TACC Technical Report TR-10-04

Incomplete Factorizations in the FLAME Framework

Victor Eijkhout*, Paolo Bientinesi† Robert van de Geijn‡

September 3, 2010

* Texas Advanced Computing Center, The University of Texas at Austin
† RWTH Aachen,Germany
‡ Computer Science Department, The University of Texas at Austin

Abstract

In this report, we apply the FLAME methodology *[The Science of Programming Matrix Computations, van de Geijn and Quintana-Ortí, 2008]* to the derivation of incomplete LU factorizations. We derive the basic ILU and MILU algorithms, and we reproduce the well-posedness theorem for ILU.

1 Introduction

In sparse matrices, it has long been observed and proved [2, 5] that an LU factorization takes substantially more storage space than the original matrix. The reason for this is the 'fill-in' that happens during the factorization: locations (i,j) where A is zero ($a_{ij} = 0$), may become nonzero during the factorization ($(L+U)_{ij} \neq 0$). For this reason, linear systems are often solved with iterative methods that use an incomplete factorization, that is, a factorization where some of the fill-in is discarded, as preconditioner. One way of interpreting this process is to consider the iterative method as a refinement process for the inexact solution that results from solving a system with the incomplete factorization of the coefficient matrix.

In this note, we will give a systematic treatment of the derivation of incomplete factorizations, using the FLAME framework [9].

1.1 The origin of incomplete factorizations

A Gaussian elimination (without pivoting) has as its basic statement

$$a_{ij} \leftarrow a_{ij} - a_{ik}a_{kk}^{-1}a_{kj}. \tag{1}$$

In the case of a sparse matrix, this introduce a nonzero value in an (i,j) location that is zero in the original matrix A. Incomplete factorizations are strategies of computing a factorization while maintaining some amount of sparsity. For instance, the update statement (1) can be amended to

$$a_{ij} \leftarrow a_{ij} - a_{ik}a_{kk}^{-1}a_{kj} \qquad \text{if } a_{ij} \neq 0. \tag{2}$$

More sophisticated strategies drop such 'fill-in' values based on a level structure, or on a drop tolerance test.

Incomplete factorizations are no longer an exact factorization of A: instead they can be considered to compute $LU = A + E$ where E is an error or residual matrix. There can be conditions on the error matrix; for instance, in 'modified incomplete factorizations' we want

$$Av = LUv \tag{3}$$

to be satisfied for some vector v. (There are various justifications for this condition, for instance expressing a conservation of mass.) Equivalently, this means $Ev = 0$.

2 ILU and MILU

First let us describe the full LU factorization. The Partitioned Matrix Expression (PME) is

$$\begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix} \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix} = \begin{pmatrix} \hat{A}_{TL} & \hat{A}_{TR} \\ \hat{A}_{BL} & \hat{A}_{BR} \end{pmatrix}$$

where $\hat{A}$ denotes the original matrix. Using the identifier A for the current contents of the array that initially stores the matrix, we can also write the PME as

$$\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} = \begin{pmatrix} \{L,U\}_{TL} & U_{TR} \\ L_{BL} & \{L,U\}_{BR} \end{pmatrix} \wedge \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & L_{BR}U_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix}$$

where $\{L,U\}_{TL}$ denotes the matrix that stores L and U.

We choose the invariant that assumes the first block row and column have been computed, and in the right under block only the update has been constructed:

$$\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} = \begin{pmatrix} \{L,U\}_{TL} & U_{TR} \\ L_{BL} & S \end{pmatrix} \wedge \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & S = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix}. \tag{4}$$

We make the usual transition from 2×2 to 3×3 form. The 'before' equation is, leaving out the trivial first clause,

$$\left(\begin{array}{c|c|c} L_{00}U_{00} = \hat{A}_{00} & u_{01} = L_{00}^{-1}\hat{a}_{01} & U_{02} = L_{00}^{-1}\hat{A}_{02} \\ \hline \ell_{10}^t = \hat{a}_{10}^t U_{00}^{-1} & \sigma_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} & s_{12}^t = \hat{a}_{12}^t - \ell_{10}^t U_{02} \\ \hline L_{20} = \hat{A}_{20}U_{00}^{-1} & s_{21} = \hat{a}_{21} - L_{20}u_{01} & S_{22} = \hat{A}_{22} - L_{20}U_{02} \end{array}\right)$$

and the 'after' equation is

$$\left(\begin{array}{c|c|c} L_{00}U_{00} = \hat{A}_{00} & u_{01} = L_{00}^{-1}\hat{a}_{01} & U_{02} = L_{00}^{-1}\hat{A}_{02} \\ \hline \ell_{10}^t = \hat{a}_{10}^t U_{00}^{-1} & \ell_{11}u_{11} + \ell_{10}^t u_{01} = \hat{a}_{11} & \ell_{11}u_{12}^t + \ell_{10}U_{02} = \hat{a}_{12}^t \\ \hline L_{20} = \hat{A}_{20}U_{00}^{-1} & \ell_{21}u_{11} + L_{20}u_{01} = \hat{a}_{21} & S_{22} = \hat{A}_{22} - L_{20}U_{02} - \ell_{21}u_{12} \end{array}\right)$$

We now reason (using the red color to indicate variables to be computed, and green for variables that have been computed):

