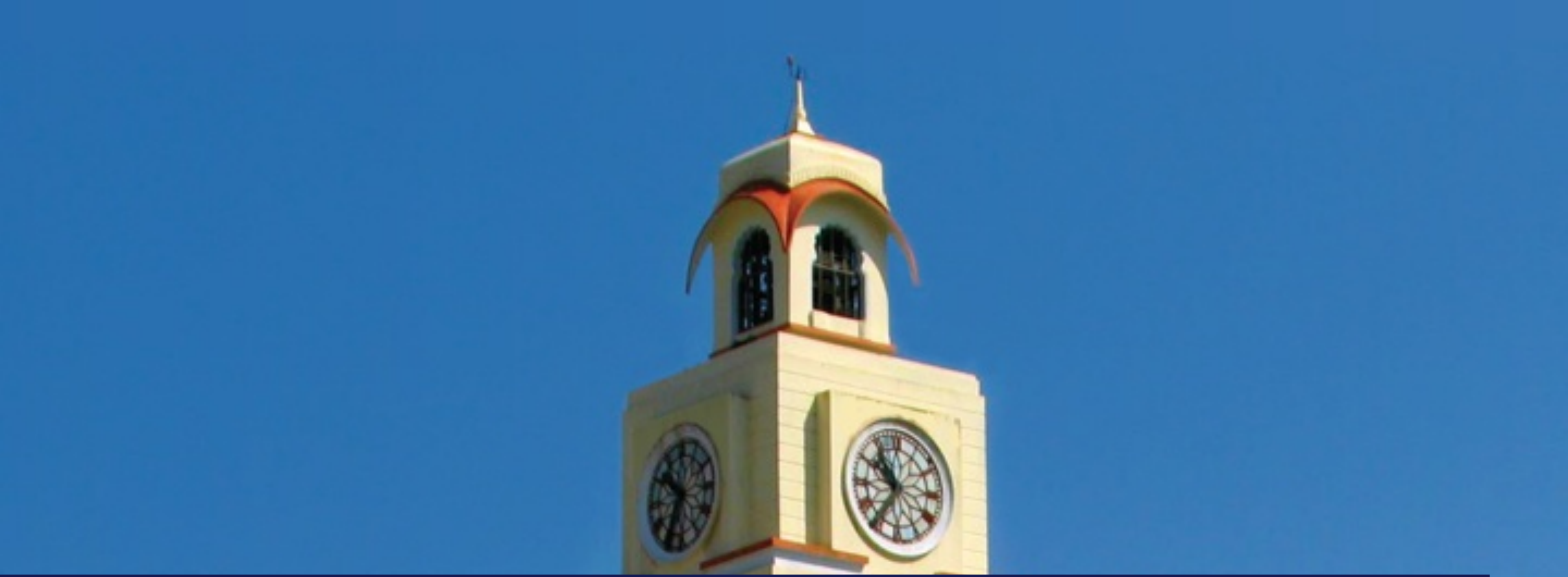
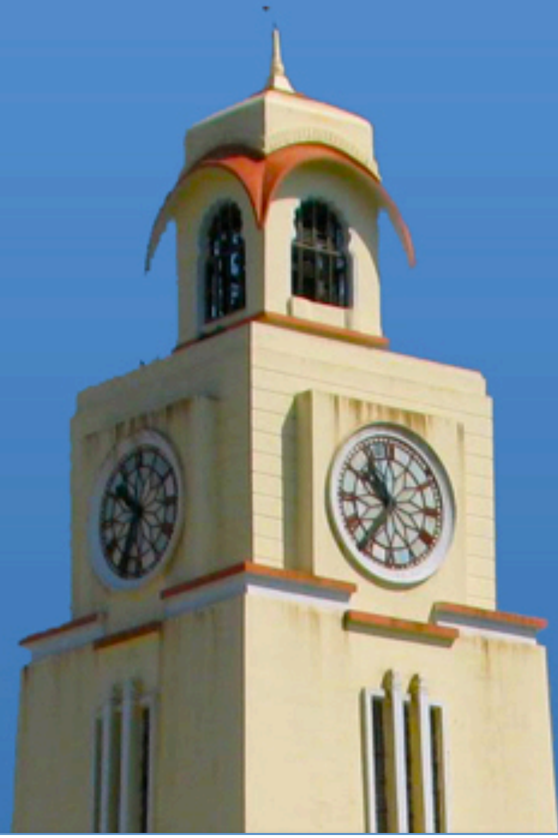


