ASSIGMENT 2. Solution of a system of linear algebraic equations

Score: 20 points
Deadline: Week 3 before your group's practical lesson.
Requirements: Implement the algorithms for the specified methods (Cramer, Gauss, Jacobi and Gauss-Seidel) and evaluate their errors.

1 General description

Systems of linear equations are fundamental in numerous fields including engineering, physics, computer science, and economics. They 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

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \cdots\cdots\cdots\cdots\cdots&\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n, \end{aligned} \tag{0.1}$$

The system of equations (0.1) can be succinctly expressed in matrix form as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{0.2}$$

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 (0.1) and (0.2) (where $i, j = \overline{1, n}$)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n, \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{0.3}$$

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 (0.1), 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

$$x_i = \frac{D_i}{D}, \quad i = \overline{1,n} \tag{0.4}$$

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 $\mathbf{b}$.

Steps to Apply Cramer's Rule:

- *Step 1. Formulate the system.* Write the system of equations in matrix form $\mathbf{Ax} = \mathbf{b}$.
- *Step 2. Calculate Determinants.*
  - Compute the determinant $D = |\mathbf{A}|$.
  - For each variable x_i, compute D_i by replacing the i-th column of $\mathbf{A}$ with $\mathbf{b}$.
- *Step 3. Solve for variables.* Use Cramer's formula to find each variable $x_1 = D_1/D$, $x_2 = D_2/D$, ..., $x_n = D_n/D$.

Example: Cramer's method

Consider a system of two linear equations in two variables x_1 and x_2:

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

the coefficient matrix and constant vector are:

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

Calculating Determinants:

1. Compute the determinant of the matrix $\mathbf{A}$:

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

2. Compute D_1 by replacing the 1-st column of $\mathbf{A}$ with $\mathbf{b}$:

$$D_1 = \left| \begin{bmatrix} 5 & 3 \\ 11 & 1 \end{bmatrix} \right| = 5 - 33 = -28.$$

3. Compute D_2 by replacing the 2-nd column of $\mathbf{A}$ with $\mathbf{b}$:

$$D_2 = \left| \begin{bmatrix} 2 & 5 \\ 4 & 11 \end{bmatrix} \right| = 22 - 20 = 2.$$

Then using Cramer's Rule (0.5):

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

Cramer's Rule is computationally efficient for small systems but becomes impractical for larger systems due to its reliance on determinants. ■

