Chapter 01

Solving linear systems
Let the system of linear equations be written in matrix form :

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Direct Methods

The methods that lead, in the absence of approximation errors and other errors, to the exact solution after a finite number of simple arithmetic operations are called **direct methods**.

Gaussian Elimination Method

The Gaussian elimination method is used to solve a system of linear equations by transforming the coefficient matrix into an upper-triangular form and then using back substitution to find the solution.

From the augmented matrix

$$\tilde{A}^{(1)} = [A|b] = [A^{(1)}|b^{(1)}]$$

we have :

$$\tilde{A}^{(1)} = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right]$$

Step 1 : Forward Elimination

The goal of forward elimination is to transform the matrix into an upper triangular form, that is, to make all elements below the main diagonal zero. For each pivot element $a_{kk}^{(k)}$, perform the following row operations for rows $i = k+1, \ldots, n$:

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad \text{then update :}$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} \cdot a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik} \cdot b_k^{(k)}$$

Repeat this process for $k = 1, 2, \ldots, n-1$ to obtain an upper triangular matrix U and a modified vector $\tilde{b}$.

Step 2 : Back Substitution

After forward elimination, the system takes the form :

$$Ux = \tilde{b}, \quad \text{where} \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Solve the system starting from the last equation :

$$\begin{aligned} x_n &= \frac{b'_n}{u_{nn}} \\ x_{n-1} &= \frac{1}{u_{n-1,n-1}} \left(b'_{n-1} - u_{n-1,n}x_n\right) \\ &\vdots \\ x_1 &= \frac{1}{u_{11}} \left(b'_1 - u_{12}x_2 - \cdots - u_{1n}x_n\right) \end{aligned}$$

Example

Solve the following system of linear equations using Gaussian elimination :

$$\begin{aligned} 3x_1 + 2x_2 - x_3 &= 1 \\ 6x_1 + 6x_2 + 2x_3 &= 12 \\ 3x_1 - 2x_2 + x_3 &= 11 \end{aligned} \tag{1}$$

Solution

We write the augmented matrix :

$$\left[\begin{array}{ccc|c} 3 & 2 & -1 & 1 \\ 6 & 6 & 2 & 12 \\ 3 & -2 & 1 & 11 \end{array}\right]$$

Eliminate the first column in R_2 and R_3 :

$$\begin{array}{l} R_2 \leftarrow R_2 - 2R_1 \\ R_3 \leftarrow R_3 - R_1 \end{array} \Rightarrow \left[\begin{array}{ccc|c} 3 & 2 & -1 & 1 \\ 0 & 2 & 4 & 10 \\ 0 & -4 & 2 & 10 \end{array}\right]$$

Eliminate the second column :

$$R_3 \leftarrow R_3 + 2R_2 \Rightarrow \left[\begin{array}{ccc|c} 3 & 2 & -1 & 1 \\ 0 & 2 & 4 & 10 \\ 0 & 0 & 10 & 30 \end{array}\right]$$

We get

$$\begin{aligned} 10x_3 &= 30 \\ 2x_2 + 4x_3 &= 10 \\ 3x_1 + 2x_2 - x_3 &= 1 \end{aligned} \tag{2}$$

Thus, the solution is :

$$\boxed{x_1 = 2, \quad x_2 = -1, \quad x_3 = 3}$$

Example

Solve the following system of linear equations using Gaussian elimination :

$$\begin{aligned} 2x_2 + x_3 &= 1 \\ 5x_1 + 3x_2 + x_3 &= 5 \\ 2x_1 - x_2 + 2x_3 &= 9 \end{aligned}$$

Solution

We write the augmented matrix :

$$\left[\begin{array}{ccc|c} 0 & 2 & 1 & 1 \\ 5 & 3 & 1 & 5 \\ 2 & -1 & 2 & 9 \end{array}\right]$$

Step 1 : Swap the first and second rows to move a nonzero element into the first column :

$$R_1 \leftrightarrow R_2 \Rightarrow \left[\begin{array}{ccc|c} 5 & 3 & 1 & 5 \\ 0 & 2 & 1 & 1 \\ 2 & -1 & 2 & 9 \end{array}\right]$$

