*ACM40290: Numerical Algorithms*

# Numerical Linear Algebra II

Dr Barry Wardell
School of Mathematics and Statistics
University College Dublin

We solved $Ax = b$ using Gaussian Elimination which required elementary row operations to be performed on both $A$ and $b$. These operation are determined by the elements of $A$ only. If we are required to solve the new equation $Ax = b'$ then *GaussElim* we would perform exactly the same operations because $A$ is the same in both equations. Hence if we have stored the multipliers $m_{ik}$ we need to perform only the final line of Algorithm *GaussElim* , i.e.,

$$b_i := b_i - m_{ik} b_k, \quad i = k+1, \ldots, n, \, k = 1, \ldots, n-1.$$

If at each stage $k$ of *GaussElim* we store $m_{ik}$ in those cells of $A$ that become zero then the $A$ matrix after elimination would be as follows

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ m_{21} & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & a_{nn}^{(n)} \end{bmatrix} = \begin{bmatrix} \diagdown & U \\ L & \diagdown \end{bmatrix}$$

We define the upper and unit lower triangular parts as

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & 1 \end{bmatrix}$$

We now modify *GaussElim* to incorporate these ideas :

**algorithm** *GaussElimLU* $(a, n)$

---

Assume that no $a_{kk} = 0$

**for** $k := 1$ **to** $n-1$ **do**
    **for** $i := k+1$ **to** $n$ **do**
        $a_{ik} := a_{ik} / a_{kk}$
        **for** $j := k+1$ **to** $n$ **do**
            $a_{ij} := a_{ij} - a_{ik} \times a_{kj}$
        **endfor** $j$
    **endfor** $i$
**endfor** $k$
**endalg** *GaussElimLU*

No diagonal elements stored - known. (1's)

$m_{ik}$.

Compare to our previous algorithm:

```
algorithm GaussElim (a,b,n)
--------------------------------------
  Assume that no a_kk = 0
      for k := 1 to n-1 do
          for i := k+1 to n do
              m_ik := a_ik / a_kk
              for j := k+1 to n do
                  a_ij := a_ij - m_ik × a_kj
              endfor j
              b_i := b_i - m_ik × b_k
          endfor i
      endfor k
endalg GaussElim
```

**Theorem 5.6** ($LU$ Decomposition)**.** *If L and U are the upper and lower triangular matrices generated by Gaussian Elimination, assuming $a_{kk}^{(k)} \neq 0$ at each stage, then*

$$\begin{aligned} A &= LU = \sum_{k=1}^{n} l_{ik} u_{kj} \\ \text{where} \quad u_{kj} &= a_{kj}^{(k)} \quad k \leq j, \quad u_{kk} = a_{kk}^{(k)} \\ l_{ik} &= m_{ik} \quad k \leq i, \quad l_{kk} = 1, \end{aligned}$$

*and this decomposition is unique.*

$$Ax = LUx = L(Ux) = Ly = b.$$

$$Ly = b \quad \text{and} \quad Ux = y$$

solutions are :

$$y = L^{-1}b, \quad Ux = L^{-1}b, \quad x = U^{-1}L^{-1}b.$$

The steps in solving $Ax = b$ using $LU$ Decomposition are as follows:

**algorithm** *SolveLU*$(a, b, n, x)$

---

1. Calculate $L$ and $U$ using *GaussElimLU*$(A, n)$, where $A = LU$.
2. Solve $Ly = b$ using *ForwSubst*$(L, b, n, y)$, where $y = L^{-1}b$.
3. Solve $Ux = y$ using *BackSubst*$(U, y, n, x)$, where $x = U^{-1}y$.

# The $LDU$ Decomposition of $A$

Gaussian Elimination also provides the decomposition

$$A = LDU',$$

where $L$ and $U'$ are unit lower and unit upper triangular and $D = [u_{ii}]$.

$$U' = D^{-1}U.$$

If $A$ is *symmetric* then

$$A = LDU' = LDL^{\mathsf{T}},$$

where $L$ is unit lower triangular.

**The Cholesky Decomposition of $A$**

If $A$ is symmetric and *positive definite* ($x^{\mathsf{T}}Ax > 0$) then we can decompose $A$ as follows :

$$A = LDL^{\mathsf{T}} = L\sqrt{D}\sqrt{D}L^{\mathsf{T}} = CC^{\mathsf{T}},$$

where $C = L\sqrt{D}$ and $\sqrt{D} = [\sqrt{d}_{ii}]_1^n$. This is possible because $x^{\mathsf{T}}Ax > 0 \Rightarrow d_{ii} > 0$. This is often called the **Cholesky Factorization** of $A$.