- Using the normalization $\ell_{11} = 1$, the $(1,1)$ location gives

$$\begin{cases} \sigma_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} \\ \ell_{11} = 1 \\ \ell_{11}u_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} \end{cases} \Rightarrow u_{11} := \ell_{11}^{-1}\sigma_{11} = \sigma_{11}$$

- Likewise,

$$\begin{cases} \ell_{11} u^t_{12} = \hat{a}^t_{12} - \ell^t_{10} U_{02} \\ \hat{a}^t_{12} - \ell^t_{10} U_{02} = s^t_{12} \end{cases} \Rightarrow u^t_{12} := s^t_{12}$$

and

$$\begin{cases} \ell_{21} u_{11} = \hat{a}_{21} - L_{20} u_{01} \\ \hat{a}_{21} - L_{20} u_{01} = s_{21} \end{cases} \Rightarrow \ell_{21} := s_{21} u_{11}^{-1}$$

- From

$$\begin{aligned} \text{Before:} \quad & S_{22} = \hat{A}_{22} - L_{20} U_{02} \\ \text{After:} \quad & S_{22} = \hat{A}_{22} - L_{20} U_{02} - \ell_{21} u^T_{12} \end{aligned}$$

we conclude that S_{22} needs to be updated as $S_{22} := S_{22} - \ell_{21} u^T_{12}$.

which is the standard right looking LU factorization.

2.1 Introducing incompleteness

We now present the incomplete factorization, where, in making the $(2,2)$ block update, we omit certain fill elements. This makes $LU \approx A$, so we introduce an error matrix $E = A - LU$, making the formal definition $LU + E = A$, or in partitioned form:

$$\begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix} \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix} + \begin{pmatrix} E_{TL} & E_{TR} \\ E_{BL} & E_{BR} \end{pmatrix} = \begin{pmatrix} \hat{A}_{TL} & \hat{A}_{TR} \\ \hat{A}_{BL} & \hat{A}_{BR} \end{pmatrix}. \tag{5}$$

In this equation, L, U, E are all outputs and they are no longer uniquely determined. We will see in the following how they are determined by decisions on what fill-in to allow.

The updated PME that includes E is:

$$\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} = \begin{pmatrix} \{L,U\}_{TL} & U_{TR} \\ L_{BL} & \{L,U\}_{BR} \end{pmatrix} \wedge \begin{pmatrix} L_{TL} U_{TL} = \hat{A}_{TL} - E_{TL} & U_{TR} = L_{TL}^{-1} (\hat{A}_{TR} - E_{TR}) \\ L_{BL} = (\hat{A}_{BL} - E_{BL}) U_{TL}^{-1} & L_{BR} U_{BR} = \hat{A}_{BR} - L_{BL} U_{TR} - E_{BR} \end{pmatrix}$$

Note that E is not stored: it is only implicitly defined as to consist of the discarded terms. It is worth remarking that this does not imply that E has a sparsity pattern that is the difference between A and LU or $L+U$. For instance, for the 5-point difference stencil for the 2D Poisson equation A has $5N$ nonzeros, a full factorization would have roughly $N^{3/2}$, but E has only $2N$.

The invariant is analogous to the full factorization case:

$$\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} = \begin{pmatrix} \{L,U\}_{TL} & U_{TR} \\ L_{BL} & S \end{pmatrix} \wedge \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} - E_{TL} & U_{TR} = L_{TL}^{-1}(\hat{A}_{TR} - E_{TR}) \\ L_{BL} = (\hat{A}_{BL} - E_{BL})U_{TL}^{-1} & S = \hat{A}_{BR} - L_{BL}U_{TR} - E_{BR} \end{pmatrix}$$

The 'before' equation is, again leaving out the trivial first clause,

$$\left(\begin{array}{c|c|c} L_{00}U_{00} = \hat{A}_{00} - E_{00} & u_{01} = L_{00}^{-1}(\hat{a}_{01} - e_{01}) & U_{02} = L_{00}^{-1}(\hat{A}_{02} - E_{02}) \\ \hline \ell_{10}^t = (\hat{a}_{10}^t - e_{10}^t)U_{00}^{-1} & \sigma_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} - \varepsilon_{11} & s_{12}^t = \hat{a}_{12}^t - \ell_{10}^t U_{02} - e_{02}^t \\ \hline L_{20} = (\hat{A}_{20} - E_{20})U_{00}^{-1} & s_{21} = \hat{a}_{21} - L_{20}u_{01} - e_{21} & S_{22} = \hat{A}_{22} - L_{20}U_{02} - E_{22} \end{array}\right)$$

and the 'after' equation is

$$\left(\begin{array}{c|c|c} L_{00}U_{00} = \hat{A}_{00} - E_{00} & u_{01} = L_{00}^{-1}(\hat{a}_{01} - e_{01}) & U_{02} = L_{00}^{-1}(\hat{A}_{02} - E_{02}) \\ \hline \ell_{10}^t = (\hat{a}_{10}^t - e_{10}^t)U_{00}^{-1} & \ell_{11}u_{11} + \ell_{10}^t u_{01} = \hat{a}_{11} - \varepsilon_{11} & \ell_{11}u_{12}^t + \ell_{10}U_{02} = \hat{a}_{12}^t - e_{12}^t \\ \hline L_{20} = (\hat{A}_{20} - E_{20})U_{00}^{-1} & \ell_{21}u_{11} + L_{20}u_{01} = \hat{a}_{21} - e_{21} & S_{22} = \hat{A}_{22} - L_{20}U_{02} - \ell_{21}u_{12} - \tilde{E}_{22} \end{array}\right)$$

