University of Mohamed Boudiaf – M'sila
Faculty of Science and Technology
Department of Civil Engineering

# Numerical Methods Course

# Chapter 7: Approximate method for solving systems of linary equations

Merini Abdelaziz*

April 20, 2026

## Contents

**1 Introduction and Definitions** **2**
1.1 Vector and Matrix Norms . . . . . 2
1.2 Examples of Vector and Matrix Norms . . . . . 2
1.3 Strictly Diagonally Dominant Matrices . . . . . 3

**2 Principle of an Iterative Method** **3**

**3 Description of the Main Iterative Methods** **5**

**4 Jacobi method** **6**
4.1 Principle of the method . . . . . 6
4.2 Jacobi algorithm . . . . . 7
4.3 Convergence of the Jacobi method . . . . . 7

**5 Gauss-Seidel Method** **9**
5.1 Principle of the Method . . . . . 9
5.2 Gauss-Seidel Algorithm . . . . . 9
5.3 Convergence and Stopping Criterion . . . . . 9

**6 Relaxation Method (SOR)** **12**
6.1 Principle of the Method . . . . . 12
6.2 Convergence of the Relaxation Method . . . . . 13

*

# 1 Introduction and Definitions

In this chapter, we introduce the iterative (or indirect) methods that provide approximate solutions to systems of linear equations:

$$Ax = b$$

where $A$ is a square matrix assumed to be invertible, $b$ is a known vector, and $x$ is the vector of unknowns. The main indirect methods used to solve the system $Ax = b$ are: 1. Jacobi method. 2. Gauss–Seidel method. 3. Relaxation method.

## 1.1 Vector and Matrix Norms

**Definition 1.** *A function* $\| \bullet \| : \mathbb{R}^n \to \mathbb{R}_+$ *is called a vector norm if:*

$$\begin{cases} 1)\|x\| \geq 0, \quad \forall x \in \mathbb{R}^n \\ 2)\|\lambda x\| = |\lambda|\|x\|, \quad \forall x \in \mathbb{R}^n, \ \forall \lambda \in \mathbb{R} \text{ or } \mathbb{C} \\ 3)\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in \mathbb{R}^n \end{cases}$$

**Definition 2.** *A function* $\| \bullet \| : M_n(\mathbb{K}) \to \mathbb{R}_+$ *is called a matrix norm if:*

$$\begin{cases} 1)\|A\| \geq 0 \\ 2)\|\lambda A\| = |\lambda|\|A\|, \quad \forall \lambda \in \mathbb{R} \text{ or } \mathbb{C} \\ 3)\|A + B\| \leq \|A\| + \|B\|, \quad \forall A, B \in M_n(\mathbb{K}) \\ 4)\|AB\| \leq \|A\| \cdot \|B\|, \quad \forall A, B \in M_n(\mathbb{K}) \end{cases}$$

## 1.2 Examples of Vector and Matrix Norms

$$\|x\|_2 = \left( \sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}}, \quad \|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

$$\|A\|_1 = \max_{1 \leq j \leq n} \left( \sum_{i=1}^{n} |a_{ij}| \right), \quad \|A\|_2 = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}}, \quad \|A\|_\infty = \max_{1 \leq i \leq n} \left( \sum_{j=1}^{n} |a_{ij}| \right)$$

—

Example 1: Vector Norms For the vector

$$x = \begin{pmatrix} 2 \\ -3 \\ 0 \\ 4 \end{pmatrix}$$

we have:

$$\|x\|_1 = |2| + |-3| + |0| + |4| = 9$$

$$\|x\|_2 = \sqrt{2^2 + (-3)^2 + 0^2 + 4^2} = \sqrt{29} \approx 5.39$$

$$\|x\|_\infty = \max(2, 3, 0, 4) = 4$$

—

Example 2: Matrix Norms For the matrix

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -3 & 2 \\ -2 & 1 & -2 \end{pmatrix}$$

we obtain:

$$\|A\|_1 = \max(3, 5, 4) = 5$$

$$\|A\|_2 = \sqrt{1^2 + 1^2 + 0^2 + 0^2 + (-3)^2 + 2^2 + (-2)^2 + 1^2 + (-2)^2} = \sqrt{24} \approx 4.90$$

$$\|A\|_\infty = \max(2, 5, 5) = 5$$

—

### 1.3 Strictly Diagonally Dominant Matrices

**Definition 3.** *Let $A = (a_{ij})_{i,j=1,\ldots,n}$ be a square matrix of order $n$. $A$ is said to have a strictly dominant diagonal if:*

