American Journal of Applied Mathematics
2024; 12(6): 214-235
http://www.sciencepublishinggroup.com/j/ajam
doi: 10.11648/j.ajam.20241206.12
ISSN: 2330-0043 (Print); ISSN: 2330-006X (Online)

Exploring Solvents and Sensitivity of Systems of Linear Equations Arising from Real World Phenomena Via Optimal Successive Over-relaxation Method

Odeli John Kigodi*, Chacha Stephen Chacha, Adeline Peter Mtunya

Department of Mathematics, Physics, and Informatics, Mkwawa University College of Education, University of Dar es Salaam, Iringa, Tanzania

Email address:

kigodimadeullah@gmail.com (Odeli John Kigodi), chacha.chacha@muce.ac.tz (Chacha Stephen Chacha),
adeline.mtunya@udsm.ac.tz (Adeline Peter Mtunya)
*Corresponding author

To cite this article:

Odeli John Kigodi, Chacha Stephen Chacha, Adeline Peter Mtunya (2024). Exploring Solvents and Sensitivity of Systems of Linear Equations Arising from Real World Phenomena Via Optimal Successive Over-relaxation Method. *American Journal of Applied Mathematics, 12*(6), 214-235. https://doi.org/10.11648/j.ajam.20241206.12

Received: 6 September 2024; **Accepted:** 4 October 2024; **Published:** 18 November 2024

Abstract: This study places a significant emphasis on assessing the efficiency of numerical methods, specifically in the context of solving linear equations of the form $Ax = b$, where A is a square matrix, x is a solvent vector, and b is a column vector representing real-world phenomena. The investigation compares the effectiveness of the Refined Successive Over Relaxation (RSOR) method to the standard Successive Over-relaxation (SOR) method. The core evaluation criteria encompass computational time (in seconds), convergence behavior, and the number of iterations necessary to approximate the solvents of five distinct real-world phenomena: Model Problem 1 (MP1) involving an Electrical Circuit, Model Problem 2 (MP2) focusing on Beam Deflection, Model Problem 3 (MP3) addressing Damped Vibrations of a Stretching Spring, Model Problem 4 (MP4) dealing with Linear Springs and Masses, and Model Problem 5 (MP5) focusing on Temperature Distribution on Heated Plate. The RSOR method generally outperforms the SOR method, particularly with a constant relaxation parameter (ω) in the range $1.0 < \omega < 1.2$. The RSOR method is favored for its robustness and efficiency with less need for fine-tuning ω, whereas the SOR method can achieve superior performance if the optimal ω is found, although this often requires time-consuming trial and error. Despite the potential for better performance with an optimal ω, the RSOR method's consistent results make it the more practical choice in many cases. The study also explores the stability of the systems of linear equations arising from these phenomena by calculating their condition numbers $(K(A))$. More interestingly, the results reveal that all systems MP1 to MP4 exhibit instability when subjected to even modest perturbations, shedding light on potential challenges in their solvents. This research not only underscores the advantages of the RSOR method but also emphasizes the importance of understanding the stability of numerical solvents in the context of real-world problems. Additionally, the results for MP5 demonstrates that tiny changes to the original matrix's coefficients have no effect on the desired solvent because the perturbed matrix's condition number is the same as the original matrix's, making the problem well-structured. The problem becomes ill-conditioned if there is an increase or decrement to the matrix's coefficients that is bigger than 10^{-5}. In summary, Sparse systems are sensitive to perturbations, resulting in instability. If the tolerance $|k(A_i^0) - k(A_i)| > 10^{-5}$ for all positive integers i, then the problem becomes poorly structured.

Keywords: Systems of Linear Equations, Solvents, SOR, RSOR, Real World Phenomena, Sensitivity Analysis

1. Introduction

According to Young, the analysis reveals that the SOR approach is more efficient than the Gauss-Seidel method in terms of number of iterations, computation time, and relaxation parameter when solving linear systems of equations resulting from partial differential equations (PDEs) of the elliptic type [33].

Evaluating the accuracy of the method at various stages poses a challenge unless the optimal accelerating factor is determined. The introduction of a new algorithm sheds light on the significance of identifying optimal relaxation factor that consistently produces satisfactory error bounds at any stage of computation. An interesting observation arises: the refined SOR method exhibits a remarkable efficiency, requiring approximately one-third of the iterations when compared to the existing Successive Over-relaxation (SOR) method proposed by Carre [6]. This reduction in the number of iterations not only signifies a potential acceleration in convergence but also emphasizes the improved computational efficiency offered by the refined SOR technique.

With reference from the study by Kulsrud, when solving the discretized Laplace's equation in rectangular coordinates via SOR method, the investigations indicate that an optimal relaxation factor is attained within the range $1 \leq \omega \leq 2$ where at desirable solvent, the quantity of iterations is relatively small [18].

In order to ensure the efficiency and effectiveness of SOR method, restricting the parameters is vital so that the spectral radius and the norms of SOR coincide with that of the Modified Successive Over-relaxation (MSOR). Optimal parameters which speed up the rate of convergence for MSOR method were obtained and revealed to be the best when solving the system of linear equations of the form $Ax = b$, [16].

The choice of relaxation factor for the SOR has been analyzed by Young, that the optimal ω is a real positive number obtained in the interval $0 < \omega < 2$. The study revealed that one iteration for the Refined Successive Over-relaxation (RSOR) is equal to twice iterations for the SOR, which means that the existing SOR converges faster than the old one [35].

The research by Hadjidimos shows that the sufficient condition of the convergence for the Syatematic Successive Over-relaxation (SSOR) is $|\omega - 1| < 1$ for all $\omega \in \Re$, which results to $\omega \in (0, 2)$. The convergence of the refined SOR is observed by varying different values of ω within the range of 0 and 2. Each iteration step of the SSOR is faster in terms of convergence because the error obtained in the Successive Over-relaxation (SOR) is bigger than that of SSOR [15].

Bai & Chi introduced an innovative adaptation of the Successive Over-relaxation (SOR) method known as the Asymptotically Successive Over-relaxation (ASOR) method for solving systems of linear equations. Their approach takes into account various factors, including the number of iterations, mesh spacing, and residual errors. Through numerical computations, it is evident that the ASOR method outperforms the traditional SOR in terms of both the number of iterations and residual errors. Furthermore, the results indicate that the ASOR method achieves faster convergence compared to both the classical Successive Over-relaxation method and the Gauss-Seidel method [5].

More interestingly, in order to minimize difficulties in solving large fuzzy system of linear equations, Allahviranloo made a numerical investigations comparing the SOR iterative method and Gauss-Seidel method, the result shows that the SOR converges faster than the Gauss-Seidel method. Additionally, the Successive Over-relaxation (SOR) method achieves improved convergence with a carefully chosen relaxation parameter ω, especially effective for systems of linear equation already convergent with the Gauss-Seidel method. The developed algorithm for fuzzy system of linear equations (FSLE) was recommended to be more effective [3].

Bai *et al.* developed the Generalized Successive Over-relaxation (GSOR) method for linear augmented systems where its convergence was proved in terms of quantity of iterations and relaxation parameter at the desirable solvent. The numerical result proves that the generalized successive over relaxation method is more efficient than the existing SOR method. Also the GSOR proved to be distinguished iterative technique in terms of quantity of iterations, computational time and residual error particularly when solving the systems of linear equations by using vector extrapolation and matrix relaxation techniques [4].