We now make the assumption that $\varepsilon_{11}, e_{12}^t, e_{21}$ do not change between the before and after state. This allows us to reason:

- Using the normalization $\ell_{11} = 1$, the $(1,1)$ location gives

$$\begin{cases} \sigma_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} - \varepsilon_{11} \\ \ell_{11}u_{11} = \hat{a}_{11} - \ell_{10}^t u_{01} - \varepsilon_{11} \end{cases} \Rightarrow u_{11} =: \ell_{11}^{-1}\sigma_{11} = \sigma_{11} = \alpha_{11}$$

- Likewise,

$$\begin{cases} \ell_{11}u_{12}^t = \hat{a}_{12}^t - \ell_{10}^t U_{02} - e_{12}^t \\ s_{12} = a_{12} = \hat{a}_{12}^t - \ell_{10}^t U_{02} - e_{12}^t \end{cases} \Rightarrow u_{12}^t := a_{12}^t$$

  and

$$\begin{cases} \ell_{21}u_{11} = \hat{a}_{21} - L_{20}u_{01} - e_{21} \\ s_{21} = a_{21} = \hat{a}_{21} - L_{20}u_{01} - e_{21} \end{cases} \Rightarrow \ell_{21} := a_{21}u_{11}{}^{-1}$$

- Finally, we consider possible updates to S_{22} and E_{22}. We have:

$$\begin{aligned} \text{Before}: &\quad S_{22} + E_{22} = \hat{A}_{22} - L_{20}U_{02} \\ \text{After}: &\quad S_{22} + E_{22} = \hat{A}_{22} - L_{20}U_{02} - l_{21}u_{12}^T \end{aligned}$$

  Here we have the freedom to split and assign $-l_{21}u_{12}^T$ to S_{22} and E_{22} anyway we want. The traditional $ILU(0)$ choice is to update
  $S_{22} := S_{22} - l_{21}u_{12}^T$ for all the indices for which $\tilde{A}_{22}$ is not zero, and
  $E_{22} := E_{22} - l_{21}u_{12}^T$ for all the indices for which $\tilde{A}_{22}$ is zero.

2.2 Modified incomplete factorizations

In the case of MILU, a vector v is chosen, typically with all positive elements, and as a constraint on the factorization $LUv = Av$ is required. Equivalently, we can impose $Ev = 0$. This requirement can be justified in a number of ways, for instance as 'mass conservation', but mathematically it can lead to an order improvement in the convergence of iterative methods [1, 3, 8].

2.2.1 Equation & PME

The governing equation is

$$\left\{ \begin{array}{l} LU + E = \hat{A} \\ LUv = \hat{A}v \end{array} \right. , \text{ or equivalently } \left\{ \begin{array}{l} LU + E = \hat{A} \\ Ev = 0 \end{array} \right. ,$$

where $\hat{A}$ and v are inputs and L, U and E are outputs. We use the second formulation. The corresponding PME is

$$\begin{array}{l} \left(\begin{array}{cc} L_{TL}U_{TL} + E_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}(\hat{A}_{TR} - E_{TR}) \\ L_{BL} = (\hat{A}_{BL} - E_{BL})U_{TL}^{-1} & L_{BR}U_{BR} + E_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{array} \right) \wedge \\ \left(\begin{array}{c} E_{TL}v_T + E_{TR}v_B = 0 \\ E_{BL}v_T + E_{BR}v_B = 0 \end{array} \right). \end{array} \tag{6}$$

2.2.2 Loop Invariant

We consider the same loop invariant as for ILU, with the extra constraint:

$$\begin{array}{l} \left(\begin{array}{cc} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{cc} \{L, U\}_{TL} & U_{TR} \\ L_{BL} & \tilde{A} \end{array} \right) \wedge \\ \left(\begin{array}{cc} L_{TL}U_{TL} = \hat{A}_{TL} - E_{TL} & U_{TR} = L_{TL}^{-1}(\hat{A}_{TR} - E_{TR}) \\ L_{BL} = (\hat{A}_{BL} - E_{BL})U_{TL}^{-1} & \tilde{A} = \hat{A}_{BR} - E_{BR} - L_{BL}U_{TR} \end{array} \right) \\ \left(\begin{array}{c} E_{TL}v_T + E_{TR}v_B = 0 \\ E_{BL}v_T + E_{BR}v_B = 0 \end{array} \right). \end{array} \tag{7}$$

We are imposing the matrix E to satisfy $Ev = 0$ at each iteration of the loop. At this stage we only need to verify that the loop invariant is *true* before the loop commences: by initializing $E = 0$, the loop invariant is satisfied ($E_{BR} \equiv E$).

