# Solving Systems of Linear Equations Ax = b, where A is a Pentadiagonal Matrix, using the Gauss-Seidel Method.

Zuzanna Sieńko

December 2023

## 1 Pentadiagonal Matrix

A pentadiagonal matrix is a special type of square matrix that has nonzero elements only on the main diagonal and the two diagonals above and below the main diagonal. This means that most elements of this matrix are zeros, and only five diagonals contain elements that may be different from zero.

A pentadiagonal matrix $A$ of dimensions $n \times n$ is defined as:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \cdots & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & \cdots & a_{n-2,n} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{n,n-1} & a_{n,n-2} \\ 0 & 0 & 0 & 0 & \cdots & a_{n-1,n} & a_{nn} \end{bmatrix} \tag{1}$$

where:

- $a_{11}, a_{22}, \ldots, a_{nn}$ are the elements on the main diagonal,
- $a_{12}, a_{23}, \ldots, a_{n-1,n}$ are the elements on the diagonal directly above the main,
- $a_{21}, a_{32}, \ldots, a_{n,n-1}$ are the elements on the diagonal directly below the main,
- $a_{13}, a_{24}, \ldots, a_{n-2,n}$ are the elements on the second diagonal above the main,
- $a_{31}, a_{42}, \ldots, a_{n,n-2}$ are the elements on the second diagonal below the main.

### 1.1 Computer Representation

For a pentadiagonal matrix of dimensions $n \times n$, we can store the nonzero diagonals as a $5 \times n$ matrix in the following way:

- The first row (Second diagonal above the main, starting from the top left corner): $[a_{1,3}, a_{2,4}, \ldots, a_{n-2,n}, 0, 0]$
- The second row (First diagonal above the main): $[a_{1,2}, a_{2,3}, \ldots, a_{n-1,n}, 0]$
- The third row (Main diagonal): $[a_{1,1}, a_{2,2}, \ldots, a_{n,n}]$
- The fourth row (First diagonal below the main): $[a_{2,1}, a_{3,2}, \ldots, a_{n,n-1}, 0]$
- The fifth row (Second diagonal below the main): $[a_{3,1}, a_{4,2}, \ldots, a_{n,n-2}, 0, 0]$

## 2 The Gauss-Seidel Method

The Gauss-Seidel method is an iterative technique used for finding solutions to systems of linear equations. It starts with a randomly chosen initial approximation of the solution, denoted as $\mathbf{x}^{(0)}$. In each subsequent iteration, this method successively adjusts individual components of this approximation to more accurately match the actual solution. The Gauss-Seidel method builds on the idea of the Jacobi method but introduces a key modification: each update utilizes the most recent approximate values of components. This approach contributes to reduced memory usage and often shortens the time needed to achieve the desired accuracy of the solution. It is important that the main matrix is diagonally dominant.

### Diagonally Dominant Matrix

A diagonally dominant matrix is a square matrix in which the absolute value of each element on the main diagonal is greater than the sum of the absolute values of the other elements in the same row. Formally, for a matrix $A = [a_{ij}]$ with dimensions $n \times n$, the matrix is diagonally dominant if:

$$\forall i \in \{1, 2, ..., n\}, \quad |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

Such a property of the matrix is particularly important in the context of iterative methods, such as the Gauss-Seidel method, because it ensures better convergence of the algorithm.

## 3 Method Description

Consider a system of $n$ linear equations with $n$ unknowns:

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

Where $\mathbf{A}$ is the matrix of coefficients, $\mathbf{x}$ is the vector of unknowns, and $\mathbf{b}$ is the vector of constants.

In the Gauss-Seidel method, a single iteration can be expressed as:

$$\mathbf{x}^{(k+1)} = (\mathbf{D} + \mathbf{L})^{-1}(-\mathbf{U}\mathbf{x}^{(k)} + \mathbf{b}),$$

where:

- $\mathbf{D}$ is a non-singular diagonal matrix,
- $\mathbf{L}$ and $\mathbf{U}$ are the lower and upper triangular matrices of $\mathbf{A}$, respectively, with zeros on the main diagonal,
- $\mathbf{A} \equiv \mathbf{D} + \mathbf{L} + \mathbf{U}$.

The index $k$ denotes the iteration number.

