MODELLING FOR ENGINEERING & HUMAN BEHAVIOUR 2022

PROCEEDINGS

Edited by

Juan Ramón Torregrosa
Juan Carlos Cortés
Antonio Hervás
Antoni Vidal
Elena López-Navarro

im²
Instituto Universitario
de Matemática Multidisciplinar

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Modelling for Engineering & Human Behaviour 2022

València, July 14th-16th, 2022

This book includes the extended abstracts of papers presented at XXIV Edition of the Mathematical Modelling Conference Series at the Institute for Multidisciplinary Mathematics *Mathematical Modelling in Engineering & Human Behaviour*.

I.S.B.N.: 978-84-09-47037-2

November 30th, 2022
Report any problems with this document to imm@imm.upv.es.

Edited by: I.U. de Matemàtica Multidisciplinar, Universitat Politècnica de València.
J.R. Torregrosa, J-C. Cortés, A. Hervás, A. Vidal-Ferràndiz and E. López-Navarro

im2
Instituto Universitario
de Matemática Multidisciplinar

Contents

Pivoting in ISM factorizations

J. Mas[♭,1] and J. Marín[♮]

(♭) Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, Spain.
(♮) Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, Spain.

1 Introduction

In this work we study pivoting techniques for the balanced incomplete factorization preconditioner (BIF) [7], for solving ill-conditioned sparse nonsingular linear systems of equations of the form

$$Ax = b,\ \ A \in \mathbb{R}^{n\times n},\ \ b \in \mathbb{R}^n \tag{1}$$

using iterative Krylov methods. There are different pivoting techniques being partial, complete and rook pivoting the more important ones [5,6]. Basically, at a given step of Gaussian elimination pivoting looks for an element sufficiently large in magnitude in the remaining submatrix, the Schur complement, to use it as the next pivot. These techniques involve row and possibly column permutations of the matrix that supposes a computational overhead. In this sense, partial pivoting is the cheapest pivoting technique, since it looks only in the first column of the Schur complement. Close behind is rook pivoting [6], it selects a pivot with maximum absolute value in his row and column, moving first to the biggest entry in magnitude in the first column, then it moves in the corresponding row, and then again in the column, and so on until the requirement is fulfilled. Finally, complete pivoting is the most expensive one, but guarantees the largest pivot at any stage however because the pivot is the entry of biggest magnitude in all the Schur complement.

BIF preconditioning is based on the incomplete Sherman-Morrison decomposition, ISM. The ISM decomposition uses recursion formulas derived from the Sherman-Morrison formula and was introduced in [7] as a method for computing approximate inverse preconditioners. In [8] and [9] the authors show that, applying the ISM algorithm to A and A^T, it is possible to compute an incomplete LDU factorization. Moreover, the inverse factors are also available and they influence the computation of the LDU factorization, and vice versa. In addition, the availability of the direct and inverse factors is exploited to implement norm based dropping rules [3]. The numerical results show that BIF is a robust algorithm comparable to other techniques as ILU(τ) [1], *ILUT* [10] and RIF [2]. Nevertheless, as mentioned above, computing stable (incomplete) factorizations for ill-conditioned problems still require the application of pivoting techniques. Here we show that with a slight modification of the ISM recursion formulas it is possible to incoporate pivoting to BIF.

[1]jmasm@imm.upv.es

2 The ISM decomposition

The ISM decomposition computes approximate inverse preconditioners since it obtains a factorization of the (shifted) inverse matrix of A, as

$$s^{-1}I - A^{-1} = s^{-2}ZD_s^{-1}V_s^T, \tag{2}$$

where $s > 0$ is a given scalar and the columns of the matrices Z and V_s are computed using the recursion formulas

$$z_k = e_k - \sum_{i=1}^{k-1} \frac{v_i^T e_k}{sr_i} z_i \quad \text{and} \quad v_k = y_k - \sum_{i=1}^{k-1} \frac{y_k^T z_i}{sr_i} v_i, \tag{3}$$

for $k = 1, 2, \ldots, n$. In (3) the vector e_k (e^k) denotes the $k-$th column (row) of the identity matrix, $y_k = (a^k - se^k)^T$ where a^k denotes the k-th row of A, and

$$r_k = 1 + y_k^T z_k/s = 1 + v_k^T e_k/s \tag{4}$$

are the entries of the diagonal matrix D_s.

