AB Journals

African Journal of Mathematics and Statistics Studies
ISSN 2689-5323 • Volume 9, Issue 3, 2026 • pp. 157–171

RESEARCH NOTE Received: 5 Sep 2026, Accepted: 20 Oct 2026, Published: 23 Nov 2026

Comparison of the Rate of Convergence of Some Iterative Methods in the Solutions of System of Linear Equations

Hannah Oluwatimilehin Elegbede[1*], Iretiolu Yemisi Alabi[2], Prof. Matthew Olufemi Alabi[1]

[1]Department of Physical Sciences, Chrisland University, Abeokuta, Ogun State.
[2]Information and Communication Technology Unit, Ladoke Akintola University of Technology, Ogbomoso.
[*]*Corresponding Author's Email: elegbedannah25@gmail.com*

Abstract

The solution to the system of linear equations cannot be connived due to its importance, however it often becomes very laborious when solving large systems of equations analytically which are very common in contemporary applications such as sparse matrix problems. To circumvent this hurdle, there is need to resort to approximate methods of solution which are often referred to as numerical solution. There is a legion of approximate methods in solving system of linear equations, but this presentation shall be limited to only three methods. The interest lies in the rate of convergency of these methods. Therefore, an attempt shall be made to apply these methods in solving system of linear equations numerically and make an appropriate recommendation on the one that is most preferred based on the rate of convergency.

To this end, the three methods under consideration are Jacobi Iterative Method, Gauss – Seidel Iterative Method and Successive Over – Relaxation Method. These three methods shall be applied to five different illustrations comprising of eight unknown quantities to compare the rate of convergence of these methods.

Keywords: Acceleration Parameter, Convergence, Determinant, Diagonally Dominant, Gaussian Elimination

How to Cite: Elegbede, H. O., Alabi, I. Y., & Alabi, M. O. (2026). Comparison of the Rate of Convergence of Some Iterative Methods in the Solutions of System of Linear Equations. *African Journal of Mathematics and Statistics Studies*, 9(3), 157–171. https://doi.org/10.52589/AJMSS-KGIHCE

© 2026 by the authors. Published by AB Journals. This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0) license.

INTRODUCTION

The solution to the system of linear equations cannot be connived due to its importance in the computational system. There are several ways of computing the solution to the system of linear equations. In the first instance, in solving system of linear equations, some of the methods adopted are analytical methods and these methods become very laborious when handling very large system of equations. To circumvent this hurdle, there comes the introduction of approximate methods often referred to as Numerical methods.

Basically, some of the analytic methods used in solving systems of linear equations are reviewed here below. It is very pertinent to note that a linear system of equations is consistent if the coefficient matrix and its augmented matrix are of the same rank [1]

Cramer's Rule

This is one of the methods that can be used to solve systems of linear equations analytically. This method involves using determinant methods. For a matrix equation of the form

$$Ax_i = b \tag{1}$$

Then

$$x_i = \frac{|A_i|}{|A|} \tag{2}$$

Where x_i, $|A_i|$, *and* $|A|$ represents the unknown variables, the determinant of the matrix formed by replacing the ith *column* of the coefficient matrix A with the column vector b. [1, 2 and 10]

This process is very cumbersome when seeking solution to large system of linear equations.

Matrix Inverse or Determinant Method

This involves finding the inverse of the matrix and finding the inverse of a large matrix is often very laborious thereby making the method to be highly intractable. The process of applying this method involves writing the system of linear equation in matrix form thus

$$Ax_i = b \tag{3}$$

where A is a square matrix of n *x* n dimension, x is a column vector and b is a column matrix of n *x* 1 dimension. The solution to the system of equation (3) then becomes

$$x_i = A^{-I}b \tag{4}$$

Where A^{-I} denotes the inverse matrix of matrix A. To determine the A^{-I}, this involves finding the cofactors of matrix A, take its transpose which is referred to as adjunct of matrix A and then divide it by the determinant of matrix A, that is

$$A^{-I} = \frac{Adj\ A}{|A|} \tag{5}$$

Where $Adj\ A$ represent the Adjunct of matrix A which is the transpose of the cofactor of matrix A. This process becomes highly intractable when the matrix is of large dimension. [1, 2 and 10]

Ower and Upper Triangular Decomposition (L U) Method

This involves resolving the coefficient matrix into Upper and Lower triangular matrix. This is as expressed below

Let the system of linear equations be written in matrix form

$$Ax_i = b \tag{6}$$

Then re-express this in the form

$$L\left(Ux_i\right) = b \tag{7}$$

Where $L\ and\ U$ represents Lower and Upper triangular matrix of the main matrix A. In applying this method in solving system of linear equations, let

$$Ux_i = y_i, \tag{8}$$

then equation (7) becomes

$$Ly_i = b, \tag{9}$$

Solving equation (9) for the unknown values of y_i and then substitute back into equation (7) to determine the values of x_i. This method is too lengthy and very difficult when a very large system of equation is involved. [1, 2 and 10]

