

Solving Systems of Linear Equations

Introduction	1
Diagonal Dominance Condition	1
Residual Vector	2
Analysis of Matrix A Cases and b vector:	2
Visualization Of A matrix and b vector	3
Comparison of Iterative Methods for a Convergent Case	3
Plot TODO: change	3
Comparison of Iterative Methods for a Divergent Matrix	4
Plot TODO: change	4
LU Decomposition	5
Plot TODO: change	5
Execution Time Comparison of Methods	6
Observations:	6
Non-Convergent Matrix plot	6
Convergent Matrix Analysis	7
Summary	7

Introduction

The objective of this report is to analyze various numerical methods used to solve systems of linear equations in the form:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Matrix **A** is a symmetric pentadiagonal matrix of size **N × N**. The diagonal values are defined as follows:

- **a₁**: Values on the main diagonal.
- **a₂ = a₃ = -1**: Values located 1 and 2 positions away from the main diagonal (both above and below).

Diagonal Dominance Condition

A matrix **A** is considered strictly diagonally dominant if, for every row, the absolute value of the diagonal element is greater than the sum of the absolute values of all other elements in that row.

The mathematical condition is expressed as: $|a_{ii}| > \sum |a_{ij}|$ for $j \neq i$

Residual Vector

For calculating the residual norm, the **L1 norm** (where **p = 1**) was used, defined by the following formula:

The residual vector **r** is defined as the difference between the right-hand side vector **b** and the product of matrix **A** and the approximated solution vector **x**:

$$\mathbf{r} = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}$$

The norm provides a numerical measure of how well the calculated solution satisfies the original system of equations. In engineering applications, the **L1 norm** (also known as the Manhattan or taxicab norm) represents the sum of the absolute values of the residual components.

Analysis of Matrix A Cases and b vector:

Matrix A Convergent Case (**a₁ = 8**):

- a. Diagonal element: $|8| = 8$
- b. Sum of off-diagonal elements: $|-1| + |-1| + |-1| + |-1| = 4$

- c. Condition: $8 > 4$ (True)
- d. Result: The matrix is strictly diagonally dominant. This guarantees that iterative methods (Jacobi, Gauss-Seidel) will converge to the correct solution.

Matrix A Divergent Case ($a_1 = 3$):

- e. Diagonal element: $|3| = 3$
- f. Sum of off-diagonal elements: $|-1| + |-1| + |-1| + |-1| = 4$
- g. Condition: $3 > 4$ (False)
- h. Result: The matrix is not diagonally dominant. In this case, the spectral radius of the iteration matrix often exceeds 1, leading to the divergence of iterative algorithms.

The matrices and vectors have sizes $\mathbf{N} \in \{100, 500, 1000, 2000, 3000\}$

Vector b is defined as follows:

$$b[i] = \sin(i \cdot (f + 1)) \text{ where } f = 8$$

Visualization Of A example matrix and b vector

Matrix A

```
[[ 8. -1. -1. ... 0. 0. 0.]
 [-1. 8. -1. ... 0. 0. 0.]
 [-1. -1. 8. ... 0. 0. 0.]
 ...
 [ 0. 0. 0. ... 8. -1. -1.]
 [ 0. 0. 0. ... -1. 8. -1.]
 [ 0. 0. 0. ... -1. -1. 8.]]
```