# Using the $LU$ Decomposition

## The Determinant of $A$

We have

$$\det(A) = \det(LU) = \det(L)\det(U).$$

$$\det(L) = 1$$

$$\det(U) = u_{11}u_{22}\ldots u_{nn}$$

$$\det(A) = u_{11}u_{22}\ldots u_{nn} = \prod_{i=1}^{n} u_{ii}.$$

Can give rise to ~~underflow~~
overflow

**Solving $AX = B$**

If $A$ is $n \times n$ and $X$ and $B$ are $n \times m$,

$x^j = j$th column of $X$ and $b^j = j$th column of $B$

$$LUx^j = b^j \quad j = 1, 2, \ldots, m.$$

**algorithm** *SolveAXB* $(A, X, B, m, n)$

---

$A$ is an $n \times n$ matrix, $X$ and $B$ are $n \times m$ matrices
$b^j$ and $x^j$ are the $j$th columns of $B$ and $X$, respectively

```
GaussElimLU(A, n)  returns L and U where A = LU
for j := 1 to m do
        ForwSubst(L, b^j, n, y)
        BackSubst(U, y, n, x^j)
endfor j
endalg SolveAXB
```

# Matrix Inversion, $A^{-1}$

Solve $AX = B$, where $B = I$.

$$Ax^j = e^j \quad j = 1, 2, \ldots, n$$

where $x^j$ is the $j$th column of $X = A^{-1}$ and $e^j$ is the $j$th column of $I$.

# simple modification of *SolveAXB*:

**algorithm** *Invert* $(A, X, n)$

---

$A$ is an $n \times n$ matrix, $X$ is the inverse of $a$
$e^j$ and $x^j$ are the $j$th columns of $I$ and $X$, respectively

```
GaussElimLU(A,n) returns L and U where A = LU
for j := 1 to n do
        ForwSubst(L, e^j, n, y)
        BackSubst(U, y, n, x^j)
endfor j
endalg Invert
```

# Pivoting and Scaling in Gaussian Elimination

If $a_{kk}^{(k)} = 0$ then we can interchange rows of the matrix $A^{(k)}$ so that $a_{kk}^{(k)} \neq 0$.

The process of interchanging rows (or columns or both) in Gaussian Elimination is called *pivoting*.

even if $a_{kk}^{(k)} \neq 0$ then a small $a_{kk}^{(k)}$ could cause problems because of roundoff.

**Example** (Need for Pivoting 1). Consider the following set of equations :

$$\begin{aligned} .0001x_1 + 1.00x_2 &= 1.00 \\ 1.00x_1 + 1.00x_2 &= 2.00 \end{aligned}$$

The exact solution is

$$\begin{aligned} x_1 &= \frac{10000}{9999} = 1.00010 \\ x_2 &= \frac{9998}{9999} = 0.99990 \end{aligned}$$

If we perform Gaussian Elimination without interchanges, using 3-digit precision we get

$$\begin{aligned} A^{(2)} = 0.000100x_1 + 1.00x_2 &= 1.00 \\ -10{,}000x_2 &= -10{,}000. \end{aligned}$$

Hence $x_2 = 1.00$ and $x_1 = 0.0$.

If we interchange rows 1 and 2 we get

$$\begin{aligned} A^{(2)} = 1.00x_1 + 1.00x_2 &= 2.00 \\ 1.00x_2 &= 1.00. \end{aligned}$$

Hence $x_2 = 1.00$ and $x_1 = 1.00$. Both of these are accurate to 3 decimal digits.

**Example** (Need for Pivoting 2.). Here is a more general example that clearly shows the need for pivoting when using finite precision arithmetic. The problem is

$$\text{Solve} \quad \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \epsilon < \boldsymbol{\varepsilon}_m = 10^{-16}, \quad \text{using d.p. floating point arithmetic.}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1/(1-\epsilon) \\ 2 - 1/(1-\epsilon) \end{bmatrix}$$

The correctly-rounded exact answer is

$$\mathrm{fl}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1/\mathrm{fl}(1-\epsilon) \\ 2 - 1/\mathrm{fl}(1-\epsilon) \end{bmatrix} = \begin{bmatrix} 1/1 \\ 2-1/1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{because} \quad \epsilon < \boldsymbol{\varepsilon}_m \,.$$

