Research Article

Different Versions of ILU and IUL Factorizations Obtained from Forward and Backward Factored Approximate Inverse Processes—Part I

Amin Rafiei and Fatemeh Shahlaei

Department of Applied Mathematics, Sabzevar Tarbiat Moallem University, Sabzevar, Iran

Correspondence should be addressed to Amin Rafiei, rafiei.am@gmail.com

Received 29 August 2011; Accepted 6 October 2011

Academic Editor: Raytcho Lazarov

Copyright © 2011 A. Rafiei and F. Shahlaei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an incomplete UL (IUL) decomposition of matrix $A$ which is extracted as a by-product of BFAPINV (backward factored approximate inverse) process. We term this IUL factorization as IULBF. We have used ILUFF and IULBF as left preconditioner for linear systems. Different versions of ILUFF and IULBF preconditioners are computed by using different dropping techniques. In this paper, we compare quality of different versions of ILUFF and IULBF preconditioners.

1. Introduction

Consider the linear system of equations

$$AX = b, \quad (1.1)$$

where the coefficient matrix $A \in \mathbb{R}^{n \times n}$ is nonsymmetric, nonsingular, large, sparse, and $X, b \in \mathbb{R}^{n}$. Suppose $M \approx A$. Linear system

$$M^{-1}AX = M^{-1}b, \quad (1.2)$$

is termed left preconditioned system of system (1.1) and matrix $M$ is called left preconditioner matrix [1]. System (1.2) is solved by Krylov subspace methods [1].

Suppose that matrix $A$ is nonsymmetric. Also, suppose that $W = [w_1^T, \ldots, w_n^T]^T$ and $Z = [z_1, \ldots, z_n]$ are unit lower and upper triangular matrices, respectively, and
\[ D = \text{diag}(d_1, \ldots, d_n) \] is a diagonal matrix. FFAPINV (forward factored approximate inverse) Algorithm [2], computes matrices \( W, Z, \) and \( D \) such that relation
\[ WAZ \approx D, \quad (1.3) \]
holds. It is possible to obtain an incomplete \( LU \) (ILU) decomposition of matrix \( A \), as a by-product of FFAPINV process, such that \( L \) is an unit lower triangular and \( U \) is an upper triangular matrix and
\[ A \approx M = LU. \quad (1.4) \]

Matrix \( M \) in (1.4) is called ILUFF preconditioner (ILU factorization obtained from forward factored approximate inverse process) [3]. The approximate inverse factors \( W, Z \) and \( D \) in (1.3) and \( L, U \) matrices in (1.4) satisfy the two following relations:
\[ L \approx W^{-1}, \quad U \approx DZ^{-1}. \quad (1.5) \]

In Algorithms 1 and 2, \( A_{:,j} \) and \( A_{j,:} \) refer to \( j \)th column and \( j \)th row of matrix \( A \), respectively.

In Section 2 of this paper, we present different dropping strategies for \( W, Z \) and \( L, U \) factors of ILUFF preconditioner. In Section 3, we first introduce the IULBF preconditioner and then, we present different dropping strategies for this preconditioner. In Section 4, we present numerical results.

2. Different Versions of ILUFF Preconditioner

Algorithm 1, which has been presented in the next page, computes the ILUFF preconditioner.

Suppose that \( \varepsilon_Z \) and \( \varepsilon_W \) are the drop tolerance parameters for \( Z \) and \( W \) matrices, respectively. We have used two strategies to drop entries of \( z_j \) and \( w_j \) vectors in ILUFF algorithm.

(i) First Dropping Strategy

In this strategy, only line 8 of Algorithm 1 will be run and line 10 will not. In this case, entries \( z_{lj} \) and \( w_{jl} \), for \( l \leq i < j \) are dropped when
\[ |z_{lj}| \leq \varepsilon_Z, \quad |w_{jl}| \leq \varepsilon_W. \quad (2.1) \]

(ii) Second Dropping Strategy

In this strategy, only line 10 of Algorithm 1 will be run and line 8 will not. In this case, the whole vectors \( z_j \) and \( w_j \) are computed as
\[ z_j = e_j - \sum_{i=1}^{j-1} \left( \frac{w_i A_{:,i}}{d_i} \right) z_i, \quad w_j = e_j^T - \sum_{i=1}^{j-1} \left( \frac{A_{j,:} z_i}{d_i} \right) w_i, \quad (2.2) \]
and then, entries \( w_{jl} \) and \( z_{lj} \), for \( l \leq j \), are dropped when criterions (2.1) are satisfied.