Dehghan & Hajarian introduced two novel preconditioner approaches aimed at addressing linear systems involving L-matrices and accurately assessing their convergence. By making reasonable assumptions about matrix A of $Ax = b$, the findings indicate that the recommended preconditioning techniques can accelerate the convergence of the Successive Over-relaxation (SOR) iterative method, as confirmed by comparison theorems. The preconditioned iteration is established to converge under specific conditions imposed on the coefficient matrix. Numerical experiments validate the proposed theorems, demonstrating the effectiveness of the method in solving linear systems of equations by achieving faster convergence compared to established iterative methods [9].

Vatti & Gonfa conducted research on how to improve the Generalized Jacobi (RGJ) approach for solving linear equation systems. By using MATLAB software to conduct experiments, the effectiveness of the RGJ approach is confirmed. The results established that the generalized Jacobi (GJ) approach does not converge as quickly as the RGJ method. The research also reveals that the spectral radius of the RGJ technique is smaller than that of the GJ method, indicating that the RGJ approach works better than the GJ method [31].

To speed up the rate of convergence for the Successive Over-relaxation (SOR) iterative method for solving linear equation systems, Ndanusa and Adeboye, introduced a preconditioned SOR. Two variations of the preconditioned SOR iteration are formulated and applied, with specific assumptions imposed on the coefficient matrix of the original linear system of equations. Convergence properties are derived and scrutinized following established protocols. The results illustrate that the rates of convergence for the preconditioned iterations surpass those of the conventional SOR technique. The theoretical findings firmly establish the superiority of the new algorithm over the existing one, a conclusion further substantiated through numerical experiments [28].

The study done by Mai & Wu, revealed that the rate of convergence in Multi-layer scheme depends on the size of spectral radius, if an iterative matrix has small spectral radius then the fast the SOR method will converge. When solving multi-layer scheme, an optimal relaxation parameter ω reduces the number of iterations in comparison to a relaxation parameter ω obtained by linear search method [23].

Additional research conducted by Youssef & Taha suggests that the Modified Successive Over-relaxation (MSOR) technique demonstrates quicker convergence in comparison to the conventional Successive Over-relaxation (SOR) method. Furthermore, the study observes that the optimal relaxation parameter falls within the range of $\omega \in (0, 2)$, where the method concurrently updates the residue along with the solvent, incorporating the most recently calculated solvent. The recommendation is made that a combination of the modified SOR and the conventional SOR could potentially enhance efficiency and effectiveness in accelerating convergence, especially with the integration of updated residue [37].

According to Kyurkchiev & Iliev, when solving the large sparse linear systems, the results indicate that the refined successive over relaxation method yields more considerable improvement in the rate of convergence compared to the existing SOR method. Also, the result indicates that the spectral radius via RSOR is finer than that of SOR [18].

The study by Emmanuel, on iterative methods for solving systems of linear equations presents the Jacobi approach and the modified Jacobi method as the two basic iterative techniques for solving a system of linear equations using real-world linear examples. The examination of the data indicates that the modified Jacobi technique is more convergent than the Jacobi method, when the number of repetitions is taken into account [11]. This demonstrates that compared to the Jacobi approach, the modified Jacobi method uses less computer storage. As a result, the modified Jacobi approach is more accurate than the Jacobi method; its rate of convergence is faster and its number of iterations is minimum.

Further study by Mayooran & Light, the results reveal that the minimum number of iterations for convergence is attained when $\omega = 1.2$. It was also recommended that relaxation factor for a minimum number of iterations lies between 1 and 2 . Finally, it was observed that the computational time (in seconds) of displaying the results for the Successive Over-relaxation (SOR) is 0.084 seconds and it is optimal in terms of memory storage [25].

More interestingly, Mayooran & Light came up with the formula for finding optimal relaxation factor. If A is a symmetric matrix with all positive eigenvalues and the relaxation parameter (ω) falls within the range $0 < \omega < 2$ subsequently, with any starting estimation $X_{(0)}$ then the SOR method will converge [25]. If in addition A is tridiagonal and $\rho(T_g)$ is the spectral radius of Iterative matrix where

$T_g = D^{-1}(L + U) < 1$ and ρ is a spectral radius, then the desirable relaxation parameter is given by

$$\omega = \frac{2}{1 + \sqrt{1 - |\rho(T_g)|^2}}. \tag{1}$$

According to Gonfa, the analysis of results showed that RGGS (refined generalized Gauss-Seidel) when solving 6×6 linear system of equations and 4×4 linear system of equations the approach is effective in terms of time, it uses 0.003789 seconds and 0.002961 seconds, respectively. The refined method converges to the exact solvent in approximately 6 iterations, outperforming other methods such as SOR (14 iterations for the first linear system, 9 iterations for the second linear system).The Refined Gauss Seidel (RGS) achieves convergence in 10 and 11 iterations for the respective linear systems, both with a specified tolerance error. Investigations prove that refined generalized Gauss Seidel uses minimum memory space in storing data because it performs few iterations towards the desired solvent in comparison to other methods; SOR and RGS respectively. This shows that refined generalized Gauss-Seidel is effective and optimal in comparison to other methods [14].

According to Muleta & Gofe, the square of the generalized accelerated successive over relaxation iterative matrix represents the novel method's iterative matrix. The findings demonstrate that compared to the current SOR method, the Refined Successive Over-relaxation (RSOR) method converges more quickly. After examining comparative examples, the findings show that the suggested method outperforms the current SOR method. The RSOR has less quantity of iterations and minimum computational time [27].

The recently introduced version of the Successive Over-relaxation method [8] exhibits clear improvements in terms of both the quantity of iterations and computational time when compared to the conventional SOR. Specifically, the refined SOR demonstrates a reduced number of iterations compared to the existing SOR. Additionally, the outcomes indicate that the effectiveness of the modified SOR, as measured by the spectral radius of the matrix, surpasses that of the existing SOR.

In order to speed up the convergence of traditional iterative algorithms like Jacobi, Gauss-Seidel, and SOR for the solution of the system of linear equations, Faruk & Ndanusa devised many types of preconditioners [13]. The study suggests adding a preconditioner to the SOR iterative matrix, and the outcomes show that the preconditioned iterative techniques outperformed the traditional SOR technique in terms of performance. Furthermore, the examination shows that the preconditioned SOR approach converges more quickly than the traditional SOR approach. According to the study, preconditioned iterative methods work better than current iterative methods.

Lotfy *et al.*, came up with the investigations on solving fuzzy-linear systems using iterative methods and recommended that the Gauss-Seidel technique is the most appropriate approach for tackling fuzzy-linear systems, the Gauss-Seidel method is effective in terms of storage, convergence, running time and accordingly computational work [22].

Furthermore, the research conducted by Mayaki & Ndanusa reveals that $\rho(G_1)$ is less than $\rho(G_{SOR})$ for relaxation factors ω ranging from 0.1 to 0.9. This suggests the effectiveness of the preconditioned SOR scheme, characterized by the matrix G_1 [24]. Additionally, the findings indicate that the preconditioned SOR exhibits faster convergence compared to the traditional SOR method, establishing its superiority over the classical SOR.