It was proved in [8] that for symmetric matrices the factorization $A = LDL^T$ and the decomposition (2) satisfy

$$D = sD_s, \quad Z = L^{-T}, \quad V_s = LD - sL^{-T}.$$

The algorithm to get the decomposition of A uses explicitly the computed factors of A^{-1}, that is, A^{-1} is implicitly factorized at the same time. Therefore, to get the LU factorization for general matrices it is necessary to compute also the ISM decomposition of A^T that gives as result

$$\tilde{Z} = L^{-T}, \quad \text{and} \quad \tilde{V}_s = LD - sU^{-1},$$

where we have denoted with tilde the factors of the ISM decomposition of A^T.

It is well known that a nonsingular matrix A has an LU factorization if there exists a lower unit triangular matrix L and an upper triangular matrix U, such that $A = LU$. The LDU factorization is obtained from the LU factorization by taking D as the diagonal matrix whose entries are the diagonal entries of U, and applying its inverse to U as $D^{-1}U$. Both factorizations are closely related with Gaussian elimination. Note that not all the nonsingular matrices have LU factorization since a zero pivot can be found during the Gaussian elimination process. However it is always possible to permute some rows, and maybe some columns of the matrix in such a way that the permuted matrix PAQ has LU factorization. Here P and Q are permutation matrices acting on rows and columns of A, respectively.

The idea is that it is possible to find permutation matrices P and Q such that at the k-th step of the Gaussian elimination process one obtains the matrix

$$(PAQ)^{(k)} = \begin{bmatrix} L_{11} & O \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ O & S^{(k)} \end{bmatrix} \tag{5}$$

where the Schur complement $S^{(k)} = A_{22} - A_{21}A_{22}^{-1}A_{12}$ is nonsingular and its first diagonal element is nonzero. Then, the permuted matrix PAQ is factorized as

$$PAQ = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}. \tag{6}$$

Note that in practice, the permutation matrices P and Q are not known in advance and therefore LU factorization algorithms determine which rows and columns must be interchanged during the elimination process.

3 Right looking ISM algorithm

To implement pivoting in the ISM decomposition it is necessary to know the Schur complement of the LU factorization. To accomplish that the vectors z_k and v_k must be computed in a different way. Instead of computing only one pair of vectors in the k-th step of the algorithm according to equations (3), the modification consist in updating also the remaining vectors, from $k+1$ to n. That is, the right part of the matrices Z and V are updated in each step. The following MATLAB code implements the new right looking version of ISM.

Algorithm 1 *The ISM right looking algorithm*

```
function [Z, V, D] = ismrl(A)
n = size(A,1);
Y = (A-eye(n))';
Z = eye(n);
V = A'-eye(n);
D = zeros(n,1);
for k=1:n-1
D(k) = 1+V(k,k);
for l = k+1:n
Z(:,l) = Z(:,l) - V(l,k)/D(k)*Z(:,k);
V(:,l) = V(:,l) - (Y(:,l)'*Z(:,k))/D(k)*V(:,k);
end
end
D(n)=1+V(n,n);
```

The next results show that the Schur complement $S^{(k)}$ is available from the matrix V. We denote by $V_{22}^{(k)}$ the $(n-k)\times(n-k)$ submatrix of V in Algorithm 1 after step k, with rows and columns with indexes in $\{k+1,\ldots,n\}$.

Theorem 12. *If A is a nonsingular matrix, then at the k-th step of the right looking ISM algorithm 1*

$$V_{22}^{(k)} = {S^{(k)}}^T - I \tag{7}$$

Corollary 2. *If the right looking algorithm is applied to A^T then*

$$\tilde{V}_{22}^{(k)} = S^{(k)} - I$$

To introduce pivoting strategies the relation

$$V_{22}^{(k)} = {S^{(k)}}^T - I$$

should be taken into account. The new pivot is looked for into the submatrix $V_{22}^{(k)} + I$ that corresponds to the transpose of the same submatrix in $A^{(k)}$ in Gaussian elimination. Thus, in partial pivoting if two columns k and $p > k$ are permuted at step k in matrix $V^{(k)}$, the rows k and p should be permuted in A.