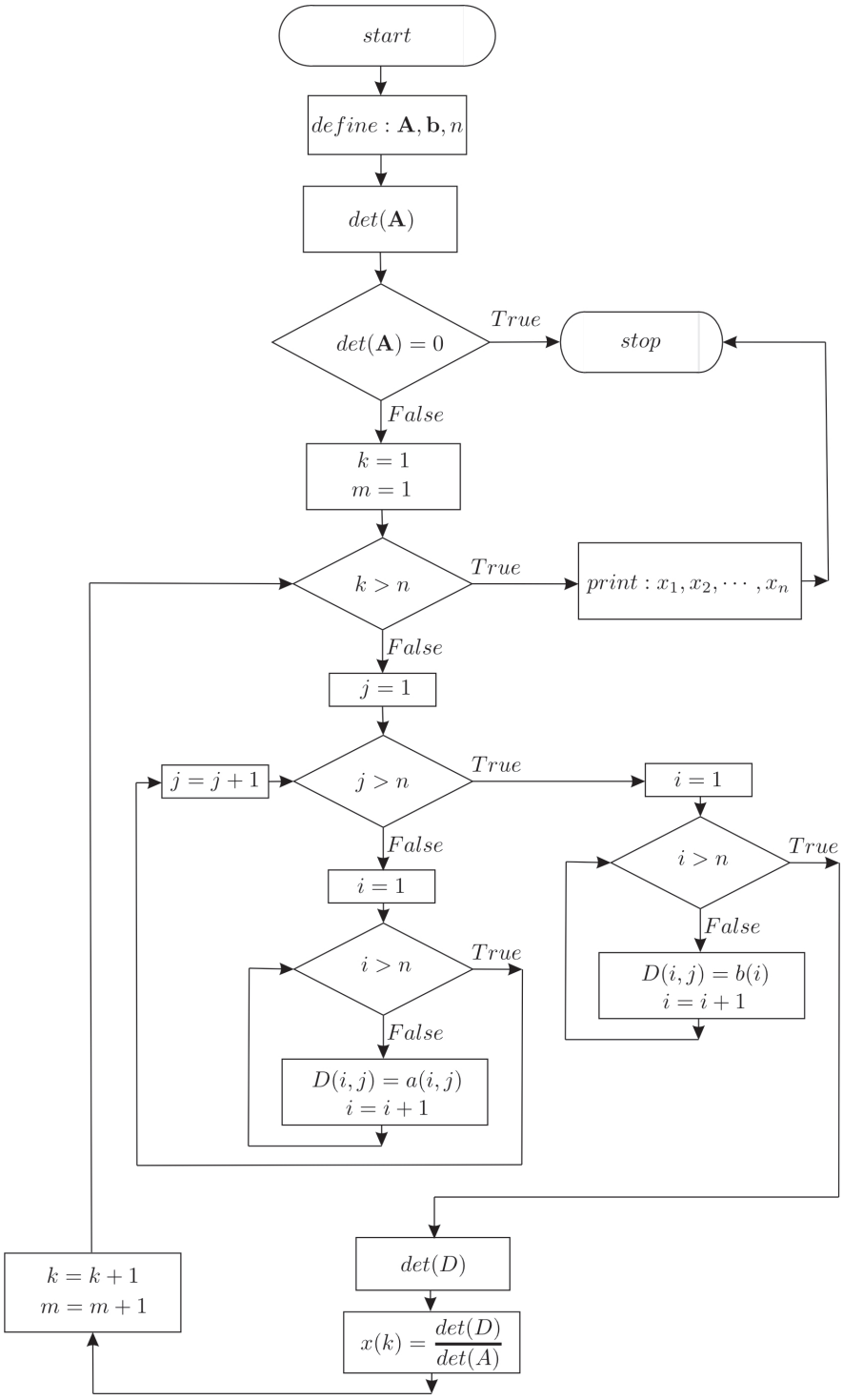

Figure 0.1: Block Diagram of Cramer's Method

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 (0.1). 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 $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is the coefficient matrix, $\mathbf{x}$ is the vector of unknowns, and $\mathbf{b}$ is the vector of constants. 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:

  - 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.

  In the first step, 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

$$c_i^{(1)} = \frac{a_{i1}}{a_{11}}, \quad i = \overline{2,n} \tag{0.5}$$

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

$$a_{ij}^{(1)} = a_{ij} - c_i^{(1)} a_{1j}, \quad b_i^{(1)} = b_i - c_i^{(1)} b_1 \quad i = \overline{2,n}, \quad j = \overline{1,n}. \tag{0.6}$$

  Then we get a system of transformed algebraic equations,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ &\vdots \\ a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n &= b_n^{(1)}, \end{aligned} \tag{0.7}$$

  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 (0.7), 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}:

$$\begin{gathered} c_i^{(k)} = \frac{a_{ik}^{k-1}}{a_{kk}^{k-1}}, \quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - c_i^{(k)} a_{kj}^{(k-1)}, \\ b_i^{(k)} = b_i^{(k-1)} - c_i^{(k)} b_k^{(k-1)}, \quad k = \overline{2,n-1}, \quad i = \overline{k+1,n}. \end{gathered} \tag{0.8}$$

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

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ &\vdots \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}. \end{aligned} \tag{0.9}$$

- *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

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \tag{0.10}$$

and moving upwards

$$x_k = \frac{1}{a_{kk}^{(k-1)}}\left(b_k^{(k-1)} - a_{k,k+1}^{(k-1)}x_{k+1} - a_{k,k+2}^{(k-1)}x_{k+2} - \cdots - a_{kn}^{(k-1)}x_n\right), \quad k = \overline{n-1,1} \tag{0.11}$$

Example: Gauss method
Consider the following system of equations:

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

Step 1. Write the augmented matrix:

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

Step 2. Perform forward elimination:
- To eliminate x_1 from Row 3 ($i = 3$ in eqns.(0.5) and (0.6)), 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 $\mathrm{b}_3^{(1)} = b_3 - c_3^{(1)}b_3$:

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

- 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 $b_3^{(2)} = b_3^{(1)} - c_3^{(2)}b_2^{(1)}$:

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

Step 3. Perform back substitution. Finally, substitute values back into original equations to find solutions for x_1, x_2, and x_3.

