ILU and IUL factorizations obtained as by-products of FFAPINV and BFAPINV algorithms

List of authors:

A. Rafiei [1]

In this paper, we propose an efficient dropping criterion for IUL factorization obtained from Backward Factored APproximate INverse (BFAPINV) and ILU factorization obtained from Forward Factored APproximate INverse (FFAPINV) algorithms.
In this paper, notations $X_{:,j}$ and $X_{j,:}$ are used as the j-th column and j-th row of matrix X, respectively.

Suppose that matrix A is nonsymmetric. Also suppose that L is an unit lower triangular matrix and U is an upper triangular matrix. Consider D as a diagonal matrix. In [2], an ILU factorization of matrix A, which is obtained as by-product of FFAPINV [3] process, has been presented. Matrices L, U and A satisfy relation

$$A \approx LU.$$

We term this ILU factorization, ILUFF (ILU factorization obtained from Forward Factored APproximate INverse). The following algorithm computes the ILUFF preconditioner.

Algorithm 1 (**ILUFF factorization**)

$w_1 = e_1^T, \quad z_1 = e_1, \quad d_1 = a_{11}..$
for $j = 2$ to n **do**
  $w_j = e_j^T, \quad z_j = e_j.$
  for $i = 1$ to $j-1$ **do**
    $L_{ji} = \frac{A_{j,:}z_i}{d_i} \quad U_{ij} = \frac{w_i A_{:,j}}{d_i}$
    apply a dropping rule to L_{ji} *and to* U_{ij}
    $z_j = z_j - (\frac{w_i A_{:,j}}{d_i})z_i, \quad w_j = w_j - (\frac{A_{j,:}z_i}{d_i})w_i$
    for all $l \leq i$ *apply a dropping rule to* z_{lj} *and to* w_{jl}
  end for
  $d_j = w_j A_{:,j}$ {not positive definite}
  $d_j = w_j A w_j^T$ {positive definite}
end for
Return $L = (L_{ij})$ and $U = (d_i U_{ij})$.

Two below dropping strategies can be used to drop entries of z_j and w_j vectors in Algorithm 1.

- **First strategy:** At each step j of Algorithm 1, entries z_{lj} and w_{jl} for $l \leq i$ are dropped when
$$|z_{lj}| \leq \varepsilon_Z, \qquad |w_{jl}| \leq \varepsilon_W. \tag{1}$$
- **Second strategy:** At each step j of Algorithm 1, the whole vectors
$$z_j = e_j - \sum_{i=1}^{j-1}(\frac{w_i A_{:,j}}{d_i})z_i, \qquad w_j = e_j^T - \sum_{i=1}^{j-1}(\frac{A_{j,:}z_i}{d_i})w_i,$$
will be computed and then, entries z_{lj} and w_{jl}, for $l \leq j$, that satisfy the former dropping criterion (1) will be dropped.

[1]Department of Applied mathematics, Sabzevar Tarbiat Moallem University, Sabzevar, Iran. Email:rafiei@sttu.ac.ir

The following proposition presents an efficient way of dropping to compute the ILUFF preconditioner [1].

Proposition 1. *Let $\varepsilon_{U,Z}$ and $\varepsilon_{L,W}$ be the same drop tolerance parameter for U, Z and L, W matrices, respectively. Suppose that at each step j of Algorithm 1, entries L_{jk} and U_{kj}, for $k < j$, be dropped when the criterions*

$$|L_{jk}|\|W_{k,:}\|_1 \leq \varepsilon_{L,W}, \qquad |U_{kj}|\|Z_{:,k}\|_\infty \leq \varepsilon_{U,Z}, \tag{2}$$

satisfied. Thus, for $1 \leq i \leq j \leq n$

- *if the first dropping strategy be applied to drop entries of Z and W matrices, then*

$$|(I - ZU)_{ij}| \leq 2(j-i)\varepsilon_{U,Z}, \qquad |(I - LW)_{ji}| \leq 2(j-i)\varepsilon_{L,W}.$$

- *if the second dropping strategy be applied to drop entries of Z and W matrices, then*

$$|(I - ZU)_{ij}| \leq (j-i+1)\varepsilon_{U,Z}, \qquad |(I - LW)_{ji}| \leq (j-i+1)\varepsilon_{L,W}.$$

Suppose that U is an unit upper triangular matrix. Consider L as a lower triangular matrix and D as a diagonal one. The following algorithm computes the incomplete IUL factorization

$$A \approx UL,$$

which is obtained as by-product of BFAPINV [4] process. We term this IUL factorization as IULBF (IUL factorization obtained from Backward Factored APproximate INverse).

