Comprehensive Journal of Science
Volume (10), Issue (38), (Dec, 2025)
ISSN: 3014-6266

مجلة العلوم الشاملة
المجلد (10) ملحق العدد 38 ديسمبر 2025
ردمد: 3014-6266

The strategies of improvement the Performance for the Jacobi method using Gaussian iterative methods

Naima Altaher Kafu

Faculty of Education-Yefren /University of Zintan

.naima.kafu@uoz.edu.ly

تاريخ الاستلام:2025/12/14 –تاريخ المراجعة: 2025/12/16 – تاريخ القبول: 2025/12/23 – تاريخ للنشر : 22/1/ 2026

Abstract

This work presents advanced methodologies for the improvement of the Jacobi method by incorporating Gauss-Seidel iterative schemes [1][2]. The Jacobi method, though simple to use [3][4], usually suffers from slow convergence, especially for large-scale linear systems [5]. In this regard, a hybrid scheme is presented that utilizes the strengths of both the Jacobi and Gauss-Seidel methods [6][7]. The important techniques that form the basis of this study include successive over-relaxation (SOR) to accelerate the convergence [2][8], adaptive step-sizing based on residual monitoring [9][10], and preconditioning techniques that enhance the numerical properties of the linear systems [11][12]. Furthermore, the effect of reordering equations for improved convergence rates is also considered [11]. Numerical experiments performed in the context of this study show that these enhancements significantly reduce the number of iterations required for acceptable solutions, thus enhancing computational efficiency [2][13]. In this context, the proposed hybrid scheme outperforms the traditional Jacobi and Gauss-Seidel methods for several scenarios and provides a robust solution for practitioners working in applied mathematics and engineering disciplines [2].

Keywords: Jacobi Method, Gauss-Seidel Method, Iterative Methods, Linear Systems, Convergence Analysis, Successive Over, Relaxation (SOR), Preconditioning.

Introduction

The Jacobi method is a classical iterative algorithm used to approximate the solution of a system of linear equations [1][2]. It is particularly useful for large systems where direct methods are computationally expensive [2]. The method iteratively refines an initial guess, using values from the previous iteration to compute new approximations [1]. However, the Jacobi method often suffers from slow convergence, especially for large-scale linear systems [3][4]. To address this limitation, hybrid approaches that integrate the strengths of both Jacobi and Gauss-Seidel methods have been developed [3][5]. The Gauss-Seidel method, an iterative technique for solving linear systems, updates the solution vector step-by-step, using the most recent values to improve convergence [5].

This study explores advanced strategies for enhancing the Jacobi method by incorporating Gauss-Seidel iterative techniques [5]. These strategies include the implementation of

Successive Over-Relaxation (SOR) to accelerate convergence [6][7], adaptive step-sizing based on residual monitoring [8][9], and preconditioning techniques to improve the numerical properties of the linear systems [10][11]. The impact of reordering equations to optimize convergence rates is also investigated [12][13]. The proposed hybrid method aims to outperform traditional Jacobi and Gauss-Seidel methods, offering a robust solution for practitioners in applied mathematics and engineering [3][5].

Literature Review

Because of its simplicity and ease of implementation, the Jacobi method has been studied and investigated in depth since its creation. Early works emphasized its foundational principles by demonstrating its ability to work under most conditions, especially on diagonally dominant matrices. However, its limitations were soon discovered by researchers, one of the major ones concerning the speed of convergence. For example, Young analyzed convergence properties of the Jacobi method and noticed that it would converge under very particular conditions depending on the properties of a matrix [1].

Considering the Jacobi method converges slowly, many improvements have been considered. A significant enhancement is that of SOR, or Successive Over-Relaxation, which improves convergence rates by introducing a relaxation factor. This was studied in-depth by W. H. Press et al., where it was shown that for an ideal choice of the relaxation factor, it is possible to achieve a substantial reduction in the number of iterations of the method, at a prescribed accuracy [2].

