Multicore Computing
Lecture10 - LU Factorization

SUNG KYUN KWAN
UNIVERSITY

남 범 석

bnam@skku.edu

Drawback of Gaussian Elimination

- Drawback of Gaussian Elimination → lots of computations
  - $n^3/3$ additions and multiplications
  - $n^2/2$ divisions.
  - Equations, especially {b}, have to be changed in each step
  - What if we want to solve the equation for a different b?
  - Can we do better?

LU Decomposition (a.k.a. LU Factorization)

- Let *A* be an *n x n* matrix
- Let *L* be a lower triangular matrix with a unit diagonal
  - $l_{ii} = 1$ for all i and $l_{ij} = 0$ for all $i < j$
- Let *U* be an upper triangular matrix
  - $u_{ij} = 0$ for all $i > j$.
- LU decomposition is the factorization of A into A = LU, with L unit lower triangular and U upper triangular.

Lower and upper triangular matrices

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ & u_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{bmatrix}$$

- Example

$$A = \begin{bmatrix} 1 & 4 & 6 \\ 2 & 10 & 17 \\ 3 & 16 & 31 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 6 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix} = LU.$$

Sequential LU decomposition

- Find x_0, x_1, x_2 such that

$$\begin{aligned} x_0 &+ 4x_1 &+ 6x_2 &= 16 \\ 2x_0 &+ 10x_1 &+ 17x_2 &= 44 \\ 3x_0 &+ 16x_1 &+ 31x_2 &= 78 \end{aligned}$$

- In matrix, solve

$$A\mathbf{x} = \mathbf{b},$$

where

$$A = \begin{bmatrix} 1 & 4 & 6 \\ 2 & 10 & 17 \\ 3 & 16 & 31 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 16 \\ 44 \\ 78 \end{bmatrix}$$

Lower and upper triangular matrices

- Triangular systems are easier to solve
- Let A = LU. Then

$$\mathrm{Ax} = \mathrm{b}$$
$$\mathrm{L(Ux)} = \mathrm{b}$$
$$\mathrm{Ly} = \mathrm{b} \text{ and } \mathrm{Ux} = \mathrm{y}$$

- $\mathrm{Ly} = \mathrm{b}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 16 \\ 44 \\ 78 \end{bmatrix} \xRightarrow{\text{Forward Substitution}} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 16 \\ 12 \\ 6 \end{bmatrix}$$

- $\mathrm{Ux} = \mathrm{y}$

$$\begin{bmatrix} 1 & 4 & 6 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 16 \\ 12 \\ 6 \end{bmatrix} \xRightarrow{\text{Backward Substitution}} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Algorithm for LU decomposition

- Some simple algebra:

$$A = LU \iff a_{ij} = \sum_{r=0}^{n-1} l_{ir} u_{rj} \quad \text{for all } i, j.$$

- Assume i ≤ j. Then:

$$\begin{aligned}
a_{ij} &= \sum_{r=0}^{n-1} l_{ir} u_{rj} = \sum_{r=0}^{i} l_{ir} u_{rj} \quad (\text{because } l_{ir} = 0 \text{ for } r > i) \\
&= \sum_{r=0}^{i-1} l_{ir} u_{rj} + l_{ii} u_{ij} = \sum_{r=0}^{i-1} l_{ir} u_{rj} + u_{ij} \\
&\iff \\
u_{ij} &= a_{ij} - \sum_{r=0}^{i-1} l_{ir} u_{rj}.
\end{aligned}$$

Computing lij and uij

- l_{ij} and u_{ij} in terms of previously computed a_{ij} , l_{ir} and u_{rj} .

- We have obtained

$$u_{ij} = a_{ij} - \sum_{r=0}^{i-1} l_{ir} u_{rj} \quad \text{for } i \leq j.$$

- Similarly,

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{r=0}^{j-1} l_{ir} u_{rj} \right) \quad \text{for } i > j.$$

Modifying the matrix A in stages

- For $0 \le k \le n$, define the intermediate matrix A(k) of stage k:

$$a_{ij}^{(k)} = a_{ij} - \sum_{r=0}^{k-1} l_{ir} u_{rj}.$$

- In this notation,

$$\begin{aligned} u_{ij} &= a_{ij} - \sum_{r=0}^{i-1} l_{ir} u_{rj} \iff u_{ij} = a_{ij}^{(i)} \\ l_{ij} &= \frac{1}{u_{jj}} \left(a_{ij} - \sum_{r=0}^{j-1} l_{ir} u_{rj} \right) \iff l_{ij} = \frac{a_{ij}^{(j)}}{u_{jj}} \end{aligned}$$

- We retrieve values u_{ij} $(i \le j)$ in stage i and l_{ij} $(i > j)$ in stage j.