Also note that the pivoting strategy should be decided looking into the Schur complement contained in V_s, or that in $\tilde{V}_s$, but not both. In contrast, for complete pivoting it is clear that V_s or $\tilde{V}_s$ produce the same pivot in exact arithmetic so any of them or both may be used.

Table 1: Test problems

Matrix	n	nz	$cond(A)$	Application
adder_dcop_06	1,813	11,224	$1.1 \cdot 10^{12}$	circuit simulation matrix
adder_dcop_19	1,813	11,224	$5.9 \cdot 10^{11}$	circuit simulation matrix
oscil_dcop_01	1,813	11,224	$5.9 \cdot 10^{12}$	circuit simulation problem
oscil_dcop_57	1,813	11,224	$1.4 \cdot 10^{21}$	circuit simulation problem
radfr1	1,048	13,299	$5.9 \cdot 10^{10}$	chemical process separation

4 Numerical experiments

In this section we report the results of some numerical experiments with a set of matrices from The Univesity of Florida Sparse Matrix Collection [4]. The matrices are listed in Table 1 where their size, number of nonzeros, condition number and application are indicated. They correspond to very ill-conditioned and highly indefinite problems for which Gaussian elimination without pivoting fails to compute good quality L and U factors, so the same is expected to be the case for incomplete LU factorizations. Partial, rook and complete pivoting techniques have been tested. The experiments have been implemented and run in MATLAB R2022. As iterative solvers the MATLAB implementation of full GMRES [11] and BiCGStab [12] were used. The iterations were stopped when the initial residual was reduced by 8 orders of magnitude with a maximum number of $1,000$ iterations. The right hand side vector was computed such that the solution was the vector of all ones. To compare the results obtained with BIF the problems were also solved with the MATLAB's incomplete LU preconditioner with partial pivoting, ILUTP.

The implementation of the BIF preconditioner is based on the algorithm described in [9] but with the right looking modification described in Section 3. For simplicity, all the experiments have been done with the the parameter s of the ISM decomposition equal to one. The algorithm is implemented such that the ISM decompositions of A and A^T are computed at the same time. Therefore, accessing to A and A^T simultaneously is needed. The pivot is choosen from the Schur complement contained in V_s rather than $\tilde{V}_s$. We note that for complete pivoting the same pivot could be obtained working either with V_s or $\tilde{V}_s$ but we choose working with V_s for simplicity.

In Table 2 the pivoting strategy is indicated with C, P and R for the complete, partial and rook pivoting strategies, respectively. *Density* is the ratio between the number of nonzeros of the preconditioner and the matrix. Column *iter* shows the number of iterations of the solver and *droptol* is the tolerance used to drop elements in BIFP and ILUTP. The other columns are self explanatory. To reduce the numbers in the tables, a blank space means that the value is the same appearing in previous rows. For instance, in Table 2 the *droptol* value for BIFP was always 10^{-6} and therefore it appears only in the first row. The same holds for the preconditioner densities which are the same for GMRES and BiCSTAB and therefore only indicated once.

Next, we will comment on the results. We note that the matrices tested can not be solved without pivoting with both BIFP and ILUTP preconditioners. Thus, pivoting is an essential tool to gain robustness for these factorizations. From the University of Florida test matrices, Table 2, we observe for the adder group that there are not big differences between the different pivoting strategies for BIFP. Density is small, except for adder_dcop_06. The same can be said for the number of iterations spent by both iterative solvers. For the rest of matrices one can see that BIFP with complete pivoting computes sparser preconditioners than partial and rook pivoting. The iteration count does not present remarkable differences except for the oscil_dcop_01 matrix for which GMRES with partial pivoting, although with larger nonzero density, doubles the number of iterations.

Finally, comparing the performance of BIFP with ILUTP we did not observed significative differences, specially with the preconditioned BiCGStab method. We recall that ILUTP uses partial pivoting and we observe that BIFP with this pivoting strategy performed closely in most cases.