Another strand of research efforts involves hybridization of iteration methods. The hybridization of the Jacobi method with the Gauss-Seidel method has attracted much interest because of its potential to exploit the respective strengths of both methods. In a recent work, Xu and Chen proposed a hybrid iterative method that uses Jacobi and Gauss-Seidel updates in an alternating manner. They have achieved faster convergence rates for certain classes of problems [3]. This hybrid method retains the simplicity of the Jacobi method and makes use of the faster convergence provided by Gauss-Seidel.

Approaches that have also been researched involve the use of preconditioning techniques to further improve the behavior of iterative methods. Preconditioning, in effect, transforms the original system into one that is more favorable; if successful, it reduces the condition number and thus improves convergence. Work by Saad has involved incomplete LU factorization as a means of preconditioning iterative methods and showed significant improvements in finding the solution for big sparse systems [4].

Another point of interest in research has been the reordering of equations in a system for better convergence. It has been observed that permutation in the equations can lead to enhanced diagonal dominance, thus improving the convergence properties of iterative methods. The work by Meijerink and Van der Vorst showed that suitable reordering may have a great effect on the performance of iterative solvers [5].

Conclusion The Jacobi method is an elementary iterative approach. Lack of efficiency in its convergence has prompted improvement efforts in great many ways: by means of relaxation techniques, hybrid methods, preconditioning, and equation reordering. This literature review has highlighted the continuing work on enhancing the efficiency and range of application of the Jacobi method for solving systems of linear equations.

1. Strategies for Enhancing the Jacobi Method Using Gauss Iterative Techniques.

The Jacobi method is an efficient iterative technique for solving linear equations. Although simple and easy to apply, in many instances, the method could be very slow in convergence, especially in large-scale linear equations. This paper investigates several approaches in improving the Jacobi method by combining the Gauss iterative methods in an attempt to accelerate the convergence. The Jacobi method can be represented mathematically in n unknowns as follows:

$$Ax = b$$

A is an $n \times n$ square matrix, x is the vector of unknowns with n elements, and b is the right-hand side vector with n elements.

1.1 Hybrid Jacobi-Gauss-Seidel Methods

The hybrid approach combines feature of both the Jacobi and Gauss-Seidel methods to achieve faster convergence. In the traditional Jacobi method, all variable values are updated at the end of each iteration using values computed in the previous iteration. Mathematically, for a system of equations with n unknowns, the update for each unknown x_i can be expressed as:

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

where k indicates the iteration number. Conversely, the Gauss-Seidel method utilizes the most recent updates immediately. This can be formulated as:

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

With a combination of these techniques, some variables are able to be solved using the Gauss-Seidel method, which relies on the latest information available, whereas others are able to be solved using the Jacobi method, which relies on older information available from the preceding iteration.

The hybrid method encourages flexible iteration strategies where, depending on the nature of the linear system, one can adjust the method for optimal performance. This is quite helpful when dealing with systems that contain a large number of unknowns, as it can help cut down on the iterations needed for converging.

In the hybrid method, the advantages of both Jacobi and Gauss-Seidel techniques are harnessed together to get the maximum results. This is done through the following methods:

This strategy includes:

1.1.1 Partial Updates

- Use the Gauss-Seidel method to update some variables and the Jacobi method to update the rest.
  Implementation:
- Select a set of variables according to certain criteria such as convergence.
- For example, if n unknowns set up in blocks, the update process using the Gauss-Seidel method should be carried out on the first block and updates using the Jacobi method should be done on the second block at the same iteration.

1.1.2 Periodic Switching

- Cycle between the Jacobi and Gauss-Seidel methods for a number of iterations.
  Implementation:
- An example is carrying out m iterations of the Jacobi iteration together with n iterations of the Gauss-Seidel iteration.
- This can be implemented as:

If $k \bmod (m + n) < m$, then update using Jacobi method; otherwise, update using Gauss-Seidel method.

1.2 Successive Over-Relaxation (SOR)

- Improve convergence speed by modifying the updates using a relaxation factor ω.

Implementation:

- Update the variables using:

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

Explore various values of w(generally, the range is $1<w<2$) to arrive at the convergence parameter suited to the linear equations.

1.3 Precondition

- Transform the existing system to improve its numerical properties, which would improve the convergence.
- Implementation:
- incomplete LU factorization or other preconditioning techniques to form the perturbed system:

$$M^{-1}\, Ax = M^{-1}b$$

- Solve the preconditioned system using a hybrid method.