The Jacobi and Gauss-Seidel methods, which are widely recognized as numerical techniques for approximating solutions to linear systems of equations, are thoroughly examined for their accuracy and efficacy. Particular attention is paid to assessing their convergence properties, including the critical metric of the maximum number of iterations necessary for convergence. Based on the research conducted from [10, 26] provides important clarifications regarding the relative effectiveness of these techniques. The findings demonstrate the superiority of the Gauss-Seidel approach in terms of efficacy and accuracy, attributed to its ability to achieve convergence within a maximum number of iterations.

Abdullahi & Ndanusa, developed a new preconditioner designed to accelerate convergence of the Accelerated Over-relaxation (AOR) method .By advancing some convergence theorems using well-known approaches, theoretical investigation of the new preconditioned AOR approach is carried out. Numerical research using example issues is conducted to verify the theoretical convergence analysis conclusions. The comparison of the suggested preconditioner with a few preconditioners that are already accessible in the literature is shown numerically. Results indicate that the suggested preconditioned AOR method converges more quickly than both the unpreconditioned AOR and the currently used preconditioned methods [1].

According to Abdullahi & Muhammad, it also revealed that optimum convergence for the refined SOR is attained at $\omega = 0.9$ where ω is a parameter for relaxation. Compared to the Successive Over-relaxation (SOR) approach, the preconditioned successive over relaxation (RPSOR) refinement sought to accelerate the convergence criteria by minimizing the spectral radius of the iterative matrix. More interestingly, the outcomes demonstrate that preconditioned successive over relaxation refinement converges more quickly than the Successive Over-relaxation (SOR) approach [2].

The study by Lisanu Assefa & Woldeselassie Teklehaymanot on Second Refinement of Accelerated Over-relaxation (SRAOR) method indicates that the decrease of spectral radius of an iterative matrix may speed up the convergence rate of the method. Numerically, it has been revealed that the second refinement over relaxation method is more efficient in comparison to the refinement over relaxation method [21].

Over time, numerous preconditioners have been developed to improve the convergence of iterative methods for solving linear systems of equations. Most of these preconditioners exhibit a limited impact on specific entries of the coefficient matrix. To address this limitation, Ndanusa and Al-Mustapha introduced a novel preconditioner that affects every element in the coefficient matrix. Specifically, this new approach eliminates the last entry in the leftmost column and reduces the size of all other entries. The resulting preconditioned iteration technique has introduced and refined convergence and comparison theorems. Through numerical examples featuring simulated solutions, it is demonstrated that the proposed method achieves faster convergence compared to the Successive Over-relaxation (SOR) method [29].

Linear systems of equations are commonly applicable in addressing real-world phenomena that necessitate the utilization of complex numerical techniques for iterative solvents. The study conducted by Xu *et al.*, the discretization of the first-order linear Fredholm integro-differential equation results to a system of linear equations. Three distinct approaches, namely the traditional Gauss-Seidel (GS) method, the Successive Over-relaxation (SOR) technique, and the refined Successive Over-relaxation (RSOR) method, are employed to solve the resulted generalized system of linear equations. The analysis indicates that the RSOR method surpasses the current GS and SOR methods in tackling large, dense linear systems, as evidenced by three criteria: the number of iterations, the execution time, and the highest absolute error [32].

The study by Yahaya *et al.* on Refined Extended Accelerated Over-relaxation (REAOR) indicates that improvements to the iteration matrix reduce the spectral radius. This reduction enhances the convergence rate of RSOR when solving systems of linear equations in the form of $Ax = b$. Also the result indicates superiority of the refined SOR over some existing methods like SOR, Gauss-Seidel and Gauss-Jacobi. When it comes to computation time and quantity of iterations, the REAOR is superior [36].

The problem's condition number expresses how unpredictable the solvent is in response to minute changes in the data input. The type of problem and the input data may affect the condition number. If the condition number of an issue is relatively minimal, it indicates that the approximate solvent agrees with the exact solvent of the provided problem, independent of perturbations. This indicates that the problem is well-structured. The problem is not well-structured if the condition number is high. The condition number during the perturbation of matrix entries of the linear systems of equations must be one or less; i.e. $\delta \leq 1$ [30].

Several strategies were devised by Epelman & Freund to calculate the condition number of a conic linear system. The problem is well-conditioned only if the condition number is small, according to the numerical results. Furthermore, the outcome demonstrates that arbitrary data perturbations can result in ill-posed problems that are both doable and infeasible. More interestingly, the study shows that, in contrast to ill-conditioned problems, well-conditioned problems have a dependable solvent [12].

According to Kiusalaas, a linear equation can be classified as well-conditioned if the condition number is closer to one; if not, the equation is ill-conditioned [17]. An ill-conditioned problem is one that is not well conditioned; it arises from minor perturbations in the matrix entries that cause departures from the original solvent.

The literature highlights the use of the Optimal SOR approach as a powerful numerical methodology for solving complex systems, emphasizing its advantages over conventional methods with regard to computing economy and convergence speed. Numerous studies and approaches are examined, demonstrating the flexibility and resilience of the Optimal SOR method in a range of real-world scenarios. Despite the advances in methodologies, current studies have not thoroughly examined the RSOR method. Key aspects such as the number of iterations, residuals, computational time, and the sensitivity of the solvent remain underexplored. This study aims to fill this gap by employing the RSOR method to evaluate these critical factors, providing new insights that existing research has yet to address.

2. Materials and Methods

The Successive Over-relaxation (SOR) technique is an iterative numerical technique used for solving systems of linear equations. Its historical development is rooted in the broader context of numerical analysis and iterative techniques. The SOR method evolved as an improvement over the Gauss-Seidel method, which is another iterative approach for solving linear systems. The Gauss-Seidel approach employs the most current values of the variables as soon as they are updated, processing each equation in turn. The Gauss-Seidel method can converge slowly for some types of matrices, despite its effectiveness. The concept of relaxation, a key element in SOR, was introduced to expedite convergence. The SOR method was independently developed and popularized by several researchers during the mid-20th century.Young is often credited with the introduction and promotion of SOR [34].

Over the years, researchers have contributed to refining and extending the SOR method. Modifications and adaptations have been proposed to enhance its applicability to various types of linear systems. Several disciplines, including computer science, physics, and engineering, use the SOR approach. Its efficiency in solving large linear systems made it a valuable tool for numerical simulations and scientific computations.The SOR method remains a classic and widely used method, especially when dealing with large, sparse linear systems.

2.1. SOR and RSOR

In this subsection, we initiate the derivation of the Successive Over-relaxation (SOR) technique, drawing inspiration from the Gauss-Seidel method. Following this, we present the Refined Successive Over-relaxation (RSOR) method, which advances the SOR method. We proceed to delineate the component-wise formulation of both the SOR and RSOR approaches. Finally, we define the convergence criteria for both the SOR and RSOR approaches.

