LECTURE 3

SOLUTION OF A SYSTEM OF LINEAR ALGEBRAIC EQUATIONS

1 Statement of the problem 2
2 Direct methods 2
  2.1 Cramer's method 2
  2.2 Gaussian method 5
  2.3 Gauss – Jordan method 9
3 Iterative methods 13
  3.1 Jacobi's iteration method 13
  3.2 Gauss – Seidel method 15
  3.3 Relaxation method 17
4 Tasks 17

1 Statement of the problem

Systems of linear equations are used to model real-world problems such as circuit analysis, optimization problems, and resource allocation. Understanding how to represent and solve these systems is crucial for both theoretical studies and practical applications in various domains.

A system of linear equations consists of n equations with n variables, which can be expressed in the following form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots&\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n. \end{aligned} \tag{1.1}$$

The system of equations (1.1) can be succinctly expressed in matrix form as

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1.2}$$

where $\mathbf{A} = a_{ij}$ is a square matrix of order n, $\mathbf{b} = b_i$ is the column vector of constants, $\mathbf{x} = x_i$ is the column vector of variables and they are roots of the system of equations (1.1) and (1.2) (where $i, j = \overline{1, n}$)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{1.3}$$

Common methods for solving systems of linear equations include *direct and iterative methods.* Direct methods, such as Gauss, Cramer and Gauss-Gordan methods, aim to find the exact solution in a finite number of steps by transforming the system into a simpler form. These methods are efficient for small to medium-sized systems but can become computationally intensive for larger ones. Iterative methods, including the Jacobi, Gauss-Seidel and Relaxation methods, generate a sequence of approximations that converge to the exact solution. They are particularly useful for large or sparse systems where direct methods may be impractical. The compact matrix representation of these systems facilitates computational implementations in programming environments and numerical analysis software, allowing for efficient handling of complex problems using optimized algorithms.

2 Direct methods

2.1 Cramer's method

Cramer's method provides an explicit formula for the solution of such systems (1.1)-(1.3), provided that the coefficient matrix is non-singular (i.e., has a non-zero determinant). Cramer's rule states that each variable x_i can be calculated using the formula

$$x_i = \frac{D_i}{D}, \quad i = \overline{1, n} \tag{1.4}$$

where D is the determinant of the coefficient matrix $\mathbf{A}$, D_i is the determinant of the matrix obtained by replacing the i^{th} column of $\mathbf{A}$ with the vector $\mathbf{b}$.

Below, Algorithm 1 presents Cramer's method for solving a system of linear algebraic equations.

Algorithm 1 Cramer's method for solving linear systems

Input:
  A ▷ $n \times n$ coefficient matrix
  $\mathbf{b}$ ▷ Right-hand side vector of size n

Initialize:
  $D \leftarrow \det(A)$ ▷ Compute determinant of A
  $\mathbf{x} \leftarrow$ zero vector of size n ▷ Solution vector

if $D = 0$ **then**
  Error "System is either singular or not compatible"
  Exit
end if

for $i \leftarrow 1$ to n **do**
  $A_i \leftarrow A$ ▷ Copy of original matrix
  Replace i-th column of A_i with $\mathbf{b}$ ▷ Create modified matrix
  $D_i \leftarrow \det(A_i)$ ▷ Compute determinant of modified matrix
  $x_i \leftarrow D_i/D$ ▷ Compute i-th component of solution
end for

Output:
  $\mathbf{x}$ ▷ Solution vector

Example: Cramer's method

Let's consider the following system of two linear equations in the variables x_1 and x_2 as

$$\begin{cases} 2x_1 + 3x_2 = 5, \\ 4x_1 + x_2 = 11. \end{cases}$$

The coefficient matrix $\mathbf{A}$ and the constant vector $\mathbf{b}$ are given by

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}.$$

Step 1. Compute the determinant D of the coefficient matrix $\mathbf{A}$ is

$$D = \det(\mathbf{A}) = (2)(1) - (3)(4) = 2 - 12 = -10.$$

Since $D \neq 0$, the system has a unique solution.

Step 2. Apply Cramer's rule to find the solution, we compute the determinants of the modified matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, where $\mathbf{A}_1$ replaces the first column of $\mathbf{A}$ with $\mathbf{b}$, $\mathbf{A}_2$ replaces the second column of $\mathbf{A}$ with $\mathbf{b}$.

1. Determinant D_1 (for x_1):

$$D_1 = \det(\mathbf{A}_1) = \begin{vmatrix} 5 & 3 \\ 11 & 1 \end{vmatrix} = (5)(1) - (3)(11) = 5 - 33 = -28.$$

2. Determinant D_2 (for x_2):

$$D_2 = \det(\mathbf{A}_2) = \begin{vmatrix} 2 & 5 \\ 4 & 11 \end{vmatrix} = (2)(11) - (5)(4) = 22 - 20 = 2.$$

Step 3. Solve for x_1 and x_2 using Cramer's rule, the solutions are

$$x_1 = \frac{D_1}{D} = \frac{-28}{-10} = 2.8, \quad x_2 = \frac{D_2}{D} = \frac{2}{-10} = -0.2.$$

The system has the unique solution as

$$\boxed{x_1 = 2.8}, \quad \boxed{x_2 = -0.2}.$$

Cramer's Rule is computationally efficient for small systems (e.g., 2×2 or 3×3). For larger systems, it becomes impractical due to the high computational cost of evaluating determinants.