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad \forall i$$

## 2 Principle of an Iterative Method

**Definition 4.** *An iterative method for solving the system $Ax = b$ consists in transforming it into an equivalent fixed-point system of the form:*

$$x = Bx + C$$

*where $B$ and $C$ are matrices (or vectors) determined from $A$ and $b$. The solution of the original system is then obtained as the limit of the sequence defined by:*

$$x^{(k+1)} = Bx^{(k)} + C, \quad x^{(0)} \text{ being an initial approximation.}$$

In general, the matrix $A$ is decomposed as:

$$A = M - N$$

where $M$ is chosen to be easily invertible. Then:

$$Ax = b \Longleftrightarrow (M - N)x = b \Longleftrightarrow Mx = Nx + b \Longleftrightarrow x = M^{-1}Nx + M^{-1}b$$

By setting:

$$B = M^{-1}N, \quad C = M^{-1}b$$

we obtain the fixed-point form:

$$x = Bx + C$$

Using the principle of the fixed-point theorem, we construct the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ as:

$$\begin{cases} x^{(k+1)} = Bx^{(k)} + C \\ x^{(0)} \text{ given} \end{cases}$$

## Convergence

The iterative method converges if and only if the spectral radius of $B$ satisfies:

$$\rho(B) < 1, \quad \rho(B) = \max_i |\lambda_i|$$

where $\lambda_i$ are the eigenvalues of $B$.

When this condition holds, the sequence $\{x^{(k)}\}$ converges to a limit $x^*$, which satisfies:

$$x^* = Bx^* + C \Longleftrightarrow Ax = b$$

Therefore, the limit $x^*$ is the exact solution of the system.

## Error Analysis

Let $e^{(k)} = x^{(k)} - x^*$ denote the error vector at iteration $k$. From the recurrence relation:

$$x^{(k)} = Bx^{(k-1)} + C, \quad x^* = Bx^* + C$$

we obtain:

$$e^{(k)} = x^{(k)} - x^* = B(x^{(k-1)} - x^*) = Be^{(k-1)}$$

By induction:

$$e^{(k)} = B^k e^{(0)}, \quad e^{(0)} = x^{(0)} - x^*$$

Thus:

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

The method converges for any initial guess $x^{(0)} \in \mathbb{R}^n$ if and only if:

$$\lim_{k \to +\infty} B^k = 0$$

which is equivalent to $\rho(B) < 1$.

**Theorem 1.** *The sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$ defined by the iterative scheme converges to $x^*$ if and only if $\rho(B) < 1$.*

**Proposition 1.** *(Relation between spectral radius and matrix norms) For any induced matrix norm $\| \cdot \|_i$ with $i \in \{1, 2, \infty\}$, we have:*

$$\rho(B) \leq \|B\|_i$$

This inequality provides a practical way to estimate convergence by computing matrix norms instead of eigenvalues.

# 3 Description of the Main Iterative Methods

Consider the following decomposition of the matrix $A$:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = D - E - F \tag{3.6}$$

where:

$$D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn})$$

is the diagonal matrix containing the main diagonal elements of $A$,

$$E = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{pmatrix}$$

is the strictly lower triangular part (below the diagonal), and

$$F = \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -a_{n-1,n} \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

is the strictly upper triangular part (above the diagonal).