Now we perform Gaussian Elimination without pivoting (row interchanges)

$$A = A^{(1)} = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}, \quad M_1 = \begin{bmatrix} 1 & 0 \\ -1/\epsilon & 1 \end{bmatrix},$$

$$A^{(2)} = M_1 A^{(1)} = \begin{bmatrix} \epsilon & 1 \\ 0 & \mathrm{fl}(1-1/\epsilon) \end{bmatrix} = \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}.$$

$$L = M_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix}, \quad \text{and} \quad \hat{U} = A^{(2)} = \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}.$$

$$L\hat{U} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix} \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix} = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix} = \hat{A} \neq A.$$

If we use the factors $L$ and $\hat{U}$ to solve the original problem we get

$$\begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix} \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Forward substitution gives

$$\begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/\epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ \mathrm{fl}(2 - 1/\epsilon) \end{bmatrix} = \begin{bmatrix} 1 \\ -1/\epsilon \end{bmatrix}.$$

Back substitution on

$$\begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1/\epsilon \end{bmatrix} \quad \text{gives} \quad \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} (1-1)/\epsilon \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

If we use pivoting then the steps are

$$A = A^{(1)} = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad P_1 A^{(1)} = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}, \quad M_1 = \begin{bmatrix} 1 & 0 \\ -\epsilon & 1 \end{bmatrix},$$

$$A^{(2)} = M_1 P_1 A^{(1)} = \begin{bmatrix} 1 & 1 \\ 0 & \mathrm{fl}(1-\epsilon) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \hat{U} \quad \text{and} \quad L = M_1^{-1} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}.$$

Checking that $L\hat{U} = PA$ gives

$$\begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \epsilon & \mathrm{fl}(1+\epsilon) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix} = PA.$$

Using these factors we get $LUx = Pb$, or

$$\begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Forward substitution gives

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\epsilon & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ \mathrm{fl}(1-\epsilon) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Back substitution gives $x_2 = 1$ and then $x_2 = 1$, the correctly-rounded exact answer.

**Partial Pivoting**

If $A$ is nonsingular then at each stage $k$ of Gaussian Elimination we are guaranteed that some $a_{ik}^{(k)} \neq 0, i \geq k$.

Partial pivoting finds the row $p_k$ for which $|a_{p_k k}^{(k)}|$ is a maximum and interchanges rows $p_k$ and $k$. Hence we must add the following statements in the outer loop of the elimination algorithm.

1. Find $p_k$ that maximizes $|a_{p_k k}^{(k)}|, \quad k \leq p_k \leq n$
2. Interchange rows $k$ and $p_k$

If we are calculating the determinant we must remember that each interchange $p_k \neq k$ causes a change of sign.

If complete rows are interchanged, i.e., multipliers and $A$-matrix elements, the array calculated by Gaussian Elimination with partial pivoting (GEPP) will contain the $LU$ decomposition of $P_{n-1} \cdots P_2 P_1 A$. Hence we have

$$P_{n-1} \cdots P_2 P_1 A = PA = LU.$$

**Theorem 5.7** ($LUP$-Decomposition). *If $A$ is an arbitrary square matrix there exists a permutation matrix $P$, a unit lower triangular matrix $L$ and an upper triangular matrix $U$ such that*

$$LU = PA.$$

*However $L$ and $U$ are not always uniquely determined by $P$ and $A$.*

**Proof 5.2.** *(see Forsythe and Moler, Section 16).*

**algorithm** *GaussElimLUP* $(a, n, p)$

---

Gaussian Elimination with partial pivoting

**for** $k := 1$ **to** $n - 1$ **do**

  $p_k :=$ `MaxPiv`$(a, k, n)$

  **for** $j := 1$ **to** $n$ **do**

    `temp` $:= a_{kj}$

    $a_{kj} := a_{p_k j}$

    $a_{p_k j} :=$ `temp`

  **endfor** $j$

  **for** $i := k + 1$ **to** $n$ **do**

    $a_{ik} := a_{ik} / a_{kk}$

    **for** $j := k + 1$ **to** $n$ **do**

      $a_{ij} := a_{ij} - a_{ik} \times a_{kj}$

    **endfor** $j$

  **endfor** $i$

**endfor** $k$

**endalg** *GaussElimLUP*

Once we have $L$, $U$, and $P$ we can solve $Ax = b$ as follows :

$$Ax = b \rightarrow PAx = Pb \rightarrow LU = Pb$$

**algorithm** *SolveLEQP*$(a, b, n, x)$

---

1. Calculate $P$, $L$ and $U$ using $\texttt{GaussElimLUP}(A, n, P)$, where $PA = LU$.
2. Calculate $b' = Pb$.
3. Solve $Ly = b'$ using $\texttt{ForwSubst}(L, b', n, y)$, where $y = L^{-1}b'$.
4. Solve $Ux = y$ using $\texttt{BackSubst}(U, y, n, x)$ , where $x = U^{-1}y$.