BITS Pilani
Pilani Campus

Mathematical Foundations for Data Science

MFDS Team



innovate achieve lead

BITS Pilani
Pilani Campus

DSECL ZC416, MFDS

Lecture No. 5

Agenda

- Gauss Elimination
  - Scaling
  - Pivoting
  - Efficiency
- LU Factorization
  - Doolittle’s method
  - Crout’s method
  - Cholesky’s method

Gauss Elimination

Solve $Ax = b$

Consists of two phases:

Forward elimination

Back substitution

Forward Elimination reduces $Ax = b$ to an upper triangular system $Tx = b'$

Back substitution can then solve $Tx = b'$ for x

$$\underbrace{\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right] \Downarrow \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}' & a_{23}' & b_2' \\ 0 & 0 & a_{33}'' & b_3'' \end{array}\right]}_{\text{Forward Elimination}}$$

$$\Downarrow$$

$$\underbrace{x_3 = \frac{b_3''}{a_{33}''} \qquad x_2 = \frac{b_2' - a_{23}' x_3}{a_{22}'} \qquad x_1 = \frac{b_1 - a_{13}x_3 - a_{12}x_2}{a_{11}}}_{\text{Back Substitution}}$$

Pitfalls of Gauss Elimination

Division by zero

It is possible that during both elimination and back-substitution phases a division by zero can occur.

For example:

$$
\begin{aligned}
2x_2 + 3x_3 &= 8 \\
4x_1 + 6x_2 + 7x_3 &= -3 \\
2x_1 + x_2 + 6x_3 &= 5
\end{aligned}
\qquad
A = \begin{matrix} 0 & 2 & 3 \\ 4 & 6 & 7 \\ 2 & 1 & 6 \end{matrix}
$$

$a_{11} = 0$
(the pivot element)

It is possible that during both elimination and back-substitution phases a division by zero can occur.

Solution: **Pivoting**

Pitfalls of Gauss Elimination

Round-off errors

Because computers carry only a limited number of significant figures, round-off errors will occur and they will *propagate* from one iteration to the next.

This problem is especially important when **large** numbers of equations (100 or more) are to be solved.

Always use **double-precision** numbers/arithmetic. It is slow but needed for correctness!

It is also a good idea to substitute your results back into the original equations and check whether a substantial error has occurred.

Ill conditioned systems

- Systems where small changes in coefficients result in large change in solution

$$x_1 + 2x_2 = 10$$
$$1.1x_1 + 2x_2 = 10.4$$

→ $x_1 = 4.0$ & $x_2 = 3.0$

$$x_1 + 2x_2 = 10$$
$$1.05x_1 + 2x_2 = 10.4$$

→ $x_1 = 8.0$ & $x_2 = 1.0$

Norms

Vector Norm – A vector norm for column vectors $x = [x_j]$ with n components is a generalized length, is denoted by $\|x\|$ satisfies postulates :

a. $\|x\|$ is a nonnegative real number

b. $\|x\| = 0$ if and only if $x = 0$

c. $\|kx\| = |k|\|x\|$ for all k

d. $\|x + y\| \leq \|x\| + \|y\|$ (Triangular Inequality)

Matrix Norm – Matrix norm corresponding to given vector is defined by

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

Matrix Norm

Norm of a matrix measures maximum stretching matrix does to any vector in given vector norm

Matrix norm corresponding to vector 1-norm is maximum absolute column sum

$$\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

Matrix norm corresponding to vector ∞ - norm is maximum absolute row sum,

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

$||\mathbf{A}||_2$ is the *Frobenius norm*

$$||\mathbf{A}||_2 = ||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij}|^2}$$