5 Conclusions

In this paper we have presented an improved version of the BIF preconditioner that incorporates pivoting. The algorithm relies on a modification of the recursion formulas such that the Schur complement of standard Gaussian elimination is available at each step of the factorization. Thus, the application of different pivoting techniques, as for instance partial, rook and complete pivoting, can be done in a straightforward maner. Incorporating pivoting turns out to be an important step in order to achieve our initial goal of obtaining a more robust preconditioner since it is able to solve very ill-conditioned and indefinite problems that it may not be possible to solve in other way. The results of the numerical experiments with several matrices arising in different applications confirm that BIF with pivoting is a robust algorithm. Partial, rook and complete pivoting has been tested. Although complete pivoting very often produces sparser preconditioners with a competitive iteration count, rook and partial pivoting perform also quite well. Taking into account that partial and rook are less expensive from a computational point of view since they need less comparisons in order to determine the pivot, these two techniques may be prefereable as default.

References

[1] M. Benzi and M. Tůma. A comparative study of sparse approximate inverse preconditioners. *Appl. Numer. Math.*, 30(2-3):305–340, 1999.

[2] M. Benzi and M. Tůma. A robust incomplete factorization preconditioner for positive definite matrices. *Numer. Linear Algebra Appl.*, 10(5-6):385–400, 2003.

[3] M. Bollhöfer. A robust and efficient *ILU* that incorporates the growth of the inverse triangular factors. *SIAM J. Sci. Comput.*, 25(1):86–103, 2003.

[4] Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, December 2011.

[5] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 2002.

[6] George Poole and Larry Neal. The rook's pivoting strategy. *Journal of Computational and Applied Mathematics*, 123:353–369, 11 2000.

[7] J. Marín R. Bru, J. Cerdán and J. Mas. Preconditioning sparse nonsymmetric linearsystems with the sherman-morrison formula. *SIAM J. Sci. Comput.*, 25(2):701–715, 2003.

[8] J. Mas R. Bru, J. Marín and M. Tuma. Balanced incomplete factorization. *SIAM J. Sci. Comput.*, 30(5):2302–2318, 2008.

[9] J. Mas R. Bru, J. Marín and M. Tuma. Improved balanced incomplete factorization. *SIAM J. Matrix Anal. Appl.*, 31(5):2431–2452, 2010.

[10] Y. Saad. ILUT: A dual threshold incomplete lu factorization. *Numer. Linear Algebra Appl.*, 1(4):387–402, 1994.

[11] Y. Saad and M. Schulz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[12] H.A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM Journal on Scientific and Statistical Computing.*, 12:631–644, 1992.

Table 2: Test results for the University of Florida matrices

Matrix	*precond*	*solver*	*droptol*	*piv*	*density*	*iter*
	BIFP	GMRES	10^{-6}	C	1.76	4
				P	1.81	3
				R	2.92	4
adder_dcop_06		BiCGStab		C		1
				P		1
				R		1
	ILUTP	GMRES	10^{-7}	P	1.67	3
		BiCGStab		P		1
	BIFP	GMRES	10^{-6}	C	0.69	3
				P	0.94	3
				R	0.72	3
adder_dcop_19		BiCGStab		C		2
				P		1
				R		2
	ILUTP	GMRES	10^{-2}	P	0.70	6
		BiCGStab		P		2
	BIFP	GMRES	10^{-7}	C	2.06	10
				P	2.64	19
				R	2.08	11
oscil_dcop_01		BiCGStab		C		1
				P		4
				R		1
	ILUTP	GMRES	10^{-9}	P	2.70	3
		BiCGStab		P		1
	BIFP	GMRES	10^{-16}	C	2.31	11
			10^{-11}	P	2.28	28
			10^{-16}	R	2.57	11
oscil_dcop_57		BiCGStab		C		1
				P		2
				R		1
	ILUTP	GMRES	10^{-16}	P	2.74	11
		BiCGStab		P		1
	BIFP	GMRES	10^{-2}	C	2.25	3
			10^{-5}	P	3.86	1
			10^{-2}	R	2.70	2
radfr1		BiCGStab		C		9
				P		3
				R		8
	ILUTP	GMRES	10^{-3}	P	2.69	2
		BiCGStab		P		10