2.2.3 Loop Invariant before and after

With respect to the derivation for ILU, the only difference in these predicates is the presence of the constraint. The partitioning and repartitioning statements are respectively

$$\left(\begin{array}{c|c} E_{TL} & E_{TR} \\ \hline E_{BL} & E_{BR} \end{array}\right) \to \left(\begin{array}{c|c|c} E_{00} & e_{01} & E_{02} \\ \hline e^t_{10} & \varepsilon_{11} & e^t_{12} \\ \hline E_{20} & e_{21} & E_{22} \end{array}\right), \left(\begin{array}{c} v_{TL} \\ \hline v_{BL} \end{array}\right) \to \left(\begin{array}{c} v_{00} \\ \hline \phi_{10} \\ \hline v_{20} \end{array}\right)$$

and

$$E_{TL} \to \begin{pmatrix} E_{00} & e_{01} \\ e^T_{10} & \varepsilon_{11} \end{pmatrix}, \ldots E_{BR} \to E_{22},\ v_T \to \begin{pmatrix} v_0 \\ \upsilon_1 \end{pmatrix}, v_B \to v_2.$$

In both the Before and After predicates, the constraint becomes

$$\begin{pmatrix} E_{00}v_0 + e_{01}\upsilon_1 + E_{02}v_2 = 0 \\ e^T_{10}v_0 + \varepsilon_{11}\upsilon_1 + e^T_{12}v_2 = 0 \\ E_{20}v_0 + e_{21}\upsilon_1 + E_{22}v_2 = 0 \end{pmatrix}. \tag{8}$$

This means that (8) is satisfied at the beginning of each iteration, and we have to identify computational updates such the equalities (8) are still *true* after their execution.

2.2.4 Updates

No differences for the updates to μ_{11}, u^T_{12} and l_{21}. Regarding the constraint, we need to show that (8) is still *true* after the execution of the updates. Since we impose that the algorithm does not update any part of E except for E_{22}, ($E_{00}, e_{01}, E_{02}, e^T_{10}, \varepsilon_{11}, e^T_{12}, E_{20}$ and e_{21} do not change from one iteration to the next), the top two equalities of (8) are automatically satisfied. It remains to show that the equality $E_{22}v_0 + e_{21}\upsilon_1 + E_{22}v_2 = 0$, which is satisfied before the updates, remains *true* after the computation. Notice that this is NOT trivial, as E_{22} changes at each iteration.

For the sake of clarity, this could be written as

$$\begin{aligned} \text{Before:} \quad & E_{20}v_0 + e_{21}\upsilon_1 + E^{\text{bef}}_{22}v_2 = 0 \\ \text{After:} \quad & E_{20}v_0 + e_{21}\upsilon_1 + E^{\text{aft}}_{22}v_2 = 0 \end{aligned}$$

Since E_{20}, e_{21} are not updated, the constraint simplifies to $E^{\text{aft}}_{22}v_2 = E^{\text{bef}}_{22}v_2$. Essentially, this means that if E_{22} is updated as $E_{22} := E_{22} + X$, then X must satisfy $Xv_2 = 0$.

We amend this as follows.

1. Construct as before $E_{22}^{(0)} := E_{22} - l_{21}u_{12}^T$ for all the indices for which $\tilde{A}_{22}$ is zero, and $S_{22}^{(0)} := S_{22} - l_{21}u_{12}^T$ for all the indices for which $\tilde{A}_{22}$ is not zero.
2. Compute the row sums of this error matrix:

$$d_2 := E_{22}^{(0)} e_2$$

where e_2 is the all-ones vector of the appropriate size.
3. Let $D_2 := \mathrm{diag}(d_2)$.
4. Now set

$$E_{22} := E_{22}^{(0)} - D_2, \quad S_{22} := S_{22}^{(0)} + D_2.$$

2.3 Fill levels

The simplest strategy for incomplete factorizations, ILU(0), is characterized by eliminating all fill-in. (An even simpler variant, sometimes denoted ILU-D, additionally disallows modification of off-diagonal elements. The two are equivalent under certain conditions on the matrix graph.) This is a special case of the class of strategies where fill-in is dropped based on *levels*. (The alternative is dropping based on size.)

Let us assign a level zero to the original elements, and compute fill levels as follows. In the statement

$$a'_{ij} = a_{ij} - a_{ij}a_{kk}^{-1}a_{kj}$$

we assign

$$\begin{cases} f(a'_{ij}) = f(a_{ij}) & \text{if } a_{ij} \neq 0 \\ f(a'_{ij}) = 1 + \max\{f(a_{ik}), f(a_{kj})\} & \text{otherwise.} \end{cases}$$

We model this by adding a PME for integer matrices that record the fill levels:

$$\begin{pmatrix} NL_{TL} & 0 \\ NL_{BL} & NL_{BR} \end{pmatrix} \begin{pmatrix} NU_{TL} & 0 \\ NU_{BL} & NU_{BR} \end{pmatrix} = \begin{pmatrix} NA_{TL} & NA_{TR} \\ NA_{BL} & NA_{BR} \end{pmatrix} - \begin{pmatrix} NE_{TL} & NE_{TR} \\ NE_{BL} & NE_{BR} \end{pmatrix} \tag{9}$$

where

$$NA_{ij} = 1 + \max_k \{NL_{ik}, NU_{kj}\}$$

if $a_{ij} \neq 0$. We can now control fill-in by adding an invariant condition that all $NA_{ij} \leq k$ for some fill level $k \geq 0$.

3 Proof techniques

We will now extend Flame worksheets to M-matrix proofs. In particular, we will prove

Theorem 1 *If A is an M-matrix, then all pivots during an LU/ILU/MILU will be positive.*

Positiveness of the pivots is trivial for the exact LU factorization, but it is long known that this statement does not hold for incomplete factorizations of general positive definite matrices [4, 6]. In this section we reproduce the proof that ILU and MILU *do* give positive pivots for the special case of M-matrices [7]. We start with a proof for LU, to introduce the proof technique.