Sequential Algorithm for LU decomposition

- **input**: $A^{(0)}$: $n \times n$ matrix.
- **output**: L : $n \times n$ lower triangular matrix,
  U : $n \times n$ upper triangular matrix,
  such that $LU = A^{(0)}$.

$$
\begin{aligned}
&\textbf{for } k := 0 \textbf{ to } n-1 \textbf{ do} \\
&\quad \textbf{for } j := k \textbf{ to } n-1 \textbf{ do} \\
&\quad\quad u_{kj} := a_{kj}^{(k)}; \\
&\quad \textbf{for } i := k+1 \textbf{ to } n-1 \textbf{ do} \\
&\quad\quad l_{ik} := a_{ik}^{(k)} / u_{kk}; \\
&\quad \textbf{for } i := k+1 \textbf{ to } n-1 \textbf{ do} \\
&\quad\quad \textbf{for } j := k+1 \textbf{ to } n-1 \textbf{ do} \\
&\quad\quad\quad a_{ij}^{(k+1)} := a_{ij}^{(k)} - l_{ik} u_{kj};
\end{aligned}
$$

Memory-efficient sequential LU decomposition

- input: $A : n \times n$ matrix, $A = A^{(0)}$.
- output: $A : n \times n$ matrix, $A = L - I_n + U$, with
  $L : n \times n$ lower triangular matrix,
  $U : n \times n$ upper triangular matrix,
  $I_n : n \times n$ identity matrix,
  such that $LU = A^{(0)}$.

for $k := 0$ **to** $n-1$ **do**
    for $i := k+1$ **to** $n-1$ **do**
        $a_{ik} := a_{ik}/a_{kk};$
    for $i := k+1$ **to** $n-1$ **do**
        for $j := k+1$ **to** $n-1$ **do**
            $a_{ij} := a_{ij} - a_{ik}a_{kj};$

Transformations of A by LU decomposition

$$A = \begin{bmatrix} 1 & 4 & 6 \\ 2 & 10 & 17 \\ 3 & 16 & 31 \end{bmatrix} \xrightarrow{(0)} \begin{bmatrix} 1 & 4 & 6 \\ 2 & 2 & 5 \\ 3 & 4 & 13 \end{bmatrix} \xrightarrow{(1)} \begin{bmatrix} 1 & 4 & 6 \\ 2 & 2 & 5 \\ 3 & 2 & 3 \end{bmatrix}$$

Hence,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 4 & 6 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix}.$$

Transformations of A by LU decomposition

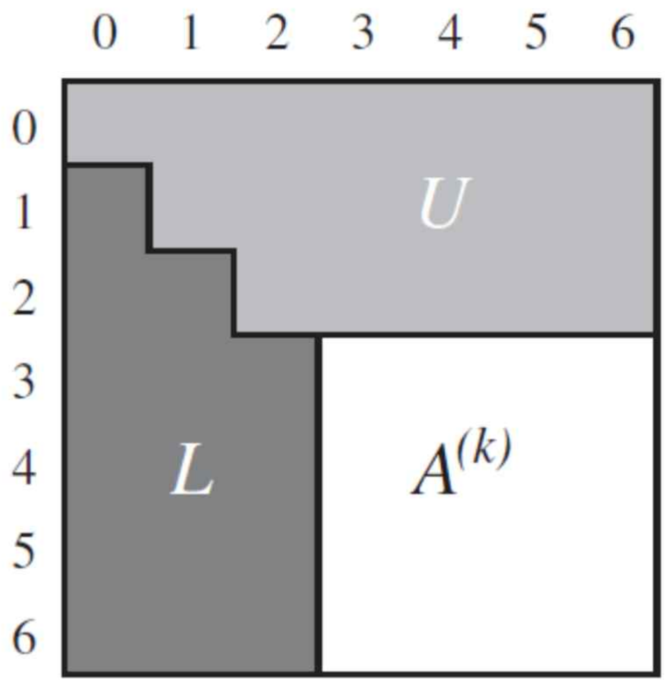
$$A = \begin{bmatrix} 3 & -7 & -2 & 2 \\ -3 & 5 & 1 & 0 \\ 6 & -4 & 0 & -5 \\ -9 & 5 & -5 & 12 \end{bmatrix} \xrightarrow{(0)} \begin{bmatrix} 3 & -7 & -2 & 2 \\ -1 & -2 & -1 & 2 \\ 2 & 10 & 4 & -9 \\ -3 & -16 & -11 & 18 \end{bmatrix} \xrightarrow{(1)} \begin{bmatrix} 3 & -7 & -2 & 2 \\ -1 & -2 & -1 & 2 \\ 2 & -5 & -1 & 1 \\ -3 & 8 & -3 & 2 \end{bmatrix}$$