According to Young, the Successive over-relaxation (SOR) is derived as follows; [35] From Gauss Seidel method

$$DX^{(k)} = b - LX^{(k)} - UX^{(k-1)} \qquad k = 1, 2, \cdots \tag{2}$$

We subtract the term $DX^{(k-1)}$ from both sides of equation 2 and get

$$\begin{aligned} D[X^{(k)} - X^{(k-1)}] &= b - LX^{(k)} - UX^{(k-1)} - DX^{(k-1)} \\ &= b - (L + D + U)X^{(k-1)} - LX^{(k)} + LX^{(k-1)} \\ &= b - AX^{(k-1)} - L[X^{(k)} - X^{(k-1)}] \end{aligned}$$

$$\text{Thus,} \qquad D[X^{(k)} - X^{(k-1)}] = b - AX^{(k-1)} - L[X^{(k)} - X^{(k-1)}] \tag{3}$$

We introduce a parameter ω called the relaxation factor on the RHS of equation 3 and get

$$D(X_{(k+1)} - X_{(k)}) = \omega\left[(b - AX_{(k)}) - L(X_{(k+1)} - X_{(k)})\right] \tag{4}$$

where; $k = 0, 1, 2, 3, \cdots$, D is called a diagonal matrix, L is called strictly lower triangular component, U is called a strictly upper triangular component, ω is called relaxation factor, b is a column matrix with $b = [b_1, b_2, b_3, \cdots, b_n]^T$.

The component-wise formulation of the Successive Over-relaxation (SOR) method involves updating each solution component iteratively using the following formula:

$$X_i^{(k+1)} = (1 - \omega)X_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}X_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}X_j^{(k)}\right), \tag{5}$$

where:

1. $X_i^{(k+1)}$ is the updated value of the i-th component in the $(k+1)$-th iteration,
2. $X_i^{(k)}$ is the current value of the i-th component in the k-th iteration,
3. a_{ij} represents the elements of the coefficient matrix A,
4. b_i is the i-th element of the column vector b,
5. ω is the relaxation factor.

The Refined Successive Over-relaxation (RSOR) method extends the SOR method by applying the relaxation technique recursively to further enhance convergence [25]. It is defined by:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - |\rho(T_g)|^2}} \tag{6}$$

where ω_{opt} is the optimal relaxation parameter and $\rho(T_g)$ is the spectral radius of Iterative matrix

2.1.1. Convergence Criteria for SOR

The study done by Young indicates that, if $0 < \omega < 2$ then the Successive Over-relaxation (SOR) technique converges for any initial vector if matrix A is positive definite and symmetric. It is noticed that, if $\omega > 2$ then the Successive Over-relaxation (SOR) technique diverges. The convergence rate for the Successive Over-relaxation (SOR) method is slower in the range of $0 < \omega < 1$ [34].

2.1.2. Convergence Criteria for RSOR

According to Abdullahi & Muhammad, the improved SOR converges more quickly than SOR and its optimal convergence occurs when $\rho(G_\omega) < 1$, where $\rho(G_\omega)$ is the spectral radius for an iteration matrix [2]. The convergence criteria for RSOR is given by

$$||\overline{\overline{X}}^{(k+1)} - \overline{\overline{X}}||_\infty \le ||G_\omega||^2_\infty ||X^{(k)} - \hat{X}||_\infty, \qquad (7)$$

where G_ω is an iteration matrix for the refined SOR, $\overline{\overline{X}}$ is a solvent of the refined SOR and $\hat{X}$ is the original solvent of the system $Ax = b$.

2.2. Numerical Investigations

In this subsection, we undertake various investigations using the SOR and RSOR method on the following five model problems arising from real world phenomena.

2.2.1. Model Problem 1 (MP1)

The problem of calculating current flowing through each resistor in a circuit shown in Figure 1 containing resistors R_1, R_2, R_3, R_4, R_5, R_6, R_7 and R_8 measured in Ohms (Ω), the potential differences V_1, V_2 and V_3 measured in volts (V) and currents i_1, i_2, i_3 and i_4 measured in ampere (A) is considered.

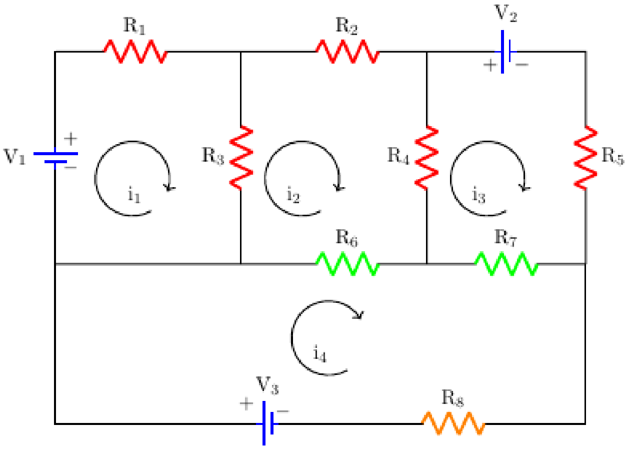

Figure 1. *Electric Circuit.*

Using the Kirchhoff's law, the following equations can be extracted from the above circuit

$$V_1 - R_1 i_1 - R_3(i_1 - i_2) = 0 \qquad (8)$$

$$-R_2 i_2 - R_4(i_2 - i_3) - R_6(i_2 - i_4) - R_3(i_2 - i_1) = 0 \qquad (9)$$

$$V_2 - R_5 i_3 - R_7(i_3 - i_4) - R_4(i_3 - i_2) = 0 \qquad (10)$$

$$V_3 - R_6(i_4 - i_2) - R_7(i_4 - i_3) - R_8 i_4 = 0 \qquad (11)$$

where

$$\begin{array}{llll} V_1 = 28V & V_2 = 36V & V_3 = 42V & \\ R_1 = 16\Omega & R_2 = 10\Omega & R_3 = 6\Omega & R_4 = 12\Omega \\ R_5 = 8\Omega & R_6 = 14\Omega & R_7 = 4\Omega & \end{array}$$

Plugging the given parameter values to the above equations, the equations may be simplified to the following equation;

$$\left\{ \begin{array}{rcl} -22i_1 + 6i_2 & = & -28 \\ 6i_1 - 42i_2 + 12i_3 + 14i_4 & = & 0 \\ 12i_2 - 24i_3 + 4i_4 & = & 36 \\ 14i_2 + 4i_3 - 23i_4 & = & -42 \end{array} \right. \qquad (12)$$

The system (12) can be rearranged as $A_1 x = b$, that is, expressed as follows

$$\begin{pmatrix} -22 & 6 & 0 & 0 \\ 6 & -42 & 12 & 14 \\ 0 & 12 & -24 & 4 \\ 0 & 14 & 4 & -23 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{pmatrix} = \begin{pmatrix} -28 \\ 0 \\ 36 \\ -42 \end{pmatrix} \qquad (13)$$

The system (13) is called linear system of equations.

2.2.2. Model Problem 2 (MP2)

An ordinary differential equation controlling the deflection of a beam with load is given by

$$E_1Cx'' = \frac{\mu Lu^2}{2} - \frac{\mu u^2}{2}, \tag{14}$$

where E_1 means Young modulus of the material, μ means linear load, C is called moment of inertia of the section, L means fixed length of the loaded part, and x represents the length of interest with boundary conditions $x(0) = x(L) = 0$. The problem of finding $x(u)$ when $E_1 = 15$Gpa, $C = 3\text{cm}^4$, $\mu = 15$kgN/m and $L = 3$m is considered. The values of $x(u)$ are computed by employing the FDM with step size $\triangle u = 1/2$.