1.4 Reordering of Equations

The reordered form of the system of linear equations may improve the performance of iterative methods for large linear equation problems. The aim here is to improve diagonal dominance.

Rationale

- **Diagonal Dominance:** A set of equations is said to be diagonally dominant if, for every equation, the largest in absolute value of the elements in the principal diagonal is greater than the sum of the absolute values of the remaining elements in the equation. If a system possesses diagonal dominance, then it will require fewer iterations in order to solve the system using the iterative method.
- **Stability:** Correct ordering of equations can improve numerical stability when solving iteratively. There can be chances of divergence.
- Implementation Steps

Step 1: Analyze the Coefficient Matrix

Given the system:

$Ax=b$, where A is the coefficient matrix:

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

- Analyzing the matrix A to find the order of the rows that maximizes the diagonal dominance.

Step 2: Rearrange the Equations

1. Sorting by
   - Calculate the absolute value of the diagonal elements |aii| for every row and measure it against the sum of the other elements' absolute values within that row:

$$|\boldsymbol{a_{ii}}| > \sum_{j\neq i} |\boldsymbol{a_{ij}}|$$

   - Swap the rows according to this criterion to maximize the diagonal dominance.

2. Use Heuristic or Algorithms:
   - Heuristically apply techniques like Gaussian Elimination or algorithms with the aim of maximizing the diagonal dominance via clever row permutations.
   - Strategies such as maximum element strategy may be employed, where the greatest available element in each column is placed on the diagonal position.

Step 3: Developing a New System

Also, after reordering, the new system can be modeled as:

$$\tilde{A}\tilde{x} = \tilde{b}$$

Where $\tilde{A}$ represents the reorganized matrix, $\tilde{x}$ is the vector of the new set of unknowns, and

$\tilde{b}$ is the updated right-hand side vector.

Example :

1. Original System:

$$\begin{aligned} 3x + y + 2z &= 5 \\ 2x + 4y + 3z &= 12 \\ 5y + 2z &= 8 \end{aligned}$$

2. Coefficient Matrix:

$$A = \begin{bmatrix} 3 & 1 & 2 \\ 2 & 4 & 3 \\ 0 & 5 & 2 \end{bmatrix}$$

3. Evaluate Diagonal Dominance:
   - Row 1: |3|>|1|+|2| (dominant)
   - Row 2: |4|>|2|+|3| (dominant)
   - Row 3: |5|>|2| (dominant)

   Because each row satisfies the condition for diagonal dominance, the set of equations can be solved by the Jacobi or the Gauss-Seidel methods.

   Impact on Convergence

   - Reordering usually results in fewer iterations being required to reach a specified level of accuracy. The reordered solution systems with improved diagonal dominance properties are likely to converge faster and with greater reliability than the unreordered systems.

   Equation rearrangement is a clever strategy for increasing the effectiveness and accuracy of iterative methods in solving linear systems. The improvement in diagonal dominance and stability achieved by this method is quite important for fast convergence rates and is, therefore, a crucial implementation step.

1.5 Adaptive Step Size

The adaptive step-size methods include changes in the increment size used in the variable update during the iteration. The strategy can be helpful for improving the rate of convergence. It can be applied during the solution of a linear system.

Rationale

- Convergence Control: With a variable step size, the control over the convergence procedure is more precise. This is especially the case when the approximate solution is close to the exact solution, where smaller steps can be used to avoid overshooting the target.
- Improvement in Efficiency: Step size can be increased in regions where the solution varies quite slowly from the convergence behavior of previous iterations or decreased when adjustments tend to create instability.

Steps to Implementation

Step 1: Check the Residuals

- Calculate the residual $r^{(k)}$ at each step, which is the difference between the left-hand side and right-hand side of the system:

$$r^{(k)} = b - Ax^{(k)}$$

- The norm of the residual can be used to judge how close a current solution is to a solution to an equation:

$$\|r^{(k)}\| = \|b - Ax^{(k)}\|$$

2. Determination of Step Size Adjustment

1. Assess Convergence:
   - "If the norm of the residual is decreasing rapidly, it may be appropriate to choose a larger step size."
   - “If the norm is not reducing satisfactorily or erratically, decrease the step size to improve stability.”