Step 2 : Eliminate the first column in R_2 and R_3 :

$$R_3 \leftarrow R_3 - \frac{2}{5}R_1 \quad \Rightarrow \left[\begin{array}{ccc|c} 5 & 3 & 1 & 5 \\ 0 & 2 & 1 & 1 \\ 0 & -\frac{11}{5} & \frac{8}{5} & 7 \end{array}\right]$$

Step 3 : Eliminate the second column in R_1 and R_3 :

$$R_3 \leftarrow R_3 + \frac{11}{10}R_2 \quad \Rightarrow \left[\begin{array}{ccc|c} 5 & 3 & 1 & 5 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{27}{10} & \frac{81}{10} \end{array}\right]$$

We get

$$\begin{aligned} \frac{27}{10}x_3 &= \frac{81}{10} \\ 2x_2 + x_3 &= 1 \\ 5x_1 + 3x_2 + x_3 &= 5 \end{aligned} \tag{3}$$

Thus, the solution is :

$$x_1 = 1, \quad x_2 = -1, \quad x_3 = 3$$

LU Decomposition and Factorization Methods

LU Decomposition Overview

LU decomposition is a method used to factorize a given square matrix A into two matrices L and U, where :
– L is a lower triangular matrix (with all elements above the diagonal being zero).
– U is an upper triangular matrix (with all elements below the diagonal being zero).
The factorization is expressed as :

$$A = LU$$

LU decomposition is useful for solving systems of linear equations $A\mathbf{x} = \mathbf{b}$ efficiently by first solving $L\mathbf{y} = \mathbf{b}$ (using forward substitution) and then solving $U\mathbf{x} = \mathbf{y}$ (using backward substitution).

Doolittle's Method

In **Doolittle's Method**, we decompose A into :
– L, a lower triangular matrix with ones on the diagonal.
– U, an upper triangular matrix.
We want to factor A such that :

$$A = LU$$

Where :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

and

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Steps for Doolittle's Method

1. Write the matrix A in terms of the unknowns of L and U.
2. Compare the elements of the product LU with the elements of A.
3. Solve for the elements of L and U one by one.

For a 3x3 matrix A :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

We equate the elements of A to the product of L and U :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

After solving, you can find L and U.

Example

Solve the following system of linear equations using Doolittle's Method :

$$\begin{aligned} x_1 + x_2 - x_3 &= 1 \\ x_1 + 2x_2 - 2x_3 &= 1 \\ -2x_1 + x_2 + x_3 &= 1 \end{aligned}$$

Solution

We start by writing the system in matrix form $A\mathbf{x} = \mathbf{b}$, where :

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

We will decompose A into L and U, where L is a lower triangular matrix with ones on the diagonal and U is an upper triangular matrix.

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

The matrix product LU should be equal to A. Therefore, we have the following system of equations :

$$A = LU = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{bmatrix}$$

Expanding the product of L and U, we get the following equations :

$$\begin{aligned} u_{11} &= 1, \\ u_{12} &= 1, \\ u_{13} &= -1, \\ l_{21} &= 1, \quad u_{22} = 1, \quad u_{23} = -1, \\ l_{31} &= -2, \quad l_{32} = -3, \quad u_{33} = -4. \end{aligned}$$

After solving for the remaining elements in L and U, we get :

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

Next, we solve for $\mathbf{x}$ by performing forward substitution with L and backward substitution with U.

First, solve $L\mathbf{y} = \mathbf{b}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & -3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

This gives the system :

$$\begin{aligned} y_1 &= 1, \\ y_1 + y_2 &= 1 \quad \Rightarrow \quad 1 + y_2 = 1 \quad \Rightarrow \quad y_2 = 0, \\ -2y_1 - 3y_2 + y_3 &= 1 \quad \Rightarrow \quad -2(1) - 3(0) + y_3 = 1 \quad \Rightarrow \quad y_3 = 3. \end{aligned}$$

Thus, $\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$.