The equation (14) is discretized to a system of the form of $Bu = c$ as follows;

$$\begin{aligned} 2E_1Cx''\Big|_{u_i} &= \mu Lu_i - \mu u_i^2 \\ \text{Recall:} \quad x''\Big|_{u_i} &= \frac{x_{i+1} - 2x_i + y_{i-1}}{\triangle u^2} \\ \Rightarrow (2E_1C)\left(\frac{x_{i+1} - 2x_i + x_{i-1}}{\triangle u^2}\right) &= \mu Lu_i - \mu u_i^2 \end{aligned}$$

But it is known that $\triangle u = 0.5$, thus

$$\begin{aligned} (8E_1C)(x_{i+1} - 2x_i + x_{i-1}) &= \mu Lu_i - \mu u_i^2, \\ x_{i+1} - 2x_i + x_{i-1} &= \frac{1}{8E_1C}(\mu Lu_i - \mu u_i^2), \end{aligned}$$

$$x_{i+1} - 2x_i + x_{i-1} = \frac{1}{8}(u_i - u_i^2). \tag{15}$$

We write equation (15) at each interior points
$u_0 = 0, \quad u_1 = 0.5, \quad u_2 = 1, \quad u_3 = 1.5, \quad u_4 = 2,$
$u_5 = 2.5, \quad u_6 = 3, \quad x_0 = 0,$ and $x_6 = 0.$

$$\text{At} \quad u_1 = 0.5: \quad -2x_1 + x_2 = 0.15625 \tag{16}$$
$$\text{At} \quad u_2 = 1: \quad x_1 - 2x_2 + x_3 = 0.25 \tag{17}$$
$$\text{At} \quad u_3 = 1.5: \quad x_2 - 2x_3 + x_4 = 0.28125 \tag{18}$$
$$\text{At} \quad u_4 = 2: \quad x_3 - 2x_4 + x_5 = 0.25 \tag{19}$$
$$\text{At} \quad u_5 = 2.5: \quad x_4 - 2x_5 = 0.15625 \tag{20}$$

Combining equations (16) up to (20) so that we may have the following system

$$\begin{cases} -2x_1 + x_2 = 0.15625 \\ x_1 - 2x_2 + x_3 = 0.25 \\ x_2 - 2x_3 + x_4 = 0.28125 \\ x_3 - 2x_4 + x_5 = 0.25 \\ x_4 - 2x_5 = 0.15625 \end{cases} \tag{21}$$

The system (21) may be expressed in the form of $A_2x = b$;

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0.15625 \\ 0.25 \\ 0.28125 \\ 0.25 \\ 0.15625 \end{pmatrix} \tag{22}$$

2.2.3. Model Problem 3 (MP3)

The differential equation below represents the damped vibrations of a stretching spring

$$m\frac{d^2x}{du^2}+u\frac{dx}{du}+\frac{k}{4}x \quad = \quad 2u, \tag{23}$$

where m is a mass of the body in kilograms, k is a spring constant, u is the distance of oscillations, x is the interval of oscillations, $m=1, k=4, 0\le u\le 0.9$ with boundary conditions $x(0)=x(0.9)=1$. The problem of finding the value of $x(0.6)$ using step size $\triangle u=0.1$ is considered.

The equation (23) is discretized to linear equation as follows;

$$\left.\frac{d^2x}{du^2}\right|_{u_i}+u_i\left.\frac{dx}{du}\right|_{u_i}+x_i \quad = \quad 2u_i.$$

$$\text{Recall:}\qquad \left.\frac{d^2x}{du^2}\right|_{u_i} \quad = \quad \frac{x_{i+1}-2x_i+x_{i-1}}{\triangle u^2},$$

$$\text{and}\qquad \left.\frac{dx}{du}\right|_{u_i} \quad = \quad \frac{x_{i+1}-x_{i-1}}{2\triangle u}, \qquad \text{then we have}$$

$$\frac{x_{i+1}-2x_i+x_{i-1}}{\triangle u^2}+u_i\left(\frac{x_{i+1}-x_{i-1}}{2\triangle u}\right)+x_i \quad = \quad 2u_i. \tag{24}$$

Simplifying equation (24) yields

$$x_{i+1}-2x_i+x_{i-1}+\frac{\triangle u}{2}u_i(x_{i+1}-x_{i-1})+\triangle u^2x_i \quad = \quad 2u_i\triangle u^2 \tag{25}$$

Substituting $\triangle u=0.1$ into equation (25) above leads to equation (26)

$$(2+(0.1u_i))x_{i+1}+(2-(0.1u_i))x_{i-1}-3.98x_i \quad = \quad 0.04u_i. \tag{26}$$

We write equation (26) at each interior points
$u_0=0,\quad u_1=0.1,\quad u_2=0.2,\quad u_3=0.3,\quad u_4=0.4,\qquad u_5=0.5,$
$u_7=0.6,\quad u_7=0.7,\quad u_8=0.8,\quad u_9=0.9\quad u_0=1,\quad u_9=1$ as follows:

$$\text{At}\quad u_1=0.1: \quad -3.98x_1+2.01x_2 \quad = \quad -1.986 \tag{27}$$

$$\text{At}\quad u_2=0.2: \quad 1.98x_1-3.98x_2+2.02x_3 \quad = \quad 0.008 \tag{28}$$

$$\text{At}\quad u_3=0.3: \quad 1.97x_2-3.98x_3+2.03x_4 \quad = \quad 0.012 \tag{29}$$

$$\text{At}\quad u_4=0.4: \quad 1.96x_3-3.98x_4+2.04x_5 \quad = \quad 0.016 \tag{30}$$

$$\text{At}\quad u_5=0.5: \quad 1.95x_4-3.98x_5+2.05x_6 \quad = \quad 0.02 \tag{31}$$

$$\text{At}\quad u_6=0.6: \quad 1.94x_5-3.98x_6+2.06x_7 \quad = \quad 0.024 \tag{32}$$

$$\text{At}\quad u_7=0.7: \quad 1.93x_6-3.98x_7+2.07x_8 \quad = \quad 0.028 \tag{33}$$

$$\text{At}\quad u_8=0.8: \quad 1.92x_7-3.98x_8 \quad = \quad -2.048 \tag{34}$$

Combining equations (27) up to (34) so that we may have the following system

$$\left\{\begin{array}{rcl} -3.98x_1+2.01x_2 & = & -1.986\\ 1.98x_1-3.98x_2+2.02x_3 & = & 0.008\\ 1.97x_2-3.98x_3+2.03x_4 & = & 0.012\\ 1.96x_3-3.98x_4+2.04x_5 & = & 0.016\\ 1.95x_4-3.98x_5+2.05x_6 & = & 0.02\\ 1.94x_5-3.98x_6+2.06x_7 & = & 0.024\\ 1.93x_6-3.98x_7+2.07x_8 & = & 0.028\\ 1.92x_7-3.98x_8 & = & -2.048 \end{array}\right. \tag{35}$$

The system (35) may be expressed in the form of $A_3x = b$;

$$\begin{pmatrix} -3.98 & 2.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.98 & -3.98 & 2.02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.97 & -3.98 & 2.03 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.96 & -3.98 & 2.04 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.95 & -3.98 & 2.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.94 & -3.98 & 2.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.93 & -3.98 & 2.07 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.92 & -3.98 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} -1.986 \\ 0.008 \\ 0.012 \\ 0.016 \\ 0.02 \\ 0.024 \\ 0.028 \\ -2.048 \end{pmatrix} \tag{36}$$

2.2.4. Model Problem 4 (MP4)

The system shown in Figure 2 having k linear springs supported by k masses. C_j, denotes the spring stiffness's are denoted, the masses weight are denoted by M_j and the expulsion of the masses are denoted by y_i, the displacements are measured from the springs before deformation. The formulation of expulsion is given by composing equilibrium function of the masses and replacing $E_i + C_iy_i = C_iy_{i+1}$ for spring forces.