■

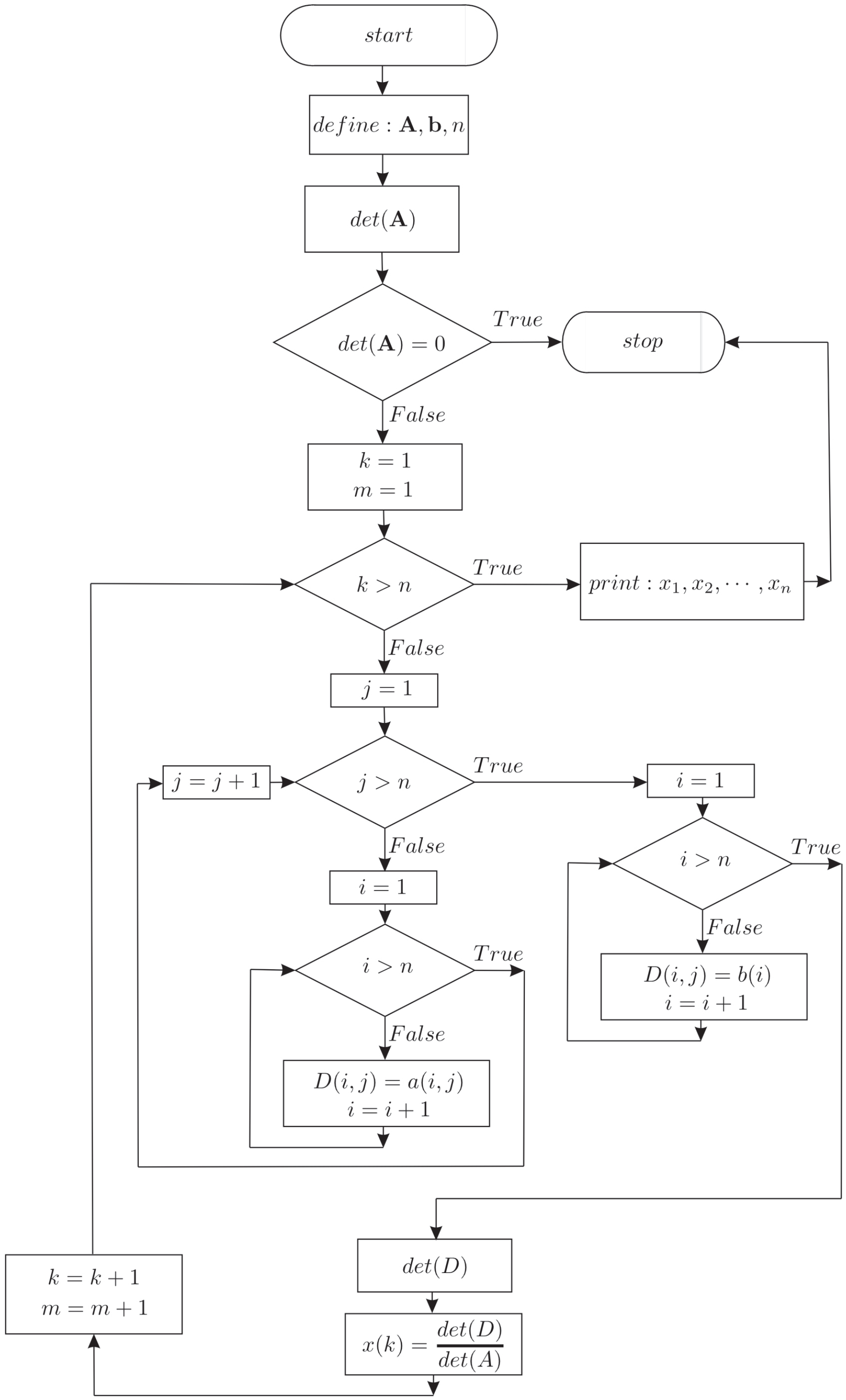

Figure 1.1: Block diagram of Cramer's method

2.2 Gaussian method

The Gaussian method (Gauss elimination method by pivoting) is a fundamental algorithm in linear algebra used to solve systems of linear equations (1.1)-(1.3). It consists of a sequence of operations on the augmented matrix that transforms it into an upper triangular form, which can then be easily solved using back substitution.

The Gauss elimination method involves the following key steps:

Step 1. Formulate the augmented matrix. Start by writing the system of linear equations in matrix form $\mathbf{Ax} = \mathbf{b}$. The augmented matrix combines both $\mathbf{A}$ and $\mathbf{b}$.

Step 2. Forward elimination. This phase aims to convert the augmented matrix into an upper triangular form. The goal is to create zeros below the pivot elements (the leading coefficients in each row). This is achieved through a series of elementary row operations as following

– Row swapping. Interchanging two rows if a diagonal element is zero.

– Row scaling. Multiplying a row by a nonzero constant factor.

– Row subtraction. Adding or subtracting a multiple of one row to/from another row.

Here, the variable x is eliminated from all equations except the first equation. To do this, we multiply the coefficients of the remaining equations by the factor

$$c_i^{(1)} = \frac{a_{i1}}{a_{11}}, \quad i = \overline{2, n} \tag{1.5}$$

and subtract them from the first equation ($j = 1$):

$$a_{ij}^{(1)} = a_{ij} - c_i^{(1)} a_{1j}, \quad b_i^{(1)} = b_i - c_i^{(1)} b_1 \quad i = \overline{2, n}, \quad j = \overline{1, n}. \tag{1.6}$$

Then we get a system of transformed algebraic equations,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ &\vdots \\ a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n &= b_n^{(1)}, \end{aligned} \tag{1.7}$$

where $a_{i1} = 0$ ($i = \overline{2, n}$) for $n - 1$ equations except the first equation.