# Gaussian Elimination and $LU$ Decomposition

The first step of Gaussian Elimination of $Ax = b$ transforms

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix} \quad \text{to} \quad A^{(2)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

This is the same as premultiplying $A^{(1)}$ by $M_1$ i.e. $A^{(2)} = M_1 A^{(1)}$, where

$$M_1 = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & 0 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad \text{and} \quad m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, 3, \ldots, n.$$

In general, at stage $k$ we have $A^{(k+1)} = M_k A^{(k)}$ where

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nk}^{(k)} & \cdots & a_{nn}^{(n)} \end{bmatrix} \qquad M_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & -m_{n,k} & & 1 \end{bmatrix}$$

The whole process of elimination is really a sequence of matrix operations on the original matrix where

$$\begin{aligned}
A^{(1)} &= A \\
A^{(2)} &= M_1 A^{(1)} \\
A^{(3)} &= M_2 A^{(2)} = M_2 M_1 A \\
&\vdots \\
A^{(n)} &= M_{n-1} A^{(n-1)} = M_{n-1} M_{n-2} \cdots M_2 M_1 A^{(1)} = MA
\end{aligned}$$

Thus $A^{(n)} = MA = U$, and the pair of matrices $(M, U)$ is called the *Elimination Form of the Decomposition*.

If we pre-multiply the equation $MA = U$ by $M^{-1}$ we get

$$\begin{aligned} A = M^{-1}U &= (M_{n-1}M_{n-2}\cdots M_2M_1)^{-1}U \\ &= M_1^{-1}M_2^{-1}\cdots M_{n-1}^{-1}U, \\ &= L_1L_2\cdots L_{n-1}U, \\ &= LU. \end{aligned}$$

However, Gaussian Elimination (G.E) computes $M_{n-1}M_{n-2}\cdots M_2M_1 = M$ and not $L_1L_2\cdots L_{n-1} = L$, and it is this matrix that we need.

Q: How do we get the components of L?

A: They are calculated by the G. E. algorithm.

follow G.W Stewart (1996), using a $4 \times 4$ matrix $A$.

We start with

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{bmatrix} \quad \text{and} \quad M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & 0 & 1 & 0 \\ -m_{41} & 0 & 0 & 1 \end{bmatrix}$$

Then stage 1 is :

$$A^{(2)} = M_1 A^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & 0 & 1 & 0 \\ -m_{41} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{bmatrix}$$

Then stage 2 is :

$$A^{(3)} = M_2 A^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_{32} & 1 & 0 \\ 0 & -m_{42} & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{bmatrix}.$$

Finally, stage 3 is :

$$A^{(4)} = M_3 A^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -m_{43} & 1 \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(4)} \end{bmatrix}.$$

## How is L obtained?

$$M_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & -m_{nk} & & 1 \end{bmatrix} \quad \text{then} \quad M_k^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & m_{nk} & & 1 \end{bmatrix}$$

(signs reversed in off-diagonal)

# Applying this to the 4 x 4 case

$$
\begin{aligned}
L = L_1L_2L_3 &= M_1^{-1}M_2^{-1}M_3^{-1} \\
&= M_1^{-1}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{32} & 1 & 0 \\ 0 & m_{42} & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & m_{43} & 1 \end{bmatrix} = M_1^{-1}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{32} & 1 & 0 \\ 0 & m_{42} & m_{43} & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & 0 & 1 & 0 \\ m_{41} & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{32} & 1 & 0 \\ 0 & m_{42} & m_{43} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix}
\end{aligned}
$$

This shows that the G.E multipliers $m_{ij}$ are the elements of the unit lower-triangular matrix $L$ for $i > j$ and that $A = LU$. The derivation above applies to matrices of any order $n$ and we have the result :

> *Gaussian Elimination* applied to the matrix $A$ gives an upper-triangular matrix $U$ and a unit-lower-triangular matrix $L$ where
>
> $$A = LU,$$
>
> and the elements of $L$ are the multipliers $m_{ij}$ calculated by Gaussian Elimination.