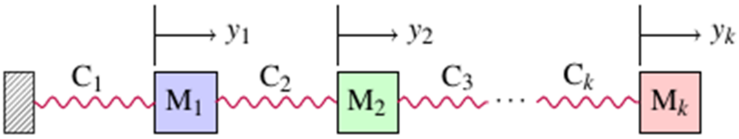

Figure 2. *Linear Springs.*

The linear equations for springs can be obtained by the system below;

$$\begin{cases} (C_1 + C_2)y_1 - C_2y_2 &= M_1 \\ -C_iy_{i-1} + (C_i + C_{i+1})y_i - C_{i+1}y_{i+1} &= M_i, \qquad 2 \le i \le k-1 \\ -C_ny_{k-1} + C_ky_k &= M_k \end{cases} \tag{37}$$

Considering six springs ($k = 6$) and suppose that $\begin{matrix} C_1 = C_2 = C_3 = 10N/mm, & C_4 = C_5 = C_6 = 5N/mm \\ M_1 = M_3 = M_5 = 100N, & M_2 = M_4 = M_6 = 50N \end{matrix}$

Plugging the above constants into equation (37) we have the system below;

$$\begin{cases} 20y_1 - 10y_2 &= 100 \\ -10y_1 + 20y_2 - 10y_3 &= 50 \\ -10y_2 + 15y_3 - 5y_4 &= 100 \\ 5y_3 + 10y_4 - 5y_5 &= 50 \\ -5y_4 + 10y_5 - 5y_6 &= 100 \\ -5y_5 + 5y_6 &= 50 \end{cases} \tag{38}$$

The system (38) may be expressed in the form of $A_4y = b$;

$$\begin{pmatrix} 20 & -10 & 0 & 0 & 0 & 0 \\ -10 & 20 & -10 & 0 & 0 & 0 \\ 0 & -10 & 15 & -5 & 0 & 0 \\ 0 & 0 & 5 & 10 & -5 & 0 \\ 0 & 0 & 0 & -5 & 10 & -5 \\ 0 & 0 & 0 & 0 & -5 & 5 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 100 \\ 50 \\ 100 \\ 50 \\ 100 \\ 50 \end{pmatrix} \tag{39}$$

2.2.5. Model Problem 5 (MP5)

The problem of determining the equilibrium temperatures at nine points labeled as $t_1 \cdots t_9$ within a trapezium plate represented in Figure 3 is considered.

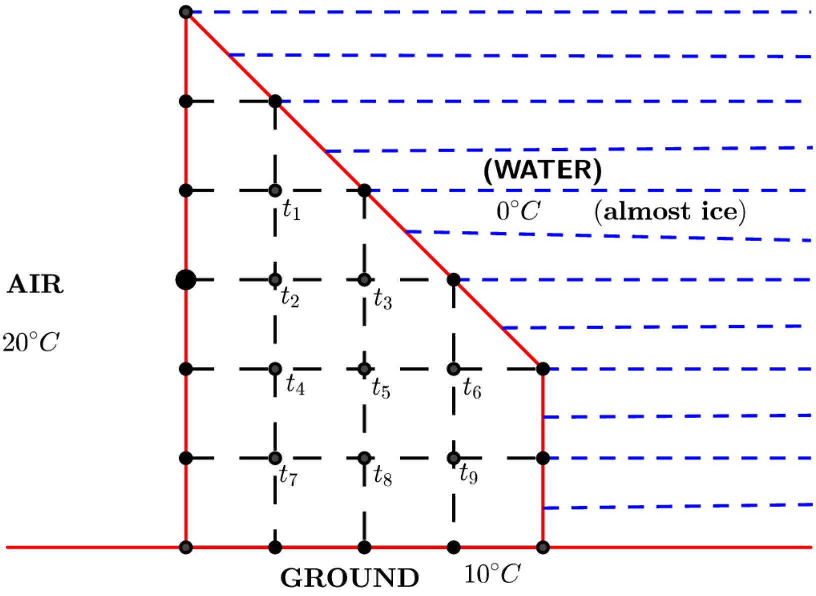

Figure 3. *Temperature Distribution on Heated Plate.*

The four nearest grid points: West, North, East, and South, have their temperatures recorded in clockwise order on the map. For example, t_1 is the average of four temperatures, or the sum of four temperatures divided by four, where West = 20, North = 0, East = 0, and South = t_2. The following nine linear equations are derived from the discretized mean value property.

$$t_1 = \frac{1}{4}(20+0+0+t_2) \Rightarrow 4t_1 - t_2 = 20 \tag{40}$$

$$t_2 = \frac{1}{4}(20+t_1+t_4+t_3) \Rightarrow -t_1 + 4t_2 - t_3 - t_4 = 20 \tag{41}$$

$$t_3 = \frac{1}{4}(t_2+0+0+t_5) \Rightarrow -t_2 + 4t_3 - t_5 = 0 \tag{42}$$

$$t_4 = \frac{1}{4}(20+t_2+t_5+t_7) \Rightarrow -t_2 + 4t_4 - t_5 - t_7 = 20 \tag{43}$$

$$t_5 = \frac{1}{4}(t_4+t_3+t_6+t_8) \Rightarrow -t_3 - t_4 + 4t_5 - t_6 - t_8 = 0 \tag{44}$$

$$t_6 = \frac{1}{4}(t_5+0+0+t_9) \Rightarrow -t_5 + 4t_6 - t_9 = 0 \tag{45}$$

$$t_7 = \frac{1}{4}(20+t_4+10+t_8) \Rightarrow -t_4 + 4t_7 - t_8 = 30 \tag{46}$$

$$t_8 = \frac{1}{4}(t_7+t_5+10+t_9) \Rightarrow -t_5 - t_7 + 4t_8 - t_9 = 10 \tag{47}$$

$$t_9 = \frac{1}{4}(t_6+t_8+10+0) \Rightarrow -t_6 - t_8 + 4t_9 = 10 \tag{48}$$

$$\tag{49}$$

Combining equations (40) up to (48) so that we may have the following system

$$\left\{\begin{array}{rcl} 4t_1 - t_2 & = & 20 \\ -t_1 + 4t_2 - t_3 - t_4 & = & 20 \\ -t_2 + 4t_3 - t_5 & = & 0 \\ -t_2 + 4t_4 - t_5 - t_7 & = & 20 \\ -t_3 - t_4 + 4t_5 - t_6 - t_8 & = & 0 \\ -t_5 + 4t_6 - t_9 & = & 0 \\ -t_4 + 4t_7 - t_8 & = & 30 \\ -t_5 - t_7 + 4t_8 - t_9 & = & 10 \\ -t_6 - t_8 + 4t_9 & = & 10 \end{array}\right. \tag{50}$$

The system (50) may be expressed in the form of $A_5 t = b$;

$$\begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{pmatrix} = \begin{pmatrix} 20 \\ 20 \\ 0 \\ 20 \\ 0 \\ 0 \\ 30 \\ 10 \\ 10 \end{pmatrix} \tag{51}$$

3. Results and Discussion

3.1. Comparison of SOR and RSOR

In this subsection, we thoroughly examine the effectiveness of refined successive over-relaxation (RSOR) and successive over-relaxation (SOR). Key performance indicators, such as calculation time (in seconds), number of iterations, and rates of error across various real-world phenomena, form the basis of the evaluation. Our experiments were executed on MATLAB R2018a 64-bit, utilizing an Intel(R) Core(TM)i5-5200U CPU@2.20GHz with a 64-bit operating system. Five different real-world phenomena are considered; they are referred to as Model Problem 1 (MP1) through Model Problem 5 (MP5). We provide detailed tables and graphs for every model problem that compare the SOR and RSOR approaches in terms of computational time, number of iterations, and error. These visual depictions serve as essential instruments for interpreting the distinctions in terms of performance between SOR and RSOR methods. Moreover, a general remark is included to provide insightful comment and enhance the overall understanding of the presented data.