2. Specify Update:

Define a parameter α α to control the growth or shrinkage of the step size. The update rule could be:

$$x^{(k+1)} = x^{(k)} + \alpha . \Delta x^{(k)}$$

Where $\Delta x^{(k)}$ is the basic update value from the Jacobi or Gauss-Seidel method

3. Adjustment
   - Define the rules for updating α:
   - Increase Step Size: If $\|r(k)\| < \epsilon$ (where ϵ is a small tolerance value), increase α (e.g., multiply by *a* factor $\beta>1$).
   - decrease the Step Size: If $||r(k)|| > \epsilon$ threshold, then decrease α by multiplying it by *a* factor $\beta<1$).

Example

1. Initial Setup:
   - Initially, an initial guess x^0 and an initial step size α =1 will be considered.

2. Iteration Process:
   - Evaluate each component of the update for each iteration of the following process:
   - Calculate the basic update expression $\Delta x^{(k)}$ using the Jacobi or Update the solution as:

$$x^{(k+1)} = x^{(k)} + \alpha . \Delta x^{(k)}$$

3. Adapt Step Size:
   - Check the residual after every iteration and allow α to converge as described in Step 2.

Benefits

- Improved Stability: The oscillation possibility can be reduced by using the rate of convergence to modify the value of the constant step size.
- Faster Convergence: Proper step size adjustment can lead to fewer iterations, since the algorithm enables faster movements to the solution if needed.

- Flexibility: It caters to different systems with different properties, thereby providing a customized convergence approach.

Adding adaptive step size techniques into some iterative algorithms, such as Jacobi and Gauss-Seidel algorithms, may yield dramatic enhancements in convergence speed and stability. Based on continuous monitoring of residuals and dynamic adjustment of step size, people could optimize solving processes for complex linear equations.

6. Acceleration Techniques

The acceleration techniques are techniques employed to increase the rate of convergence of iterative formulas. This is carried out by altering the iterative formula so as to reduce the number of iterations necessary for obtaining a satisfactory solution.

Key Acceleration Techniques

1. Aitken's Delta-Squared Process
   We will examine the steps involved in the
   - Objective: To accelerate the convergence of a sequence of approximations.
   - How It Works: Aitken's method enhances the series by eliminating inaccuracies in each iteration.
   - Formula

$$\hat{x}^{(k)} = x^{(k)} + \frac{(x^{(k)} - x^{(k-1)})^2}{x^{(k)} - x^{(k-1)} - x^{(k-2)}}$$

   - Application: After each iterative step, apply Aitken's process if the three most recent approximations are available, producing a refined estimate.

2. Richardson Extrapolation
   - Objective: To enhance accuracy via linear combinations of solutions obtained from iterations with different parameters.
   - How It Works: Cells from iterative processes with different relaxation factors ω_1 and ω_2 are combined to get a more accurate result.
   - Formula:

$$x_{final} = \frac{\omega_1 x^{(k)} - \omega_2 x^{(k-1)}}{\omega_1 - \omega_2}$$

   - Application: Usage of conventional and adjusted relaxation factors in the process of iteration, calculation of the linear combination to get a better estimate
3. Deflation Techniques
   - Objective: Removes the impact of slowly converging components on solution iterations.
   - How It Works: The specific eigenvalues or modes causing slow convergence can be pinpointed and tamed, allowing the other modes to converge faster.
   - Implementation:

- Spectral techniques to be used to identify modes to be removed, or "Modify the iterative algorithm to ignore these modes so that other components can converge faster.

4. Preconditioning
   - Objective: The objective of this step is to change the form of the equations into something more easily solvable.
   - How It Works: Preconditioners are used to help improve the condition number of the coefficient matrix which speeds up convergence.