We consider the Gauss-Seidel method written in matrix form:

$$\mathbf{x}^{(k+1)} = \mathbf{B}_{GS}\mathbf{x}^{(k)} + \mathbf{c}_{GS}, \quad k = 0, 1, \ldots, \tag{2}$$

where

$$\mathbf{B}_{GS} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U} \tag{3}$$

and

$$\mathbf{c}_{GS} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{b}. \tag{4}$$

## 4 Iterative Formula

Breaking down the above formula into components, we obtain the formula used in numerical implementations:

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

In this iteration, each element $x_i^{(k+1)}$ is calculated using the most recent available values. For elements $x_j$ with index $j < i$, we use values from the current iteration $(k+1)$, while for $j > i$ we use values from the previous iteration $k$. This sequential updating of elements is a key feature that distinguishes the Gauss-Seidel method from the Jacobi method.

### 4.1 Preference for the Sequential Form in the Gauss-Seidel Method

In the Gauss-Seidel method, a sequential approach is typically preferred over the matrix form for several key reasons:

1. Utilization of updated values: The sequential form allows for the use of already updated values from the current iteration when calculating subsequent components of the solution vector, which can lead to faster convergence compared to the matrix form, which relies solely on values from the previous iteration.

2. Computational complexity: The matrix form requires inverting a matrix or solving a linear system in each iteration, which is computationally expensive, especially for large matrices. In the sequential approach, each vector component is updated directly, reducing computational complexity.

3. Memory efficiency: The sequential method is more memory-efficient as it does not require storing additional matrices that are necessary in the matrix form.

4. Numerical stability: Direct updates made on the solution vector can be more numerically stable than matrix operations, especially in the case of ill-conditioned matrices.

5. Practical application: In many numerical applications, especially in solving large systems of linear equations with sparse matrices, the sequential approach is more efficient than the matrix approach, avoiding the computation and storage of full matrices.

## 5 Program Description

To simulate the Gauss-Seidel method solving the system of linear equations Ax = b, where A is a pentadiagonal matrix, the following methods were implemented:

- **gaussSeidel(A, b, x0, tolerance, maxIterations)**: Function calculating the solution of the system of equations $B\mathbf{x} = \mathbf{b}$, where $B$ is a pentadiagonal matrix. The function accepts the following parameters:
  - $A$: Computer representation of the diagonals of matrix $B$, where each row corresponds to one of the diagonals.
  - $\mathbf{b}$: The vector of constants in the system of equations.
  - $\mathbf{x}_0$: Initial approximation of the result.
  - *tolerance*: The computational error for which we end the iterative search for the solution.
  - *maxIterations*: The maximum number of iterations that guarantees the end of the computational process if a result meeting the specified level of tolerance cannot be achieved.

  The function returns the solution of the system of equations using the Gauss-Seidel method.

- **convertToFiveDiagonal(A)**: Function transforming a matrix that represents the diagonals of a pentadiagonal matrix into its corresponding pentadiagonal matrix. The function accepts the following parameters:
  - $A$: Matrix where each row represents one of the diagonals of the pentadiagonal matrix.

  The function returns matrix $B$, which is the full representation of the pentadiagonal matrix.

- **compare(A, B, b, x0, tolerance, maxIterations)**: Function comparing the results of the Gauss-Seidel method for pentadiagonal matrices with the built-in method for solving systems of equations in MATLAB. The function accepts the following parameters:
  - $A$: Computer representation of the diagonals of matrix $B$.
  - $B$: Transformed computer representation of $A$ into matrix $B$.
  - $\mathbf{b}$: The vector of constants in the system of equations.
  - $\mathbf{x}_0$: Initial approximation of the result.
  - *tolerance*: The computational error for which we end the iterative search for the solution.
  - *maxIterations*: The maximum number of iterations that guarantees the end of the computational process if a result meeting the specified level of tolerance cannot be achieved.

  The function returns a table containing the execution time, computational error, and comparative error of both methods.

## 6 Examples

In this section, I present the results of applying the Gauss-Seidel method to solve a system of linear equations in the form $Ax = b$.

### 6.1 Example 1

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 11 & 2 & 1 & 0 & 0 \\ 2 & 12 & 2 & 1 & 0 \\ 1 & 2 & 13 & 2 & 1 \\ 0 & 1 & 2 & 14 & 2 \\ 0 & 0 & 1 & 2 & 15 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

and $x$ is the vector of unknowns.