Gaussian Elimination Method

This is another method used in solving system of linear equations analytically. It is a standard elimination method for solving linear systems that proceeds systematically irrespective of features of the coefficient matrix. This method has been proved to be of great practical importance. [1]. The method involves a step – by step algorithm used in linear algebra to handle the solution of system of linear equations. This is done by transforming the coefficient matrix into an upper triangular matrix through row reduction then using backward substitution to determine the value of the unknowns. However, this method is very efficient but nonetheless it becomes intractable when handling very large system of linear equations and when the coefficient matrix has a lot of zeros, that is sparse matrix. [2 ,10]

From the foregoing, it is evident that there is a lot of work that needs to be done when solving system of linear equations analytically, then to circumvent this hurdle, then comes the introduction of approximate of numerical solution. The numerical solution is in the form of iterative process. The iterative process involves utilizing the previous value to determine the next value. This process is far better than the analytical method of solving system of linear equations.

The analytical methods in solving system of linear equations as enumerated above and with the problems encounter in applying each of the methods brought about the approximate methods. The desirability of approximate methods in solving system of linear equations cannot

be overemphasized. Linear system of large dimensions often arises in the solution of partial differential equations. The coefficient matrix for these systems is often in form of sparce matrix in which nonzero entries form a pattern, in such cases, iterative process provides an efficient method for solving such a system of linear equations [4]. The iterative method is an adaptive technique that dynamically adjusts parameters in iterative process to optimize performance across different hardware architectures and problem sets [14]. The iterative method of solving system of linear equations is beneficial not only due to its lower memory computation requirement but because of its adaptability to parallel computation architectures crucial for handling modern computational challenges [5]. In using iterative methods in solving system of linear equations, it is expedient that the coefficient matrix must be diagonally dominant. [1, 2, 4,9]

Several authors have applied different approximate techniques in solving system of linear equations and found out that the method is beneficial. In like manner, it has been shown that combination of some iterative methods can accelerate the rate of convergency of the method and can equally handle situation where the coefficient matrix is a large sparce matric [9]. Also adjusting the parameters of the iterative methods to optimized performance of the method across different hardware architecture and problem sets can equally improve the convergency of the solution of the system of linear equations [14].

In this presentation, the optimum focus is to compare the rate of convergency of three carefully selected iterative methods in solving system of linear equations. The three methods are:

i. Jacobi Iterative Method

This is one of approximate methods that can be used in solving system of linear equation. The method was developed by a German Mathematician Carl Gustav Jacob Jacobi. The iterative method is

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{i=1}^{n} a_{ij} x_j^{(k)} \right) \tag{10}$$

Where $x^{(k)}$ *and* $x^{(k+1)}$ are the values of x in the successive iterations. [7,9,13]

ii. Gauss-Seidel

This is an iterative method developed by two mathematicians, Carl Friedrich Gauss and Philip L. Seidel. This method makes use of the immediate value to generate the next iterative value. It is an improvement on Jacobi method because it takes care of inadequacies encountered in applying Jacobi method in solving system of linear equation. The iterative method is [9]

$$x_i^{(n+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)} \right) \tag{11}$$

iii. Successive Over Relaxation Method

This method is an extension of Gauss Sidel method with the introduction of relaxation factor which accelerates the rate of convergency of the method. The selection of the optimum relaxation factor depends heavily on the spectral properties of the matrix, thereby making the method more complex but potentially more efficient in terms of convergency [10]. The iterative method is

$$x_i^{(n+1)} = \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}\right) - (\omega - 1)x_i^{(n)} \quad (12)$$

Where ω represents the relaxation parameter and for convergence's sake, it often recommends that this parameter should be in the range $1 \leq \omega < 2$. It is often recommended that the value of ω be

$$\omega = \frac{2}{1+\sqrt{1-\rho}} \quad (13)$$

Where ρ is the spectral radius of the coefficient matrix. [1,9]. In other words, the optimal choice of the parameter ω is based on the study of eigenvalues of the iterative matrices for linear systems. [4]

It is of note that poor convergence or divergence of solution occurs in some cases if the coefficient matrix does not satisfy some specific criteria such as diagonally dominant or symmetric remains a critical challenge [6]. Likewise, the choice of acceleration parameterω must be carefully chosen. If the parameter is in the interval $0 < 1 \leq 1$, this is referred to as underrelaxation parameter and this is employed to make non convergency system to converge, in order words, it increases the rate of convergence by damping oscillation. However, if $1 \leq \omega \leq 2$, the parameter ω is referred to over relaxation parameter and this is used to accelerate the rate of convergency of the system. This is very important because it will save time and use less computer memory. [3, 8]

Experimental Illustration

At this juncture, some illustrations shall be presented in which the three methods under consideration, that is Jacobi Method, Gauss-Seidel Method and Successive Over -Relaxation Method (SOR) shall be applied to be able to compare their rate of convergence. Note that the initial guess for all the illustrations is zero and the acceleration parameter for SOR method $\omega = 1.05$ for all the Illustrations. The illustrations are

