# CE 601: Numerical Methods
# Lecture 5

Course Coordinator:
Dr. Suresh A. Kartha,
Associate Professor,
Department of Civil Engineering,
IIT Guwahati.

# Elimination Methods

- For a system [A]{x} = {b} where [A] is an n X n matrix, the number of operations = $(2/3)n^3+(3/2)n^2-(7/6)n$.
- Note: The formula $(2/3)n^3+(3/2)n^2-(7/6)n$ will always yield <u>integer values</u>.
- Like Gauss elimination, you might have studied Gauss-Jordan elimination method. The objective of Gauss-Jordan scheme was to convert [A]{x} = {b} => [I]{x} = {x}, [I] -> Identity matrix, through systematic elimination process.
- We are not going to discuss on this method in the class and request you to work on your our referring literature.

- The Gauss-Jordan scheme is computationally less efficient than the Gauss elimination scheme. The no. of operations involved in Gauss-Jordan scheme = $n^3+n^2-n$.
- Like Gauss elimination, Gauss-Jordan, etc. you might have studied matrix inverse methods to solve linear systems

  i.e. $[A]\{x\} = \{b\}$

  so, $\{x\} = [A]^{-1}\{b\}$

  There are two evaluations in this scheme

→ First evaluate the inverse of matrix [A]

→Second, to evaluate the product of $[A]^{-1}\{b\}$

- In both the evaluations, there are arithmetic operations involved. For matrix inverse it takes $2n^2-n$ operations i.e. total of $2n^3$ operations.
- Based on the significant digits assigned to the variables or components, round-off errors may creep in the solutions while using Gauss elimination scheme.

  → These errors can be minimized by performing partial pivoting or scaled partial pivoting.

- LU Decomposition

- We have discussed that matrix can be factored i.e., it can be given as product of two different matrix.

  [A] = [B][C]

- There can be many possibilities of obtaining factor matrices.

  → However if we specify the diagonal elements of either [L] or [U], then we will have a unique a factorization for [A].

  →The LU decomposition methods like Doolittle and Crout work on these principles.

- In a similar tone, one can also factorize [A] as product of [L] and [U] i.e., [A]= [L][U] where [L] is lower triangular and [U] is a upper triangular matrix.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

- <u>Property of LU decomposition</u>

○ $[A]\{x\} = \{b\}$

$[L][U]\ \{x\} = \{b\}$

Multiply left and right side by $[L]^{-1}$

$[L]^{-1}[L][U]\ \{x\} = [L]^{-1}\ \{b\}$

$\Rightarrow [I][U]\{x\} = [L]^{-1}\ \{b\}$

$\Rightarrow [U\}\{x\} = [L]^{-1}\ \{b\}$

○ Let us define the system $[U]\{x\} = \{c\}$

So, we get $\{c\} = [L]^{-1}\ \{b\}$

or, $[L]\{c\} = \{b\}$

- The meaning here is:
- ✓ Lower triangular matrix [L] will transform RHS vector {b} to {c} using the relation $[L]^{-1}\{c\} = \{b\}$.
- ✓ On obtaining {c} , using the relation $[U]\{x\} = \{c\}$, we obtain the solution vector.
- ✓ So if we can suitable algorithm such that process involved in LU decomposition are”
  - Factorize $[A] = [L][U]$
  - Forward substitution to evaluate {c}
  - Backward substitution to evaluate {x]
- ✓ If the no. of arithmetic operations in factorization can be reduced then LU decomposition becomes efficient.

- <u>The Advantage of LU Decomposition</u>

- If there are many linear systems with same coefficient matrix i.e.,

  [A]{x} = {b}

  [A]{y} = {c}

  [A]{z} = {d}

- Then if we adopt Gauss elimination, for each system it may take $(2/3)n^3+(3/2)n^2-(7/6)n$ operations. The purpose may become tedious.
- If we do LU decomposition, then you may see that only once we need to decompose [A]=[L][U]
- After that for each system, we need to utilize only the forward and backward substitutions.

- <u>Doolittle LU Decomposition</u>
- [A] = [L][U]
- If we factorize in such a way that

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

i.e. diagonal elements of [L] are 1, then the approach is Doolittle's method.

- If

$$[A] = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

the approach is called Crout's method.

# Algorithm for Doolittle's LU Decomposition

- Doolittle algorithm is developed using knowledge of Gauss elimination.
- Recall Gauss elimination at any step *k*

$$\left.\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - l_{ik} a_{kj}^{(k-1)} \\ l_{ik} &= a_{ik}^{(k-1)} \big/ a_{kk}^{(k-1)} \end{aligned}\right\} k = 1,2,3,\ldots,n-1;\ j = k, k+1,\ldots,n; i = k+1, k+2,\ldots,n$$