Vector b

```
[ 0. 0.4121 -0.7510 ... -0.3039 0.6695 -0.9161 ]
```

Comparison of Iterative Methods for a Convergent Case

Conclusions

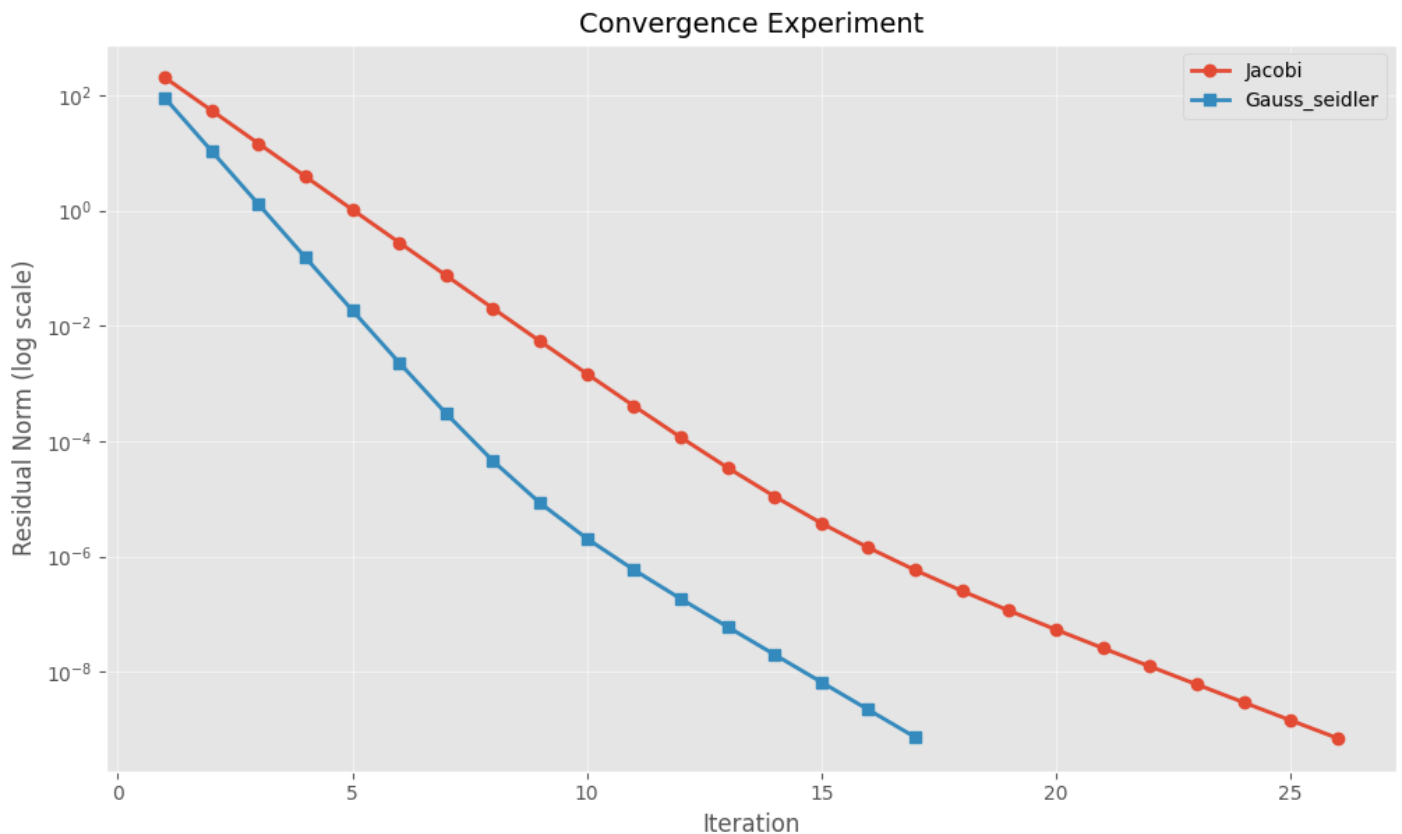
For the matrix with $\mathbf{a1} = 8$ and $\mathbf{N} = 1200$, Jacobi and Gauss-Seidel iterative methods were applied. The termination criterion for the algorithm was reaching a residual norm of less than 10^{-9} .

- **Jacobi Method:** Achieved the required residual after **26 iterations**.
- **Gauss-Seidel Method:** Was faster, concluding after **17 iterations**.
- **Execution Times:** Differences in processing times in favor of Jacobi is result of vectorizing calculations which in NumPy (written in c) is much faster.

- **Jacobi:** 0.0769871 s
- **Gauss-Seidel:** 0.1260815s

On the plots, we observe the decrease of the residual norm in subsequent iterations on a logarithmic scale.