Applying the above operations to the equations other than the first and second equations in the system of equations (1.7), we eliminate the variable x_2 from the equations starting from the third to the last n-th equation. In the same way, we can eliminate the variables x_3,.., x_{n-1}:

$$\begin{gathered} c_i^{(k)} = \frac{a_{ik}^{k-1}}{a_{kk}^{k-1}}, \quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - c_i^{(k)} a_{kj}^{(k-1)}, \\ b_i^{(k)} = b_i^{(k-1)} - c_i^{(k)} b_k^{(k-1)}, \quad k = \overline{2, n-1}, \quad i = \overline{k+1, n}. \end{gathered} \tag{1.8}$$

Then we obtain a system of equations reduced to triangular form as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ &\vdots \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}. \end{aligned} \tag{1.9}$$

Step 3. Back substitution. Once the matrix is in upper triangular form, back substitution is used to solve for the unknowns starting from the last equation

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \tag{1.10}$$

and moving upwards

$$x_k = \frac{1}{a_{kk}^{(k-1)}} \left(b_k^{(k-1)} - a_{k,k+1}^{(k-1)} x_{k+1} - a_{k,k+2}^{(k-1)} x_{k+2} - \cdots - a_{kn}^{(k-1)} x_n \right), \quad k = \overline{n-1, 1}. \tag{1.11}$$

The Gaussian elimination algorithm is systematically summarized in Algorithm 2, while its forward elimination and backward substitution components are presented separately in the block diagrams Figure 1.2 and Figure 1.3, respectively.

Example: Gauss method

Let's consider the following system of three linear equations with three unknowns (x_1, x_2, x_3)

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

Step 1. We represent the system in augmented matrix form $[A|\mathbf{b}]$:

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

The vertical bar separates the coefficient matrix A from the constant terms $\mathbf{b}$. Each row corresponds to one equation in the system.

Step 2. Perform forward elimination. To eliminate x_1 from 3rd row ($i = 3$ in eqns.(1.5) and (1.6)), perform $c_3^{(1)} = a_{31}/a_{11} = 3/1 = 3$, $a_{3j}^{(1)} = a_{3j} - c_3^{(1)} a_{1j}$ for $j = \overline{1, 3}$ and $b_3^{(1)} = b_3 - c_3^{(1)} b_3$, we get the updated matrix

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

Next, to eliminate x_2, perform $c_3^{(2)} = a_{32}^{(1)}/a_{22}^{(1)} = -2/2 = -1$, $a_{3j}^{(2)} = a_{3j}^{(1)} - c_3^{(1)} a_{2j}^{(1)}$ for $j = \overline{2, 3}$ and $b_3^{(2)} = b_3^{(1)} - c_3^{(2)} b_2^{(1)}$ and we find the final upper triangular matrix

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

Step 3. Perform back substitution by substituting values back into original equations to find solutions for x_1, x_2, and x_3. From 3rd row, compute $x_3 = b_3^{(2)}/a_{33}^{(2)}$:

$$3x_3 = 1 \Longrightarrow x_3 = \frac{1}{3}$$

In order to find x_2 substitute x_3 into row 2 and compute $x_2 = 1/a_{22}^{(1)} \cdot \left(b_2^{(1)} - a_{23}^{(1)} x_3\right)$ as

$$x_2 = 1/2 \cdot \left(6 - 5\left(\frac{1}{3}\right)\right) \Longrightarrow x_2 = \frac{13}{6}$$

Finally, substitute x_2 and x_3 into row 1 and compute $x_1 = 1/a_{11} \cdot (b_1 - a_{12}x_2 - a_{13}x_3)$:

$$x_1 = 1/1 \cdot \left(4 - 2\left(\frac{13}{6}\right) - \left(\frac{1}{3}\right)\right) \Longrightarrow x_1 = -\frac{2}{3}$$

The solution to the system of equations is $(x_1, x_2, x_3) = (-\frac{2}{3}, \frac{13}{6}, \frac{1}{3})$.