$$\xrightarrow{(2)} \begin{bmatrix} 3 & -7 & -2 & 2 \\ -1 & -2 & -1 & 2 \\ 2 & -5 & -1 & 1 \\ -3 & 8 & 3 & -1 \end{bmatrix}$$

Hence, $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & -5 & 1 & 0 \\ -3 & 8 & 3 & 1 \end{bmatrix}$ $U = \begin{bmatrix} 3 & -7 & -2 & 2 \\ 0 & -2 & -1 & 2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

Storing L, U, $A^{(k)}$ in the space of A

- At the start of stage k = 3:
  rows 0, 1, 2 of U and columns 0, 1, 2 of L below the diagonal have already been computed.



Row permutations needed

- LU decomposition breaks down immediately in stage 0 for

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

  because we try to divide by 0.

  - A solution is to permute the rows suitably (pivot element=largest).
  - Thus, we compute a permuted LU decomposition,

$$PA = LU.$$

  - P is a permutation matrix, obtained by permuting the rows of I_n.
  - Output of LU decomposition of A: L, U, P.

Row permutations needed

- Find the PA = LU factorization $\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}$
  - The first permutation step is trivial since the pivot element 10 is already the largest.
  - The first elimination step is:
- 1st elimination step:
  - row 2 ← row 2 − (−3/10)(row 1)
  - row 3 ← row3 − (5/10) (row1)

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -3/10 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & -1/10 & 6 \\ 0 & 5/2 & 5 \end{pmatrix}$$

Row permutations needed

- 2^{nd} permutation step:
  - -1/10 is smaller than 5/2. So, swap rows 2 and 3

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -3/10 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & 5/2 & 5 \\ 0 & -1/10 & 6 \end{pmatrix}$$

- 2^{nd} elimination step:
- row 3 $\leftarrow$ row 3 $-$ $(-1/25)$(row 2)

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -3/10 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/25 & 1 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & 5/2 & 5 \\ 0 & 0 & 31/5 \end{pmatrix}$$

Row permutations needed

- The operations can be reorganized as follows:

$$PA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -3/10 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/25 & 1 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & 5/2 & 5 \\ 0 & 0 & 31/5 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ -3/10 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/25 & 1 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & 5/2 & 5 \\ 0 & 0 & 31/5 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ -3/10 & -1/25 & 1 \end{pmatrix} \begin{pmatrix} 10 & -7 & 0 \\ 0 & 5/2 & 5 \\ 0 & 0 & 31/5 \end{pmatrix} = LU$$

Block-Cyclic Decomposition for Parallel Execution

- One approach
  - Assume we have a square number of processors
  - Divide the matrix into blocks – storing one block per processor
  - Need to modify LU decomposition algorithm

P0	P1	P2	P3
P4	P5	P6	P7
P8	P9	P10	P11
P12	P13	P14	P15

Constructing the Block LU Factorization

A00	A01	A02		L00	**0**	**0**		U00	U01	U02
A10	A11	A12	=	L10	**1**	**0**	*	0	?11	?12
A20	A21	A22		L20	**0**	**1**		0	?21	?22

- $A_{00} = L_{00}*U_{00}$ (compute by L_{00}, U_{00} by LU factorization)
- $A_{01} = L_{00}*U_{01} \rightarrow U_{01} = L_{00}^{-1}A_{01}$
- $A_{02} = L_{00}*U_{02} \rightarrow U_{02} = L_{00}^{-1}A_{02}$
- $A_{10} = L_{10}*U_{00} \rightarrow L_{10} = A_{10}U_{00}^{-1}$
- $A_{20} = L_{20}*U_{00} \rightarrow L_{20} = A_{20}U_{00}^{-1}$
- $A_{11} = L_{10}*U_{01} + ?_{11} \rightarrow ?_{11} = A_{11} – L_{10}*U_{01}$