We recall the definition of an M-matrix:

Definition 1 **Z-matrix:** *A is a Z-matrix, written $Z(A)$ iff*

$a_{ij} \leq 0$ *for* $i \neq j$.

Definition 2 **M-matrix:** *A is an M-matrix iff*

- $Z(A)$, *and*
- $\exists u > 0 : v := Au \wedge v > 0$, *where are comparisons are in a pointwise sense.*

We rely on the following statement:

Theorem 2 *If A is a Z-matrix, then A is an M-matrix iff A is positive definite. In particular, if A is an M-matrix, its leading pivot will be positive.*

We prove the above theorem by showing inductively that every A_{BR} block during the factorization is a Z-matrix, and a vector $u_B > 0$ can be found such that $A_{BR}u_B > 0$. This then implies that the leading pivots are positive.

We will go through the steps needed to fill in the proof worksheet of figure 1.

This theorem is a little departure with respect to the worksheet structure, as the statement (Step 1a, Precondition) requires parts of the matrix that are normally exposed only at a later stage (Step 3, Partitioning).

Algorithm side	Proof side	Step
$[D,E,F,\ldots] = \mathrm{op}(A,B,C,D,\ldots)$	Theorem name	
$\{\ P_{\rm pre}\ \}$	{ Theorem statement }	1a
Partition **where** …	Initialization	3
$\{\ P_{\rm inv}\ \}$	{ Base Case }	2
While G **do**		4
Repartition **where** …	…	5a
$\{\ P_{\rm before}\ \}$	$\{InductiveHypothesis\}$	6
S_U	Proof	8
$\{\ P_{\rm after}\ \}$	$\{InductiveStep\}$	7
Continue with		5b
endwhile		
$\{\ P_{\rm post}\ \}$		1b

Figure 1: An empty proof worksheet

0. Description We will simultaneously derive the algorithm for $LU = A$ and the theorem that 'M-matrix has positive pivots'.

1. Precondition Our algorithm precondition is $A = \hat{A} \wedge M\text{-matrix}(\hat{A})$, and the proof precondition is **Shouldn't this be null? This sounds more like the postcondition. VLE**

$$M\text{-matrix}(\hat{A}) \Rightarrow \\ \text{diag}(U) > 0$$

3. Partitioning We partition matrices as $A \rightarrow \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix}, L \rightarrow \begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}, U \rightarrow \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix}$, and the auxiliary vectors as $u \rightarrow \begin{pmatrix} u_T \\ u_B \end{pmatrix} v \rightarrow \begin{pmatrix} v_T \\ v_B \end{pmatrix}$.

2. Invariant Our algorithm invariant corresponds to the right-looking factorization where S_{BR} has been formed. We abbreviate the full form (4) as:

$$\begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & A_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix}$$

The proof invariant states in the base case

$$A_{BR} \equiv \hat{A} \text{ is an M-matrix, that is,} \\ \begin{cases} Z(A_{BR}) \\ \exists u_B > 0 \colon v_B := A_{BR}u_B \wedge v_B > 0 \end{cases}$$

4. Loop guard The loop guard is, as usual, $A_{TL} <> A$.

5. Repartition The usual: $\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10} & \alpha_{11} & a_{12} \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), U \rightarrow \ldots$ and $\left(\begin{array}{c} u_T \\ \hline u_B \end{array}\right) \rightarrow \left(\begin{array}{c} u_0 \\ \hline \upsilon_1 \\ \hline u_2 \end{array}\right), \begin{pmatrix} v_T \\ v_B \end{pmatrix} \rightarrow \ldots$.

6. Before update The algorithm state is $\left(\begin{array}{c|c|c} \star & \star & \star \\ \hline \star & \alpha_{11}=\sigma_{11} & a_{12}^t = s_{12}^t \\ \hline \star & a_{21}=s_{21} & A_{22}=S_{22} \end{array}\right)$, the proof induction hypothesis is

$$\begin{cases} Z\begin{pmatrix} \alpha_{11} & a_{12}^T \\ a_{21} & A_{22} \end{pmatrix} \wedge \\ \exists \begin{pmatrix} \mu_1 \\ u_2 \end{pmatrix} > 0\colon \begin{cases} \nu_1 := \alpha_{11}\mu_1 + a_{12}^T u_2 > 0 \\ v_2 := a_{21}\mu_1 + A_{22}u_2 > 0 \end{cases} \end{cases}.$$

7. After update After the update, the algorithm state is $\left(\begin{array}{c|c|c} \star & \star & \star \\ \hline \star & \upsilon_{11}=\alpha_{11} & u_{12}^t = a_{12}^t \\ \hline \star & \ell_{21} = a_{21}\alpha_{11}^{-1} & A_{22} := S_{22} - \ell_{21}u_{12}^t \end{array}\right)$, and the proof state is

$$\begin{cases} Z(S_{22}) \\ \exists u_2 > 0\colon\ v_2 := S_{22}u_2 \wedge v_2 > 0 \end{cases}.$$

8. Update The algorithm update is as derived above $\begin{bmatrix} \upsilon_{11} := \alpha_{11} \\ u_{12}^T := a_{12}^T \\ l_{21} := a_{21}/\mu_{11} \\ A_{22} := S_{22} - l_{21}u_{12}^T \end{bmatrix}$; with this update we can prove the induction step that A_{22} (which is the A_{BR} block after the unpartition) is again an M-matrix:

$$\begin{cases} \textbf{Proof} \\ Z(A_{22}) \text{ since } Z(S_{22}) \text{ and } \ell_{21}u_{12}^t \geq 0; \\ \text{Use } u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2, \\ \text{substitute in } S_{22}u_2 = v_2 - s_{21}u_1 \\ \text{to find } A_{22}u_2 = (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2 \end{cases}.$$

The full worksheet is in Figure 2.