We have used two strategies to drop entries of \( L \) and \( U \) matrices in ILUFF algorithm.
(i) Inverse-Based Dropping Strategy

Let $\varepsilon_{L,W}$ be the same drop tolerance parameter for $L$ and $W$ matrices and $\varepsilon_{U,Z}$ be the same drop tolerance parameter for $U$ and $Z$ matrices. Consider $\varepsilon_{L,W}$ as $\varepsilon_W$ and $\varepsilon_{U,Z}$ as $\varepsilon_Z$. We drop entries $z_{ij}$ and $w_{jl}$, for $l \leq i < j$, when criterions (2.1) hold. Then, in line 6 of Algorithm 1, entries $L_{ij}$ and $U_{ij}$, for $i < j$, are dropped when

$$ |L_{ij}|\|W_{ij}\|_1 \leq \varepsilon_{L,W}, \quad |U_{ij}|\|Z_{ij}\|_\infty \leq \varepsilon_{U,Z}. $$

\[ (2.3) \]
(ii) Simple Dropping Strategy

Let \( \varepsilon_L \) and \( \varepsilon_U \) be the drop tolerance parameters for \( L \) and \( U \) matrices. In line 6 of Algorithm 1, entries \( L_{ji} \) and \( U_{ij} \), for \( i < j \), are dropped when

\[
|L_{ji}| \leq \varepsilon_L, \quad |U_{ij}| \leq \varepsilon_U.
\] (2.4)

Different versions of ILUFF preconditioners are computed by using different dropping strategies in Algorithm 1.

(i) ILUFF1

In Algorithm 1, first dropping strategy is used to drop entries of \( W \) and \( Z \) matrices and simple dropping strategy is used to drop entries of \( L \) and \( U \) matrices.

(ii) ILUFF2

In Algorithm 1, first dropping strategy is used to drop entries of \( W \) and \( Z \) matrices and inverse-based dropping strategy is used to drop entries of \( L \) and \( U \) matrices.

(iii) ILUFF3

In Algorithm 1, second dropping strategy is used to drop entries of \( W \) and \( Z \) matrices and simple dropping strategy is used to drop entries of \( L \) and \( U \) matrices.

(iv) ILUFF4

In Algorithm 1, second dropping strategy is used to drop entries of \( W \) and \( Z \) matrices and inverse-based dropping strategy is used to drop entries of \( L \) and \( U \) matrices.

3. IULBF Preconditioner and Its Different Versions

Suppose that \( W = [w_1^T, \ldots, w_n^T]^T \) and \( Z = [z_1, \ldots, z_n] \) are unit upper and lower triangular matrices, respectively, and \( D = \text{diag}(d_1, \ldots, d_n) \) is a diagonal matrix. BFAPINV algorithm [2, 4] computes matrices \( W, Z, \) and \( D \) such that relation (1.3) holds. We obtain an IUL decomposition of matrix \( A \), as a by-product of BFAPINV process, such that \( L \) is a lower triangular and \( U \) is an unit upper triangular matrix and

\[
A \approx M = UL.
\] (3.1)

Matrix \( M \) in relation (3.1) is called IULBF preconditioner (IUL factorization obtained from backward factored approximate inverse process). Algorithm 2 computes the IULBF preconditioner. The approximate inverse factors \( W, Z, \) and \( D \) in (1.3) and \( L, U \) matrices in (3.1) satisfy the two following relations:

\[
U \approx W^{-1}, \quad L \approx DZ^{-1}.
\] (3.2)
Suppose that \( \varepsilon_Z \) and \( \varepsilon_W \) are the drop tolerance parameters for \( Z \) and \( W \) matrices, respectively. We have used two strategies to drop entries of \( z_j \) and \( w_j \) vectors in IULBF algorithm.

(i) First Dropping Strategy

In this strategy, only line 8 of Algorithm 2 will be run and line 10 will not. In this case, entries \( z_{lj} \) and \( w_{jl} \), for \( j < i \leq l \), are dropped when criteria

\[
\| z_{lj} \| \leq \varepsilon_Z, \quad \| w_{jl} \| \leq \varepsilon_W,
\]

hold.