Types:

- Incomplete LU Factorization: Approximate the matrix A as LU, where L is a lower triangular matrix and U is an upper triangular matrix.
- Symmetric Successive Over-Relaxation (SSOR): It is a combination of relaxation methods and a preconditioning technique used for

System Transformation:

$$\boldsymbol{M^{-1}Ax = M^{-1}b}$$

- Application: Jacobi or Gauss-Seidel iterative methods can then be applied to the new system of equations. There will be faster rates of

5. **Hybrid Methods**

- Objective: To merge the advantages of a variety of iterative methods to ensure better performance.
- How It Works: Through a cycling between techniques (such as Jacobi and Gauss-Seidel) or incorporating other techniques (such as Successive Over-Relaxation), the overall efficiency can be improved.

Advantages of Acceleration Techniques

- Quicker Convergence: These methods result in faster solutions because the process converges quicker compared to traditional methods.
- Stability: They assist in dampening oscillations and facilitating convergence towards a solution.
- Flexibility: Can be adapted to different systems and convergence tendencies, thus offering an adapted solution according to the needs.

Methods such as the Aitken Delta-Squared process, Richardson extrapolation, deflation methods, and hybrid techniques have improved the efficiency of the Jacobi and Gauss-Seidel methods. With the use of these methods, an individual would be able to solve linear systems in a much more efficient way.

Example:

- Start by employing Jacobi iterative techniques that can run in parallel, then switch to Gauss-Seidel iterative techniques which guarantee faster convergence.

Hybrid Method-Example: Jacobi Method Followed by Gauss-Seidel Method

Problem Setup

Consider the following system of linear equations:

$$\mathbf{4x + y + z = 7 \qquad (1)}$$

$$\mathbf{x + 3y + 2z = 13 \qquad (2)}$$

$$\mathbf{2x - 4y + 5z = -3 \quad (3)}$$

Step 1: Initial Guess

We will start with an initial guess for the solution vector:

$$x^{(0)} = [0 , 0 ,0]$$

Step 2: Jacobi Method Iterations

The Jacobi update formulas for the given system are derived as follows:

1. From equation (1):

$$x^{(k+1)} = \frac{1}{4}(7 - y^{(k)} - z^{(k)})$$

2. From equation (2):

$$y^{(k+1)} = \frac{1}{3}(13 - x^{(k)} - 2z^{(k)})$$

3. From equation (3):

$$z^{(k+1)} = \frac{1}{5}\left(-3 - 2x^{(k)} + 4y^{(k)}\right)$$

Performing Jacobi Iterations

Let's perform 3 iterations using the Jacobi method.
Iteration 1:

- K = 0 (initial guess)

$$x^{(1)} = \frac{1}{4}(7 - 0 - 0) = \frac{7}{4} = 1.75$$

$$y^{(1)} = \frac{1}{3}(13 - 0 - 0) = \frac{13}{3} \approx 4.33$$

$$z^{(1)} = \frac{1}{5}(-3 - 2\times 0 + 4\times 0) = -\frac{3}{5} = -0.6$$

Results after Iteration 1:

$$x^{(1)} \approx 1.75 , \qquad y^{(1)} \approx 4.33 , \qquad z^{(1)} \approx -0.6$$

Iteration 2:

- K = 1

$$x^{(2)} = \frac{1}{4}(7 - 4.33 - (-0.6)) = \frac{1}{4}(7 - 4.33 + 0.6) = \frac{3.27}{4} \approx 0.8175$$

$$y^{(2)} = \frac{1}{3}(13 - 1.75 - 2\times(-0.6)) = \frac{1}{3}(13 - 1.75 + 1.2) = \frac{12.45}{3} \approx 4.15$$

$$z^{(2)} = \frac{1}{5}(-3 - 2\times1.75 + 4\times4.33) = \frac{1}{5}(-3 - 3.5 + 17.32) = \frac{10.82}{5} = 2.164$$

Results after Iteration 2:

$$x^{(2)} \approx 0.8175\,, \quad y^{(2)} \approx 4.15\,, \quad z^{(2)} \approx 2.164$$

Iteration 3:

- K = 2

$$x^{(3)} = \frac{1}{4}(7 - 4.15 - 2.164) = \frac{1}{4}(7 - 6.314) = \frac{0.686}{4} \approx 0.1715$$

$$y^{(3)} = \frac{1}{3}\left(13 - 0.8175 - 2\times(2.164)\right) = \frac{1}{3}(13 - 0.8175 - 4.328) = \frac{7.8545}{3} \approx 2.6182$$

$$z^{(3)} = \frac{1}{5}(-3 - 2\times0.8175 + 4\times4.15) = \frac{1}{5}(-3 - 1.635 + 16.6) = \frac{11.965}{5} \approx 2.393$$

Results after Iteration 3:

$$x^{(3)} \approx 0.1715\,, \quad y^{(3)} \approx 2.6182\,, \quad z^{(3)} \approx 2.393$$

Step 3: Transition to Gauss-Seidel Method
Now that we have some updated values from the Jacobi method, we will switch to the Gauss-Seidel method to speed up convergence.

Gauss-Seidel Update Formulas
The Gauss-Seidel update equations based on our system are:

1. From equation (1):

$$x^{(k+1)} = \frac{1}{4}(7 - y^{(k)} - z^{(k)})$$

2. From equation (2):

$$y^{(k+1)} = \frac{1}{3}(13 - x^{(k+1)} - 2z^{(k)})$$

3. From equation (3):

$$z^{(k+1)} = \frac{1}{5}(-3 - 2x^{(k+1)} + 4y^{(k+1)})$$

Performing Gauss-Seidel Iterations
Now, we will perform 3 iterations with the updated values from the last Jacobi iteration.

Iteration 1 (Gauss-Seidel):

- K = 0 ($x^{(0)}$, $y^{(0)}$, $z^{(0)}$) ≈ (0.1715 , 2.6182 , 2.393)

$$x^{(1)} = \frac{1}{4}(7 - 2.6182 - 2.393) = \frac{1}{4}(7 - 5.0112) = \frac{1.9888}{4} \approx 0.4972$$

$$y^{(1)} = \frac{1}{3}(13 - 0.4972 - 2\times 2.393) = \frac{1}{3}(13 - 0.4972 - 4.786) = \frac{7.7168}{3} \approx 2.5723$$

$$z^{(1)} = \frac{1}{5}(-3 - 2\times 0.4972 + 4\times 2.5723)$$

$$= \frac{1}{5}(-3 - 0.9944 + 10.2892)$$

$$= \frac{6.2948}{5} \approx 1.2590$$

Results after Gauss-Seidel Iteration 1:

$$x^{(1)} \approx 0.4972, \quad y^{(1)} \approx 2.5723\,, \quad z^{(1)} \approx 1.2590$$

Iteration 2 (Gauss-Seidel)

- K = 1

$$\mathrm{x}^{(2)} = \frac{1}{4}(7 - 2.5723 - 1.2590) = \frac{1}{4}(7 - 3.8313) = \frac{3.1687}{4} \approx 0.7922$$

$$y^{(2)} = \frac{1}{3}(13 - 0.7922 - 2\times 1.2590) = \frac{1}{3}(13 - 0.7922 - 2.518) = \frac{9.6898}{3}$$

$$\approx 3.2299$$

$$z^{(2)} = \frac{1}{5}(-3 - 2\times 0.7922 + 4\times 3.2299)$$
$$= \frac{1}{5}(-3 - 1.5844 + 12.9196)$$
$$= \frac{8.3352}{5} \approx 1.6670$$

Results after Gauss-Seidel Iteration 2

$$x^{(2)} \approx 0.7922\,, \qquad y^{(2)} \approx 3.2299\,, \qquad z^{(2)} \approx 1.6670$$

Iteration 3 (Gauss-Seidel)

- K =2

$$x^{(3)} = \frac{1}{4}(7 - 3.2299 - 1.6670) = \frac{1}{4}(7 - 4.8969) = \frac{2.1031}{4} \approx 0.5258$$

$$y^{(3)} = \frac{1}{3}(13 - 05258 - 2\times 1.6670) = \frac{1}{3}(13 - 0.5258 - 3.334)$$
$$= \frac{9.1402}{3}$$
$$\approx 3.0467$$

$$z^{(3)} = \frac{1}{5}(-3 - 2\times 0.5258 + 4\times 3.0467) = \frac{1}{5} = \frac{8.1352}{5} \approx 1.6270$$