Condition Number

Condition number of square nonsingular matrix A defined by

$$cond(A) = \|A\| \cdot \|A\|^{-1}$$

By convention, cond $(A) = \infty$ if A singular

Example: $A = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 0 & 1 \\ 3 & -1 & 4 \end{bmatrix}$ $\|A\|_1 = 6$ $\|A\|_\infty = 8$

$$A^{-1} = \begin{bmatrix} 0.5 & 1.5 & -0.5 \\ -0.5 & 2.5 & -0.5 \\ -0.5 & -0.5 & 0.5 \end{bmatrix} \quad \|A^{-1}\|_1 = 4.5 \quad \|A^{-1}\|_\infty = 3.5$$

cond_1 (A) = 6 x 4.5 = 27

cond_∞ (A) = 8 x 3.5 = 28

Techniques for Improving the solution

Use of more significant figures – double precision arithmetic

Pivoting
If a pivot element is zero, normalization step leads to division by zero. The same problem may arise, when the pivot element is close to zero. Problem can be avoided:

Partial pivoting

- Switching the rows below so that the largest element is the pivot element.

Complete pivoting

- Searching for the largest element in all rows and columns then switching.
- This is rarely used because switching columns changes the order of x's and adds significant complexity and overhead ➔ costly

Scaling **-** used to reduce the round-off errors and improve accuracy

Partial Pivoting – Example

Pivoting Example

Example 14: Solve the following system using Gauss Elimination with pivoting.

$$
\begin{aligned}
2x_2 + \quad\quad + x_4 &= 0 \\
2x_1 + 2x_2 + 3x_3 + 2x_4 &= -2 \\
4x_1 - 3x_2 \quad\quad + x_4 &= -7 \\
6x_1 + x_2 - 6x_3 - 5x_4 &= 6
\end{aligned}
$$

Step 0: Form the augmented matrix

$$
\left[\begin{array}{cccc|c}
0 & 2 & 0 & 1 & 0 \\
2 & 2 & 3 & 2 & -2 \\
4 & -3 & 0 & 1 & -7 \\
6 & 1 & -6 & -5 & 6
\end{array}\right]
$$

Step 1: Forward Elimination

(1.1) Eliminate x_1 . But the pivot element is 0. We have to interchange the 1^{st} row with one of the rows below it. Interchange it with the 4^{th} row because 6 is the largest possible pivot.

Partial Pivoting – Example

(1.1) Eliminate x_1 . But the pivot element is 0. We have to interchange the 1st row with one of the rows below it. Interchange it with the 4th row because 6 is the largest possible pivot.

6	1	-6	-5	|	6
2	2	3	2	|	-2
4	-3	0	1	|	-7
0	2	0	1	|	0

Now eliminate x_1

6	1	-6	-5	|	6
0	1.6667	5	3.6667	|	-4
0	-3.6667	4	4.3333	|	-11
0	2	0	1	|	0

(1.2) Eliminate x_2 .from the 3rd and 4th eqns. Pivot element is 1.6667. There is no division by zero problem. Still we will perform pivoting to reduce round-off errors. Interchange the 2nd and 3rd rows. Note that complete pivoting would interchange 2nd and 3rd columns.

6	1	-6	-5	|	6
0	-3.6667	4	4.3333	|	-11
0	1.6667	5	3.6667	|	-4
0	2	0	1	|	0

Eliminate x_2

6	1	-6	-5	|	6
0	-3.6667	4	4.3333	|	-11
0	0	6.8182	5.6364	|	-9.0001
0	0	2.1818	3.3636	|	-5.9999

(1.3) Eliminate x_3 . 6.8182 > 2.1818, therefore no pivoting is necessary.