- You can now write

$$a_{ij}^{(k)} - a_{ij}^{(k-1)} = -l_{ik} a_{kj}^{(k-1)}$$

$$\text{i.e., } a_{ij}^{(k)} - a_{ij}^{(k-2)} = -l_{ik} a_{kj}^{(k-1)} - l_{i(k-1)} a_{(k-1)j}^{(k-2)}$$

$$\text{Extending, } a_{ij}^{(k)} - a_{ij} = -\sum_{m=1}^{k} l_{im} a_{mj}^{(m-1)}$$

$$\text{or, } a_{ij} = a_{ij}^{(k)} + \sum_{m=1}^{k} l_{im} a_{mj}^{(m-1)}; i = k+1, k+2,\ldots,n;\ j = k, k+1,\ldots,n$$

- This is nothing but

$$[A]=[L][U]$$

$$\Rightarrow a_{ij}^{(k)} = a_{ij} - \sum_{m=1}^{k} l_{im} a_{mj}^{(m-1)}$$

where $l_{ij} \rightarrow$ elements of lower triangular matrix

$$\text{such that } l_{ij} = \begin{cases} l_{ik} & ; \text{for } i > k; k = 1,2,3,\ldots,n-1 \\ 1 & ; \text{for } i = j \\ 0 & ; \text{for } i < j \end{cases}$$

- So you get

$$[L] = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \ldots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}, [U] = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{pmatrix}$$

where $l_{21} = a_{21}/a_{11}, l_{32} = a_{32}^{(1)}/a_{22}^{(1)}$, etc.

- So the steps involved are:
  - No. of steps as in Gauss elimination

    *k = 1,2,3,…,n-1*
  - At any *k* , *i= k+1,k+2,…,n* and *j=k,k+1,…,n*

$$l_{kk} = 1$$

$$l_{ik} = 0; i < k$$

$$u_{ij} = a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik} a_{kj}^{(k-1)}$$

$$u_{ij} = 0$$

$$l_{ik} = a_{ik}^{(k-1)} \Big/ a_{kk}^{(k-1)}; i > k$$

- Forward substitution for *c*

$$c_1 = b_1$$

$$c_i = b_i - \sum_{m=1}^{k-1} l_{im} c_m ; i = 2, 3, \ldots, n$$

- Back substitution for *x*

$$x_n = c_n / u_{nn}$$

$$x_{n-1} = \frac{c_{n-1} - u_{(n-1)n} x_n}{u_{(n-1)(n-1)}}$$

$$\text{In genral, } x_i = \frac{c_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}} ; i = (n-1), (n-2), \ldots, 2, 1$$

- <u>Example</u>

$$\begin{pmatrix} 80 & -20 & -20 \\ -20 & 40 & -20 \\ -20 & -20 & 130 \end{pmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 20 \\ 20 \\ 20 \end{Bmatrix}$$

$$A = L \; U$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix}, \; U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

$$l_{21} = -1/4, l_{31} = -1/4, l_{32} = -5/7$$

$$u_{11} = 80, u_{12} = -20, u_{13} = -20$$

$$u_{21} = 0, u_{22} = 40 - (-1/4)(-20) = 35$$

$$u_{23} = -20 - (-1/4)(-20) = -25$$

$$u_{31} = u_{32} = 0$$

$$u_{33} = 130 - \sum_{m=1}^{k=2} l_{3m} a_{mj}^{(m-1)}$$

$$= 750/7$$

Forward Substitution:

$$c_1 = b_1 = 20$$

$$c_2 = 20 - (-20 / 4) = 25$$

$$c_3 = 20 - (-20 / 4) - (-5 \times 25 / 7) = 300 / 7$$

$$[U]\{x\} = \{c\}$$

$$\text{i.e.,} \begin{pmatrix} 80 & -20 & -20 \\ 0 & 35 & -25 \\ 0 & 0 & 750/7 \end{pmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 20 \\ 25 \\ 300/7 \end{Bmatrix}$$

Now you can do back substitution easily.

- <u>Number of operations</u>
- In Doolittle's algorithm,
  - to evaluate $l_{ik}$ for any k$^{th}$ step for $i > k$, it takes ($n$-$k$) operations.
  - To evaluate $u_{ij} = a_{ij}^{(k)}$ it takes $2(n-k)^2$ operations.
  - In ($n$-$k$) elimination steps to form [L] and [u], it takes, $\sum_{p=1}^{n-1}(n-p)(2n-2p+1)$

$$= \frac{n}{6}(n-1)(4n+1) = \frac{2}{3}n^3 - \frac{n^2}{2} - \frac{n}{6}$$

- To perform forward substitution,

  as $c_1 = b_1$ (No operation required)

$$c_i = b_i - \sum_{m=1}^{i-1} l_{im} c_m ; i = 2, 3, 4, \ldots, n$$

There are *2i* operations for each i.

So, no. of operations = $\sum_{i=2}^{n} 2i = n^2 - n$

- To perform backward substitution, it requires $n^2$ operations (as in Gauss)
- Total no. of operations = $\frac{2}{3}n^3 - \frac{n^2}{2} - \frac{n}{6} + 2n^2 - n$

$$= \frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n$$

- No. of operations is same as Gauss elimination method.
- However if there are many systems involving [A], then you may need to just add the no. of operations for forward & backward substitution for each system.