3.1 Incomplete LU

Incomplete LU differs only minimally from exact LU. In the update, we replace the update $A_{22} := S_{22} - l_{21}u_{12}^T$ by $A_{22} := S_{22} - F(l_{21}u_{12}^T)$ where $F(\cdot)$ is a sparsification operator.

In the proof this means that $S_{22} \geq A_{22} = S_{22} - F(l_{21}u_{12}^T) \geq S_{22} - l_{21}u_{12}^T$ (with a pointwise comparison of the matrices), so $A_{22}u_2 \geq (S_{22} - l_{21}u_{12}^T)u_2 \geq v_2$.

Algorithm side	Proof side	Step
$LU = A$	M-matrix has positive pivots	
$\{ A = \hat{A} \wedge M\text{-matrix}(\hat{A}) \}$	$\left\{ \begin{array}{l} M\text{-matrix}(\hat{A}) \Rightarrow \\ \text{diag}(U) > 0 \end{array} \right\}$	1a
Partition $A \rightarrow \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix}, L \rightarrow \begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}, U \rightarrow \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix}, \quad u \rightarrow \begin{pmatrix} u_T \\ u_B \end{pmatrix} v \rightarrow \begin{pmatrix} v_T \\ v_B \end{pmatrix}$ **where** A_{TL}, L_{TL}, U_{TL} are 0×0.		3
$\left\{ \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & A_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} A_{BR} \equiv \hat{A} \text{ is an M-matrix, that is,} \\ \left\{ \begin{array}{l} Z(A_{BR}) \\ \exists u_B > 0 \colon v_B := A_{BR}u_B \wedge v_B > 0 \end{array} \right. \end{array} \right\}$	2
While $A_{TL} <> A$ **do**		4
Repartition $\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} \rightarrow \begin{pmatrix} A_{00} & a_{01} & A_{02} \\ a_{10} & \alpha_{11} & a_{12} \\ A_{20} & a_{21} & A_{22} \end{pmatrix}, U \rightarrow \ldots \quad \begin{pmatrix} u_T \\ u_B \end{pmatrix} \rightarrow \begin{pmatrix} u_0 \\ \upsilon_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_T \\ v_B \end{pmatrix} \rightarrow \ldots$ **where** $\ldots$		5a
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \alpha_{11} = \sigma_{11} & a_{12}^t = s_{12}^t \\ \star & a_{21} = s_{21} & A_{22} = S_{22} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z\begin{pmatrix} \alpha_{11} & a_{12}^T \\ a_{21} & A_{22} \end{pmatrix} \wedge \\ \exists \begin{pmatrix} \mu_1 \\ u_2 \end{pmatrix} > 0 \colon \left\{ \begin{array}{l} \nu_1 := \alpha_{11}\mu_1 + a_{12}^T u_2 > 0 \\ v_2 := a_{21}\mu_1 + A_{22}u_2 > 0 \end{array} \right. \end{array} \right\}$	6
$\upsilon_{11} := \alpha_{11}$ $u_{12}^T := a_{12}^T$ $l_{21} := a_{21}/\mu_{11}$ $A_{22} := S_{22} - l_{21}u_{12}^T$	**Proof** $Z(A_{22})$ since $Z(S_{22})$ and $\ell_{21}u_{12}^t \geq 0$; Use $u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2$, substitute in $S_{22}u_2 = v_2 - s_{21}u_1$ to find $A_{22}u_2 = (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2$	8
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \upsilon_{11} = \alpha_{11} & u_{12}^t = a_{12}^t \\ \star & \ell_{21} = a_{21}\alpha_{11}^{-1} & A_{22} := S_{22} - \ell_{21}u_{12}^t \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z(S_{22}) \\ \exists u_2 > 0 \colon v_2 := S_{22}u_2 \wedge v_2 > 0 \end{array} \right\}$	7
Continue with		5b
endwhile		
$\{ LU = \hat{A} \}$		1b

Figure 2: Proof worksheet for M-matrix LU factorization

$$\text{Our update is } \begin{cases} \upsilon_{11} := \alpha_{11} \\ u_{12}^T := a_{12}^T \\ l_{21} := a_{21}/\mu_{11} \\ A_{22} := S_{22} - F(l_{21}u_{12}^T) \end{cases},$$

$$\text{and the proof step is } \begin{cases} \textbf{Proof} \\ Z(A_{22}) \text{ since } Z(S_{22}) \text{ and } \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0; \\ \text{Use } u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2, \\ \text{Combine } \begin{cases} \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0 \\ A_{22} = S_{22} - F(\ell_{21}u_{12}^t) \\ S_{22}u_2 = v_2 - s_{21}u_1 \end{cases} \\ \text{to find } A_{22}u_2 \geq (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2 \end{cases}.$$

The full worksheet is in Figure 3.