6	1	-6	-5	|	6
0	-3.6667	4	4.3333	|	-11
0	0	6.8182	5.6364	|	-9.0001
0	0	0	1.5600	|	-3.1199

Partial Pivoting – Example

Step 2: Back substitution

x_4 = -3.1199 / 1.5600 = **-1.9999**

x_3 = [-9.0001 – 5.6364*(-1.9999)] / 6.8182 = **0.33325**

x_2 = [-11 – 4.3333*(-1.9999) – 4*0.33325] / -3.6667 = **1.0000**

x_1 = [6 – (-5)*(-1.9999) – (-6)*0.33325 – 1*1.0000] / 6 = **-0.50000**

Exact solution is x = $[-2 \quad 1/3 \quad 1 \quad -0.5]^T$. Use more than 5 sig. figs. to reduce round-off errors.

Scaling

- Normalize the equations so that the maximum coefficient in every row is equal to 1.0. That is, divide each row by the coefficient in that row with the maximum magnitude.

- It is advised to scale a system before calculating its determinant. This is especially important if we are calculating the determinant to see if the system is ill-conditioned or not.

- Consider the following systems

$$\begin{aligned} 2x_1 - 3x_2 &= 5 \\ 3.98x_1 - 6x_2 &= 7 \end{aligned} \qquad\qquad \begin{aligned} 20x_1 - 30x_2 &= 50 \\ 39.8x_1 - 60x_2 &= 70 \end{aligned}$$

- They are actually the same system. In the second one the equations are multiplied by 10.
- Determinant of the 1st system is 2(-6) – (-3)(3.98) = -0.06 , which is close to zero.
- Determinant of the 2nd system is 20(-60) – (-30)(39.8) = -6 , which is not that close to zero.
- So is this system ill-conditioned or not?
  - Scale this system. They are the same, use the 1st one.

$$\begin{aligned} -0.6667\ x_1 + x_2 &= -1.6667 \\ -0.6633\ x_1 + x_2 &= -1.1667 \end{aligned}$$

  - Now calculate the determinant as -0.6667*1 – 1*(-0.6633) = -0.0034

Scaled Partial Pivoting

- Scaling is also useful when some rows have coefficients that are large compared to those in other rows.

- Consider the following system

$$
\begin{aligned}
2x_1 + 100000x_2 &= 100000 \\
x_1 + x_2 &= 2
\end{aligned}
$$

Exact solution is (1.00002, 0.99998)

(a) If we solve this system with Gauss Elimination, no pivoting is necessary (2 is larger than 1). Use only 3 sig. figs to emphasize the round-off errors.

$$
\begin{aligned}
2.00*10^{0}\, x_1 + 1.00*10^{5}\, x_2 &= 1.00*10^{5} \\
- 5.00*10^{4}\, x_2 &= -5.00*10^{4}
\end{aligned}
\longrightarrow
$$

$x_1 = 0.00*10^{1}$ WRONG

0

$x_2 = 1.00*10^{1}$ OK

(b) First scale the system and than solve with Gauss Elimination.

$$
\begin{aligned}
2.00*10^{-5}\, x_1 + x_2 &= 1 \\
x_1 + x_2 &= 2
\end{aligned}
$$

This system now needs pivoting. Interchange the rows and solve.

$$
\begin{aligned}
2.00*10^{-5}\, x_1 + x_2 &= 1.00*10^{0} \\
x_1 + x_2 &= 2.00*10^{0}
\end{aligned}
\longrightarrow
\begin{aligned}
x_1 + x_2 &= 2 \\
x_2 &= 1
\end{aligned}
\longrightarrow
\begin{aligned}
x1 &= 1.00 \\
x2 &= 1.00
\end{aligned}
$$

x1 = 1.00 OK

x2 = 1.00 OK

Conclusion: Scaling showed that pivoting is necessary. But scaling itself is not necessary (pivot the original system and solve). Scaling also introduces additional round-off errors. Therefore use scaling to decide whether pivoting is necessary or not but than use the original coefficients.

Scaled Partial Pivoting

Example 15: Solve the following system using Gauss Elimination with scaled partial pivoting. Keep numbers as fractions of integers to eliminate round-off errors.

```
 3   -13     9     3   | -19
-6     4     1   -18   | -34
 6    -2     2     4   |  16
12    -8     6    10   |  26
```

Start by forming the scale vector. It has the largest coefficient (in magnitude) of each row.

```
SV = {13   18   6   12}
```

This scale vector will be updated if we interchange rows during pivoting.