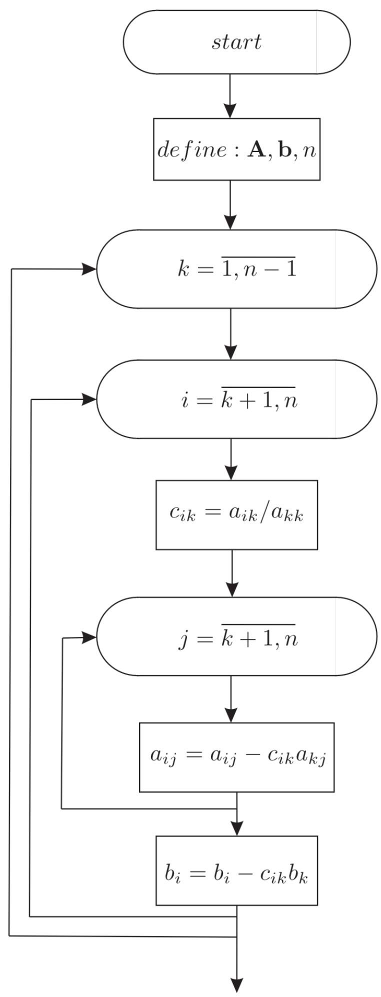

Figure 1.2: Block diagram of forward elimination in the Gaussian method

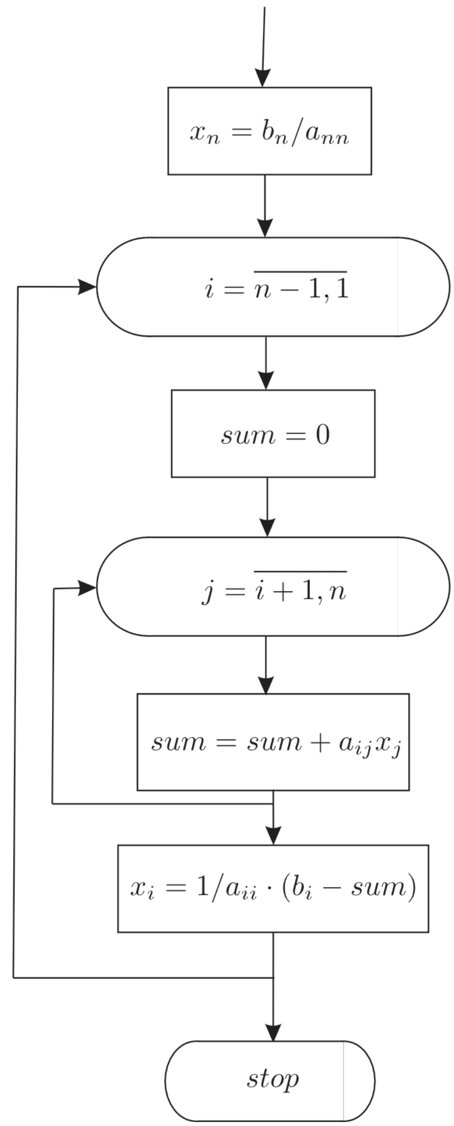

Figure 1.3: Block diagram of back substitution in the Gaussian method

2.3 Gauss – Jordan method

The Gauss-Jordan method is an extension of the Gaussian elimination technique used to solve systems of linear equations. It transforms the augmented matrix of the system into a form known as reduced row echelon form (identity matrix) through a series of elementary row operations. This method is particularly useful because it allows for the direct extraction of solutions without the need for back substitution.

The complete procedure for solving linear systems via the Gauss-Jordan method is formally presented in Algorithm 3, which systematically transforms the augmented matrix into reduced row echelon form through sequential row operations. Given below example demonstrates the complete process from initial matrix setup to final solution, including all intermediate steps.

Algorithm 2 Gaussian elimination with partial pivoting

Input:
  A ▷ $n \times n$ coefficient matrix
  $\mathbf{b}$ ▷ Right-hand side vector of size n

Initialize:
  Augmented matrix $[A|\mathbf{b}] \leftarrow$ combine A and $\mathbf{b}$
  $\mathbf{x} \leftarrow$ zero vector of size n ▷ Solution vector

for $k \leftarrow 1$ to $n-1$ **do**
  Partial pivoting:
    Find row p with maximum $|A_{ik}|$ where $i \in [k, n]$
    If $p \neq k$, swap rows p and k in $[A|\mathbf{b}]$

  Elimination:
  for $i \leftarrow k+1$ to n **do**
    factor $\leftarrow A_{ik}/A_{kk}$
    for $j \leftarrow k$ to n **do**
      $A_{ij} \leftarrow A_{ij} - \text{factor} \times A_{kj}$
    end for
    $b_i \leftarrow b_i - \text{factor} \times b_k$
  end for
end for

Back substitution:
for $i \leftarrow n$ downto 1 **do**
  sum $\leftarrow b_i$
  for $j \leftarrow i+1$ to n **do**
    sum $\leftarrow$ sum $- A_{ij} \times x_j$
  end for
  $x_i \leftarrow \text{sum}/A_{ii}$
end for

Output:
  $\mathbf{x}$ ▷ Solution vector

Algorithm 3 Gauss-Jordan elimination method

Input:
  A ▷ $n \times n$ coefficient matrix
  $\mathbf{b}$ ▷ Right-hand side vector of size n

Initialize:
  Augmented matrix $[A|\mathbf{b}] \leftarrow$ combine A and $\mathbf{b}$
  $n \leftarrow$ number of rows in A

for $k \leftarrow 1$ to n **do**
  Partial pivoting:
    Find row p with maximum $|A_{ik}|$ where $i \in [k, n]$
    If $p \neq k$, swap rows p and k in $[A|\mathbf{b}]$

  Normalization:
    pivot $\leftarrow A_{kk}$
    $R_k \leftarrow R_k/\text{pivot}$ ▷ Normalize pivot row

  Elimination:
  for $i \leftarrow 1$ to n where $i \neq k$ **do**
    factor $\leftarrow A_{ik}$
    $R_i \leftarrow R_i - \text{factor} \times R_k$ ▷ Eliminate variable x_k from all other rows
  end for
end for

Output:
  $\mathbf{x} \leftarrow$ last column of $[A|\mathbf{b}]$ ▷ Solution vector

Example: Gauss-Jordan method

We are given the following system of three linear equations with three unknowns

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 9 \\
2x_1 - 3x_2 + 4x_3 &= 13 \\
3x_1 + 4x_2 + 5x_3 &= 40
\end{aligned}
$$

The goal is to solve this system using the Gauss-Jordan elimination method, which transforms the augmented matrix into reduced row echelon form.

