International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 2, Mar-Apr 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Sor vs Gauss-Seidel: Racing Towards Convergence

Aakar Roghelia, Dipti Patel

Department of Mathematics, BVM Engineering College, Vallabh Vidyanagar, Gujarat, India

ABSTRACT

This paper presents a comparative analysis of the Gauss-Seidel and Successive Over-Relaxation (SOR) methods for solving large systems of linear equations. While direct methods like Gaussian elimination provide exact solutions, iterative methods are often preferred for large, sparse systems due to their computational efficiency. The SOR method, an enhanced version of the Gauss-Seidel method, introduces a relaxation parameter (ω) to accelerate convergence. A computational program was developed to determine the optimal relaxation factor and minimize the number of iterations.

KEYWORDS: *Gauss-Seidel method, SOR method, System of linear equations*

How to cite this paper: Aakar Roghelia | Dipti Patel "Sor vs Gauss-Seidel: Racing Towards Convergence" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-2, April 2025, pp.1067-1070, URL: www.ijtsrd.com/papers/ijtsrd78603.pdf

IJTSRD78603

1. INTRODUCTION

The solution of systems of linear equations is essential in mathematics, engineering, natural sciences, finance and many more. E.g. in discretization of ordinary differential equations (ODE) and partial differential equations (PDE) large sparse systems arises. To solve such systems we have direct and indirect (iterative) methods. Direct methods such as Gauss elimination method, LU decomposition methods, although provide exact solution, (except round off errors) are lengthy and time consuming in case of large systems. In such cases it is advisable to apply iterative methods, which provide approximate solution with desired accuracy. Among iterative methods Jacobi and Gauss Seidel method are the oldest and famous. Now a days many improved versions of these methods are available.

In this paper, we discuss one of the improved version of Gauss-Seidel method namely, Successive Over-Relaxation (SOR) method. In section 2, we discuss about the SOR method and its importance. In section 3, we give outline of the program to decide the optimal value of ω and in section 4 we compute two numerical examples through programming to obtain appropriate value of the relaxation factor ω in order to obtain rapid convergence.

2. Successive Over Relaxation method

Successive Over Relaxation (SOR) method was first introduced by Dr. David M. young in his doctoral thesis in 1950 to be used on digital computers. SOR method is an improvement or modification of Gauss-Seidel method, which increase the rate of convergence by introducing a relaxation (scaling) parameter ω [2]. It was first used to solve linear systems that appear in elliptic type particle differential equation [3]. Consider a system of n linear equation in n unknowns. In the matrix format is it denoted by AX = B, where A is a coefficient matrix of order n × n, X is a variable vector to be determined and B is real constant vector of order n × 1. Here the system must be diagonally dominant. If not then equations must be rearranged to make the system diagonally dominant.

Let

$$A = L + \left(1 - \frac{1}{\omega}\right) D + \frac{1}{\omega} D + U$$

$$= \frac{(D + \omega L) + \{(\omega - 1) D + \omega U\}}{\omega}.$$

If $A = \left[a_{ij}\right]_{n\times n}$, then $L = \left[l_{ij}\right]_{n\times n}$ with $l_{ij} = \begin{cases} 0 & if\ i \le j \\ a_{ij} & if\ i > j \end{cases}$, $U = \left[u_{ij}\right]_{n\times n}$ with $u_{ij} = \begin{cases} 0 & if\ i \ge j \\ a_{ij} & if\ i < j \end{cases}$, $D = \left[d_{ij}\right]_{n\times n}$ with $d_{ij} = \begin{cases} a_{ij} & if\ i = j \\ 0 & otherwise \end{cases}$.

So,

$$AX = B$$
$$\Rightarrow (D+\omega L)X + \left[(\omega-1)D + \omega U\right]X = \omega B$$
$$\Rightarrow X = (D+\omega L)^{-1}\left[\left\{(1-\omega)D - \omega U\right\}X + \omega B\right]$$

Which give the iterative formula, $X^{(k+1)} = (D+\omega L)^{-1}\left[\left\{(1-\omega)D - \omega U\right\}X^{(k)} + \omega B\right]$.