Step 1: Forward Elimination

(1.1) Compare scaled coefficients 3/13, 6/18, 6/6, 12/12. Third one is the largest (actually fourth one is the same but we use the first occurance). Interchange rows 1 and 3.

```
 6    -2     2     4   |  16
-6     4     1   -18   | -34
 3   -13     9     3   | -19
12    -8     6    10   |  26
```

Update the scale vector `SV = {6 18 13 12}`

Scaled Partial Pivoting

(1.1) Compare scaled coefficients 3/13, 6/18, 6/6, 12/12. Third one is the largest (actually fourth one is the same but we use the first occurance). Interchange rows 1 and 3.

6	-2	2	4	|	16
-6	4	1	-18	|	-34
3	-13	9	3	|	-19
12	-8	6	10	|	26

Update the scale vector `SV = {6 18 13 12}`

Eliminate x_1.

Subtract (-6/6) times row 1 from row 2.

Subtract (3/6) times row 1 from row 3.

Subtract (12/6) times row 1 from row 4.

Resulting system is

6	-2	2	4	|	16
0	2	3	-14	|	-18
0	-12	8	1	|	-27
0	-4	2	2	|	-6

Scaled Partial Pivoting

(1.2) Compare scaled coefficients 2/18, 12/13, 4/12. Second one is the largest. Interchange rows 2 and 3.

$$\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -12 & 8 & 1 & -27 \\ 0 & 2 & 3 & -14 & -18 \\ 0 & -4 & 2 & 2 & -6 \end{array}$$

Update the scale vector `SV = {6 13 18 12}`

Eliminate x_2.

Subtract (2/(-12)) times row 2 from row 3.

Subtract ((-4)/(-12)) times row 2 from row 4.

Resulting system is

$$\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -12 & 8 & 1 & -27 \\ 0 & 0 & 13/3 & -83/6 & -45/2 \\ 0 & 0 & -2/3 & 5/3 & 3 \end{array}$$

Scaled Partial Pivoting

(1.3) Compare scaled coefficients (13/3)/18, (2/3)/12. First one is larger. No need for pivoting.

Scale vector remains the same `SV = {6 13 18 12}`

Eliminate x_3.

Subtract ((-2/3)/(13/3)) times row 3 from row 4.

Resulting system is

6	-2	2	4	|	16
0	-12	8	1	|	-27
0	0	13/3	-83/6	|	-45/2
0	0	0	-6/13	|	-6/13

Step 2: Back substitution

Equation 4 $\rightarrow$ $x_4 = 1$

Equation 3 $\rightarrow$ $x_3 = -2$

Equation 2 $\rightarrow$ $x_2 = 1$

Equation 1 $\rightarrow$ $x_1 = 3$

Gauss Elimination with Rounding

$$0.0004x_1 + 1.402x_2 = 1.406$$
$$0.4003x_1 - 1.502x_2 = 2.501$$

Original solution of the system is $x_1 = 10, x_2 = 1$

Picking the first of given equation as pivot equation , we have to multiply this equation by $m = 0.4003/0.0004 = 1001$ and subtract result from the second equation , obtaining

$$-1405x_2 = -1404 \rightarrow x_2 = 0.9993$$

From first equation we get $x_1 = 12.5$

The failure occurs because $|a_{11}|$ is small compared to $|a_{12}|$ so that a small round off error in x_2 led to a large error in x_1

Gauss Elimination with Rounding

$$0.0004x_1 + 1.402x_2 = 1.406$$
$$0.4003x_1 - 1.502x_2 = 2.501$$

Picking the second of the given equations as the pivot equation, we have to multiply this equation by $0.0004/0.4003 = 0.0009993$ and subtract the result from the first equation obtaining

$$1.404x_2 = 1.404$$

$x_2 = 1$ and $x_1 = 10$

Note $|a_{21}|$ is not very small compared to $|a_{22}|$ so that a small round off error in x_2 would not lead to a large error in x_1

Operation Count – Gauss Elimination

Important factors in judging the quality of a numerical method are

- Amount of storage
- Amount of time (= number of operations)

Consider Augmented Matrix of Ax = b, where $a_{in+1} = b_i$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n+1} \\ a_{21} & a_{22} & \cdots & a_{2n+1} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn+1} \end{bmatrix}$$

Operation Count – Gauss Elimination

In elimination procedure to get the rank we will make all elements below main diagonal zero.