- From row 3, 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 $x_2 = 1/a_{22}^{(1)} \cdot \left(b_2^{(1)} - a_{23}^{(1)} x_3\right)$:

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

- 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 $(x_1, x_2, x_3) = (-\frac{2}{3}, \frac{13}{6}, \frac{1}{3})$.

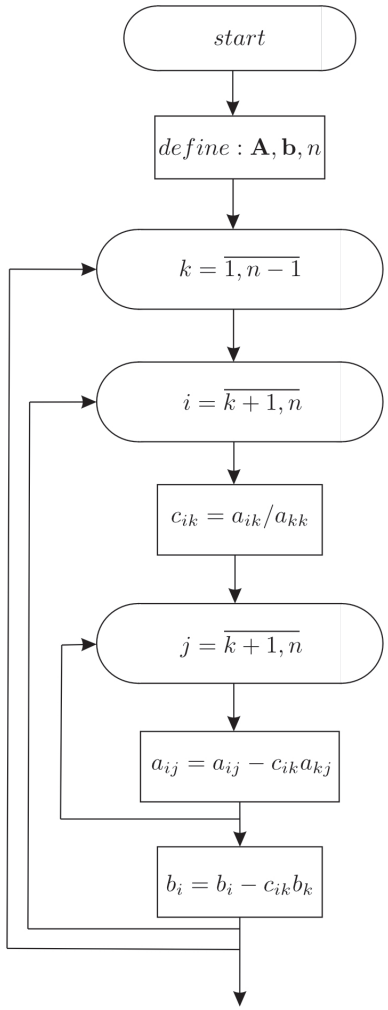

Figure 0.2: Block Diagram of forward elimination in the Gaussian Method

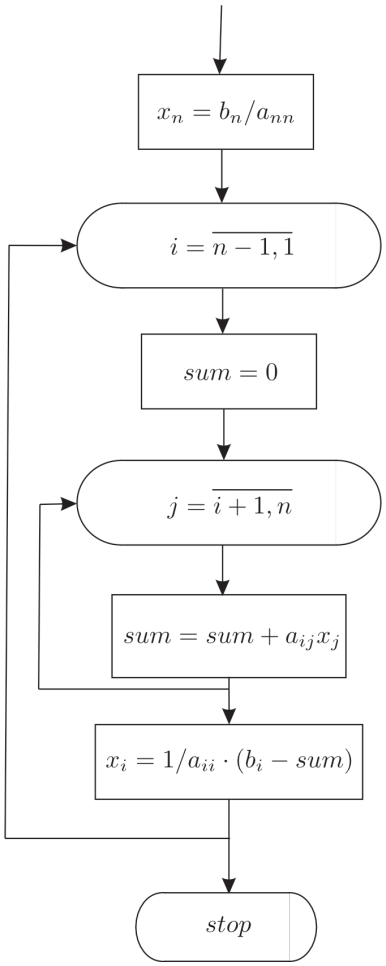

Figure 0.3: Block Diagram of back substitution in the Gaussian Method

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. This method is particularly useful because it allows for the direct extraction of solutions without the need for back substitution.

The Gauss-Jordan elimination process involves a sequence of elementary row operations, which include:

- *Step 1. Interchanging Rows.* Swap two rows to position a non-zero row above any zero rows.
- *Step 2. Scaling Rows.* Multiply or divide a row by a non-zero constant to make the leading coefficient equal to 1.
- Step 3. Row Addition. Add or subtract multiples of one row from another to eliminate entries above and below the leading coefficients.