(ii) Second Dropping Strategy

In this strategy, only line 10 of Algorithm 2 will be run and line 8 will not. In this case, the whole vectors \( z_j \) and \( w_j \) are computed as

\[
z_j = e_j - \sum_{i=j+1}^{n} \left( \frac{w_i A_i}{d_i} \right) z_i, \quad w_j = e_j^T - \sum_{i=j+1}^{n} \left( \frac{A_i z_i}{d_i} \right) w_i,
\]

and then, entries \( w_{jl} \) and \( z_{lj} \), for \( l \geq j \), are dropped when criteria (3.3) are satisfied.

We have used two strategies to drop entries of \( L \) and \( U \) matrices in IULBF algorithm.

(i) Inverse-Based Dropping Strategy

Let \( \varepsilon_{U,W} \) be the same drop tolerance parameter for \( U \) and \( W \) matrices and \( \varepsilon_{L,Z} \) be the same drop tolerance parameter for \( L \) and \( Z \) matrices. Consider \( \varepsilon_{U,W} \) as \( \varepsilon_W \) and \( \varepsilon_{L,Z} \) as \( \varepsilon_Z \). We drop entries \( z_{ij} \) and \( w_{ji} \), for \( j < i \leq l \), when criteria (3.3) hold. Then, in line 6 of Algorithm 2, entries \( L_{ij} \) and \( U_{ji} \), for \( i > j \), are dropped when

\[
\| L_{ij} \| \| Z_{:j} \|_\infty \leq \varepsilon_{L,Z}, \quad \| U_{ji} \| \| W_{:i} \|_1 \leq \varepsilon_{U,W}.
\]

(ii) Simple Dropping Strategy

Let \( \varepsilon_L \) and \( \varepsilon_U \) be the drop tolerance parameters for \( L \) and \( U \) matrices. In line 6 of Algorithm 2, entries \( L_{ij} \) and \( U_{ji} \), for \( i > j \), are dropped when

\[
\| L_{ij} \| \leq \varepsilon_L, \quad \| U_{ji} \| \leq \varepsilon_U.
\]

Different versions of IULBF preconditioner are computed by using different dropping strategies in Algorithm 2.

(i) IULBF1

In Algorithm 2, first dropping strategy is used to drop entries of \( W \) and \( Z \) matrices and simple dropping strategy is used to drop entries of \( L \) and \( U \) matrices.
Table 1: Information of GMRES(16) method without preconditioning and matrix properties.

| Matrix    | n  | nnz  | PD | Itime | it |
|-----------|----|------|----|-------|----|
| hor-131   | 434| 4182 | No | 67.594| 4273|
| sherman2  | 1080| 23094| No | +     | +   |
| cavity05  | 1182| 32632| No | 0.875 | 27  |
| cavity06  | 1182| 29675| No | +     | +   |
| sherman4  | 1104| 3786 | No | 0.531 | 23  |
| epb0      | 1794| 7764 | No | +     | +   |
| pde2961   | 2961| 14585| yes| 0.734 | 18  |

(ii) IULBF2

In Algorithm 2, first dropping strategy is used to drop entries of $W$ and $Z$ matrices and inverse-based dropping strategy is used to drop entries of $L$ and $U$ matrices.

(iii) IULBF3

In Algorithm 2, second dropping strategy is used to drop entries of $W$ and $Z$ matrices and simple dropping strategy is used to drop entries of $L$ and $U$ matrices.

(iv) IULBF4

In Algorithm 2, second dropping strategy is used to drop entries of $W$ and $Z$ matrices and inverse-based dropping strategy is used to drop entries of $L$ and $U$ matrices.

4. Numerical Results

In this section, we report results of left preconditioned GMRES(16) method [1]. Preconditioners are ILUFF1, ILUFF2, ILUFF3, ILUFF4, IULBF1, IULBF2, IULBF3, and IULBF4. All coefficient matrices are nonsymmetric and from University of Florida Sparse Matrix Collection [5]. Vector $b$ is $Ae$ in which $e = [1, \ldots, 1]^T$. We have written codes of ILUFF1, ILUFF2, ILUFF3, ILUFF4, IULBF1, IULBF2, IULBF3, IULBF4, and GMRES(16) in MATLAB, and we have run all the experiments on a machine with 1GB of RAM memory. In all the experiments, if the pivot element $d_i$ (lines 11 and 12 of ILUFF and IULBF algorithms) is less than the machine precision, then we replace it by $10^{-4}$. Density of preconditioners is defined as

$$\text{density} = \frac{\text{nnz}(L) + \text{nnz}(U)}{\text{nnz}(A)},$$

in which $\text{nnz}(L)$, $\text{nnz}(U)$, and $\text{nnz}(A)$ refer to the number of nonzero entries of $L$, $U$, and $A$ matrices, respectively. In all the experiments, we have selected $\varepsilon_L, \varepsilon_U, \varepsilon_W, \varepsilon_Z, \varepsilon_{LZ}, \varepsilon_{LU}, \varepsilon_{LW}$, and $\varepsilon_{UZ}$ equal to 0.1.