3.1.1. Comparison of Computational Time (in Seconds), Number of Iterations (NIT) and Error for MP1

Numerical simulations for Model Problem 1 were executed using MATLAB software, with a maximum of 100 iterations and a tolerance of 10^{-5}. The results are as presented in Table 1.

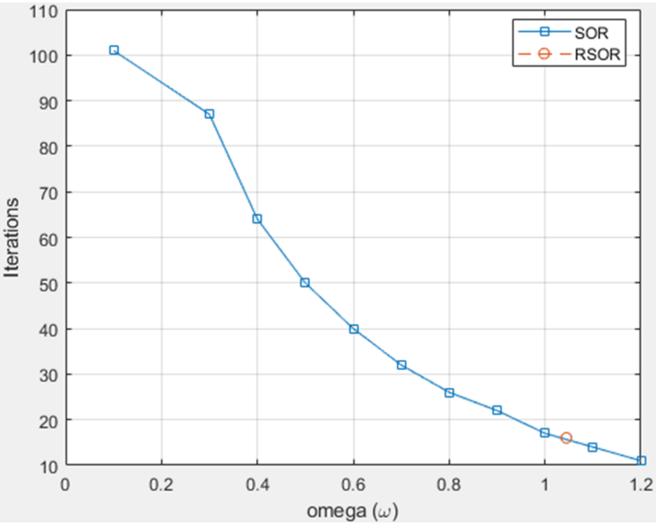

Figure 4. *Variations of omega against number of Iterations for MP1.*

The comparison between SOR and RSOR is visually depicted in Figures 4, 5, and 6, as derived from the data presented in Table 1.

Table 1. *Comparison of SOR and RSOR methods for MP1.*

		SOR				RSOR	
ω	NIT	Time (s)	Error	ω	NIT	Time (s)	Error
0.1	101	3.32×10^{-2}	1.19×10^{-3}	1.0446	16	0.00018	5.19×10^{-6}
0.3	87	4.20×10^{-4}	9.88×10^{-6}				
0.4	64	1.12×10^{-4}	9.36×10^{-6}				
0.5	50	2.50×10^{-4}	8.23×10^{-6}				
0.6	40	4.18×10^{-4}	7.75×10^{-6}				
0.7	32	7.63×10^{-5}	8.72×10^{-6}				
0.8	26	9.53×10^{-5}	9.08×10^{-6}				
0.9	22	2.07×10^{-5}	5.86×10^{-6}				
1	17	1.72×10^{-5}	9.28×10^{-6}				
1.1	14	1.34×10^{-5}	4.03×10^{-6}				
1.2	11	1.17×10^{-5}	6.29×10^{-6}				

In Figure 4, it is evident that the number of iterations decreases with an increase in the value of omega for the SOR method. From an iteration efficiency perspective, at the desirable solvent, the graphical depiction clearly shows that the SOR approach is superior to the RSOR method. Notably, the graph reveals that RSOR achieves faster convergence when the relaxation parameter (ω) is set to $\omega = 1.0446$, whereas the SOR method converges at $\omega = 1.2$ and diverges at $\omega = 0.1$.

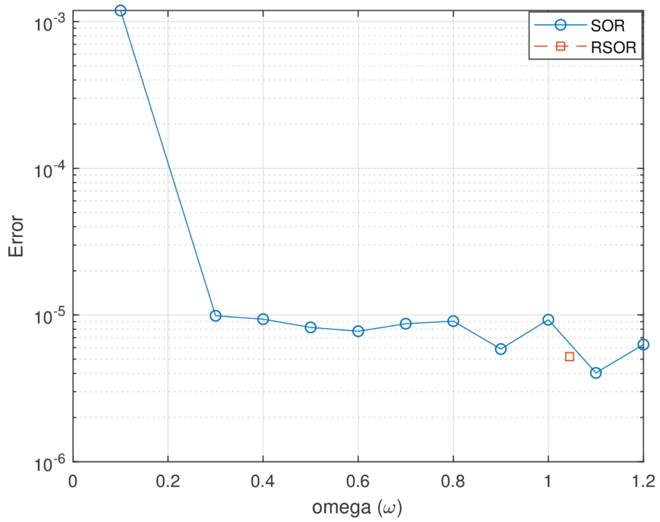

Figure 5. *Variations of omega against Error for MP1.*

The depicted figure above illustrates a diminishing trend in the error associated with the SOR method as the relaxation parameter (ω)increases. Specifically, in Figure 5, it is evident that the error for the RSOR method at the optimal relaxation factor is consistently lower than that of the SOR method. This observation underscores the superior performance of the SOR method in terms of error magnitude. More interestingly, the error for SOR method is less than that of RSOR method at $\omega = 1.1$, implying that the SOR converges faster than the RSOR method when $\omega = 1.1$. However, it's worth noting that the SOR method exhibits reduced efficiency at varying relaxation parameters compared to the RSOR method.

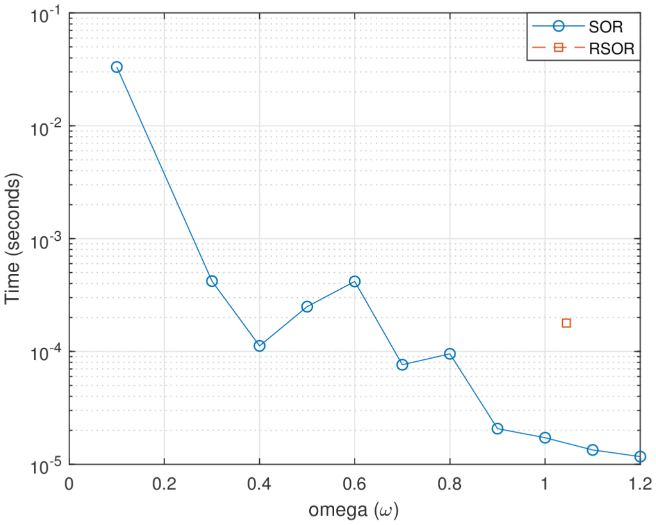

Figure 6. *Variations of omega against computational time for MP1.*

Referring to Figure 6, it is evident that changes in the relaxation parameter (ω) significantly impact the computational time for the SOR method. The results reveal that the increase of relaxation parameter leads to the decrease of computational time for SOR method. Notably, at the optimal relaxation factor, the computational time for the SOR technique proves to be superior to that of the RSOR technique. In summary, considering three key criteria: computational time, number of iterations, and error; the results strongly support the trial-and-error technique (SOR) for determining the relaxation factor(ω) over using a formula. This underscores the superiority of the SOR method compared to the RSOR technique. Interestingly, the RSOR technique demonstrates greater efficiency than the SOR technique across all relaxation parameters except for the optimal (ω) . For RSOR, the optimal relaxation parameter lies in the range $\omega = 1.0$ to $\omega = 1.2$, specifically at $\omega = 1.0446$.