Example: Gauss-Jordan method
Consider the following system of equations:

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

- *Step 1. The augmented matrix* for this system is

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

- *Step 2. Apply Gauss-Jordan Elimination*
  - Eliminate x_1 from Rows 2 and 3, compute

$$R_2 = R_2 - 2R_1$$

and

$$R_3 = R_3 - 3R_1$$

This results in:

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

  - To eliminate x_2 from Row 3, compute:

$$R_3 = R_3 + \frac{1}{5}R_2$$

Now we have:

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

  - Operate $R_2 = -R_2$ and $R_3 = 5R_3$. This results in:

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

  - Operate $R_2 = R_2 + \frac{1}{6}R_3$ and $R_3 = \frac{1}{12}R_3$. The result is

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

– Operate $R_2 = \frac{1}{5}R_2$, 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 $R_1 = R_1 - R_2 - R_3$, this gives

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

Thus, the solution to the system is $(x_1, x_2, x_3) = (1, 3, 5)$

3 Iteration 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:

$$2x_1 + x_2 + x_3 = 6 \tag{0.12}$$
$$x_1 + 3x_2 - x_3 = 0 \tag{0.13}$$
$$-x_1 + x_2 + 2x_3 = 3 \tag{0.14}$$

- *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):
$$x_1 = 1/2 \cdot (6 - x_2 - x_3) \tag{0.15}$$
  - From equation (1.13):
$$x_2 = 1/3 \cdot (x_3 - x_1) \tag{0.16}$$
  - From equation (1.14):
$$x_3 = 1/2 \cdot (3 + x_1 - x_2) \tag{0.17}$$

- *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.

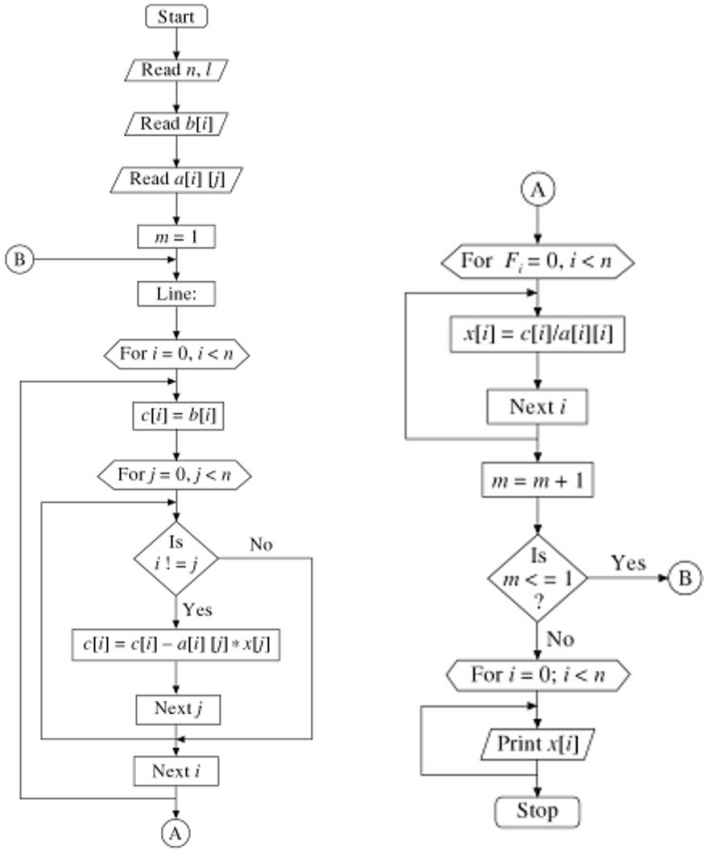

Figure 0.4: Block diagram of Jacobi's iteration method

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:

$$20x_1 + x_2 - 2x_3 = 17 \tag{0.18}$$
$$3x_1 + 20x_2 - x_3 = -18 \tag{0.19}$$
$$2x_1 - 3x_2 + 20x_3 = 25 \tag{0.20}$$

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 $x_3 = 1$.