Step 1. We begin by constructing the augmented matrix $[A|\mathbf{b}]$ of the system

$$
\left[\begin{array}{ccc|c}
1 & 1 & 1 & 9 \\
2 & -3 & 4 & 13 \\
3 & 4 & 5 & 40
\end{array}\right]
$$

Step 2. Apply Gauss-Jordan elimination to eliminate x_1 from rows 2 and 3. We first use the element $a_{11} = 1$ as our pivot to eliminate x_1 from the rows below. Compute row operations $R_2 \leftarrow R_2 - 2R_1$

$$\begin{aligned}[2,-3,4|13] - 2\times[1,1,1|9] &= [2-2,-3-2,4-2|13-18]\\ &= [0,-5,2|-5]\end{aligned}$$

and $R_3 \leftarrow R_3 - 3R_1$

$$\begin{aligned}[3,4,5|40] - 3\times[1,1,1|9] &= [3-3,4-3,5-3|40-27]\\ &= [0,1,2|13]\end{aligned}$$

After these operations, our matrix becomes

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 9\\ 0 & -5 & 2 & -5\\ 0 & 1 & 2 & 13\end{array}\right].$$

To eliminate x_2 from row 3, now we use $a_{22} = -5$ as our next pivot. First, we'll eliminate x_2 from row 3 by computing row operation $R_3 \leftarrow R_3 + \frac{1}{5}R_2$

$$\begin{aligned}[0,1,2|13] + \frac{1}{5}\times[0,-5,2|-5] &= [0,1-1,2+\frac{2}{5}|13-1]\\ &= [0,0,\frac{12}{5}|12]\end{aligned}$$

The matrix now appears as

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 9\\ 0 & -5 & 2 & -5\\ 0 & 0 & \frac{12}{5} & 12\end{array}\right].$$

Operate $R_2 \leftarrow -R_2$, $R_3 \leftarrow 5R_3$ and these result in

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 9\\ 0 & 5 & -2 & 5\\ 0 & 0 & 12 & 60\end{array}\right].$$

We begin by normalizing the last row to make the leading coefficient 1 and work upwards to eliminate x_3 from the rows above. Operate $R_2 \leftarrow R_2 + \frac{1}{6}R_3$ and $R_3 \leftarrow \frac{1}{12}R_3$ and the matrix becomes

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 9\\ 0 & 5 & 0 & 15\\ 0 & 0 & 1 & 5\end{array}\right].$$

Operate $R_2 \leftarrow \frac{1}{5}R_2$, we get

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

Now we can proceed to eliminate entries above each pivot. Operate $R_1 \leftarrow R_1 - R_2 - R_3$ and after all these operations, we obtain the reduced row echelon form

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

Thus, the solution can be directly read from the final matrix

$$x_1 = 1, \quad x_2 = 3, \quad x_3 = 5.$$

3 Iterative methods

3.1 Jacobi's iteration method

The Jacobi method is an iterative algorithm used to solve systems of linear equations, particularly those that are strictly diagonally dominant. Named after the German mathematician Carl Gustav Jacob Jacobi, this method is effective for approximating solutions when direct methods may be computationally expensive or impractical.

The Jacobi method operates under two main assumptions:

1. The system of equations has a unique solution.

2. The coefficient matrix has no zeros on its main diagonal.

where $x_i^{(k+1)}$ is the updated value of variable x_i at iteration $k+1$, and a_{ij} are the coefficients from the matrix $\mathbf{A}$.

The Jacobi algorithm can be summarized in the following steps:

- *Step 1. Initialization.* Start with an initial guess for the solution vector $\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}\right)^T$, where T is the transpose operator. Initial guess can be a zero vector or any reasonable estimate.

- *Step 2. Iteration.* For each iteration k update each variable using the formula above based on values from the previous iteration. The iterative formula is expressed as:

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

- *Step 3. Convergence check.* After updating all variables, check if changes are below a predetermined threshold to determine if convergence has been achieved:

$$|x_{k+1} - x_k| \leq \epsilon$$

- *Step 4. Repeat.* If convergence is not reached, return to Step 2 and continue iterating until the solution stabilizes or a maximum number of iterations is reached.

Example: Jacobi's method

Consider the following system of linear equations:

$$2x_1 + x_2 + x_3 = 6 \tag{1.12}$$
$$x_1 + 3x_2 - x_3 = 0 \tag{1.13}$$
$$-x_1 + x_2 + 2x_3 = 3 \tag{1.14}$$

- *Step 1: Initialization.* We start with an initial guess for the variables:

$$x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 0$$

- *Step 2: Iteration.* To apply the Jacobi method, we first need to express each variable in terms of the others.
  - From equation (1.12):
  $$x_1 = 1/2 \cdot (6 - x_2 - x_3) \tag{1.15}$$
  - From equation (1.13):
  $$x_2 = 1/3 \cdot (x_3 - x_1) \tag{1.16}$$
  - From equation (1.14):
  $$x_3 = 1/2 \cdot (3 + x_1 - x_2) \tag{1.17}$$

- *Step 3-4: Iteration process.* Now we will use the equations derived to perform iterations.
  - *Iteration 1:* Using initial guesses in equations (4), (5), and (6):
    * Calculate $x_1^{(1)}$: $x_1^{(1)} = (6 - 0 - 0)/2 = 3$
    * Calculate $x_2^{(1)}$: $x_2^{(1)} = (0 - 0)/3 = 0$
    * Calculate $x_3^{(1)}$: $x_3^{(1)} = (3 + 0 - 0)/2 = 1.5$
  - *Iteration 2:* Using values from iteration 1:
    * Calculate $x_1^{(2)}$: $x_1^{(2)} = (6 - 0 - 1.5)/2 = 2.25$
    * Calculate $x_2^{(2)}$: $x_2^{(2)} = (1.5 - 3)/3 = -0.5$
    * Calculate $x_3^{(2)}$: $x_3^{(2)} = (3 + 3 - 0))/2 = 3$
  - *Iteration $k+1$.* Continue iterating until convergence $\left|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right| \leq \epsilon$ is achieved.