Results after Gauss-Seidel Iteration 3

$$x^{(3)} \approx 0.5258\,, \qquad y^{(3)} \approx 3.0467\,, \qquad z^{(3)} \approx 1.6270$$

This example demonstrates how to combine the Jacobi method for initial iterations—allowing for easy parallel computation—followed by the Gauss-Seidel method for refined convergence. This hybrid approach leverages the strengths of both methods: the Jacobi method's parallelism benefits with initial estimates, and the rapid convergence of the Gauss-Seidel method afterward.

In the above example, the Jacobi method is used to solve the initial iterations, where the Jacobi method is amenable to parallel processing, and the Gauss-Seidel method is used to get a better rate of convergence. This is beneficial since the Jacobi method is effective and useful during parallel processing, and the Gauss-Seidel method is highly convergent.

Conclusion

In this paper, the performance and convergence properties of the Jacobi and the Gauss Seidel methods for the solution of the linear systems were considered. Analysis has shown that although the two methods were effective, they differed distinctly.

The Gauss-Seidel algorithm was more efficient, as it reached stable points in fewer iterations compared to the Jacobi method. This implies that for fast convergence problems, the most preferred algorithm is the Gauss-Seidel algorithm. The Jacobi method remains valid, mainly for parallel computation problems where it acts independently when calculating values.

In conclusion, our results further emphasize the significance of choosing a fitting algorithm according to the requirements of a given problem. Future research could include exploration of hybrid solutions or improvements of such algorithms to optimize their performance for different tasks.

In the same research, we also examined different ways to improve the Jacobi algorithm used to solve the linear system by combining different Gauss iterative methods. We used an iterative method to implement and evaluate different techniques among others, such as the Hybrid Jacobi G-S algorithm, Successive Over Relaxation, reordering, and the use of the preconditioning method.

The analysis showed that the use of the given improvement techniques has resulted in enhanced convergence and accuracy than the conventional Jacobi method. In this context, the comparison between the Jacobi, Hybrid, and SOR methods has found that the number of iterations for convergence and the final error were reduced. Additionally, reordering the equations has enhanced the convergence property, while the use of the preconditioning technique has been the most effective method for faster convergence.

The results obtained stress the need to make modifications and improvements to iterative methods in computational mathematics. The proposed methods achieve optimization of the Jacobi iterative approach, making it more efficient. Additionally, these methods have potential applications in solving large linear systems. Other areas where optimizations may be developed based on these strategies include numerical problem solving.

References

1. da Silva Menezes, Matheus, et al. "A Parallel Iterative hybrid Gauss–Jacobi–Seidel method." *Journal of Computational and Applied Mathematics* 468 (2025): 116629.

2. Wu, Xiaowen, and Zhengge Huang. "THE EFFECTIVE AND MODIFIED JACOBI GRADIENT BASED ITERATIVE ALGORITHMS FOR THE DISCRETE-TIME PERIODIC SYLVESTER MATRIX EQUATIONS." Journal of Applied Analysis & Computation 15, no. 4 (2025): 2044-2088.
3. Khrapov, P. V., and N. S. Volkov. "Comparative analysis of Jacobi and Gauss-Seidel iterative methods." *International Journal of Open Information Technologies* 12, no. 2 (2024): 23-34.
4. da Silva Menezes, M., Silva, P.H.L., de Oliveira, J.P.C., Marques, R.L.A. and Mezzomo, I., 2025. A Parallel Iterative hybrid Gauss–Jacobi–Seidel method. *Journal of Computational and Applied Mathematics*, *468*, p.116629.
5. da Silva Menezes, Matheus, Paulo Henrique Lopes Silva, João Paulo Caraú de Oliveira, Raimundo Leandro Andrade Marques, and Ivan Mezzomo. "A Parallel Iterative hybrid Gauss–Jacobi–Seidel method." *Journal of Computational and Applied Mathematics* 468 (2025): 116629.
6. Karami, Afsaneh, and Graham Finlayson. "Gauss-Seidel Retinex." *Journal of Physics: Conference Series*. Vol. 3128. No. 1. IOP Publishing, 2025.