Example 3
Let

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 5 \end{pmatrix}$$

Then we can write the decomposition:

$$A = D - E - F$$

with

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \quad E = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & -1 & 0 \end{pmatrix}, \quad F = \begin{pmatrix} 0 & -1 & -2 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

Thus, the matrix $A$ is decomposed into:
* $D$: the diagonal matrix containing the main diagonal entries,
* $E$: the strictly lower triangular part (below the diagonal),
* $F$: the strictly upper triangular part (above the diagonal).

This decomposition is the basis for constructing the iteration matrices used in the Jacobi, Gauss–Seidel, and Relaxation methods.

# 4 Jacobi method

## 4.1 Principle of the method

The matrix $A$ is decomposed into the form

$$A = M - N = D - (E + F)$$

with $D$ = diagonal of A.

$E$ : lower matrix $\begin{cases} -a_{ij} & \text{for } j > i \\ 0 & \text{pour } i \leq j \end{cases}$

$F$ : upper matrix $\begin{cases} -a_{i,j} & \text{for } i < j \\ 0 & \text{for } i \leq j \end{cases}$

the system $Ax = b$ then becomes: $Dx = (E + F)x + b$
. Jacobi's iterative method is therefore written :

$$Dx^{(k+1)} = (E + F)x^{(k)} + b$$

Let: $x^{(k+1)} = D^{-1}(E+F)x^{(k)} + D^{-1}b\}$ Note 1. This method assumes pivots $a_{ii} \neq 0$ (if this is not the case, a simple exchange of cloths is often sufficient to verify this condition).

## 4.2 Jacobi algorithm

$\begin{cases} x^{(k+1)} = Jx^{(k)} + C, \text{ where} J = D^{-1}(E + F) \text{ and } C = D^{-1}b \\ x^{(0)} \text{ given} \end{cases}$
Hence Jacobi's algorithm:
$\begin{cases} x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_j - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right] \\ x^{(0)} \text{ given} \end{cases}$

## 4.3 Convergence of the Jacobi method

theorem 1. Jacobi's algorithm converges if and only if the spectral radius of J is strictly less than 1 , i.e. :
$\rho(J) = \max_i \{|\lambda_i|\}$, where $\lambda_i$ are the eigenvalues of the $J$ matrix.
Remark 2. 1. In practice, calculating $\rho(J)$ is very complicated. It is then sufficient to check whether $\|J\| < 1$, since $\rho(J) < \|J\|$.
2. Jacobi's method converges if one of the following conditions is met:
if $\|J\| < 1$.
If $A$ is a strictly dominant a-diagonal matrix $\left( |a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ii}| \quad \forall i \text{ fixed}, \right.$
$\left. 1 \leq i \leq n \right)$
Calculation stop criterion for the Jacobi method Calculations for this method stop when the absolute difference between two successive iterations is less than a given $\epsilon$ .

$$\left| x^{(k+1)} - x^{(k)} \right| < \epsilon$$

Here, you need to check the difference for all the components one by one: $\left| x_1^{(k+1)} - x_1^{(k)} \right| < \epsilon, \left| x_2^{(k+1)} - x_2^{(k)} \right| < \epsilon, \ldots, \left| x_n^{(k+1)} - x_n^{(k)} \right| < \epsilon.$

## Example: Jacobi Iteration

Consider the linear system:

$$\begin{cases} 10x_1 + 2x_2 - x_3 = 27 \\ -3x_1 - 6x_2 + 2x_3 = -61.5 \\ x_1 + x_2 + 5x_3 = -21.5 \end{cases}$$

The coefficient matrix and right-hand side are:

$$A = \begin{pmatrix} 10 & 2 & -1 \\ -3 & -6 & 2 \\ 1 & 1 & 5 \end{pmatrix}, \quad b = \begin{pmatrix} 27 \\ -61.5 \\ -21.5 \end{pmatrix}$$

We decompose $A = D - (E + F)$, where:

$$D = \begin{pmatrix} 10 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \quad E = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix}, \quad F = \begin{pmatrix} 0 & -2 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

The Jacobi iteration formula is:

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b$$

Explicitly, the updates are:

$$\begin{cases} x_1^{(k+1)} = \dfrac{1}{10}\left(27 - 2x_2^{(k)} + x_3^{(k)}\right) \\ x_2^{(k+1)} = \dfrac{1}{-6}\left(-61.5 + 3x_1^{(k)} - 2x_3^{(k)}\right) \\ x_3^{(k+1)} = \dfrac{1}{5}\left(-21.5 + x_1^{(k)} + x_2^{(k)}\right) \end{cases}$$

## First Iterations

Starting with $x^{(0)} = (0, 0, 0)^T$, we compute:

$$x^{(1)} = \left(\frac{27}{10}, \frac{-61.5}{-6}, \frac{-21.5}{5}\right)^T = (2.7,\ 10.25,\ -4.3)^T$$

Next iteration:

$$x^{(2)} = \begin{cases} x_1^{(2)} = \frac{1}{10}\left(27 - 2(10.25) + (-4.3)\right) = 0.72 \\ x_2^{(2)} = \frac{1}{-6}\left(-61.5 + 3(2.7) - 2(-4.3)\right) = 9.42 \\ x_3^{(2)} = \frac{1}{5}\left(-21.5 + 2.7 + 10.25\right) = -1.71 \end{cases}$$

So:

$$x^{(2)} = (0.72,\ 9.42,\ -1.71)^T$$

Repeating this process produces successive approximations $x^{(3)}, x^{(4)}, \ldots$ that converge to the true solution of $Ax = b$.

## Observation

The Jacobi method uses only the diagonal entries of $A$ for inversion, while the off-diagonal terms are treated explicitly in each iteration. Convergence is guaranteed if the spectral radius $\rho(J) < 1$, which is often satisfied when $A$ is strictly diagonally dominant.

# 5 Gauss-Seidel Method

## 5.1 Principle of the Method

The matrix $A$ is decomposed into the form:

$$A = M - N = (D - E) - F$$

with:
$D$: diagonal matrix of $A$.
$E$: strictly lower triangular matrix $\begin{cases} -a_{ij} & \text{for } j < i \\ 0 & \text{for } i \leq j \end{cases}$
$F$: strictly upper triangular matrix $\begin{cases} -a_{ij} & \text{for } i < j \\ 0 & \text{for } j \leq i \end{cases}$

The system $Ax = b$ then becomes: $(D - E)x = Fx + b$.
The Gauss-Seidel iterative method is therefore written as:

$$(D - E)x^{(k+1)} = Fx^{(k)} + b$$

Which leads to the matrix iterative form:

$$x^{(k+1)} = (D - E)^{-1}Fx^{(k)} + (D - E)^{-1}b$$

**Note 1.** Similar to Jacobi, this method requires that the diagonal elements $a_{ii} \neq 0$.

## 5.2 Gauss-Seidel Algorithm

The iterative scheme is:

$$\begin{cases} x^{(k+1)} = G_s x^{(k)} + C \\ G_s = (D - E)^{-1}F \\ C = (D - E)^{-1}b \end{cases}$$

In scalar form, the Gauss-Seidel algorithm uses the updated values as soon as they are available:

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

## 5.3 Convergence and Stopping Criterion

**Theorem 1.** The Gauss-Seidel algorithm converges if and only if the spectral radius $\rho(G_s) < 1$.

**Remark 2.** Gauss-Seidel converges if $A$ is strictly diagonally dominant or if $A$ is symmetric and positive definite.

**Stopping Criterion:** The calculation stops when the absolute difference between two successive iterations is less than $\epsilon$:

$$\|x^{(k+1)} - x^{(k)}\|_\infty < \epsilon \implies \max_i |x_i^{(k+1)} - x_i^{(k)}| < \epsilon$$

## Example: Gauss-Seidel Iteration Process

Consider the linear system:

$$\begin{cases} 10x_1 + 2x_2 - x_3 = 27 \\ -3x_1 - 6x_2 + 2x_3 = -61.5 \\ x_1 + x_2 + 5x_3 = -21.5 \end{cases}$$

Starting with $x^{(0)} = (0, 0, 0)^T$:

- **Iteration 1 ($k = 0$):**

$$\begin{aligned} x_1^{(1)} &= \frac{27}{10} = 2.7 \\ x_2^{(1)} &= \frac{-61.5 + 3(2.7)}{-6} = 8.9 \\ x_3^{(1)} &= \frac{-21.5 - 2.7 - 8.9}{5} = -6.62 \end{aligned}$$

  $\implies x^{(1)} = (2.7,\ 8.9,\ -6.62)^T$.