After performing sufficient iterations, round the final approximate solutions and print them.

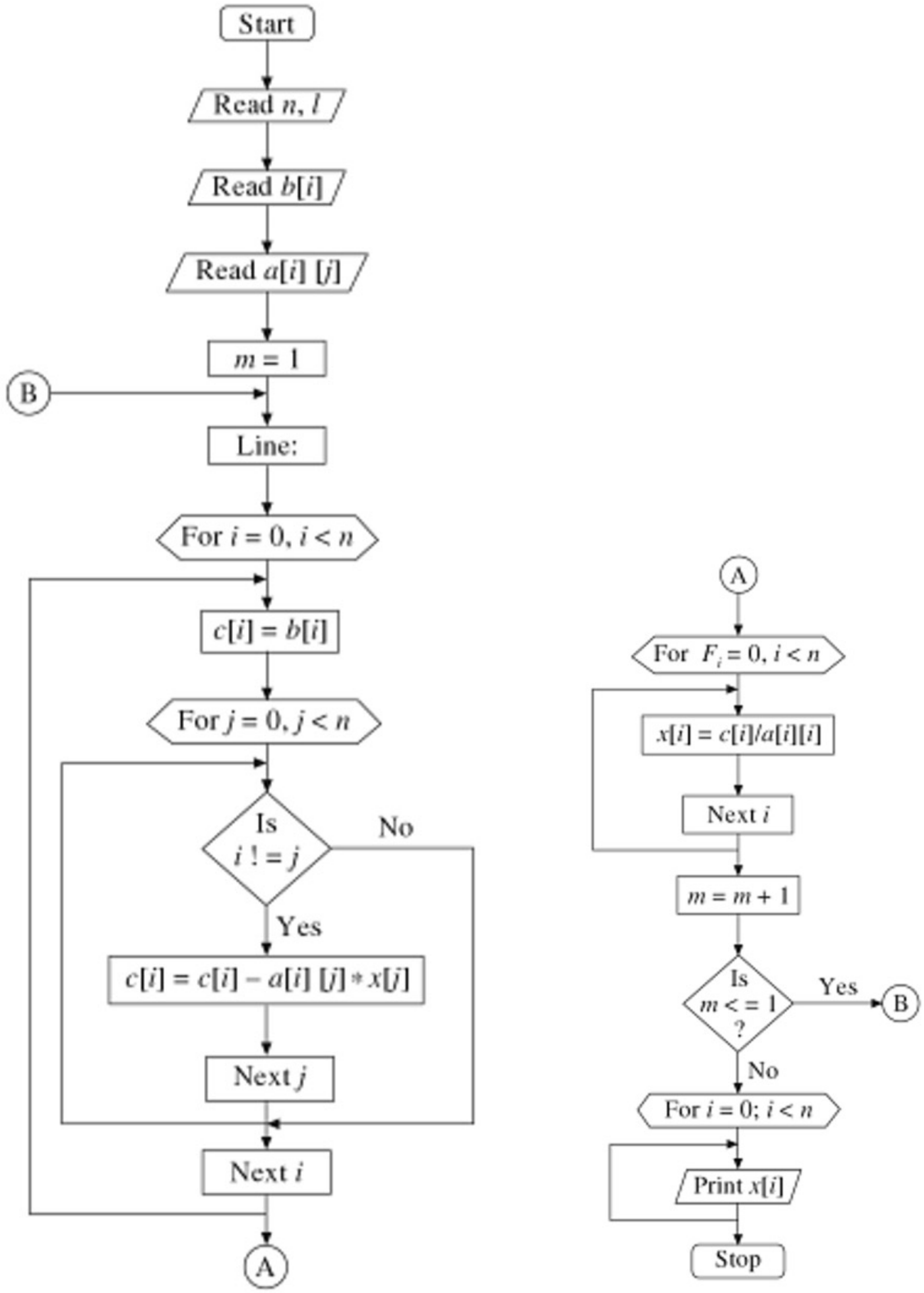

Figure 1.4: Block diagram of Jacobi's iteration method

3.2 Gauss – Seidel method

The method improves upon the Jacobi method by using the most recently computed values in subsequent calculations, which can lead to faster convergence.

The Gauss-Seidel algorithm can be summarized in the following steps:

- *Step 1. Initialization.* Start with an initial guess for the solution vector $\mathbf{x}^{(0)}$. This can often be a zero vector or any other reasonable estimate.
- *Step 2. Iteration.* For each iteration k, update each variable x_i in the solution vector using the formula:

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

  where the first summation uses updated values, while the second uses values from the previous iteration.
- *Step 3. Convergence check.* After updating all variables, check for convergence by evaluating if the difference between successive iterations is below a predetermined threshold $\left|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right| \leq \epsilon$.
- *Step 4. Repeat.* If convergence is not achieved, return to step 2 and repeat until the solution stabilizes.