3.2 Modified ILU

Since we are adding a nonnegative diagonal to S_{22}, we can no longer state that $A_{22} \geq S_{22} - l_{21}u_{12}^T$, but we still have $Z(A_{22})$. However, $(D_{22} - F(\ell_{21}u_{12}^T))u_2 = -\ell_{21}u_{12}^T)u_2$, so $A_{22}u_2 = (S_{22} + D_{22} - \ell_{21}u_{12}^T)u_2 = (S_{22} - \ell_{21}u_{12}^T)u_2 = v_2$.

$$\text{Our update is } \begin{cases} \upsilon_{11} := \alpha_{11} \\ u_{12}^T := a_{12}^T \\ \ell_{21} := a_{21}/\mu_{11} \\ A_{22} := S_{22} + D_{22} - F(\ell_{21}u_{12}^T) \\ \text{where } D_{22}u_2 = \ell_{21}u_{12}^T u_2 - F(\ell_{21}u_{12}^T)u_2 \end{cases},$$

$$\text{and the proof step is } \begin{cases} \textbf{Proof} \\ Z(A_{22}) \text{ since } Z(S_{22}) \text{ and } \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0; \\ \text{Use } u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2, \\ \text{Combine } \begin{cases} \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0 \\ A_{22} = S_{22} + D_{22} - F(\ell_{21}u_{12}^t), \quad \text{so } Z(A_{22}) \\ S_{22}u_2 = v_2 - s_{21}u_1 \end{cases} \\ \text{to find } A_{22}u_2 = (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2 \end{cases}.$$

The full worksheet is in Figure 4.

Algorithm side	Proof side	Step
$LU = A$	M-matrix has positive pivots	
$\{ A = \hat{A} \wedge M\text{-matrix}(\hat{A}) \}$	$\left\{ \begin{array}{l} M\text{-matrix}(\hat{A}) \Rightarrow \\ \text{diag}(U) > 0 \end{array} \right\}$	1a
Partition $A \rightarrow \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix}, L \rightarrow \begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}, U \rightarrow \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix},$	$u \rightarrow \begin{pmatrix} u_T \\ u_B \end{pmatrix} v \rightarrow \begin{pmatrix} v_T \\ v_B \end{pmatrix}$ **where** A_{TL}, L_{TL}, U_{TL} are 0×0.	3
$\left\{ \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & A_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} A_{BR} \equiv \hat{A} \text{ is an M-matrix, that is,} \\ \left\{ \begin{array}{l} Z(A_{BR}) \\ \exists u_B > 0\colon v_B := A_{BR}u_B \wedge v_B > 0 \end{array} \right. \end{array} \right\}$	2
While $A_{TL} <> A$ **do**		4
Repartition $\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} \rightarrow \begin{pmatrix} A_{00} & a_{01} & A_{02} \\ a_{10} & \alpha_{11} & a_{12} \\ A_{20} & a_{21} & A_{22} \end{pmatrix}, U \rightarrow \ldots$	$\begin{pmatrix} u_T \\ u_B \end{pmatrix} \rightarrow \begin{pmatrix} u_0 \\ \upsilon_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_T \\ v_B \end{pmatrix} \rightarrow \ldots$ **where** $\ldots$	5a
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \alpha_{11} = \sigma_{11} & a_{12}^t = s_{12}^t \\ \star & a_{21} = s_{21} & A_{22} = S_{22} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z\begin{pmatrix} \alpha_{11} & a_{12}^T \\ a_{21} & A_{22} \end{pmatrix} \wedge \\ \exists \begin{pmatrix} \mu_1 \\ u_2 \end{pmatrix} > 0\colon \left\{ \begin{array}{l} \nu_1 := \alpha_{11}\mu_1 + a_{12}^T u_2 > 0 \\ \nu_2 := a_{21}\mu_1 + A_{22}u_2 > 0 \end{array} \right. \end{array} \right\}$	6
$\upsilon_{11} := \alpha_{11}$ $u_{12}^T := a_{12}^T$ $l_{21} := a_{21}/\mu_{11}$ $A_{22} := S_{22} - F(l_{21}u_{12}^T)$	**Proof** $Z(A_{22})$ since $Z(S_{22})$ and $\ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0$; Use $u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2$, Combine $\left\{ \begin{array}{l} \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0 \\ A_{22} = S_{22} - F(\ell_{21}u_{12}^t) \\ S_{22}u_2 = v_2 - s_{21}u_1 \end{array} \right.$ to find $A_{22}u_2 \geq (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2$	8
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \upsilon_{11} = \alpha_{11} & u_{12}^t = a_{12}^t \\ \star & \ell_{21} = a_{21}\alpha_{11}^{-1} & A_{22} := S_{22} - \ell_{21}u_{12}^t \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z(S_{22}) \\ \exists u_2 > 0\colon v_2 := S_{22}u_2 \wedge v_2 > 0 \end{array} \right\}$	7
Continue with		5b
endwhile		
$\{ LU = \hat{A} \}$		1b