Now, solve $U\mathbf{x} = \mathbf{y}$:

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

This gives the system :

$$-4x_3 = 3 \quad \Rightarrow \quad x_3 = -\frac{3}{4} = -0.75.$$

$$x_2 + (0.75) = 0 \quad \Rightarrow \quad x_2 = -0.75.$$

$$x_1 - 0.75 + 0.75 = 1 \quad \Rightarrow \quad x_1 = 1.$$

Thus, the solution is :

$$\boxed{x_1 = 1, \quad x_2 = -0.75, \quad x_3 = -0.75}$$

Steps for Crout's Method

1. Write the matrix A in terms of the unknowns of L and U.
2. Use the relations derived from multiplying L and U to solve for the elements of L and U.

For a 3x3 matrix A :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The product LU will be :

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

This will give the system of equations from which the values of L and U can be determined.

example

Solve the following system of linear equations using Crout's Method :

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 4x_1 + 3x_2 - x_3 &= 6 \\ 3x_1 + 5x_2 + 3x_3 &= 4 \end{aligned}$$

Solution

We start by writing the system in matrix form $A\mathbf{x} = \mathbf{b}$, where :

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 6 \\ 4 \end{bmatrix}$$

We will decompose A into L and U, where L is a lower triangular matrix and U is an upper triangular matrix.

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

The matrix product LU should be equal to A. Therefore, we have the following system of equations :

$$A = LU = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$$

Expanding the product of L and U, we get the following equations :

$$
\begin{aligned}
l_{11} &= 1 \\
l_{11}u_{12} &= 1 \Rightarrow u_{12} = \frac{1}{l_{11}} = 1 \\
l_{11}u_{13} &= 1 \Rightarrow u_{13} = \frac{1}{l_{11}} = 1 \\
l_{21} &= 4 \\
l_{21}u_{12} + l_{22} &= 3 \Rightarrow 4 \cdot 1 + l_{22} = 3 \Rightarrow l_{22} = -1 \\
l_{21}u_{13} + l_{22}u_{23} &= -1 \Rightarrow 4 \cdot 1 + (-1) \cdot u_{23} = -1 \Rightarrow u_{23} = 5 \\
l_{31} &= 3 \\
l_{31}u_{12} + l_{32} &= 5 \Rightarrow 3 \cdot 1 + l_{32} = 5 \Rightarrow l_{32} = 2 \\
l_{31}u_{13} + l_{32}u_{23} + l_{33} &= 3 \Rightarrow 3 \cdot 1 + 2 \cdot 5 + l_{33} = 3 \Rightarrow l_{33} = -10
\end{aligned}
$$

After solving for the remaining elements in L and U, we get :

$$
L = \begin{bmatrix} 1 & 0 & 0 \\ 4 & -1 & 0 \\ 3 & 2 & -10 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}
$$

Next, we solve for $\mathbf{x}$ by performing forward substitution with L and backward substitution with U.

First, solve $L\mathbf{y} = \mathbf{b}$:

$$
\begin{bmatrix} 1 & 0 & 0 \\ 4 & -1 & 0 \\ 3 & 2 & -10 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 4 \end{bmatrix}
$$

This gives the system :

$$
\begin{aligned}
y_1 &= 1, \\
4y_1 - y_2 &= 6 \quad \Rightarrow \quad 4(1) - y_2 = 6 \quad \Rightarrow \quad y_2 = -2, \\
3y_1 + 2y_2 - 10y_3 &= 4 \quad \Rightarrow \quad 3(1) + 2(-2) - 10y_3 = 4 \quad \Rightarrow \quad y_3 = -\frac{1}{2}.
\end{aligned}
$$

Thus, $\mathbf{y} = \begin{bmatrix} 1 \\ -2 \\ -\frac{1}{2} \end{bmatrix}$.

Now, solve $U\mathbf{x} = \mathbf{y}$:

$$
\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ -\frac{1}{2} \end{bmatrix}
$$

This gives the system :

$$x_3 = -\frac{1}{2},$$
$$x_2 + 5x_3 = -2 \quad \Rightarrow \quad x_2 + 5(-\frac{1}{2}) = -2 \quad \Rightarrow \quad x_2 = \frac{1}{2},$$
$$x_1 + x_2 + x_3 = 1 \quad \Rightarrow \quad x_1 + \frac{1}{2} - \frac{1}{2} = 1 \quad \Rightarrow \quad x_1 = 1.$$