After applying the Gauss-Seidel method, the following solution for the vector $x$ was obtained:

$$x = \begin{pmatrix} 0.075826482691243 \\ 0.058052428667271 \\ 0.049803833061781 \\ 0.052110224486704 \\ 0.056398381197654 \end{pmatrix} \approx \begin{pmatrix} 0.0758 \\ 0.0581 \\ 0.0498 \\ 0.0521 \\ 0.0564 \end{pmatrix}$$

The efficiency and accuracy of the Gauss-Seidel method and the built-in method for solving linear systems of equations were analyzed using the following parameters:

- Size of matrix $A$: $5 \times 5$.
- Initial approximation: zero vector $[0, 0, 0, 0, 0]$.
- Maximum number of iterations for Gauss-Seidel: 1000.
- Error tolerance: $10^{-16}$.

The obtained results are as follows:

| Method | Time [s] | Error |
|---|---|---|
| Gauss-Seidel | 0.0013331 | 0 |
| Built-in | 0.0001202 | $2.72 \times 10^{-16}$ |

The Gauss-Seidel method, despite a higher execution time, achieved a zero computational error. Meanwhile, the built-in method, being significantly faster, showed a minimal error of the order of $10^{-16}$.

### 6.2 Example 2

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 87 & 3 & 1 & 0 & 0 \\ 4 & 22 & 1 & 5 & 0 \\ 1 & 6 & 38 & 6 & 2 \\ 0 & 2 & 13 & 11 & 5 \\ 0 & 0 & 7 & 2 & 7 \end{pmatrix}, \quad b = \begin{pmatrix} 234 \\ 80 \\ 190 \\ 301 \\ 107 \end{pmatrix}$$

and $x$ is the vector of unknowns.

After applying the Gauss-Seidel method, the following approximate solutions for the vector were obtained:

1. For the initial approximation $x_0 = [0, 0, 0, 0, 0]$:

$$x_1 = \begin{pmatrix} 2.748093728685375 \\ -2.114250614565031 \\ 1.258597448067495 \\ 22.852508231524340 \\ 7.497828771496979 \end{pmatrix} \approx \begin{pmatrix} 2.75 \\ -2.11 \\ 1.26 \\ 22.85 \\ 7.50 \end{pmatrix}$$

2. For the initial approximation randomly generated $x_0 \approx [1.53, -0.77, 0.37, -0.23, 1.12]$:

$$x_2 = \begin{pmatrix} 2.7481 \\ -2.1143 \\ 1.2586 \\ 22.8525 \\ 7.4978 \end{pmatrix} \approx \begin{pmatrix} 2.75 \\ -2.11 \\ 1.26 \\ 22.85 \\ 7.50 \end{pmatrix}$$

The efficiency and accuracy of the Gauss-Seidel method and the built-in method for solving linear systems of equations were analyzed using the following parameters:

- Size of matrix $A$: $5 \times 5$.
- Vector $b$: $[234, 80, 190, 301, 107]$.
- Initial approximation 1: $x_0 = [0, 0, 0, 0, 0]$.
- Initial approximation 2: $x_0 = [1.53, -0.77, 0.37, -0.23, 1.12]$.
- Maximum number of iterations for Gauss-Seidel: 1000.
- Error tolerance: $10^{-16}$.

latex Copy code The obtained results are as follows:

| Method | Time 1 [s] | Error 1 | Time 2 [s] | Error 2 |
|---|---|---|---|---|
| Gauss-Seidel | $6.68 \times 10^{-5}$ | $5.68 \times 10^{-14}$ | 0.000105 | $5.68 \times 10^{-14}$ |
| Built-in | $6.61 \times 10^{-5}$ | $3.18 \times 10^{-14}$ | 0.0002069 | $3.18 \times 10^{-14}$ |

It's noteworthy that both the Gauss-Seidel method and the built-in method achieved very low computational errors, indicating their high accuracy. The execution time is also at a very low level, suggesting the efficiency of these methods. Importantly, the results do not differ significantly depending on the choice of initial approximation $x_0$. This is significant because, in the Gauss-Seidel method, the choice of initial approximation $x_0$ does not affect the convergence of the method itself. By definition, $x_0$ can be any vector, which offers great flexibility in applying this method to various problems. This property is especially beneficial in cases where there are no clear guidelines for the optimal choice of initial approximation, allowing the method to be applied across a wide range of applications without the need for detailed analysis of initial conditions.