Here $(D+\omega L)^{-1}\left[(1-\omega)D - \omega U\right]X$ is called iteration matrix and $(D+\omega L)^{-1}\omega B$ is called iteration vector. The formula for SOR is obtained from above relation which is as follows

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

The following table gives information regarding the solution for different values of ω.

Value of ω	Effect of ω
$\omega = 1$	The method reduces to Gauss-Seidel method
$0 < \omega < 1$	Under relaxation method
$1 < \omega < 2$	Over relaxation method
$\omega \ge 2$	System becomes divergent.

So, we consider $0 < \omega < 2$ for convenience.

Development of program

The value of ω affects the rate of convergence of the method but it is not easy to choose ω by trial and error method. An inappropriate choice will lead to divergence of the solution. To overcome this problem we have developed a program which gives us an optimal value of ω to reduce the number of iterations and accelerate the rate of convergence.

Program Outline

```
Sub SOR_Method_From_Excel()

Dim A(1 To 10, 1 To 10) As Double
Dim b(1 To 10) As Double
Dim x(1 To 10) As Double
Dim x_old(1 To 10) As Double
Dim omega(1 To 200) As Double
Dim i As Long, j As Long, k As Long, m As
Long, n As Long
Dim max_iter As Long
Dim tolerance As Double
Dim filePath As String
Dim externalWB As Workbook
Dim dataSheet As Worksheet
Dim resultSheet As Worksheet

filePath = "E:SOR method new.xlsx" '
Set externalWB = Workbooks.Open(filePath)

Set dataSheet = externalWB.Sheets("Sheet2")

For i = 1 To 10
  For j = 1 To 10
    A(i, j) = dataSheet.Cells(i, j).Value
  Next j
Next i

For i = 1 To 10
  b(i) = dataSheet.Cells(i, 11).Value
Next i

On Error Resume Next
Set resultSheet = externalWB.Sheets("SOR
Results")
  If resultSheet Is Nothing Then
    Set resultSheet =
externalWB.Sheets.Add(After:=externalWB.Sheet
s(externalWB.Sheets.Count))
    resultSheet.Name = "SOR Results"
End If
On Error GoTo 0

resultSheet.Cells.Clear
resultSheet.Cells(1, 1).Value = "Iteration"
resultSheet.Cells(1, 2).Value = "Omega"
resultSheet.Cells(1, 3).Value = "Convergence
Iteration"
resultSheet.Cells(1, 4).Value = "Solution Vector"

For m = 2 To 200
n = 1000

omega(1) = 0
omega(m) = omega(m - 1) + 0.01
max_iter = 100
tolerance = 0.000001

For i = 1 To 10
  x(i) = 0
Next i

For k = 1 To max_iter
  For i = 1 To 10
    x_old(i) = x(i)
Next i

    For i = 1 To 10
    Dim sum1 As Double, sum2 As Double
    sum1 = 0
```

```
    sum2 = 0
    For j = 1 To i - 1
    sum1 = sum1 + A(i, j) * x(j)
    Next j
    For j = i + 1 To 10
    sum2 = sum2 + A(i, j) * x_old(j)
    Next j
    x(i) = (1 - omega(m)) * x_old(i) +
    ((omega(m)) * (b(i) - sum1 - sum2)) / A(i, i)
    Next i

    Dim max_diff As Double
    max_diff = 0
    For i = 1 To 10
        Dim diff As Double
        diff = Abs(x(i) - x_old(i))
        If diff > max_diff Then
            max_diff = diff
        End If
    Next i

    If max_diff < tolerance Then
            Exit For
        End If
    Next k

    If n > k Then
        n = k
    End If

    resultSheet.Cells(m, 1).Value = m - 1
    resultSheet.Cells(m, 2).Value = omega(m)
    resultSheet.Cells(m, 3).Value = n
    Next m

    externalWB.Save
    externalWB.Close

MsgBox "Results saved successfully in the same file."

End Sub
```

3. Findings

We have developed a computational program capable of solving linear systems using both the Successive Over-Relaxation (SOR) method and the Gauss-Seidel method. The program supports systems with up to 10 variables and can be generalized for larger systems by adjusting matrix dimensions and computational parameters. Users can input system matrices.

The program determines optimal relaxation factor ω for the given system of equations and according to given convergence criteria.