Thus, the solution is :

$$x_1 = 1, \quad x_2 = \frac{1}{2}, \quad x_3 = -\frac{1}{2}.$$

Cholesky Decomposition

Definition : Positive Definite Matrix

Definition 6.22. A matrix $A \in \mathbb{R}^{n\times n}$ is called **positive definite** if it is symmetric and satisfies

$$x^T Ax > 0 \quad \text{for all } x \in \mathbb{R}^n,\ x \neq 0.$$

Example : Verifying Positive Definiteness

Let

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$, $x \neq 0$. Then :

$$x^T Ax = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2x_1^2 - 2x_1x_2 + 2x_2^2$$

This simplifies to :

$$x^T Ax = 2(x_1^2 - x_1x_2 + x_2^2)$$

Now note that :

$$x_1^2 - x_1x_2 + x_2^2 = \frac{1}{2}\left[(x_1 - x_2)^2 + x_1^2 + x_2^2\right] > 0 \quad \text{for all } x \neq 0$$

Therefore :

$$x^T Ax > 0 \quad \text{for all } x \neq 0$$

Conclusion : A is symmetric and $x^T Ax > 0$, so A is **positive definite**.

Theorem : Cholesky Decomposition

If $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, then there exists a unique lower triangular matrix L with positive diagonal entries such that

$$A = LL^T.$$

Algorithm : Cholesky Decomposition

Given : A symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ **Goal :** Compute a lower triangular matrix L such that $A = LL^T$

- **Step 1 :** For $i = 1$ to n
  - **Step 1.1 :** Compute diagonal entries :

$$L_{ii} = \sqrt{A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2}$$

  - **Step 1.2 :** For $j = i + 1$ to n, compute :

$$L_{ji} = \frac{1}{L_{ii}} \left(A_{ji} - \sum_{k=1}^{i-1} L_{jk} L_{ik} \right)$$

- **Step 2 :** All elements above the diagonal are zero.

Detailed Example : Solve $Ax = b$ Using Cholesky Method

Given :

$$A = \begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix}, \quad b = \begin{bmatrix} 14 \\ -101 \\ 155 \end{bmatrix}$$

Step 1 : Verify Symmetry

$$A = A^T \quad \text{so } A \text{ is symmetric.}$$

Step 2 : Compute Cholesky Factor L (such that $A = LL^T$)

We compute L step by step :

$$\textbf{Compute } L_{11} = \sqrt{4} = 2$$

$$\textbf{Compute } L_{21} = \frac{1}{2}(2) = 1, \quad L_{31} = \frac{1}{2}(14) = 7$$

$$\textbf{Compute } L_{22} = \sqrt{17 - 1^2} = \sqrt{16} = 4$$

$$\textbf{Compute } L_{32} = \frac{1}{4}(-5 - (7)(1)) = \frac{-12}{4} = -3$$

$$\textbf{Compute } L_{33} = \sqrt{83 - 7^2 - (-3)^2} = \sqrt{83 - 49 - 9} = \sqrt{25} = 5$$

Thus, the matrix L is :

$$L = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{bmatrix}$$

Step 3 : Solve $Ly = b$ (Forward Substitution)

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ -101 \\ 155 \end{bmatrix}$$

$$\Rightarrow \begin{cases} 2y_1 = 14 \Rightarrow y_1 = 7 \\ 1 \cdot 7 + 4y_2 = -101 \Rightarrow y_2 = \frac{-108}{4} = -27 \\ 7 \cdot 7 - 3 \cdot (-27) + 5y_3 = 155 \Rightarrow 49 + 81 + 5y_3 = 155 \Rightarrow y_3 = \frac{25}{5} = 5 \end{cases}$$

$$\Rightarrow y = \begin{bmatrix} 7 \\ -27 \\ 5 \end{bmatrix}$$

Step 4 : Solve $L^T x = y$ (Backward Substitution)

$$\begin{bmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -27 \\ 5 \end{bmatrix}$$

$$\Rightarrow \begin{cases} 5x_3 = 5 \Rightarrow x_3 = 1 \\ 4x_2 - 3 \cdot 1 = -27 \Rightarrow x_2 = \frac{-24}{4} = -6 \\ 2x_1 + 1 \cdot (-6) + 7 \cdot 1 = 7 \Rightarrow 2x_1 = 6 \Rightarrow x_1 = 3 \end{cases}$$

$$\Rightarrow x = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}$$

Iterative Methods for Solving Linear Systems

Iterative methods are numerical techniques used to find approximate solutions to systems of linear equations :

$$Ax = b$$

They are particularly useful in the following situations :
- When the number of equations is large (e.g., hundreds or thousands).
- When the coefficient matrix A is sparse, i.e., it contains many zero entries.
- When direct methods like Gaussian elimination or LU/Cholesky decomposition are computationally expensive or impractical.