## 6.3 Example 3

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 10 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 10 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 10 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 10 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 10 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 100 \\ 100 \\ \vdots \\ \vdots \\ \vdots \\ 100 \\ 100 \end{pmatrix}$$

The obtained results are as follows:

| Method | Time [s] | Error |
|---|---|---|
| Gauss-Seidel | 0.0003415 | $2.46 \times 10^{-14}$ |
| Built-in | 0.0001647 | $2.84 \times 10^{-14}$ |

The results show that both the Gauss-Seidel method and the built-in method provide very close solutions with a small computational error. The Gauss-Seidel method achieved a minimal error of the order of $10^{-14}$, similar to the built-in method.

## 6.4 Example 4

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 2 & 0 & 0 \\ 1 & 2 & 3 & 3 & 3 & 0 \\ 0 & 2 & 3 & 4 & 4 & 4 \\ 0 & 0 & 3 & 4 & 5 & 5 \\ 0 & 0 & 0 & 4 & 5 & 6 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

and $x$ is the vector of unknowns.

After applying the Gauss-Seidel method, the following approximate solution for the vector $x$ was obtained:

$$x = \begin{pmatrix} 7.96430643303806 \times 10^{17} \\ -1.103666860068391 \times 10^{18} \\ -1.698632310366 \times 10^{16} \\ 1.164278713758498 \times 10^{18} \\ -6.75038327181797 \times 10^{17} \\ -2.13653869854167 \times 10^{17} \end{pmatrix} \approx \begin{pmatrix} 7.96 \times 10^{17} \\ -1.10 \times 10^{18} \\ -1.69 \times 10^{16} \\ 1.16 \times 10^{18} \\ -6.75 \times 10^{17} \\ -2.14 \times 10^{17} \end{pmatrix}$$

while the actual result for the vector $x$ is:

$$x = \begin{pmatrix} 1.428571428571428 \\ -0.224489795918367 \\ -0.204081632653061 \\ 0.714285714285714 \\ 0.163265306122449 \\ 0.387755102040816 \end{pmatrix} \approx \begin{pmatrix} 1.43 \\ -0.22 \\ -0.20 \\ 0.71 \\ 0.16 \\ 0.39 \end{pmatrix}$$

The efficiency and accuracy of the Gauss-Seidel and built-in methods for solving linear systems of equations were analyzed using the following parameters:

- Size of matrix $A$: $6 \times 6$.
- Vector $b$: $[1, 2, 3, 4, 5, 6]$.
- Initial approximation: zero vector $[0, 0, 0, 0, 0, 0]$.
- Maximum number of iterations for Gauss-Seidel: 100.
- Error tolerance: $10^{-16}$.

The obtained results are as follows:

| Method | Time [s] | Error |
|---|---|---|
| Gauss-Seidel | 0.0003383 | $1.49 \times 10^{18}$ |
| Built-in | 0.0001818 | 0 |

The results show that the Gauss-Seidel method did not achieve satisfactory accuracy in this case, obtaining an error of the order of $10^{18}$, while the built-in method provided an exact solution.

This example perfectly illustrates that the Gauss-Seidel method is effective and convergent only for diagonally dominant matrices. For matrices that do not meet this condition, this method may diverge or converge very slowly, leading to inaccurate or unstable results.

## 6.5 Example 5

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 100 & -5 & -2 & \cdots & 0 & 0 & 0 & 0 \\ 4 & 100 & 2 & 1 & \cdots & 0 & 0 & 0 \\ -3 & -1 & 100 & -5 & -2 & \cdots & 0 & 0 \\ 0 & 4 & 4 & 100 & 2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 4 & -1 & 100 & -5 & 1 \\ 0 & 0 & \cdots & 0 & -3 & 4 & 100 & 2 \\ 0 & 0 & 0 & \cdots & 0 & 4 & -1 & 100 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

and $x$ is the vector of unknowns.