Illustration 1: Determine the solution to the system of linear equations below:

$$\begin{bmatrix} 20 & -3 & 1 & -5 & 2 & 4 & -1 & 2 \\ 2 & 26 & -3 & 1 & -4 & 1 & 5 & 1 \\ 5 & 3 & -32 & 2 & -1 & 4 & -7 & -2 \\ 1 & 2 & 0 & 24 & -1 & -3 & 5 & -1 \\ 0 & 1 & -4 & 2 & 25 & 1 & 0 & -5 \\ 3 & 0 & 2 & 0 & 11 & -32 & 8 & -5 \\ 4 & -1 & -3 & 0 & 7 & 2 & -26 & 5 \\ -2 & 4 & -5 & 6 & 0 & 1 & -3 & 24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 42 \\ 40 \\ 60 \\ 32 \\ 25 \\ 48 \\ 42 \\ 36 \end{bmatrix}$$

The solutions to Illustration 1 are as shown in Tables 1 to 3

Table 1: Numerical output to Illustration 1 using Jacobi Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.1000	1.5385	-1.8750	1.3333	1.0000	-1.5000	-1.6154	1.5000
2	2.7271	1.5738	-1.6268	1.9333	0.8918	1.7148	-0.6928	0.5552
3	3.0644	1.3815	-1.4531	1.2870	0.7015	-1.2995	-0.8535	0.6273
4	2.7860	1.3835	-1.4027	1.3965	0.7867	-1.3738	-0.8199	0.8481
5	2.7971	1.4075	-1.4729	1.4872	0.8331	-1.3935	-0.8089	0.8150
6	2.8300	1.4021	-1.4644	1.4681	0.8078	-1.3730	-0.7954	0.7768
7	2.8269	1.3955	-1.4544	1.4500	0.8024	-1.3688	-0.8037	0.7878
8	2.8191	1.3978	-1.4568	1.4573	0.8078	-1.3741	-0.8041	0.7941
9	2.8212	1.3987	-1.4587	1.4597	0.8082	-1.3742	-0.8029	0.7909
10	2.8224	1.3982	-1.4579	1.4579	0.8070	-1.3732	-0.8028	0.7901
11	2.8219	1.3980	-1.4576	1.4577	0.8071	-1.3733	-0.8031	0.7908
12	2.8217	1.3982	-1.4578	1.4581	0.8073	-1.3735	-0.8031	0.7909
13	2.8218	1.3982	-1.4579	1.4581	0.8073	-1.3734	-0.8030	0.7907
14	2.8218	1.3982	-1.4579	1.4581	0.8073	-1.3734	-0.8030	0.7907

Table 2: **Numerical output to Illustration 1 using Gauss-Seidel Iterative Method**

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.1000	1.3769	-1.4178	1.1311	0.6276	-1.1760	-1.1032	0.7785
2	2.6996	1.4477	-1.4519	1.5335	0.7898	-1.4636	-0.8384	0.7540
3	2.8695	1.4013	-1.4515	1.4358	0.8062	-1.3720	-0.8038	0.8009
4	2.8152	1.3998	-1.4608	1.4627	0.8083	-1.3756	-0.8017	0.7884
5	2.8240	1.3975	-1.4574	1.4565	0.8071	-1.3725	-0.8032	0.7915
6	2.8211	1.3983	-1.4580	1.4584	0.8073	-1.3736	-0.8030	0.7906
7	2.8220	1.3981	-1.4578	1.4579	0.8073	-1.3733	-0.8031	0.7908
8	2.8218	1.3982	-1.4578	1.4580	0.8073	-1.3734	-0.8031	0.7907
9	2.8218	1.3981	-1.4578	1.4580	0.8073	-1.3734	-0.8031	0.7908
10	2.8218	1.3981	-1.4578	1.4580	0.8073	-1.3734	-0.8031	0.7908

Table 3: Numerical output to Illustration 1 using Successive Over Relaxation (SOR) Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.1010	1.3775	-1.4183	1.1316	0.6278	-1.1765	-1.1035	0.7786
2	2.7002	1.4477	-1.4518	1.5338	0.7900	-1.4637	-0.8382	0.7540

3	2.8697	1.4012	-1.4515	1.4357	0.8062	-1.3719	-0.8037	0.8010
4	2.8151	1.3998	-1.4608	1.4627	0.8083	-1.3756	-0.8017	0.7882
5	2.8241	1.3975	-1.4574	1.4564	0.8071	-1.3725	-0.8032	0.7915
6	1.8211	1.3983	-1.4580	1.4584	0.8073	-1.3736	-0.8030	0.7905
7	2.8220	1.3981	-1.4578	1.4579	0.8073	-1.3733	-0.8031	0.7908
8	2.8220	1.3981	-1.4578	1.4579	0.8073	-1.3733	-0.8031	0.7908