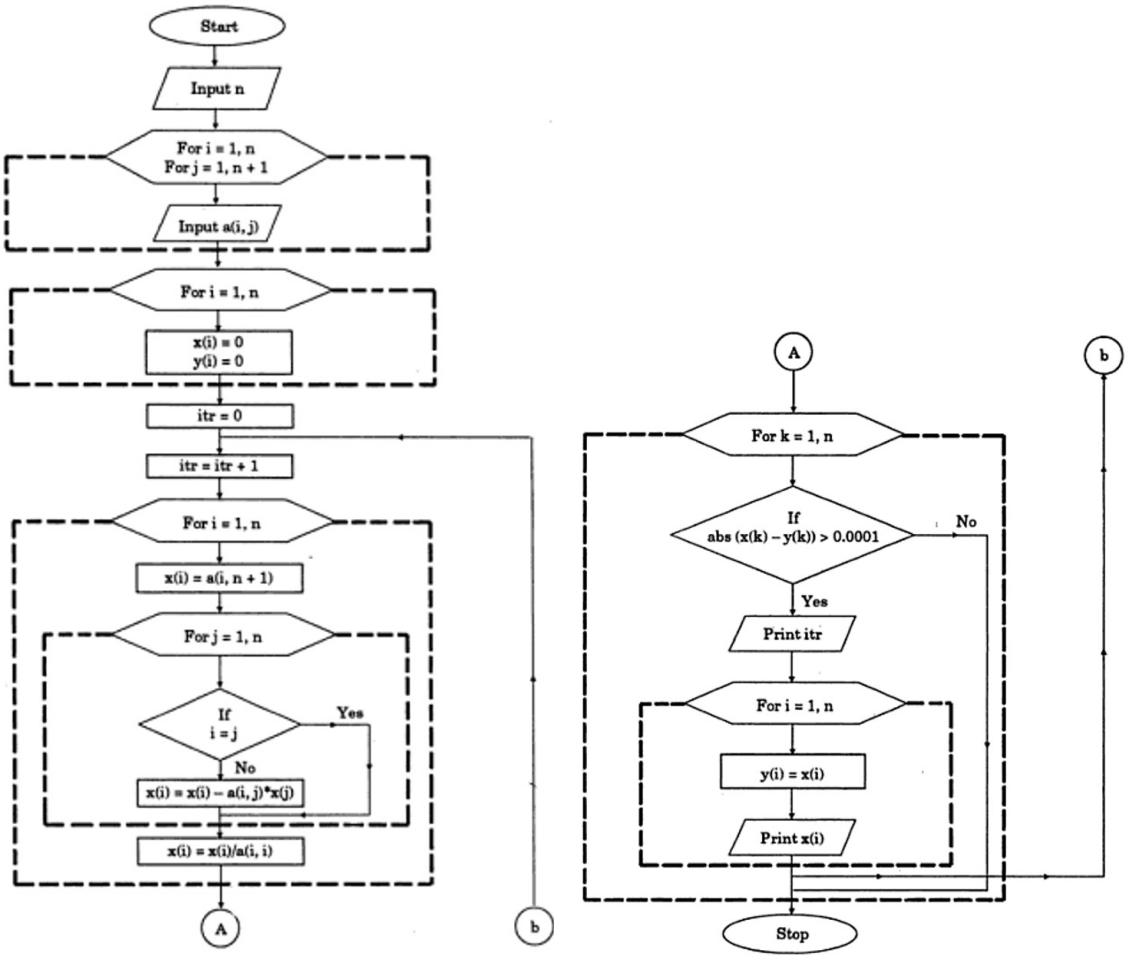

Figure 0.5: Block diagram of Gauss-Seidel method

3.3 Relaxation method

The relaxation method builds upon the principles of the Jacobi and Gauss-Seidel methods, introducing an extrapolation factor, often denoted as ω, which can accelerate convergence.

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 Ill-Conditioned Equations

Ill-conditioned equations refer to systems of linear equations that are sensitive to small changes in the input data, leading to disproportionately large changes in the output solutions. This sensitivity makes it difficult to trust the solutions obtained from numerical methods applied to such systems.

On the contrary, a system is well-conditioned if small changes in the coefficients of the system also produce small changes in the solution. We often come across ill-conditioned systems in practical applications. Ill- conditioning of a system is usually expected when the determinant of the coefficient matrix is small. The coefficient matrix of an ill-conditioned sys- tem is called an ill-conditioned matrix.

5 Tasks

1. Solve the following system of equations using Cramer, Gauss, Jacobi, and Gauss-Seidel methods and compare the final results:

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

6 Resources

- B.S. Grewal, "Numerical Methods in Engineering and Science", Khanna Publication, Ed. 9th., 2014 (Chapters 3.4-3.7)