Algorithm 2 **(IULBF factorization)**

1. $w_n = e_n^T, \quad z_n = e_n, \quad d_n = a_{nn}.$
2. **for** $j = n-1$ to 1 **do**
3. $\quad w_j = e_j^T, \quad z_j = e_j.$
4. $\quad$ **for** $i = j+1$ to n **do**
5. $\quad\quad U_{ji} = \frac{A_{j,:}z_i}{d_i} \quad L_{ij} = \frac{w_i A_{:,j}}{d_i}$
6. $\quad\quad$ *apply a dropping rule to* U_{ji} *and to* L_{ij}
7. $\quad\quad z_j = z_j - (\frac{w_i A_{:,j}}{d_i})z_i, \quad w_j = w_j - (\frac{A_{j,:}z_i}{d_i})w_i$
8. $\quad\quad$ *for all* $l \geq i$ *apply a dropping rule to* z_{lj} *and to* w_{jl}
9. $\quad$ **end for**
10. $\quad d_j = w_j A_{:,j}$ {not positive definite}
11. $\quad d_j = w_j A w_j^T$ {positive definite}
12. **end for**
13. Return $L = (d_i L_{ij})$ and $U = (U_{ij})$.

Two below dropping strategies can be used to drop entries of vectors z_j and w_j in Algorithm 2.

- **First strategy:** At each step j of Algorithm 2, entries z_{lj} and w_{jl}, for $l \geq i$, are dropped when criterion (1) is satisfied.
- **Second strategy:** At each step j of Algorithm 2, the whole vectors

$$z_j = e_j - \sum_{i=j+1}^{n} (\frac{w_i A_{:,j}}{d_i})z_i, \qquad w_j = e_j^T - \sum_{i=j+1}^{n} (\frac{A_{j,:}z_i}{d_i})w_i,$$

will be computed and then, entries z_{lj} and w_{jl}, for $l \geq j$, that satisfy the dropping criterion (1) will be dropped.

The following proposition presents an efficient way of dropping to compute the IULBF preconditioner [1].

Proposition 2. *Let $\varepsilon_{U,W}$ and $\varepsilon_{L,Z}$ be the same drop tolerance parameter for U, W and L, Z matrices, respectively. Suppose that at each step j of Algorithm 2, entries L_{kj} and U_{jk}, for $k > j$, are dropped when the criterions*

$$|L_{kj}|\|Z_{:,k}\|_\infty \leq \varepsilon_{L,Z}, \qquad |U_{jk}|\|W_{k,:}\|_1 \leq \varepsilon_{U,W}. \tag{3}$$

satisfied. Thus, for $1 \leq j \leq i \leq n$

- *if the first dropping strategy be applied to drop entries of Z and W matrices, then*

$$|(I - UW)_{ji}| \leq 2(i-j)\varepsilon_{U,W}, \qquad |(I - ZL)_{ij}| \leq 2(i-j)\varepsilon_{L,Z}.$$

- *if the second dropping strategy be applied to drop entries of Z and W matrices, then*

$$|(I - UW)_{ji}| \leq (i-j+1)\varepsilon_{U,W}, \qquad |(I - ZL)_{ij}| \leq (i-j+1)\varepsilon_{L,Z}.$$

References

[1] A. Rafiei, ILU and IUL factorizations obtained from Forward and Backward Factored APproximate INVerse algorithms *Submitted*, 2010.

[2] D.K. Salkuyeh, A. Rafiei and H. Roohani, *ILU Preconditioning Based on the FAPINV Algorithm*, http://www.arxiv.org, arXiv:1010.2812.

[3] J. Zhang, *A procedure for computing factored approximate inverse, M.Sc. dissertation, Department of Computer Science, University of Kentucky*, 1999.

[4] J. Zhang, *A sparse approximate inverse technique for parallel preconditioning of general sparse matrices*, Appl. Math. Comput., **130** (2002) 63-85.