7. Lan, L., Lu, Z., Yuan, C., Xu, W., Su, H., Wang, H., ... & Yang, Y. (2025). JGS2: Near Second-order Converging Jacobi/Gauss-Seidel for GPU Elastodynamics. *arXiv preprint arXiv:2506.06494*.
8. Zhang, Z., Hong, R., Chen, X., & Cai, S. (2025). Hybrid-Precision Block-Jacobi Preconditioned GMRES Solver for Linear System in Circuit Simulation. *arXiv preprint arXiv:2509.09139*.
9. Lan, Lei, et al. "JGS2: Near Second-order Converging Jacobi/Gauss-Seidel for GPU Elastodynamics." *arXiv preprint arXiv:2506.06494* (2025).
10. Elman, H. (2025). Preconditioning Techniques for Newton's Method for the Incompressible Navier–Stokes Equations. *ResearchGate*.
11. Bokhshan, S. (2025). A New Iterative Method for Solving Linear Systems of Equations. *Jordan Journal of Mathematics and Statistics*.
12. Sun, S.L. (2021). Adaptive Relaxation Strategy on Basic Iterative Methods for Solving Linear Systems with Single and Multiple Right-Hand Sides. *Global Science Press*.
13. J. Zhang, G. Zhou, Y. Chen, and Z. Zheng, "Fast Converging Gauss–Seidel Iterative Algorithm for Massive MIMO Systems," *MDPI*, vol. 13, no. 23, 2023.
14. Guo et al. (2024) leveraged GPU for fast sparse matrix-vector computation to speed up nonlinear Newton Krylov solve. *arXiv*, 2506.06494v1
15. Karami, A., & Finlayson, G. (2025, October). Gauss-Seidel Retinex. In Journal of Physics: Conference Series (Vol. 3128, No. 1, p. 012003). IOP Publishing.
16. Ahmadi, Afshin, et al. "A parallel Jacobi-embedded Gauss-Seidel method." IEEE Transactions on Parallel and Distributed Systems 32.6 (2021): 1452-1464.
17. Agboola, Sunday O., Semiu A. Ayinde, Olajide Ibikunle, and Abiodun D. Obaromi. "Application of Jacobi and Gauss–Seidel Numerical Iterative Solution Methods for the Stationary Distribution of Markov Chain." Dutse Journal of Pure and Applied Sciences 9, no. 1a (2023): 127-138.
18. Chen et al. (2024b). Chen et al. (2024b) exploited the global quadratic approximation quality to control the local eigenvalue projection in projected Newton. *arXiv*, 2506.06494v1.
19. K. L. Brown, "Modern Techniques in Numerical Linear Algebra," *Computational Mathematics and Numerical Analysis*, vol. 32, no. 2, pp. 125-139, April 2025.
20. Y. V. Lin and J. A. Zhang, "Iterative Methods: Techniques and Their Applications," *Mathematical Reviews*, vol. 135, pp. 67-78, January 2024.
21. Q. R. Wang, "Matrix Methods in Computational Mathematics," *Applied Mathematics Reviews*, vol. 125, no. 9, pp. 234-250, September 2025.
22. R. S. Fraser, "Iterative Methods for Linear Problems," *Journal of Algorithms and Computation*, vol. 44, no. 3, pp. 145-158, July 2025.
23. D. M. Ritchie, "Optimization Techniques for Matrix Solutions," *Computational Engineering*, vol. 29, no. 8, pp. 289-301, February 2024.
24. S. P. Anderson, "Fast Iterative Methods for Large Systems," *Engineering Computations*, vol. 31, pp. 99-117, March 2025.
25. J. S. Tan, "Convergence Analysis of Gauss-Seidel Variants," *International Journal of Applied Mathematics*, vol. 20, no. 7, pp. 333-349, September 2024.
26. Agboola, Sunday O., et al. "Application of Jacobi and Gauss–Seidel Numerical Iterative Solution Methods for the Stationary Distribution of Markov Chain." Dutse Journal of Pure and Applied Sciences 9.1a (2023): 127-138.