To evaluate the performance of the methods, we applied the program to two different examples. In one case (Illustration A), the SOR method demonstrated superior performance with faster convergence, particularly by optimizing the relaxation factor. In the other case (Illustration B), the Gauss-Seidel method outperformed SOR due to the favorable properties of the system matrix, achieving rapid convergence without additional parameter tuning.

This flexible and efficient program serves as a valuable tool for researchers and engineers dealing with large-scale linear systems. It offers detailed insights into the iterative processes of both methods, making it easier to choose the most suitable approach for specific problem types.

Illustration A.

$$\begin{bmatrix} 5 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 7 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 9 & -6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 8 & -5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 6 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 10 & -7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -3 & 11 & -8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -4 & 12 & -9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 & 13 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 9 \\ 18 \\ 25 \\ 35 \\ 45 \\ 60 \\ 70 \\ 80 \\ 90 \\ 100 \end{bmatrix}$$

After applying the developed program with a tolerance value of 0.000001, it was observed that the Gauss-Seidel method converged to the solution in 87 iterations, while the Successive Over-Relaxation (SOR) method achieved convergence in only 27 iterations using an optimal relaxation factor of ω = 1.41. This demonstrates the significant efficiency of the SOR method, reducing the number of iterations by approximately 69% compared to the Gauss-Seidel method. The results highlight the advantage of using the SOR method for faster convergence, particularly when an appropriate relaxation factor is applied.

Illustration B.

$$\begin{bmatrix} 40 & -5 & 3 & 2 & 1 & -1 & 4 & 3 & 2 & -2 \\ -3 & 42 & -7 & 5 & 4 & 1 & -6 & 2 & 1 & 3 \\ 4 & -6 & 45 & -4 & 3 & 2 & 5 & -1 & -3 & 2 \\ -5 & 3 & 4 & 47 & -6 & 1 & -2 & 4 & 5 & -3 \\ 2 & 5 & -3 & 4 & 50 & -7 & 3 & 1 & -5 & 6 \\ -4 & 1 & 3 & 5 & -6 & 53 & -8 & 4 & 2 & 1 \\ 3 & -2 & 5 & -6 & 2 & 4 & 55 & -7 & 3 & -4 \\ 4 & 2 & -5 & 3 & 1 & -8 & 6 & 57 & -4 & 2 \\ -2 & 3 & 4 & 5 & -7 & 1 & 2 & -5 & 60 & -6 \\ 3 & -4 & 2 & -5 & 6 & 4 & -7 & 2 & 1 & 63 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 50 \\ 60 \\ 70 \\ 80 \\ 90 \\ 100 \\ 110 \\ 120 \\ 130 \\ 140 \end{bmatrix}$$

After applying the developed program with a tolerance value of 0.000001, it was observed that the Gauss-Seidel method converged to the solution in 8 iterations. In contrast, the Successive Over-Relaxation (SOR) method did converge, but it did not achieve convergence in fewer than 8 iterations, regardless of the relaxation factor (ω) used. This suggests that while SOR can still provide a solution, it did not offer any improvement in iteration count compared to Gauss-Seidel for this particular system, indicating that the optimal relaxation factor did not significantly accelerate convergence in this case.

4. Conclusion:

The comparative analysis of the Gauss-Seidel and Successive Over-Relaxation (SOR) methods highlights that the performance of each method depends on the nature of the linear system. In some cases, the SOR method significantly reduces the number of iterations when an optimal relaxation factor is used. However, there are cases where Gauss-Seidel performs equally well or even better, especially in systems with favorable matrix properties. The developed program effectively demonstrates these differences and can be adapted for larger systems. Overall, both methods have their strengths, and selecting the appropriate one requires consideration of the specific problem characteristics. After applying program to various examples it is observed that SOR method is more effective in tridiagonal and penta diagonal systems.

5. REFERENCES

[1] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[2] D. M. Young, Iterative Methods for Solving Partial Difference Equations of Elliptic Type, ph.d. dissertation, Harvard University, 1950.

[3] D. M. Young, Iterative methods for solving partial difference equations of elliptic type, Transactions of the American Mathematical Society, 76 (1954), pp. 92–111.

[4] D. M. Young, Iterative Solution of Large Linear Systems, Dover Publications, 1971.