Illustration 2: Seek the solution to the system of linear equations hereunder

$$\begin{bmatrix} 15 & 0 & 3 & -1 & 2 & 4 & -2 & 1 \\ 1 & -22 & 4 & -2 & 0 & -1 & 5 & 3 \\ 0 & 1 & -16 & 1 & -3 & 2 & -1 & 4 \\ 3 & 0 & 1 & 20 & -1 & 3 & -4 & 5 \\ -1 & 3 & 0 & 2 & -25 & 0 & -2 & 8 \\ 2 & 1 & -3 & 0 & 1 & -24 & 5 & -2 \\ 5 & -1 & 2 & -3 & 2 & 1 & -18 & 0 \\ 4 & 1 & 3 & -5 & 1 & 0 & 2 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 30 \\ 40 \\ 32 \\ 36 \\ 30 \\ 38 \\ 24 \\ 26 \end{bmatrix}$$

The output to Illustration 2 is in tables 4 to 6

Table 4: Numerical output to Illustration 2 using Jacobi Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.0000	-1.8182	-2.000	1.8000	-1.2000	-1.5833	-1.3333	1.3000
2	2.8378	-2.3803	-1.5657	1.1858	-1.1955	-1.6785	-1.4203	1.9342
3	2.6809	-2.1407	-1.4880	0.8770	-1.3133	-1.7572	-1.0105	1.5846
4	2.7594	-2.0602	-1.5931	1.0719	-1.3497	-1.6604	-1.0248	1.4800
5	2.7775	-2.1109	-1.5822	1.0724	-1.3307	-1.6332	-1.0503	1.5280
6	2.7590	-2.1074	-1.5719	1.0489	-1.3335	-1.6436	-1.0376	1.5270
7	2.7603	-2.1015	-1.5750	1.0553	-1.3352	-1.6437	-1.0388	1.5220
8	2.7617	-2.1036	-1.5751	1.0562	-1.3342	-1.6429	-1.0404	1.5238
9	2.7611	-2.1037	-1.5747	1.0552	-1.3342	-1.6433	-1.0399	1.5239
10	2.7611	-2.1034	-1.5748	1.0554	-1.3343	-1.6433	-1.0399	1.5237
11	2.7611	-2.1034	-1.5748	1.0554	-1.3343	-1.6433	-1.0399	1.5237

Table 5: Numerical output to Illustration 2 using Gauss-Siedel Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.0000	-1.7273	-2.1080	1.6054	-1.3588	-1.2818	-1.4058	1.9124
2	2.7367	-2.2817	-1.3817	0.8236	-1.3284	-1.7852	-0.9840	1.4447
3	2.7569	-2.0456	-1.6278	1.1112	-1.3303	-1.6162	-1.0575	1.5451
4	2.7640	-2.1192	-1.5632	1.0417	-1.3351	-1.6506	-1.0352	1.5183
5	2.7610	-2.0998	-1.5778	1.0590	-1.3342	-1.6413	-1.0410	1.5250
6	2.7613	-2.1044	-1.5740	1.0546	-1.3343	-1.6437	-1.0396	1.5234
7	2.7612	-2.1036	-1.5750	1.0556	-1.3343	-1.6431	-1.0400	1.5238
8	2.7612	-2.1036	-1.5748	1.0554	-1.3343	-1.6433	-1.0399	1.5237
9	2.7612	-2.1036	-1.5748	1.0554	-1.3343	-1.6433	-1.0399	1.5237

Table 6: Numerical output to Illustration 2 using Successive Over- Relaxation (SOR) Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.0010	-1.7281	-2.1091	1.6061	-1.3596	-1.2822	-1.4065	1.9137
2	2.7374	-2.2822	-1.3809	0.8227	-1.3285	-1.7858	-0.9833	1.4440
3	2.7570	-2.0452	-1.6282	1.1118	-1.3303	-1.6158	-1.0577	1.5454
4	2.7640	-2.1194	-1.5630	1.0415	-1.3351	-1.6507	-1.0351	1.5183
5	2.7610	-2.0998	-1.5779	1.0590	-1.3342	-1.6413	-1.0410	1.5250
6	2.7613	-2.1045	-1.5740	1.0545	-1.3343	-1.6437	-1.0396	1.5234
7	2.7612	-2.1033	-1.5750	1.0556	-1.3343	-1.6431	-1.0400	1.5238
8	2.7612	-2.1033	-1.5750	1.0556	-1.3343	-1.6431	-1.0400	1.5238

Illustration 3: Determine the solution to the system of linear equations as shown below

$$\begin{bmatrix} 22 & 4 & -1 & 2 & -3 & 0 & -1 & 6 \\ 1 & 18 & -1 & 3 & -1 & 2 & 5 & -2 \\ 3 & 4 & -17 & 2 & -1 & 1 & -2 & 3 \\ 2 & 0 & 3 & 16 & -4 & -1 & 0 & 1 \\ 5 & 2 & -1 & 4 & -22 & 3 & -1 & 2 \\ 4 & -3 & 7 & 0 & -1 & 24 & -2 & 1 \\ 1 & -4 & -1 & 5 & 1 & 0 & -25 & 2 \\ 4 & -5 & 1 & 1 & -2 & 1 & -3 & 24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 40 \\ 32 \\ 35 \\ 30 \\ 42 \\ 36 \\ 35 \\ 38 \end{bmatrix}$$