Total number of multiplications and additions required to determine the rank by elimination procedure are

$$2.\sum_{k=1}^{n-1}(n-k)(n-k+1)=O(n^3)$$

Total number of divisions is

$$\sum_{k=1}^{n-1}(n-k)=O(n^2)$$

Operation Count – Gauss Elimination

In back substitution total number of additions, multiplications and divisions required are

$$\left(2.\sum_{k=1}^{n}(n-k)\right)+n=O(n^2)$$

If an operation takes 10^{-9} sec, then

Algorithm	$n = 1000$	$n = 10000$
Elimination	0.7 sec	11 min
Back substitution	0.001 sec	0.1 sec

LU Factorization

We write square matrix A as

$$A = LU$$

Doo Little's Method : L is lower triangular matrix diag(L) = 1, $l_{ii} = 1$ and U is upper triangular matrix

Crout's Method : U is upper triangular matrix with diag(U) = 1, $u_{ii} = 1$ and L is lower triangular matrix

Cholesky's Method: $U = L^T$

Benefits of LU Decomposition

$A = LU$, Thus, the system $Ax = B$, is

$LUx = B$

Let $Ux = y$, then

$Ly = B$

Algorithm :-

Step-I Solve $Ly = B$, to find y.

Step-II Then solve $Ux = y$ to find x

•Methods of LU Factorization

Doolittle Method: The Factors L, U are defined as

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

$$l_{ij} = 1, \quad \text{for } i = j$$
$$l_{ij} = 0, \quad \text{for } i < j$$
$$u_{ij} = 0, \quad \text{for } i > j$$

Crout's Method: The Factors L, U are defined as

$$L = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \quad U = \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$l_{ij} = 0, \quad \text{for } i < j$$
$$u_{ij} = 1, \quad \text{for } i = j$$
$$u_{ij} = 0, \quad \text{for } i > j$$

Crout's Method

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = LU$$

where $L = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$ $U = \begin{bmatrix} 1 & U_{12} & U_{13} \\ 0 & 1 & U_{23} \\ 0 & 0 & 1 \end{bmatrix}$

Crout's Method

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = LU$$

$$\text{where } L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 2 & 0 \\ 2 & 2 & 3 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Cholesky Method

- Cholesky decomposition is a technique that is designed for a system where the matrix $\mathbf{A}$ is symmetric and positive definite.
- The symmetric matrix refers to the matrix with the element of $a_{ij} = a_{ji}$ for all $i \neq j$. In other words, $\mathbf{A} = \mathbf{A}^T$.
- Cholesky decomposition method offer computational advantages because only half of the storage and computation time are required.
- In Cholesky method, a symmetric matrix $\mathbf{A}$ is decomposed as

$$\mathbf{A} = \mathbf{U}^T\mathbf{U}$$

Cholesky Method

- Decompose A such that $A = U^TU$. Hence, we may have $U^TUx = b$
- Set up and solve $U^Td = b$, where d can be obtained by using forward substitution
- Set up and solve $Ux = d$, where x can be obtained by using backward substitution

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2}$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}}{u_{ii}} \quad \textbf{for } j = i+1,\ldots,n$$

Computational Complexity

The LU decomposition is computed directly without solving simultaneous equations

- It is more economical to produce the LU Factorization
- This is followed by solving two simpler linear systems

1. To perform LU Factorization , we need about $\frac{n^3}{3}$ operations
2. To solve the Lower triangular system Ly=b we need $O(n^2)$ operations
3. To solve the Upper triangular system Ux=y we need $O(n^2)$ operations