Plot



Comparison of Iterative Methods for a Divergent Matrix

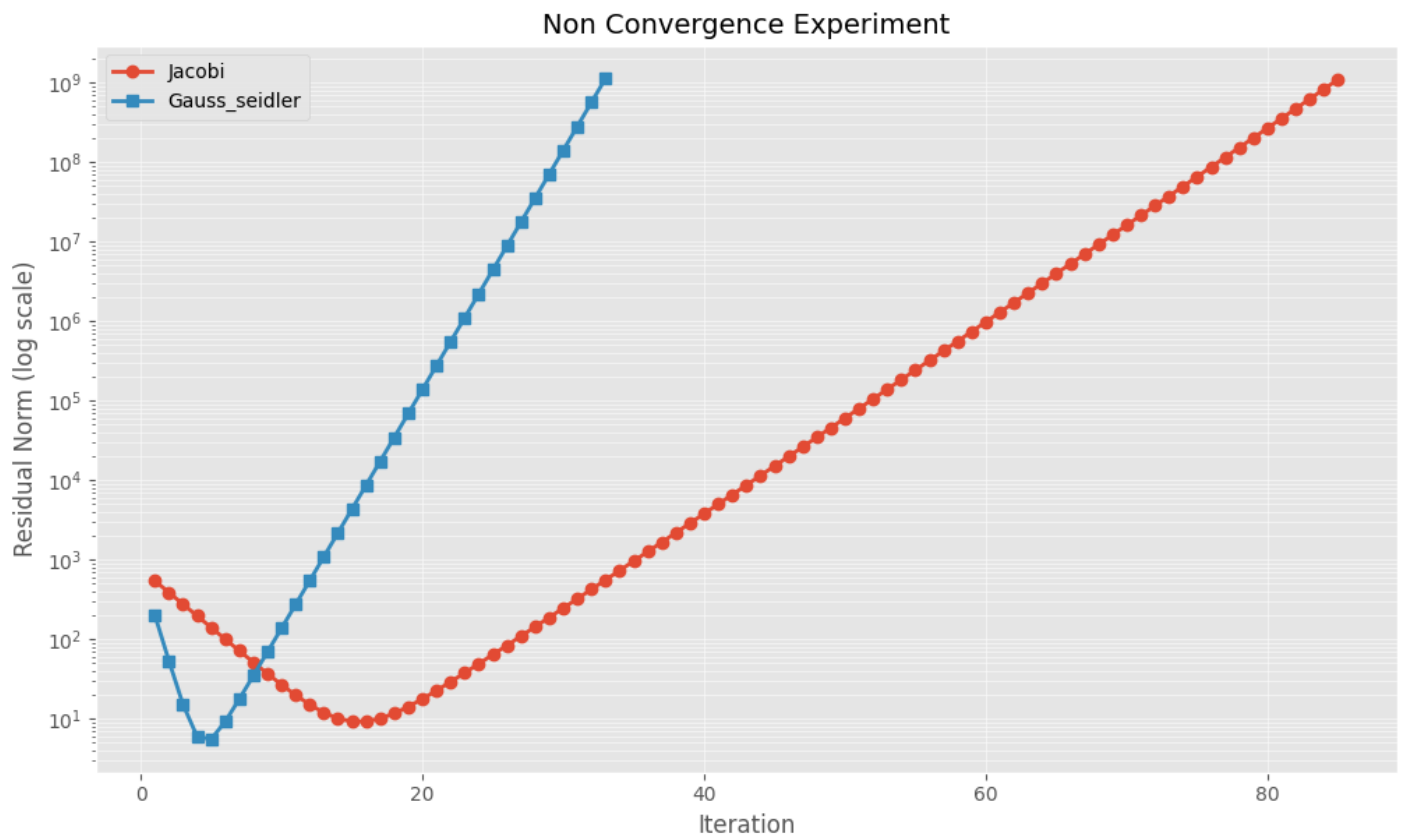
Conclusions:

For the matrix where **a1 = 3** and **N = 1200**, the behavior of iterative methods was tested in the absence of the convergence condition.

- **Jacobi Method:** Reached a residual in the order of 10^9 after approximately 86 iterations.
- **Gauss-Seidel Method:** Reached a similar residual level after only 32 iterations.
- In this case, the methods are **non-convergent**, which aligns with the theory for matrices that do not satisfy the diagonal dominance criterion.
- A temporary decrease in the residual can be observed during the initial stage of the algorithm execution.
- **Execution Times:** Times Difference are similar in this case.
 - **Jacobi:** 0.2242017 s

○ **Gauss-Seidel:** 0.2419733 s

Plot



LU Decomposition

A direct method for solving the system of equations using LU decomposition was implemented.

For the divergent matrix where **a1 = 3**, **N = 1200**:

- **Residual Norm:** $7.48 \cdot 10^{-13}$
- **Execution Time:** 3.436337 s

Conclusion:

The LU method provides high accuracy, but is significantly slower than iterative methods for large matrix sizes.