Tables 7 to 9 show the output to Illustration 3

Table 7: Numerical output to Illustration 3 using Jacobi Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	1.8182	1.7778	-2.0588	1.8750	-1.9091	1.5000	-1.4000	1.6923
2	0.4454	1.5541	-0.4352	1.5445	-0.4777	1.7530	-1.0953	1.1857
3	0.9371	1.6861	-0.9635	1.8169	-0.9694	1.5864	-1.2288	1.5980
4	0.6788	1.6581	-0.7060	1.6954	-0.7512	1.6262	-1.1413	1.4541
5	0.7796	1.6744	-0.8187	1.7455	-0.8579	1.6131	-1.1845	1.5384
6	0.7275	1.6711	-.07657	1.7213	-0.8115	1.6196	-1.1661	1.5041
7	0.7492	1.6740	-0.7891	1.7320	-0.8335	1.6173	-1.1755	1.5218
8	0.7384	1.6733	-0.7780	1.7269	-0.8235	1.6184	-1.1715	1.5143
9	0.7431	1.6738	-0.7829	1.7292	-0.8282	1.6180	-1.1735	1.5180
10	0.7408	1.6737	-0.7806	1.7281	-0.8261	1.6182	-1.1726	1.5164
11	0.7418	1.6738	-0.7817	1.7286	-0.8271	1.6181	-1.1730	1.5172
12	0.7413	1.6737	-0.7812	1.7284	-0.8266	1.6181	-1.1729	1.5168
13	0.7415	1.6737	-0.7814	1.7285	-0.8268	1.6181	-1.1729	1.5170
14	0.7414	1.6737	-0.7813	1.7284	-0.8267	1.6181	-1.1729	1.5169
15	0.7414	1.6737	-0.7813	1.7284	-0.8267	1.6181	-1.1729	1.5169

Table 8: Numerical output to Illustration 3 using Gauss-Seidel Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	1.8182	1.6768	-1.3434	1.8996	-0.9370	1.7594	-1.1994	1.2257
2	0.7630	1.5660	-0.8162	1.7318	0.8355	1.6208	-1.1764	1.4905
3	0.7650	1.6672	-0.7819	1.7252	-0.8244	1.6145	-1.1736	1.5097
4	0.7452	1.6740	-0.7825	1.7290	-0.8269	1.6181	-1.1732	1.5160
5	0.7415	1.6735	-0.7814	1.7285	-0.8268	1.6181	-1.1729	1.5169
6	0.7415	1.6735	-0.7814	1.7285	-0.8268	1.6181	-1.1729	1.5169

Table 9: Numerical output to Illustration 3 using Successive Over- Relaxation (SOR) Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	1.8191	1.6776	-1.3438	1.9005	-0.9370	1.7603	-1.1999	1.3057
2	0.7404	1.5759	-0.8033	1.7271	-0.8337	1.6187	-1.1739	1.4662
3	0.7712	1.6653	-0.7866	1.7272	-0.8253	1.6157	-1.1744	1.4797
4	0.7712	1.6653	-0.7866	1.7272	-0.8253	1.6157	-1.1744	1.4797

Illustration 4: Investigate the solution to the system of linear equations as expressed in the matrix form

$$\begin{bmatrix} 20 & 3 & 0 & 2 & 1 & -7 & 4 & 1 \\ 1 & -16 & 1 & -1 & 3 & 1 & -5 & 3 \\ 2 & 0 & 23 & -5 & 2 & -3 & 1 & 4 \\ 4 & -1 & 0 & 14 & -1 & 1 & -2 & 8 \\ 1 & 3 & -1 & 1 & -16 & 2 & -1 & 4 \\ 0 & -1 & 1 & 4 & -5 & 20 & -3 & 2 \\ 3 & 2 & -6 & 0 & 3 & -1 & 24 & 6 \\ 5 & 1 & 3 & -6 & -2 & 3 & 0 & 25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 46 \\ 32 \\ 46 \\ 26 \\ 30 \\ 32 \\ 36 \\ 40 \end{bmatrix}$$