- **Iteration 2 ($k = 1$):**

$$\begin{aligned} x_1^{(2)} &= \frac{1}{10}(27 - 2(8.9) + (-6.62)) = 0.258 \\ x_2^{(2)} &= \frac{-61.5 + 3(0.258) - 2(-6.62)}{-6} = 7.9143 \\ x_3^{(2)} &= \frac{-21.5 - 0.258 - 7.9143}{5} = -5.9345 \end{aligned}$$

  $\implies x^{(2)} = (0.258,\ 7.9143,\ -5.9345)^T$.

**Observation:** Gauss-Seidel incorporates new information immediately ($x_1^{(k+1)}$ used for $x_2^{(k+1)}$), typically leading to faster convergence than Jacobi.

## Derivation of the Gauss-Seidel Iteration Matrix

For the system $Ax = b$ defined above:

### 1. Matrix Decomposition

We decompose $A$ into $A = D - E - F$:

- $D = \begin{pmatrix} 10 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 5 \end{pmatrix}$
- $E = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix}$
- $F = \begin{pmatrix} 0 & -2 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$

### 2. Iteration Matrix $G_s$

The matrix is defined by $G_s = (D - E)^{-1}F$. First, calculate $(D - E)$:

$$D - E = \begin{pmatrix} 10 & 0 & 0 \\ -3 & -6 & 0 \\ 1 & 1 & 5 \end{pmatrix}$$

### 3. Inverse of $(D - E)$

The inverse of the lower triangular matrix is:

$$(D - E)^{-1} = \begin{pmatrix} 0.1 & 0 & 0 \\ -0.05 & -0.1667 & 0 \\ -0.01 & 0.0333 & 0.2 \end{pmatrix}$$

### 4. Final $G_s$ Matrix

$$G_s = \begin{pmatrix} 0.1 & 0 & 0 \\ -0.05 & -0.1667 & 0 \\ -0.01 & 0.0333 & 0.2 \end{pmatrix} \begin{pmatrix} 0 & -2 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$G_s = \begin{pmatrix} 0 & -0.2 & 0.1 \\ 0 & 0.1 & 0.2834 \\ 0 & 0.02 & -0.0766 \end{pmatrix}$$

**Conclusion:** The first column of zeros in $G_s$ reflects that $x_1^{(k+1)}$ is independent of $x_1^{(k)}$, a characteristic of the Gauss-Seidel update.

# 6 Relaxation Method (SOR)

The Relaxation method, also known as the Successive Over-Relaxation (SOR) method, is an iterative technique for solving linear systems $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is a square matrix and $x, b \in \mathbb{R}^n$.

## 6.1 Principle of the Method

Decompose the matrix $A$ as $A = D - E - F$, where:

- $D$ is the diagonal matrix of $A$.
- $E$ is the strictly lower triangular matrix ($E_{ij} = -a_{ij}$ for $i > j$).
- $F$ is the strictly upper triangular matrix ($F_{ij} = -a_{ij}$ for $i < j$).

