles on a massively parallel machine.

There are a couple of ifs and buts about this. You have to calculate the inverses $(D_j^k)^{-1}$, which are size $N_x \times N_x$. As we have shown in section 2 this will take ${}^2\log N_x$ recursive cycles on a massively parallel machine. So the whole operation will take $O(\log N_x \times \log N_y)$ cycles. The memory requirements will be prohibitive though: $3 \times N_x \times N_y^2 \times {}^2\log N_x$ to store the intermediate matrices, because starting from A^1 the blocks will be full.

5 Incomplete Cyclic Block Reduction

To reduce the memory requirements one has to do an incomplete cyclic block reduction. Instead of using equations 13-17 we only retain the main, super and subdiagonal in every nonzero block. This gives the following set of recursions: For the diagonal blocks:

$$D_j^{k+1} = D^k - \mathcal{E}_1(U^k_{j,j+2^k}\mathcal{E}_3(D^k_{j+2^k}{}^{-1})L^k_{j+2^k,j}), \qquad j = 1,\ldots,2^k \tag{18}$$

$$\begin{aligned} D_j^{k+1} = D^k &- \mathcal{E}_1(L^k_{j,j-2^k}\mathcal{E}_3(D^k_{j-2^k}{}^{-1})U^k_{j-2^k,j})+ \\ &- \mathcal{E}_1(U^k_{j,j+2^k}\mathcal{E}_3(D^k_{j+2^k}{}^{-1})L^k_{j+2^k,j}), \qquad j = 2^k+1,\ldots,N_y-2^k \end{aligned} \tag{19}$$

$$D_j^{k+1} = D^k - \mathcal{E}_1(L^k_{j,j-2^k}\mathcal{E}_3(D^k_{j-2^k}{}^{-1})U^k_{j-2^k,j}), \qquad j = N_x - 2^k+1,\ldots,N_y \tag{20}$$

for the subdiagonal blocks

$$L^{k+1}_{j+2^{k+1},j} = \mathcal{E}_1(L^k_{j+2^{k+1},j+2^k}\mathcal{E}_3(D^k_{j+2^k}{}^{-1})L^k_{j+2^k,j}), j = 1,\ldots,N_y-2^{k+1} \tag{21}$$

and for the super diagonal blocks

$$U^{k+1}_{j,j+2^{k+1}} = \mathcal{E}_1(U^k_{j,j+2^k}\mathcal{E}_3(D^k_{j+2^k}{}^{-1})U^k_{j+2^k,j+2^{k+1}}), \quad j = 1,\ldots,N_y-2^{k+1} \tag{22}$$

It is easily verified, that the structure of all nonzero blocks remains tridiagonal.

6 The Magnificent Seven

The problem remains, however, how to calculate $\mathcal{E}_3(A^{-1})$ if A is $M \times M$ and tridiagonal. We recall from section 2 how the inverse was constructed:

$$A^{-1} = (D^{k+1})^{-1}E^kE^{k-1}\ldots E^1E^0 \tag{23}$$

with $E^j = (L^j + D^j + U^j)(D^j)^{-1}$ and $2^{k-1} < M \leq 2^k$. The typical structure of E^j is tridiagonal, with a main diagonal and a super and subdiagonal 2^j places removed. Let us denote the pattern of such a matix by $d_{2^j} + d_0 + d_{-2^j}$. The product of matrices of such a structure has a specific property, that will be the subject of the next lemma.

Lemma 1. *The pattern of nonzero diagonals of the product* $P^j = E^k \cdot E^{k-1} \cdot \dots \cdot E^{k-j}$ *is given by* $d_{m2^{k-j}}, m \in \mathbb{Z}, -M < m2^{k-j} < M$. *All other diagonals are 0.*

Proof. The lemma holds apparently for $j = 0$. Suppose it holds up to j. Since E^{k-j-1} has pattern $d_{2^{k-j-1}} + d_0 + d_{-2^{k-j-1}}$ the resulting matrix has a diagonal pattern that is a result of multiplying any existing diagonal by one of the three diagonals of E^{k-j-1}. One easily verifies, that multiplying a matrix consisting of a single diagonal d_n by a matrix consisting of a single diagonal d_m results in a matrix consisting of a single diagonal d_{n+m}. Now let $d_{m2^{k-j}}, m \in \mathbb{Z}$ be any diagonal of P^j. Multiplication by d_0 results in the same pattern: $d_{m2^{k-j}} = d_{(2m)2^{k-j-1}}$. Multiplication by $d_{2^{k-j-1}}$ results in a diagonal $d_{(2m+1)2^{k-j-1}}$ and multiplication by $d_{-2^{k-j-1}}$ in a diagonal $d_{(2m-1)2^{k-j-1}}$. □

The Lemma is instrumental in the following theorem that enables us to calculate $\mathcal{E}_3(A^{-1})$ in a parallel way.

Theorem 1. *To calculate 7 central nonzero diagonals of* P^k*, it is sufficient to retain 3 central nonzero diagonals of* $P^j, j = 1, k-2$ *and 5 central nonzero diagonals of* P^{k-1}.

Proof. Following the reasoning of Lemma 1 it will be clear, that to calculate the seven diagonals $d_{-3}, d_{-2}, \dots, d_2, d_3$ of P^k one needs $d_{-4}, d_{-2}, d_0, d_2, d_4$ of P^{k-1} and d_{-1}, d_0, d_1 of E^0. The five diagonals of P^{k-1} can be calculated from d_{-4}, d_0, d_4 of P^{k-2} and d_{-2}, d_0, d_2 of E^1. And the 3 central diagonals d_{-2^j}, d_0, d_{2^j} of P^{k-j} can be calculated from the 3 central diagonals $d_{-2^{j+1}}, d_0, d_{2^{j+1}}$ of P^{k-j-1} and d_{-2^j}, d_0, d_{2^j} of E^j. □

Remark It will be evident, that again there are only ${}^2\log M$ recursive cycles in this calculation and that within such a cycle all calculations can be executed in parallel.

References

1. O. Axelsson, S. Brinkkemper, and V.P. Il'in. On some versions of incomplete block matrix factorization iterative methods. *Lin. Algebra Appl. 59*, pages 3–15, 1984.
2. P. Concus, G.H. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comp. 6*, pages 220–252, 1985.
3. P.W. Hemker, R. Kettler, P. Wesseling, and P.M. De Zeeuw. Multigrid methods: Development of fast solvers. *Appl. Math. Comp.*, 13:311–326, 1983.
4. Robert Kettler. *Linear Multigrid Methods for Numerical Reservoir Simulation*. University of Technology, Delft, december 1987. Thesis.
5. G. Meurant. The block preconditioned conjugate gradient method on vector computers. *BIT*, 24:623–633, 1984.
6. Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.
7. H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.