The output to Illustration is 4 is contained in Tables 10 to 12

Table 10: Numerical output to Illustration 4 using Jacobi Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.3000	-2.0000	2.0000	1.8571	-1.8750	1.6000	1.5000	1.6000
2	2.6880	-23676	2.2320	0.9089	-1.6089	0.6248	1.7802	1.0837
3	2.4532	-2.4328	1.9194	0.9374	-1.9959	0.9447	1.8755	0.9037
4	2.5813	-2.5736	2.0485	0.9758	-2.0124	0.8869	1.9388	0.9283
5	2.5653	-2.5834	2.0326	0.9394	-2.0415	0.8686	1.9604	0.9076
6	2.5622	-2.5990	2.0289	0.9470	-2.0544	0.8742	1.9672	0.9043
7	2.5652	-2.6045	2.0330	0.9467	-2.0574	0.8703	1.9707	0.9061
8	2.5640	-2.6057	2.0319	0.9459	-2.0588	0.8694	1.9716	0.9054
9	2.5639	-2.6065	2.0319	0.9463	-2.0594	0.8694	1.9718	0.9056
10	2.5639	-2.6065	2.0319	0.9463	-2.0594	0.8694	1.9718	0.9056

Table 11: Numerical output to Illustration 4 using Gauss-Seidel Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.3000	-1.8562	1.8000	1.0674	-2.1251	0.6724	2.1107	1.0037
2	2.3409	-2.6985	2.0346	1.0256	-2.0947	0.8508	1.9873	0.9721
3	2.5587	-2.6161	2.0381	0.9433	-2.0489	0.8673	1.9569	0.9067
4	2.5674	-2.5993	2.0303	0.9444	-2.0570	0.8683	1.9699	0.9048
5	2.5630	-2.6059	2.0316	0.9466	-2.0596	0.8689	1.9721	0.9060
6	2.5636	-2.6069	2.0320	0.9463	-2.0596	0.8691	1.9719	0.9058
7	2.5639	-2.6068	2.0320	0.9462	02.0595	0.8691	1.9719	0.9057
8	2.5639	-2.6068	2.0320	0.9462	02.0595	0.8691	1.9719	0.9057

Table 12: Numerical output to Illustration 4 using Successive Over – Relaxation (SOR) Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	2.3011	-1.8581	1.8008	1.0676	-2.1263	0.6724	2.1122	1.0039
2	2.3408	-2.6995	2.0348	1.0256	-2.0949	0.8510	1.9873	0.9721
3	2.5590	-2.6161	2.0381	0.9432	-2.0488	0.8674	1.9569	0.9066
4	2.5674	-2.5992	2.0303	0.9444	-2.0570	0.8683	1.9699	0.9048
5	2.5630	-2.6059	2.0316	0.9466	-2.0596	0.8689	1.9721	0.9060
6	2.5636	-2.6069	2.0320	0.9463	-2.0596	0.8691	1.9719	0.9058
7	2.5636	-2.6069	2.0320	0.9463	-2.0596	0.8691	1.9719	0.9058

Illustration 5: Determine the solution to the system of linear equations as expressed in the matrix form below

$$\begin{bmatrix} 10 & -1 & -1 & 0 & 2 & 1 & -1 & 2 \\ 3 & -14 & 1 & 3 & 0 & 1 & 1 & 1 \\ 4 & 0 & 18 & 4 & -1 & 2 & 0 & 3 \\ 5 & -1 & 3 & 20 & 2 & 0 & 2 & 1 \\ 3 & 2 & 0 & 1 & -25 & 6 & -3 & 4 \\ 1 & -3 & 4 & 0 & -6 & 22 & 4 & 2 \\ 2 & 1 & -1 & 4 & -1 & 3 & -15 & 1 \\ 5 & -4 & 1 & 2 & -3 & 1 & 6 & 24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 46 \\ 24 \\ 26 \\ 32 \\ 40 \\ 36 \\ 25 \\ 38 \end{bmatrix}$$

Table 13: Numerical output to Illustration 5 using Jacobi Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.2727	-1.7143	1.4444	1.6000	-1.6000	1.6364	-1.6667	1.4167
2	0.2436	-1.1108	0.5215	1.4853	-0.8210	0.8655	-0.8858	1.0291
3	0.2477	-1.2345	0.7470	1.5246	-1.1216	1.2226	-1.0505	1.1181
4	0.2477	-1.1890	0.6661	1.5256	-1.0097	1.1044	-.09654	1.0726
5	0.2480	-1.2002	0.6928	1.5226	-1.0518	1.1445	-0.9908	1.0811
6	0.2482	-1.1972	0.6852	1.5243	-1.0388	1.1305	-0.9827	1.0777
7	0.2482	-1.1980	0.6876	1.5236	-1.0434	1.1346	-0.9854	1.0786
8	0.2482	-1.1978	0.6869	1.5239	-1.0420	1.1333	-0.9846	1.0783
9	0.2482	-1.1979	0.6871	1.5238	-1.0424	1.1333	-0.9849	1.0784
10	0.2482	-1.1979	0.6871	1.5238	-1.0424	1.1333	-0.9849	1.0784