Let $\omega \in \mathbb{R}\backslash\{0\}$ be a relaxation parameter. The system $Ax = b$ is manipulated as follows:

$$\begin{aligned} Ax = b &\iff (D - E - F)x = b \\ &\iff \omega(D - E - F)x = \omega b \\ &\iff (D - \omega E - \omega F)x + (\omega - 1)Dx = (\omega - 1)Dx + \omega b \\ &\iff (D - \omega E)x = [(1 - \omega)D + \omega F]x + \omega b \end{aligned}$$

This leads to the iterative sequence $(x^{(k)})_{k\in\mathbb{N}}$:

$$(D - \omega E)x^{(k+1)} = [(1 - \omega)D + \omega F]x^{(k)} + \omega b$$

The iteration matrix $R_\omega$ is given by:

$$x^{(k+1)} = \underbrace{(D - \omega E)^{-1}[(1 - \omega)D + \omega F]}_{R_\omega} x^{(k)} + \underbrace{\omega(D - \omega E)^{-1}b}_{C}$$

In scalar form, the update for each component is:

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

**Remarks:**

- If $\omega > 1$: Over-relaxation (used to accelerate convergence).
- If $\omega = 1$: The method reduces to the Gauss-Seidel method.
- If $\omega < 1$: Under-relaxation (used to stabilize divergent systems).

## 6.2 Convergence of the Relaxation Method

1. **Necessary Condition:** For any invertible matrix $A$, a necessary condition for convergence is $\omega \in ]0, 2[$.
2. **Strictly Diagonally Dominant Matrix:** If $A$ is strictly diagonally dominant, the method converges for $\omega \in ]0, 1]$.
3. **Symmetric Positive Definite (SPD) Matrix:** If $A$ is SPD, the method converges if and only if $\omega \in ]0, 2[$.

### Example: SOR Iteration

Consider the same system from previous examples, with $\omega = 1.1$:

$$\begin{cases} 10x_1 + 2x_2 - x_3 = 27 \\ -3x_1 - 6x_2 + 2x_3 = -61.5 \\ x_1 + x_2 + 5x_3 = -21.5 \end{cases}$$

**Iteration 1 ($k = 0$) with $x^{(0)} = (0, 0, 0)^T$:**

- **Calculate $x_1^{(1)}$:** $x_1^{(GS)} = \frac{27}{10} = 2.7$
  $x_1^{(1)} = (1 - 1.1)(0) + 1.1(2.7) = \mathbf{2.97}$
- **Calculate $x_2^{(1)}$:** $x_2^{(GS)} = \frac{-61.5+3(2.97)}{-6} = 8.765$
  $x_2^{(1)} = (1 - 1.1)(0) + 1.1(8.765) = \mathbf{9.6415}$
- **Calculate $x_3^{(1)}$:** $x_3^{(GS)} = \frac{-21.5-2.97-9.6415}{5} = -6.8223$
  $x_3^{(1)} = (1 - 1.1)(0) + 1.1(-6.8223) = \mathbf{-7.5045}$

**Result:** $x^{(1)} = (2.97,\ 9.6415,\ -7.5045)^T$.

*Observation:* With $\omega = 1.1$, the values "jump" further in the direction of the solution, which can significantly reduce the number of iterations required to reach a specific tolerance $\epsilon$.

## Comparison of Iterative Methods

| **Feature** | **Jacobi Method** | **Gauss-Seidel Method** | **SOR Method** |
|---|---|---|---|
| **Matrix Form** | $Dx^{(k+1)} = (E+F)x^{(k)} + b$ | $(D-E)x^{(k+1)} = Fx^{(k)} + b$ | $(D-\omega E)x^{(k+1)} = [(1-\omega)D + \omega F]x^{(k)} + \omega b$ |
| **Component Update** | $x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij}x_j^{(k)})$ | $x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)})$ | $x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega x_i^{GS}$ |
| **Iteration Matrix** | $J = D^{-1}(E+F)$ | $G_s = (D-E)^{-1}F$ | $R_\omega = (D-\omega E)^{-1}[(1-\omega)D + \omega F]$ |
| **Convergence Rate** | $\rho(J) < 1$ (Slowest) | $\rho(G_s) < 1$ (Generally $2\times$ faster) | $\rho(R_\omega) < 1$ (Fastest if $\omega$ is optimal) |
| **Parallelization** | Very Easy (Independent) | Difficult (Sequential) | Difficult (Sequential) |

Table 1: Detailed landscape comparison of iterative linear solvers.