Unlike direct methods, iterative methods generate a sequence of approximations :

$$x^{(0)}, x^{(1)}, x^{(2)}, \dots$$

which ideally converge to the exact solution x.

Common Iterative Methods

- **Jacobi Method**
- **Gauss-Seidel Method**
- **Successive Over-Relaxation (SOR) Method**

Each method constructs the next approximation using the previous one :

$$x^{(k+1)} = Bx^{(k)} + c$$

where B and c depend on how the matrix A is decomposed.

In the next sections, we define and derive each method, present convergence conditions, and solve examples.

Matrix Decomposition : $A = D - L - U$

Let A be an $n \times n$ matrix. We can decompose it as :

$$A = D - L - U$$

where :
- D is the diagonal matrix containing the diagonal elements of A,
- $-L$ is the strictly lower triangular part of A,
- $-U$ is the strictly upper triangular part of A.

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \quad L = -\begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}, \quad U = -\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Therefore :

$$A = D - L - U$$

This decomposition is especially useful when deriving the Jacobi and Gauss-Seidel iterative formulas :

– Jacobi : $B_J = D^{-1}(L + U)$, $c = D^{-1}b$
– Gauss-Seidel : $B_{GS} = (D - L)^{-1}U$, $c = (D - L)^{-1}b$

Fundamental norms of matrix spaces

To analyze convergence of iterative methods, we make use of **vector norms** and **induced matrix norms**.

Vector Norms

Let $x \in \mathbb{R}^n$. Common vector norms include :
– 1-norm :

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

– 2-norm (Euclidean norm) :

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}$$

– Infinity norm :

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

Matrix Norms

Let $A \in \mathbb{R}^{n \times n}$. The **induced matrix norm** $\|A\|$ is defined from a vector norm :

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

Important induced norms :
– 1-norm :

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}| \quad \text{(maximum column sum)}$$

– Infinity norm :

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \quad \text{(maximum row sum)}$$

Jacobi Method

The Jacobi method is an iterative technique for solving linear systems of the form $Ax = b$, where A is decomposed as :

$$A = D - L - U$$

so the iteration formula becomes :

$$x^{(k+1)} = D^{-1}(L + U)x^{(k)} + D^{-1}b$$

Equivalently, for each component $i = 1, 2, \dots, n$:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

Jacobi Algorithm

1. Start with an initial guess $x^{(0)}$
2. For $k = 0, 1, 2, \dots$, compute :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n$$

3. Repeat until $\|x^{(k+1)} - x^{(k)}\|$ is smaller than a desired tolerance.

Example : Solve Using Jacobi Method

Solve the system :

$$A = \begin{bmatrix} 4 & 1 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 7 \\ 5 \end{bmatrix}$$

Initial guess :

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Iteration formula :

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}(4 - x_2^{(k)} - 2x_3^{(k)}) \\ x_2^{(k+1)} &= \frac{1}{5}(7 - x_1^{(k)} - x_3^{(k)}) \\ x_3^{(k+1)} &= \frac{1}{3}(5 - 2x_1^{(k)} - x_2^{(k)}) \end{aligned}$$

First iteration :

$$x^{(1)} = \begin{bmatrix} 1 \\ 1.4 \\ 1.6667 \end{bmatrix}$$

Second iteration :

$$\begin{aligned} x_1^{(2)} &= \frac{1}{4}(4 - 1.4 - 2 \cdot 1.6667) = -0.1833 \\ x_2^{(2)} &= \frac{1}{5}(7 - 1 - 1.6667) = 0.8667 \\ x_3^{(2)} &= \frac{1}{3}(5 - 2 \cdot 1 - 1.4) = 0.5333 \end{aligned}$$

Continue until the change is below tolerance.