Example: Gauss-Seidel method

Consider the following system:

$$20x_1 + x_2 - 2x_3 = 17 \tag{1.18}$$

$$3x_1 + 20x_2 - x_3 = -18 \tag{1.19}$$

$$2x_1 - 3x_2 + 20x_3 = 25 \tag{1.20}$$

1. *Step 1. Initialization.* We start with an initial guess for the variables:

$$x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 0,$$

2. *Step 2-3. Iterations and convergence check.* Using initial guesses, calculate the first iteration:

$$x_1^{(1)} = (17 - x_2^{(0)} + 2x_3^{(0)})/20 = 0.8500$$

$$x_2^{(1)} = (-18 - 3x_1^{(1)} + x_3^{(0)})/20 = -1.0275$$

$$x_3^{(1)} = (25 - 2x_1^{(1)} + 3x_3^{(0)})/20 = 1.0109$$

   In the second iteration, we get:

$$x_1^{(2)} = (17 - x_2^{(1)} + 2x_3^{(1)})/20 = 1.0025$$

$$x_2^{(2)} = (-18 - 3x_1^{(2)} + x_3^{(1)})/20 = -0.9998$$

$$x_3^{(2)} = (25 - 2x_1^{(2)} + 3x_3^{(1)})/20 = 0.9998$$

In the third iteration, we get:

$$x_1^{(3)} = (17 - x_2^{(2)} + 2x_3^{(2)})/20 = 1.0000$$

$$x_2^{(3)} = (-18 - 3x_1^{(3)} + x_3^{(2)})/20 = -1.0000$$

$$x_3^{(3)} = (25 - 2x_1^{(3)} + 3x_3^{(2)})/20 = 1.0000$$

The values in the second and third iterations being partically the same for the tolerance $\epsilon = 0.002$. Hence $x_1 = 1$, $x_2 = -1$ and $x_3 = 1$.

3.3 Relaxation method

The relaxation method is an iterative technique used to solve systems of linear equations, particularly when standard iterative methods (like Jacobi or Gauss-Seidel) exhibit slow convergence. The method accelerates convergence by introducing a relaxation parameter ω that adjusts the step size during iterations.

The relaxation method is designed to iteratively refine the solution to a system of equations of the form $A\mathbf{x} = \mathbf{b}$. The key idea is to relax the requirement that each variable must be computed exactly from the previous iteration, allowing for a weighted average that can lead to faster convergence.

Relaxation method's algorithm can be summarized in the following steps:

- *Step 1. Initialization.* Start with an initial guess for the solution vector $\mathbf{x}^{(0)}$.
- *Step 2. Iteration.* For each variable x_i in the system, update its value using the formula:

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

  where k indicates the current iteration, ω is a relaxation factor typically chosen in the range $(0, 2)$.
- *Step 3. Convergence check.* After updating all variables, check if changes are below a predetermined threshold to determine convergence.
- *Step 4. Repeat.* If convergence is not achieved, return to step 2 and continue iterating until stability is reached.