Table 14: Numerical output to Illustration 5 using Gauss-Seidel Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.2727	-1.6558	1.3838	1.2415	-1.6501	0.6965	-1.2526	1.0006
2	0.2643	-1.2610	0.7744	1.5950	-1.1278	1.1409	-0.9717	1.0407
3	0.2507	-1.1771	0.6714	1.5357	-1.0457	1.1392	-0.9800	1.0791
4	0.2483	-1.1956	0.6835	1.5243	-1.0412	1.1339	-0.9842	1.0788
5	0.2481	-1.1979	0.6869	1.5236	-1.0423	1.1335	-0.9848	1.0784
6	0.2482	-1.1979	0.6871	1.5238	-1.0423	1.1335	-0.9848	1.0784
7	0.2482	-1.1978	0.6871	1.5238	-1.0423	1.1336	-0.9848	1.0784
8	0.2482	-1.1978	0.6871	1.5238	-1.0423	1.1336	-0.9848	1.0784

Table 15: Numerical output to Illustration 5 using Successive Over – Relaxation (SOR) Iterative Method

Number of Iterations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.2741	-1.6638	1.3905	1.2459	-1.6586	0.6953	-1.2581	1.0037
2	0.2643	-1.2578	0.7690	1.5989	-1.1238	1.1462	-0.9676	1.0405
3	0,2505	-1.1756	0.6698	1.5349	-1.0445	1.1393	-0.9802	1.0799
4	0.2482	-1.1959	0.6836	1.5240	-1.0411	1.1338	-0.9843	1.0789
5	0.2481	-1.1980	0.6870	1.5236	-1.0423	1.1334	-0.9849	1.0784
6	0.2481	-1.1980	0.6870	1.5236	-1.0423	1.1334	-0.9849	1.0784

DISCUSSION OF RESULTS

At this juncture, the discussion of the results shall be based on the result generated for each illustration as presented Tables 1 to 15. In order of discussion, five illustrations were presented in which the three methods under consideration that is Jacobi Iterative method, the Gauss–Seidel Iterative method and the Successive Over – Relation Iterative method (SOR) were used for each illustration to see the rate of convergency of each method.

From Table 1, which shows the output generated for illustration one using Jacobi Iterative method. It was discovered that the 13th and 14th iteration yield the same results. This shows that the method converges at the 14th iteration. On the other hand, when Gauss–Seidel iterative method was used on the same illustration 1and the results is contained in Table 2. It was discovered that convergence starts at the 9th iteration and the iterative process stop at the 10th iteration. This shows that the solution to the system of linear equation under consideration is achieved at the 9th iteration. Also, the Successive Over – Relaxation (SOR) iterative method was equally applied to illustration 1and the output was as shown in Table 3. It was found out that convergence starts at the 7th iteration and stops at the 8th iteration. This shows that the solution to the system of equations as expressed in illustration 1 was achieved at the 7th

iteration. From this illustration, it shows that Successive Over – Relaxation iterative method converges faster that other two methods under review.

To be more assertive about this claim, another illustration was considered. This leads to considering Illustration 2. The output to Illustration 2 using Jacobi Iterative method was presented in Table 4. The result on this table shows that convergence starts at the 10th iteration and stops at the 11th iteration. This shows that the solution to the system of linear equations as contained in Illustration 2 was achieved at the 10th iteration. Table 5 shows the output to Illustration 2 using Gauss-Seidel Iterative method. It shows that convergence starts at the 9th iteration and stops at the 10th iteration. This shows that the solution to the system of linear equations as contained in Illustration 2 was achieved at the 9th. Table 6 contains the results generated by Illustration 2 using SOR iterative method. In this case, convergence starts 7th iteration and stops at the 8th. This shows that solution is achieved at the 7th iteration when SOR iterative method was used. This further confirms that SOR converges faster that both Jacobi and Gauss-Seidel Iterative methods.

It will be inadequate to draw a conclusion based on results from only two illustrations; therefore, another Illustration is considered. Considering Illustration 3 in which the results are presented in Tables 7 to 9.

Table 7 shows the output to Illustration 3 using Jacobi Iterative method and this shows that convergency starts at the 14th iteration and ends at the 15th iterations. This shows that the solution to the system of linear equations as illustrated in illustration 3 was achieved at the 15th iteration. On the other hand, when Gauss–Seidel Iterative method was used to determine the solution to the system of equations under consideration, the output as contained in Table 8 shows that convergency starts at the 5th iteration and ends at the 6th iteration which means that the solution was achieved at the 6th iteration and this shows a highly significant in the rate of convergency as compared to Jacobi Iterative method. Also, SOR Iterative method was applied to the same Illustration 3, and the results are as presented in Table 9. This shows a significant improvement on the rate of convergency because the solution to the system of linear equations considered here was achieved at the 4th.

For further confirmation of the desirability of SOR iterative method in solving system of linear equations, another illustration was considered. This leads to Illustration 4. The three iterative methods under consideration were applied to Illustration and the results are as presented in Tables 10 to 12.

Table 10 shows the output to Illustration 4 using Jacobi Iterative method and this shows that the solution was achieved at the 10th iteration whereas when Gauss–Seidel Iterative method was used, the solution was achieved at the 8th iteration but when SOR Iterative method was applied to the same Illustration 4, convergency to the solution was achieved at the 7th iteration which shows a significant improvement on the number of iterations taken to achieve the solution to the system of linear equations under consideration.