Table 1 reports results of GMRES(16) method without preconditioning. In this table, $n$ indicates the dimension of the matrix and $PD$ column indicates whether or not the matrix is positive definite. Yes (no) in this column means that the matrix is (is not) positive definite.
Table 2: properties of ILUFF1, ILUFF2, ILUFF3, and ILUFF4 preconditioners.

| Method    | ILUFF1 |       | ILUFF2 |       | ILUFF3 |       | ILUFF4 |       |
|-----------|--------|-------|--------|-------|--------|-------|--------|-------|
|           | Density| Ptime | Density| Ptime | Density| Ptime | Density| Ptime |
| hor-131   | 0.984696| 22.031| 1.098996| 43.86 | 0.990435| 32.422| 1.075562| 32.64 |
| sherman2  | 0.463237| 466.531| 0.682342| 742.432| 0.462847| 637.203| 0.687668| 666.312|
| cavity05  | 0.272646| 736.343| 0.338992| 1567.223| 0.295636| 1079.63| 0.368013| 1139.59|
| cavity06  | 0.291794| 678.782| 0.376243| 1484.22| 0.295636| 993| 0.407515| 817.828|
| sherman4  | 1.243001| 266.203| 1.312467| 804.922| 1.243001| 319.172| 1.321447| 574.328|
| epb0      | 1.575348| 943.093| 1.981968| 2360.44| 1.750386| 1274.88| 2.248583| 1777.27|
| pde2961   | 1.234763| 6996.83| 1.327048| 10863| 1.248269| 5879.3| 1.334248| 9262.47|

Jtime, indicates the iteration time of GMRES(16) without preconditioning and it is the number of iterations of GMRES(16) method. Jtime is in seconds. In this table, + means that there is no convergence after 10,000 iterations. In all the experiments, the stopping criterion is

\[
\frac{\|r_k\|_2}{\|r_0\|_2} \leq 10^{-8},
\]

in which \(r_k\) is the kth residual vector of the system and \(r_0\) is the initial residual vector. In all the experiments, the initial guess is the zero vector.

In Table 2, the information of ILUFF1, ILUFF2, ILUFF3, and ILUFF4 preconditioners are presented and also in Table 3, the information of IULBF1, IULBF2, IULBF3, and IULBF4 preconditioners are presented. In Tables 2 and 3, Ptime is the preconditioning time and density is the density of preconditioner. Ptime is also in seconds.

In Table 4, results of left preconditioned systems by using different versions of ILUFF preconditioner have been presented, and also in Table 5 results of left preconditioned systems by using different versions of IULBF preconditioner have been presented. In Tables 4 and 5, Ttime is the total time which is the sum of preconditioning time and iteration time, and it is the number of iterations of left preconditioned GMRES(16). In these tables, + indicates that no convergence has been obtained in 5000 iterations.

Table 3: Properties of IULBF1, IULBF2, IULBF3, and IULBF4 preconditioners.

| Method    | IULBF1 |       | IULBF2 |       | IULBF3 |       | IULBF4 |       |
|-----------|--------|-------|--------|-------|--------|-------|--------|-------|
|           | Density| Ptime | Density| Ptime | Density| Ptime | Density| Ptime |
| hor-131   | 0.893352| 34.422| 1.104017| 62.11 | 1.046628| 26.219| 1.889527| 63.328|
| sherman2  | 1.434745| 766.187| 2.834069| 1166.11| 0.823720| 562.094| 1.335498| 1263.34|
| cavity05  | 0.682030| 1596.86| 1.478304| 1887.42| 0.686106| 803.25| 1.538000| 2071.98|
| cavity06  | 0.651693| 663.953| 1.549958| 1870.22| 0.657894| 791.422| 1.628745| 1337.67|
| sherman4  | 1.270470| 789.687| 1.345483| 870.672| 1.386952| 459.625| 2.055203| 622.031|
| epb0      | 0.850721| 1077.48| 0.853941| 3577.97| 1.009274| 951.688| 1.356775| 2924.58|
| pde2961   | 1.278642| 7344.64| 1.357011| 15909.7| 1.309427| 4634.16| 2.180871| 9440.33|