Gauss-Seidel Method

The Gauss-Seidel method improves upon the Jacobi method by using the most recent values of the solution as soon as they are available.

We again start from the decomposition :

$$A = D - L - U$$

the iteration formula becomes :

$$x^{(k+1)} = (D - L)^{-1} U x^{(k)} + (D - L)^{-1} b$$

Component-wise, the update formula is :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$$

Notice that unlike the Jacobi method, we use the latest available values during each iteration.

Gauss-Seidel Algorithm

1. Start with an initial guess $x^{(0)}$
2. For $k = 0, 1, 2, \ldots$, update each component :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$$

3. Repeat until convergence.

Example : Solve Using Gauss-Seidel Method

Solve the same system :

$$A = \begin{bmatrix} 4 & 1 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 7 \\ 5 \end{bmatrix}$$

Initial guess :

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Iteration formula :

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}(4 - x_2^{(k)} - 2x_3^{(k)}) \\ x_2^{(k+1)} &= \frac{1}{5}(7 - x_1^{(k+1)} - x_3^{(k)}) \\ x_3^{(k+1)} &= \frac{1}{3}(5 - 2x_1^{(k+1)} - x_2^{(k+1)}) \end{aligned}$$

First iteration :

$$x_1^{(1)} = \frac{1}{4}(4 - 0 - 0) = 1$$
$$x_2^{(1)} = \frac{1}{5}(7 - 1 - 0) = 1.2$$
$$x_3^{(1)} = \frac{1}{3}(5 - 2 \cdot 1 - 1.2) = 0.6$$

Second iteration :

$$x_1^{(2)} = \frac{1}{4}(4 - 1.2 - 2 \cdot 0.6) = 0.4$$
$$x_2^{(2)} = \frac{1}{5}(7 - 0.4 - 0.6) = 1.2$$
$$x_3^{(2)} = \frac{1}{3}(5 - 2 \cdot 0.4 - 1.2) = 1$$

Repeat until the solution converges.

Successive Over-Relaxation (SOR) Method

The Successive Over-Relaxation (SOR) method is an improved iterative technique that accelerates the convergence of the Gauss-Seidel method by introducing a relaxation parameter ω, where $0 < \omega < 2$.

Starting from the decomposition :

$$A = D - L - U$$

the SOR method uses :

$$M = \frac{1}{\omega}D - L, \quad N = \left(\frac{1-\omega}{\omega}\right) D + U$$

The general iteration becomes :

$$x^{(k+1)} = (D - \omega L)^{-1} \left[\omega b + (\omega U + (1-\omega)D)x^{(k)}\right]$$

Component-wise SOR Iteration

For each component $i = 1, 2, \ldots, n$:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right)$$

SOR Algorithm

1. Choose an initial guess $x^{(0)}$
2. Choose a relaxation factor $\omega \in (0, 2)$
3. For $k = 0, 1, 2, \ldots$, update :

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right)$$

4. Repeat until convergence.

Example : Solve Using SOR Method

Solve the same system :

$$A = \begin{bmatrix} 4 & 1 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 7 \\ 5 \end{bmatrix}$$

Initial guess :

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \omega = 1.25$$

Iteration formulas :

$$x_1^{(k+1)} = (1-\omega)x_1^{(k)} + \frac{\omega}{4}(4 - x_2^{(k)} - 2x_3^{(k)})$$
$$x_2^{(k+1)} = (1-\omega)x_2^{(k)} + \frac{\omega}{5}(7 - x_1^{(k+1)} - x_3^{(k)})$$
$$x_3^{(k+1)} = (1-\omega)x_3^{(k)} + \frac{\omega}{3}(5 - 2x_1^{(k+1)} - x_2^{(k+1)})$$

First iteration with $\omega = 1.25$:

$$x_1^{(1)} = 1.25 \cdot \frac{1}{4}(4 - 0 - 0) = 1.25$$
$$x_2^{(1)} = 1.25 \cdot \frac{1}{5}(7 - 1.25 - 0) = 1.4375$$
$$x_3^{(1)} = 1.25 \cdot \frac{1}{3}(5 - 2 \cdot 1.25 - 1.4375) = 0.4427$$

Continue the process until convergence.