After applying the Gauss-Seidel method, the following solution for the vector $x$ was obtained:

$$x = 1.0 \times 10^{-18} \times \begin{pmatrix} 0.012015620215212 \\ 0.010129491809161 \\ -0.001121702353096 \\ 0.010334177908174 \\ \vdots \\ \vdots \\ 0.000000000249650 \end{pmatrix}$$

The efficiency and accuracy analysis of the Gauss-Seidel and built-in methods for solving linear systems of equations was conducted using the following parameters:

- Size of matrix $A$: $100 \times 100$.
- Initial approximation: zero vector $[0, 0, \ldots, 0]$ (100 elements).
- Maximum number of iterations for Gauss-Seidel: 100.
- Error tolerance: $10^{-16}$.

The obtained results are as follows:

| Method | Time [s] | Error |
|---|---|---|
| Gauss-Seidel | 0.0002141 | $2.27 \times 10^{-8}$ |
| Built-in | 0.0006254 | 0 |

The Gauss-Seidel method, despite a longer execution time, achieved a computational error of the order of $10^{-8}$. However, the implemented Gauss-Seidel method is 3 times faster than the built-in method. This suggests that for larger matrices, the Gauss-Seidel method may be the preferred choice for solving systems of equations.

## 6.6 Example 6

Consider the system of equations $Ax = b$, where $A$ and $b$ are defined as:

$$A = \begin{pmatrix} 5 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 1 & 5 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 5 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 & 5 & 1 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 1 & 5 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 5 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

and $x$ is the vector of unknowns.

After applying the Gauss-Seidel method, the following solution for the vector $x$ was obtained:

$$x = \begin{pmatrix} 0.154770871069376 \\ 0.127322003750035 \\ 0.098823640903087 \\ \vdots \\ \vdots \\ \vdots \\ 0.127322003750035 \\ 0.154770871069376 \end{pmatrix}$$

The efficiency and accuracy analysis of the Gauss-Seidel and built-in methods for solving linear systems of equations was conducted using the following parameters:

- Size of matrix $A$: $1000 \times 1000$.
- Initial approximation: zero vector $[0, 0, \ldots, 0]$ (1000 elements).
- Maximum number of iterations for Gauss-Seidel: 1000.
- Error tolerance: $10^{-16}$.

The obtained results are as follows:

| Method | Time [s] | Error |
|---|---|---|
| Gauss-Seidel | 0.0010275 | $4.84 \times 10^{-16}$ |
| Built-in | 0.0487751 | $3.91 \times 10^{-15}$ |

In the context of solving linear systems of equations with a matrix size of $1000 \times 1000$, the Gauss-Seidel method shows a significant advantage in both speed and accuracy compared to the built-in method. Specifically, the Gauss-Seidel method is over 40 times faster than the built-in method, marking a significant improvement in performance. Additionally, the results obtained using the Gauss-Seidel method are more accurate, making it the preferred choice in applications requiring both speed and computational precision.

## 7 Practical Applications

The Gauss-Seidel method, due to its efficiency in solving large (even on the order of millions) systems of linear equations, finds broad application in advanced areas of Machine Learning and Data Science:

1. **Linear and Logistic Regression:** In these fundamental machine learning techniques, the Gauss-Seidel method can be used for efficient computation of model weights, especially in cases where the feature matrix is large and sparse. For example, in predictive analysis of the financial market, where regression models are often applied, the speed and accuracy of this method can significantly accelerate the learning process.

2. **Large Dataset Analysis:** In cases where large datasets are analyzed, there is often a need to solve large systems of linear equations. The Gauss-Seidel method can be used for efficient data processing, especially when fast iterative optimization is required.

3. **Neural Networks:** In deep learning, particularly in optimization algorithms such as backpropagation, the Gauss-Seidel method can be used for efficient calculation of gradients and weight updates. For large neural networks, where calculations can be highly complex, this method can contribute to increased efficiency.

4. **Recommendation Systems:** In recommendation systems, where techniques such as matrix factorization are often used, the Gauss-Seidel method can be utilized for efficient solving of linear equations, which is crucial for quickly generating recommendations.

5. **Statistical Modeling and Econometrics:** In statistical modeling and econometric analysis, where regression techniques and other estimation methods are frequently applied, the Gauss-Seidel method can contribute to increased computational efficiency in the model estimation process.

These examples illustrate how the Gauss-Seidel method can be utilized in various aspects of Machine Learning and Data Science, offering quick and efficient solutions for complex computational problems.