Execution Time Comparison of Methods

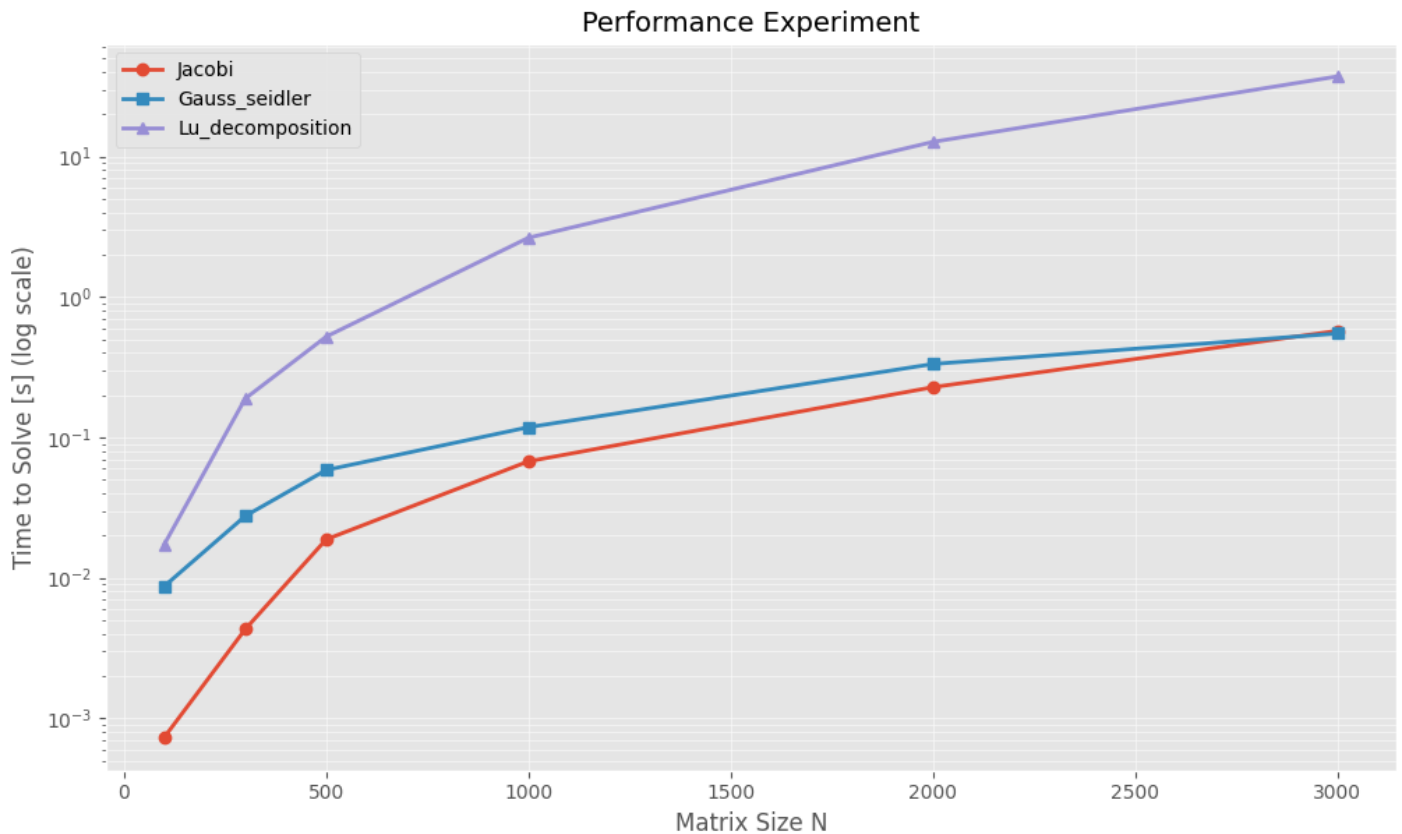
The Jacobi, Gauss-Seidel, and LU methods were analyzed regarding execution time for various sizes of the convergent matrix where $a_1 = 8$.

Matrix Sizes: $N \in \{100, 500, 1000, 2000, 3000\}$

Observations:

- Gauss-Seidel Method: This is the fastest method—even for $N = 3000$, the execution time remains below 1 s.
- Jacobi Method: Slightly slower for larger matrices, though marginally faster for smaller sizes.
- LU Method: Drastically slower for $N = 3000$, with the execution time exceeding 30 s.

Convergent Matrix Plot



Summary

Based on the numerical experiments, the following conclusions regarding the performance and stability of the algorithms have been formulated:

- **Performance of Iterative Methods:** The Jacobi and Gauss-Seidel methods prove to be highly efficient for systems where the matrix \mathbf{A} is strictly diagonally dominant ($|a_{ii}| > 4$). In such cases, these methods reach the required tolerance (10^{-9}) in a relatively low number of iterations, making them suitable for large-scale sparse systems.
- **Method Efficiency Comparison:** The **Gauss-Seidel** method consistently outperforms the Jacobi method in terms of convergence rate. Because Gauss-Seidel utilizes the most recently updated values of \mathbf{x} within the same iteration, it typically requires approximately half as many iterations to reach the same residual norm level.
- **Convergence Failure:** For the non-convergent case ($\mathbf{a}_1 = 3$), the iterative methods failed to produce a valid solution. The residual norm increased exponentially after a short initial period, confirming that diagonal dominance (or the spectral radius $\rho(\mathbf{M}) < 1$) is a critical requirement for these solvers.

- **Direct vs. Iterative Solvers:** The **LU decomposition** method provided the most accurate results, with a residual norm near the limits of double-precision floating-point arithmetic ($\sim 10^{-13}$). However, its computational complexity of $O(N^3)$ results in a dramatic increase in execution time for large **N**. While iterative methods solved the **N = 3000** system in less than a second, the LU method required over **30 seconds**, highlighting the trade-off between absolute precision and computational cost.

The report successfully validates the theoretical properties of these numerical methods. It demonstrates that while direct methods like LU are robust and accurate for any non-singular matrix, iterative methods offer superior performance for large, diagonally dominant systems.