Note on ω

- If $\omega = 1$: the method reduces to Gauss-Seidel - If $\omega < 1$: under-relaxation (slow convergence but more stable) - If $\omega > 1$: over-relaxation (faster convergence, but may become unstable if ω is too large)

Convergence Analysis of Iterative Methods

Convergence Theorem Using Matrix Norm

Consider the iterative method :

$$x^{(k+1)} = Bx^{(k)} + c$$

Let $\|\cdot\|$ be any matrix norm. Then :

theorem

If $\|B\| < 1$, then the iterative method converges for any initial guess $x^{(0)}$.

Idea of the Proof. Suppose x is the true solution. Then :

$$e^{(k)} = x^{(k)} - x = B^k(x^{(0)} - x)$$

Taking norms :

$$\|e^{(k)}\| \leq \|B^k\| \cdot \|x^{(0)} - x\| \leq \|B\|^k \cdot \|x^{(0)} - x\|$$

Since $\|B\| < 1$, we get $\|e^{(k)}\| \to 0$ as $k \to \infty$, so the method converges. □

Norm vs Spectral Radius

$$\rho(B) \leq \|B\| \quad \text{for any induced matrix norm}$$

Hence, $\|B\| < 1 \Rightarrow \rho(B) < 1$, so matrix norm condition is stronger than necessary, but easier to verify.

An iterative method of the form :

$$x^{(k+1)} = Bx^{(k)} + c$$

converges to the exact solution x for any initial guess $x^{(0)}$ if and only if the **spectral radius** of the iteration matrix B satisfies :

$$\rho(B) < 1$$

where the spectral radius $\rho(B)$ is the maximum absolute value of the eigenvalues of B :

$$\rho(B) = \max_{1 \leq i \leq n} |\lambda_i(B)|$$

Sufficient Conditions for Convergence

- **Diagonal Dominance :** If the matrix A is strictly or irreducibly diagonally dominant, then both Jacobi and Gauss-Seidel methods converge.
  A matrix A is **strictly diagonally dominant** if :

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \text{for all } i$$

- **Symmetric Positive Definite (SPD) Matrix :** If A is symmetric positive definite, then :
  - Gauss-Seidel always converges
  - SOR converges for any $0 < \omega < 2$

Convergence Behavior

- The closer $\rho(B)$ is to 0, the faster the convergence.
- If $\rho(B) \geq 1$, the iteration diverges.
- For the SOR method, the optimal ω (i.e., minimizing the number of iterations) depends on the matrix and is usually found experimentally.

Comparison of Convergence

Method	Converges If
Jacobi	$\rho(B_J) < 1$ or A is strictly diagonally dominant
Gauss-Seidel	$\rho(B_{GS}) < 1$ or A is SPD
SOR	$\rho(B_{SOR}) < 1$ and $0 < \omega < 2$

Example : Convergence Check

Consider the linear system :

$$A = \begin{bmatrix} 10 & -1 & 2 \\ -1 & 11 & -1 \\ 2 & -1 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 25 \\ -11 \end{bmatrix}$$

Check for Diagonal Dominance :

$$|a_{11}| = 10 > |-1| + |2| = 3$$
$$|a_{22}| = 11 > |-1| + |-1| = 2$$
$$|a_{33}| = 10 > |2| + |-1| = 3$$

The matrix A is strictly diagonally dominant, so both Jacobi and Gauss-Seidel methods are guaranteed to converge.

Compute Iteration Matrix for Jacobi :

We decompose $A = D - L - U$, where :

$$D = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 11 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad L + U = \begin{bmatrix} 0 & 1 & -2 \\ 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}$$

The Jacobi iteration matrix is :

$$B_J = D^{-1}(L+U) = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{2}{10} \\ \frac{1}{11} & 0 & \frac{1}{11} \\ -\frac{2}{10} & \frac{1}{10} & 0 \end{bmatrix}$$

$$\rho(B_J) \approx 0.3 < 1$$

Therefore, the Jacobi method converges.

Conclusion : Since A is diagonally dominant and $\rho(B_J) < 1$, both Jacobi and Gauss-Seidel methods will converge for this system.