4 Tasks

1. Solve the following system using Cramer's Rule:

$$\begin{cases} 3x - 2y = 5 \\ x + 4y = 11 \end{cases}$$

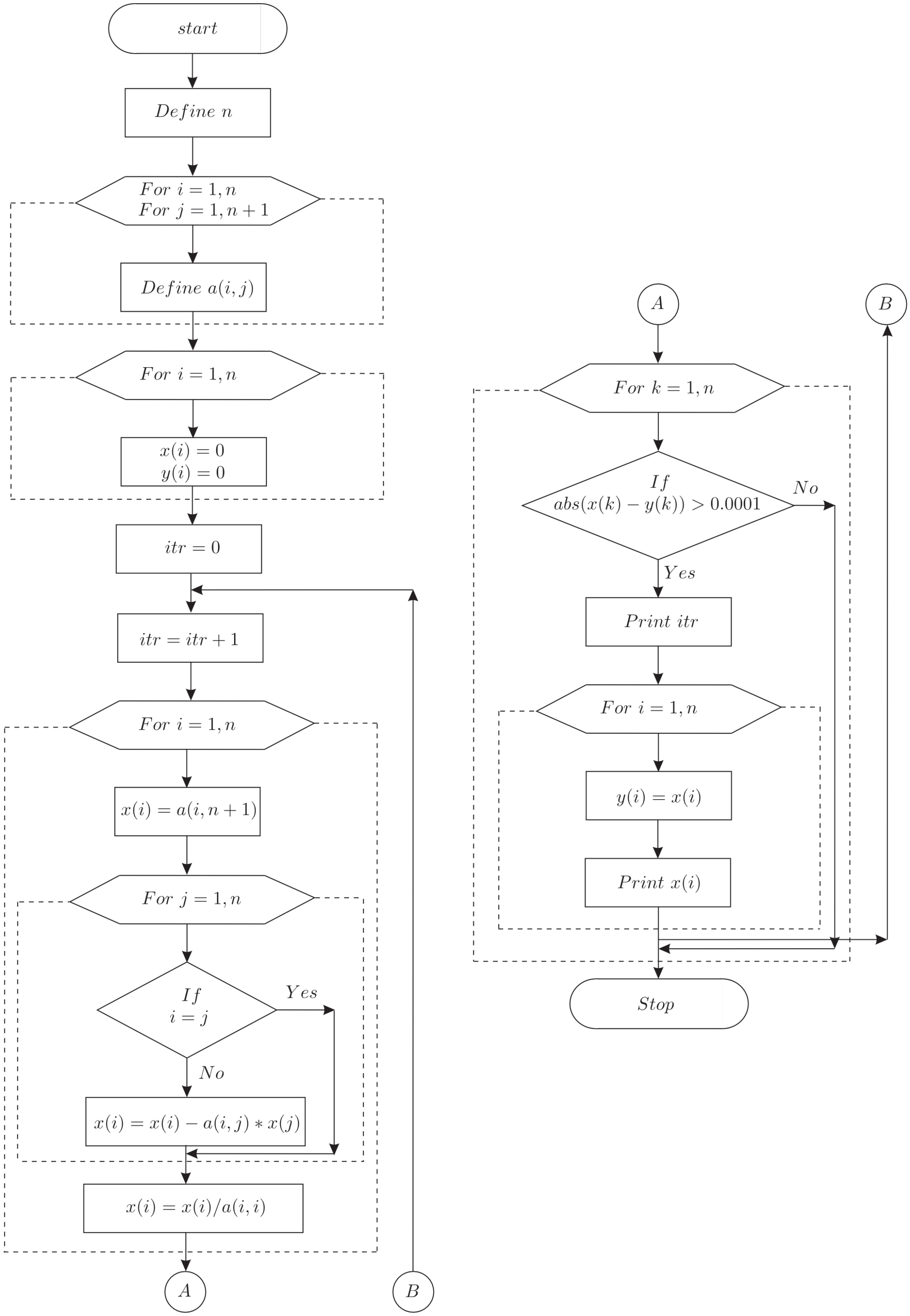

Figure 1.5: Block diagram of Gauss-Seidel method

Algorithm 4 Relaxation method for solving a system of linear algebraic equations

Input:
A ▷ $n \times n$ coefficient matrix
$\mathbf{b}$ ▷ Right-hand side vector of size n
$\mathbf{x}^{(0)}$ ▷ Initial guess vector
ω ▷ Relaxation parameter ($1 < \omega < 2$)
ϵ ▷ Tolerance for stopping criterion
$N_{\max}$ ▷ Maximum number of iterations

Initialize:
$k \leftarrow 0$ ▷ Iteration counter
$\mathbf{x} \leftarrow \mathbf{x}^{(0)}$ ▷ Current solution vector

repeat
  $k \leftarrow k + 1$
  for $i \leftarrow 1$ to n **do**
    $\sigma \leftarrow 0$
    for $j \leftarrow 1$ to $i - 1$ **do**
      $\sigma \leftarrow \sigma + A_{ij} x_j^{(k)}$ ▷ Use updated values
    end for
    for $j \leftarrow i + 1$ to n **do**
      $\sigma \leftarrow \sigma + A_{ij} x_j^{(k-1)}$ ▷ Use old values
    end for
    $x_i^{(k)} \leftarrow (1 - \omega) x_i^{(k-1)} + \frac{\omega}{A_{ii}}(b_i - \sigma)$
  end for
  Compute residual $\|\mathbf{r}^{(k)}\| \leftarrow \|\mathbf{b} - A\mathbf{x}^{(k)}\|$
until $\|\mathbf{r}^{(k)}\| < \epsilon$ **or** $k \geq N_{\max}$

Output:
$\mathbf{x}^{(k)}$ ▷ Approximate solution
k ▷ Number of iterations performed

(a) Compute the determinant D of the coefficient matrix.

(b) Compute D_x by replacing the first column with the constants.

(c) Compute D_y by replacing the second column with the constants.

(d) Solve for x and y.

2. Solve the following system using Gaussian elimination method, where the system can be represented as an augmented matrix as

$$\left[\begin{array}{cccc|c} 6.5 & 2.2 & 3.0 & 2.8 & 1.6 \\ 4.0 & 3.2 & 1.2 & 4.3 & 4.0 \\ 3.2 & 3.3 & 4.0 & 2.0 & 4.5 \\ 4.6 & 3.4 & 1.1 & 3.8 & 3.2 \end{array}\right]$$

(a) Perform row operations to reach upper triangular form.

(b) Apply back substitution to find x_1, x_2, x_3 and x_4.

3. Solve the system using Gauss-Jordan method:

$$\begin{cases} 2x_1 + 2x_2 - x_3 + x_5 = 0 \\ -x_1 - x_2 + 2x_3 - 3x_4 + x_5 = 0 \\ x_1 + x_2 - 2x_3 - x_5 = 0 \\ x_3 + x_4 + x_5 = 0 \end{cases}$$

(a) Write the augmented matrix form.

(b) Perform row operations to reach reduced row echelon form.

(c) Read the solution directly from the matrix.

4. Solve the system using the Jacobi method (perform 3 iterations), starting with initial guess $(0, 0, 0)$:

$$\begin{cases} 4x + y + z = 7 \\ x + 5y + 2z = 10 \\ x + 2y + 6z = 14 \end{cases}$$

Compute the approximate solution after each iteration.

5. Solve the same system as above using the Gauss-Seidel method (perform 3 iterations), starting with $(0, 0, 0)$:

$$\begin{cases} 4x + y + z = 7 \\ x + 5y + 2z = 10 \\ x + 2y + 6z = 14 \end{cases}$$

Compare the convergence speed with Jacobi's method.

6. Use the Relaxation method with $\omega = 1.2$ to solve the system (perform 3 iterations):

$$\begin{cases} 3x + y - z = 3 \\ x + 4y + z = 6 \\ -x + y + 5z = 7 \end{cases}$$

   Start with $(0, 0, 0)$ and show updated values at each step.

7. Solve the following system of linear equations using four different methods and compare your results:

$$\begin{aligned} 3x_1 - 5x_2 + 47x_3 + 20x_4 &= 18 \\ 11x_1 + 16x_2 + 17x_3 + 10x_4 &= 26 \\ 56x_1 + 22x_2 + 11x_3 - 18x_4 &= 34 \\ 17x_1 + 66x_2 - 12x_3 + 7x_4 &= 82 \end{aligned}$$

   (a) Rearrange the equations to ensure diagonal dominance for Jacobi and Gauss-Seidel methods.
   (b) Use the same initial guess for all iterative methods.
   (c) Compute the relative error after each iteration.
   (d) Create a table comparing all four solutions.
   (e) Discuss which method was most efficient
   (f) Comment on accuracy and convergence rates
   (g) Explain any discrepancies between methods