Figure 3: Proof worksheet for M-matrix ILU factorization

Algorithm side	Proof side	Step
$LU = A$	M-matrix has positive pivots	
$\left\{\ A = \hat{A} \wedge M\text{-matrix}(\hat{A})\ \right\}$	$\left\{ \begin{array}{l} M\text{-matrix}(\hat{A}) \Rightarrow \\ \text{diag}(U) > 0 \end{array} \right\}$	1a
Partition $A \rightarrow \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix}, L \rightarrow \begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}, U \rightarrow \begin{pmatrix} U_{TL} & U_{TR} \\ 0 & U_{BR} \end{pmatrix}, \quad u \rightarrow \begin{pmatrix} u_T \\ u_B \end{pmatrix} v \rightarrow \begin{pmatrix} v_T \\ v_B \end{pmatrix}$ **where** A_{TL}, L_{TL}, U_{TL} are 0×0.		3
$\left\{ \begin{pmatrix} L_{TL}U_{TL} = \hat{A}_{TL} & U_{TR} = L_{TL}^{-1}\hat{A}_{TR} \\ L_{BL} = \hat{A}_{BL}U_{TL}^{-1} & A_{BR} = \hat{A}_{BR} - L_{BL}U_{TR} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} A_{BR} \equiv \hat{A} \text{ is an M-matrix, that is,} \\ \left\{ \begin{array}{l} Z(A_{BR}) \\ \exists u_B > 0 \colon v_B := A_{BR}u_B \wedge v_B > 0 \end{array} \right. \end{array} \right\}$	2
While $A_{TL} <> A$ **do**		4
Repartition $\begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} \rightarrow \begin{pmatrix} A_{00} & a_{01} & A_{02} \\ a_{10} & \alpha_{11} & a_{12} \\ A_{20} & a_{21} & A_{22} \end{pmatrix}, U \rightarrow \ldots \quad \begin{pmatrix} u_T \\ u_B \end{pmatrix} \rightarrow \begin{pmatrix} u_0 \\ \upsilon_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_T \\ v_B \end{pmatrix} \rightarrow \ldots$ **where** $\ldots$		5a
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \alpha_{11} = \sigma_{11} & a_{12}^t = s_{12}^t \\ \star & a_{21} = s_{21} & A_{22} = S_{22} \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z\begin{pmatrix} \alpha_{11} & a_{12}^T \\ a_{21} & A_{22} \end{pmatrix} \wedge \\ \exists \begin{pmatrix} \mu_1 \\ u_2 \end{pmatrix} > 0 \colon \left\{ \begin{array}{l} \nu_1 := \alpha_{11}\mu_1 + a_{12}^T u_2 > 0 \\ v_2 := a_{21}\mu_1 + A_{22}u_2 > 0 \end{array} \right. \end{array} \right\}$	6
$\upsilon_{11} := \alpha_{11}$ $u_{12}^T := a_{12}^T$ $\ell_{21} := a_{21}/\mu_{11}$ $A_{22} := S_{22} + D_{22} - F(\ell_{21}u_{12}^T)$ where $D_{22}u_2 = \ell_{21}u_{12}^T u_2 - F(\ell_{21}u_{12}^T)u_2$	**Proof** $Z(A_{22})$ since $Z(S_{22})$ and $\ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0$; Use $u_1 = \sigma_{11}^{-1}v_1 - \sigma_{11}^{-1}s_{12}^t u_2$, Combine $\left\{ \begin{array}{l} \ell_{21}u_{12}^t \geq F(\ell_{21}u_{12}^t) \geq 0 \\ A_{22} = S_{22} + D_{22} - F(\ell_{21}u_{12}^t), \quad \text{so } Z(A_{22}) \\ S_{22}u_2 = v_2 - s_{21}u_1 \end{array} \right.$ to find $A_{22}u_2 = (S_{22} - \ell_{21}u_{12}^t)u_2 = v_2 - a_{21}\alpha_{11}^{-1}v_1 \geq v_2$	8
$\left\{ \begin{pmatrix} \star & \star & \star \\ \star & \upsilon_{11} = \alpha_{11} & u_{12}^t = a_{12}^t \\ \star & \ell_{21} = a_{21}\alpha_{11}^{-1} & A_{22} := S_{22} - \ell_{21}u_{12}^t \end{pmatrix} \right\}$	$\left\{ \begin{array}{l} Z(S_{22}) \\ \exists u_2 > 0 \colon v_2 := S_{22}u_2 \wedge v_2 > 0 \end{array} \right\}$	7
Continue with		5b
endwhile		
$\left\{\ LU = \hat{A}\right\}$		1b

Figure 4: Proof worksheet for M-matrix MILU factorization

References

[1] R. Beauwens. Approximate factorizations with S/P consistently ordered *M*-factors. *BIT*, 29:658–681, 1989.

[2] Alan George and Joseph H-W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1981.

[3] Ivar Gustafsson. An incomplete factorization preconditioning method based on modification of element matrices. *BIT*, 36:86–100, 1996.

[4] D.S. Kershaw. The incomplete cholesky-conjugate gradient method for the iterative solution of systems of linear equations. *J. Comp. Phys.*, 26:43–65, 1978.

[5] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.*, 16:346–358, 1979.

[6] T.A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Math. Comp.*, 34:473–497, 1980.

[7] J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric *m*-matrix. *Math Comp*, 31:148–162, 1977.

[8] H. Stone. Iterative solution of implicit approximations of multidimensional partial differetntial equations. *SIAM J. Numer. Anal.*, 5:530–558, 1968.

[9] Robert A. van de Geijn and Enrique S. Quintana-Ortí. *The Science of Programming Matrix Computations.* `www.lulu.com`, 2008.