Constructing the Block LU Factorization

$A_{00} = L_{00}*U_{00}$ (compute by L_{00}, U_{00} by LU factorization)

$A_{01} = L_{00}*U_{01} \rightarrow U_{01} = L_{00}^{-1}A_{01}$

$A_{02} = L_{00}*U_{02} \rightarrow U_{02} = L_{00}^{-1}A_{02}$

$A_{10} = L_{10}*U_{00} \rightarrow L_{10} = A_{10}U_{00}^{-1}$

$A_{20} = L_{20}*U_{00} \rightarrow L_{20} = A_{20}U_{00}^{-1}$

$A_{11} = L_{10}*U_{01} + ?_{11} \rightarrow ?_{11} = A_{11} - L_{10}*U_{01}$

$A_{12} = L_{10}*U_{02} + ?_{12} \rightarrow ?_{12} = A_{12} - L_{10}*U_{02}$

$A_{21} = L_{20}*U_{01} + ?_{21} \rightarrow ?_{21} = A_{21} - L_{20}*U_{01}$

$A_{22} = L_{20}*U_{02} + ?_{22} \rightarrow ?_{22} = A_{22} - L_{20}*U_{02}$

In the general case:

$A_{nm} = L_{n0}*U_{0m} + ?_{nm} \rightarrow ?_{nm} = A_{nm} - L_{n0}*U_{0m}$

Summary First Stage

A00	A01	A02
A10	A11	A12
A20	A21	A22

=

L00	**0**	**0**
L10	**1**	**0**
L20	**0**	**1**

*

U00	U01	U02
0	?11	?12
0	?21	?22

- First step: LU factorize uppermost block diagonal
- Second step: a) compute $U_{0n} = L_{00}^{-1} A_{0n}$ n>0
- b) compute $L_{n0} = A_{n0} U_{00}^{-1}$ n>0
- Third step: compute $?_{nm} = A_{nm} - L_{n0} * U_{0m}$, (n,m>0)

Now Factorize Lower Right Corner Blocks

- We repeat the algorithm recursively for two lower right corner blocks.

$$\begin{bmatrix} ?11 & ?12 \\ ?21 & ?22 \end{bmatrix} = \begin{bmatrix} L11 & \mathbf{0} \\ L21 & \mathbf{1} \end{bmatrix} * \begin{bmatrix} U11 & U12 \\ 0 & ??22 \end{bmatrix}$$

End Result

$$
\begin{bmatrix} A00 & A01 & A02 \\ A10 & A11 & A12 \\ A20 & A21 & A22 \end{bmatrix}
=
\begin{bmatrix} L00 & 0 & 0 \\ L10 & L11 & 0 \\ L20 & L21 & L22 \end{bmatrix}
*
\begin{bmatrix} U00 & U01 & U02 \\ 0 & U11 & U12 \\ 0 & 0 & U22 \end{bmatrix}
$$

Parallel Algorithm

P0: $A_{00} = L_{00} * U_{00}$
(compute by L_{00}, U_{00} by LU factorization)

P1: $U_{01} = L_{00}^{-1} A_{01}$

P2: $U_{02} = L_{00}^{-1} A_{02}$

P3: $L_{10} = A_{10} U_{00}^{-1}$

P6: $L_{20} = A_{20} U_{00}^{-1}$

P4: $A_{11} \leftarrow A_{11} - L_{10} * U_{01}$

P5: $A_{12} \leftarrow A_{12} - L_{10} * U_{02}$

P7: $A_{21} \leftarrow A_{21} - L_{20} * U_{01}$

P8: $A_{22} \leftarrow A_{22} - L_{20} * U_{02}$

P0	P1	P2
P3	P4	P5
P6	P7	P8

In the general case:

$A_{nm} = L_{n0} * U_{0m} + ?_{nm} \rightarrow ?_{nm} = A_{nm} - L_{n0} * U_{0m}$

Parallel Algorithm

P0: $A_{00} = L_{00} * U_{00}$
(compute by L_{00}, U_{00} by LU factorization)