To be able to draw a convincing conclusion on which iterative method under consideration is most desired, another illustration was considered. This leads to considering Illustration 5. The three iterative methods were applied to Illustration 5, and the output was presented in Tables 13 to 15.

Table 13 contains the output to Illustration 5 when Jacobi Iterative method was used. It shows that convergence to the solution was achieved at the 10th iteration whereas when Gauss–Seidel Iterative method was applied to the same illustration, convergence to the solution was achieved at the 8th iteration as shown in Table 14. SOR iterative method was equally applied to determine the solution to system of linear equations as shown in Illustration 5. The result is as presented in Table 15, and this shows that convergence was achieved at the 6th iteration.

CONCLUSION

From the analysis of results as presented to all the Illustrations and results as presented in Tables 1 to 15, it shows very clearly that in all the Illustrations considered, Successive Over Relaxation (SOR) iterative method gives the least number of iterations and more computationally efficient results. One important thing to note when using iterative method in solving any mathematical problem is the number of iterations and how accurate the method is when compared with the analytical solution. Also, it is very paramount to equally consider the space in the memory of the computer and the time taken. The greater the number of iterations the more the space will be occupied in the computer memory and the more time taken before convergency is achieved. From the going, it can be concluded unequivocally that Successive Over Relaxation (SOR) Iterative method is more desired in solving system of linear equations.

Also, it is also expedient that whenever an iterative method is to be used in determining the solution to system of linear equations such an equation must the diagonally dominant. It is also advisable to choose the initial guess very near zero to minimize the number of iterations.

REFERENCES

Anzt, H. et al. (2020). Preparing Sparse Solvers for Exascale Computing. *A Philosophical Transaction of The Royal Society A: Mathematical, Physical and Engineering Sciences, 378(2166).* *https://royalsocietypublishing.org/rsta/article/378/2166/20190053/111581/preparing-sparse-solvers-for-exascale*

Anzt, H., Dongarra, J., Flegar,G., and Quintana-Orti, E.S. (2017). Adaptive Precision In Block-Jacobi Preconditioning for Iterative Sparse Linear System Solvers. Concurrency and Computation: Practice and Experience, 29(9). https://netlib.org/utk/people/jackDongarra/PAPERS/Anzt-et-al-2018-concurrencypdf

Barrett, R. et al. (1994). Templates for The Solution of Linear Systems. (2nd Edition) Society for Industrial and Applied Mathematics. https://netlib.org/templates/templates.pdf

Enyew, T.K. et al. (2019). Second Refinement of Jacobi Iterative Method for Solving Linear System of Equations. International Journal of Computing Science and Applied Mathematics, 5(2), [41-51]. iptek.its.ac.id

Grewal, B. S. (2007). *Numerical Methods in Engineering and Science with Programs in FORTRAN 77, C and C++*. Khanna Publishers dn76000/eu.archive.org/0/items/numerical-methods-in-engineering-b-s-grewalpdf/ numerical-methods-in-engineering-b-s-grewalpdf.pdf

Jain, M. K., Iyengar, S. R. K., & Jain, R. K. (2022). *Numerical Methods for Scientific and Engineering Computation*. New Age International Publisher. Dokumen.pub/numerical-methods-scientific-and-engineering-computation.html

Kreyszig, E. (2006). *Advanced Engineering Mathematics*. John Wiley & Sons. http://www.wiley.com/college/kreyszig/

Mathews, J. H. (1992). *Numerical Methods for Mathematics, Science a Engineering.* Prentice-Hall .pdfbeast.live/download/4868367-numerical-methods-for-mathematics-science-and-engineering-john-h-mathews-pdf

Patil, P. B., & Veima, U. P. (2009). *Numerical computational methods*. Narosa Publishing Home. narosa.com/print.asp?catgcode=978-81-7319-951-6

Saad, Y. (2003). Iterative methods for Sparse Linear Systems. https://www.lmn.pub.ro/~daniel/Electromagnetic ModellingDoctoral/Books/Numerical%20methods/Saaditerativemethods.pdf

Smith, G. D. (1978). *Numerical Solution of Partial Differential Equations*. Oxford University Press. wp.kntu.ac.ir/ghoreishif/smith.pdf

Stroud, K. A. (1996). Further Engineering Mathematics Programmes and Problems. Palgrave Publishers. archive.org/details/furtherengineeri000stro-y5m43ed

Yang, X. & Mittal, R. (2014). Acceleration of the Jacobi Iterative Method by Factors Exceeding 100 using Scheduled Relaxation. Journal of Computational Physics, 274, [695 708]. https://www.sciencedirect.com/science/article/abs/pii/S0021999114004173

Young, D.M. (1971). Iterative Solution of Large Linear Systems. Academic Press. https://archive.org/details/iterativesolutio00youn/page/n5/mode/2up