...
Table 4: Information of preconditioned GMRES(16) method by using different versions of ILUFF preconditioner.

| Method          | ILUFF1 | ILUFF2 | ILUFF3 | ILUFF4 |
|-----------------|--------|--------|--------|--------|
| hor-131         | it     | T time | it     | T time | it     | T time | it     | T time |
| Sherman2        | +      | +      | +      | +      | +      | +      | +      | +      |
| cavity05       | +      | +      | +      | +      | +      | +      | 191    | 2324.7 |
| cavity06       | +      | +      | +      | +      | +      | +      | 96     | 926.188 |
| sherman4       | 3      | 270.324| 3      | 810.734| 3      | 322.687| 3      | 577.485 |
| pde2961        | 12     | 978.405| 8      | 2385.63| 13     | 1335.89| 8      | 1795.58 |

Table 5: Information of preconditioned GMRES(16) method by using different versions of IULBF preconditioner.

| Method          | IULBF1 | IULBF2 | IULBF3 | IULBF4 |
|-----------------|--------|--------|--------|--------|
| hor-131         | it     | T time | it     | T time | it     | T time |
| Sherman2        | +      | +      | +      | +      | +      | +      |
| cavity05       | 3      | 1600.69| 2      | 1891.52| 3      | 808.703| 1      | 2074.26 |
| cavity06       | 3      | 668.093| 2      | 1872.95| 3      | 795.438| 1      | 1339.13 |
| sherman4       | 3      | 792.609| 2      | 874.141| 2      | 462.016| 2      | 626.094 |
| pde2961        | 24     | 1142.06| 23     | 3673.17| 22     | 1025.64| 22     | 3010.25 |

5. Conclusion

Results of Tables 1 and 4 show that ILUFF1, ILUFF2, ILUFF3, and ILUFF4 preconditioners are useful tools to decrease the number of iterations of GMRES(16) method and results of Tables 1 and 5 show that IULBF1, IULBF2, IULBF3, and IULBF4 preconditioners are also useful tools to decrease the number of iterations of GMRES(16) method.

Comparison of columns 2 and 6 of Table 4 indicates that sometimes ILUFF3 preconditioner decreases the number of iterations of GMRES(16) method a little bit more than ILUFF1 preconditioner and some other times it is vice versa. Comparison of columns 2 and 4 and columns 6 and 8 of this table, also shows that ILUFF2 preconditioner decreases the number of iterations of GMRES(16) method more than ILUFF1 preconditioner and ILUFF4 preconditioner decreases the number of iterations of GMRES(16) method more than ILUFF3 preconditioner.

Comparison of columns 2 and 6 of Table 5 indicates that IULBF3 preconditioner decreases the number of iterations of GMRES(16) method a little bit more than IULBF1 preconditioner. Comparison of columns 2 and 4 and columns 6 and 8 of this table, also shows that IULBF2 preconditioner decreases the number of iterations of GMRES(16) method more than IULBF1 preconditioner and IULBF4 preconditioner decreases the number of iterations of GMRES(16) method more than IULBF3 preconditioner.

Comparison of columns of Tables 4 and 5 indicate that (except for matrix epb0) different versions of IULBF preconditioner decrease the number of iterations of GMRES(16) method more than different versions of ILUFF preconditioner.
References

[1] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, Mass, USA, 1996.
[2] D. K. Salkuyeh, “A sparse approximate inverse preconditioner for nonsymmetric positive definite matrices,” *Journal of Applied Mathematics & Informatics*, vol. 28, no. 5-6, pp. 1131–1141, 2010.
[3] D. K. Salkuyeh, A. Rafiei, and H. Roohani, “ILU preconditioning based on the FAPINV algorithm,” [http://arxiv.org/abs/1010.2812](http://arxiv.org/abs/1010.2812).
[4] J.-C. Luo, “A new class of decomposition for inverting asymmetric and indefinite matrices,” *Computers & Mathematics with Applications*, vol. 25, no. 4, pp. 95–104, 1993.
[5] T. Davis, “University of Florida Sparse Matrix Collection,” [http://www.cise.ufl.edu/research/sparse/matrices/](http://www.cise.ufl.edu/research/sparse/matrices/).
Submit your manuscripts at http://www.hindawi.com