P1: $U_{01} = L_{00}^{-1} A_{01}$

P2: $U_{02} = L_{00}^{-1} A_{02}$

P3: $L_{10} = A_{10} U_{00}^{-1}$

P6: $L_{20} = A_{20} U_{00}^{-1}$

P4: $A_{11} \leftarrow A_{11} - L_{10} * U_{01}$

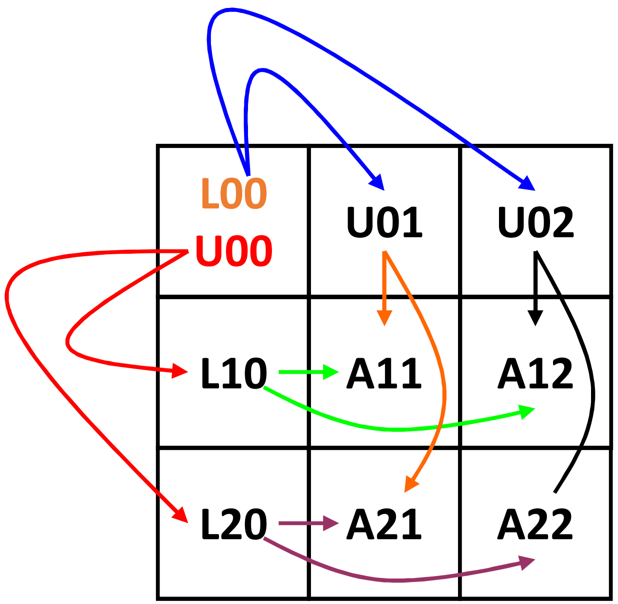
P5: $A_{12} \leftarrow A_{12} - L_{10} * U_{02}$

P7: $A_{21} \leftarrow A_{21} - L_{20} * U_{01}$

P8: $A_{22} \leftarrow A_{22} - L_{20} * U_{02}$

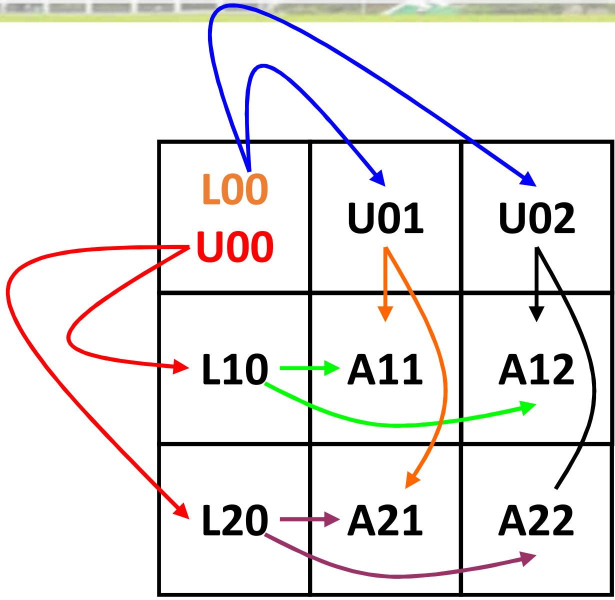
In the general case:

$A_{nm} = L_{n0} * U_{0m} + ?_{nm} \rightarrow ?_{nm} = A_{nm} - L_{n0} * U_{0m}$



Communication Summary

P0	P1	P2
P3	P4	P5
P6	P7	P8



P0: L_{00}, U_{00} =lu(A)

P1: $U_{01} = L_{00}^{-1}A_{01}$
P2: $U_{02} = L_{00}^{-1}A_{02}$

P3: $L_{10} = A_{10}U_{00}^{-1}$
P6: $L_{20} = A_{20}U_{00}^{-1}$

P4: $A_{11} \leftarrow A_{11} - L_{10}{}^*U_{01}$
P5: $A_{12} \leftarrow A_{12} - L_{10}{}^*U_{02}$
P7: $A_{21} \leftarrow A_{21} - L_{20}{}^*U_{01}$
P8: $A_{22} \leftarrow A_{22} - L_{20}{}^*U_{02}$

P0: sends L_{00} to P1,P2
sends U_{00} to P3,P6

P1: sends U_{01} to P4,P7
P2: sends U_{02} to P5,P8

P3: sends L_{10} to P4,P5
P4: sends L_{20} to P7,P8