3.1.2. Comparison of Computational Time (in Seconds), Number of Iterations (NIT) and Error for MP2

Numerical simulations for Model Problem 2 were executed using MATLAB software, with a maximum of 100 iterations and a tolerance of 10^{-5}. The results are as presented in Table 2.

Table 2. *Comparison of SOR and RSOR methods for MP2.*

		SOR				RSOR	
ω	**NIT**	**Time (s)**	**Error**	**ω**	**NIT**	**Time (s)**	**Error**
0.1	101	9.32×10^{-2}	3.75×10^{-3}				
0.8	55	1.11×10^{-1}	9.09×10^{-6}				
0.9	45	7.84×10^{-5}	9.62×10^{-6}				
1.0	37	1.46×10^{-4}	9.34×10^{-6}				
1.1	30	5.99×10^{-3}	9.56×10^{-6}				
1.2	24	5.30×10^{-5}	8.22×10^{-6}	1.0518	34	0.0074	7.44×10^{-6}
1.3	18	8.07×10^{-5}	5.20×10^{-6}				

SOR				RSOR			
ω	NIT	Time (s)	Error	ω	NIT	Time (s)	Error
1.4	15	2.41×10^{-5}	6.51×10^{-6}				
1.5	18	9.30×10^{-6}	6.89×10^{-6}				
1.6	24	1.08×10^{-5}	3.70×10^{-6}				
1.7	33	1.63×10^{-5}	8.14×10^{-6}				
1.8	53	5.09×10^{-5}	8.63×10^{-6}				

The comparison between SOR and RSOR is visually depicted in Figures 7, 8, and 9, as derived from the data presented in Table 2.

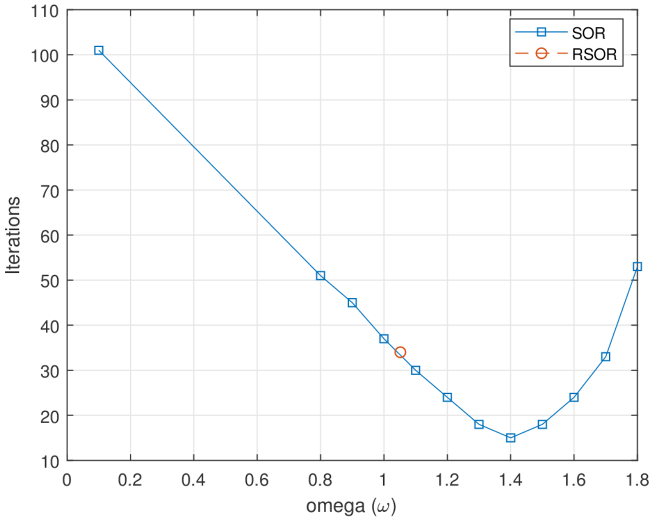

Figure 7. Variations of omega against number of Iterations for MP2.

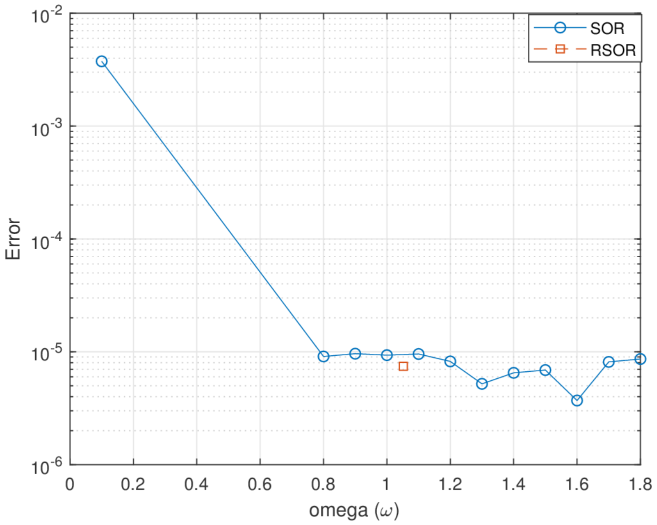

Figure 8. Variations of omega against Error for MP2.

Figure 7 illustrates that, with the increase of the relaxation factor (ω) in the Successive Over-relaxation (SOR) technique, the number of iterations decreases until reaching the optimal solvent, beyond which divergence occurs immediately. The graph also highlights that the Refined Successive Over-relaxation (RSOR) method exhibits superior performance with the minimum number of iterations compared to the traditional SOR method. Moreover, the graph demonstrates that SOR achieves faster convergence when $\omega = 1.4$, whereas RSOR converges at $\omega = 1.0518$. More interestingly, the graph depicts that the SOR method diverges when $\omega = 0.1$ and immediately at $\omega > 1.4$.

Upon examining Figure 8, it is evident that at the optimal solvent point, the error obtained by the Successive Over-relaxation (SOR) technique surpasses that of the Refined Successive Over-relaxation (RSOR) technique. Despite this, the RSOR technique consistently exhibits remarkable performance, maintaining its superiority even when the parameter for relaxation is varied in the SOR technique. Furthermore, the graph highlights slight differences in errors, particularly within a range $0.8 \leq \omega \leq 1.2$. Notably, during this range, the RSOR method demonstrates faster convergence compared to the conventional SOR technique. The optimal convergence for SOR is attained when $\omega = 1.6$.

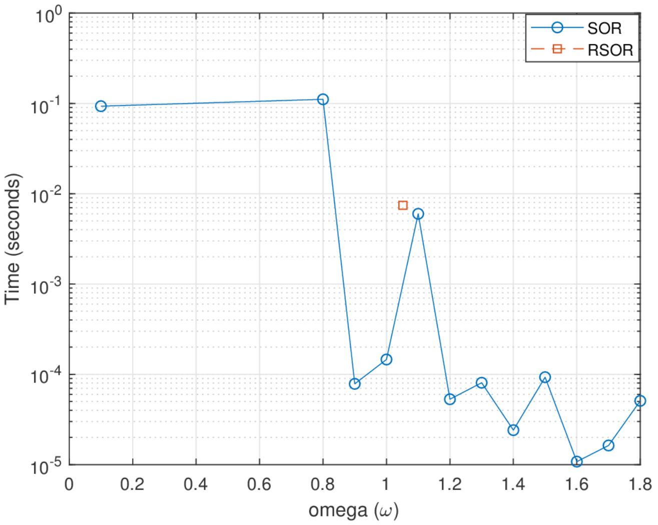

Figure 9. Variations of omega against computational time for MP2.

When $\omega > 1.1$, the Successive Over-relaxation (SOR) method demonstrates greater efficiency in comparison to Refined Successive Over-relaxation (RSOR) technique, particularly in terms of computational time, as depicted in Figure 9. More interestingly, the SOR technique exhibits reduced efficiency compared to RSOR in computational time when the parameter for relaxation falls within the range $0 <$

$\omega \leq 0.9$. In a nutshell, the optimal parameter for relaxation falls within the range $1.0 < \omega < 1.2$. Analyzing the results at desirable solvent presented in Figures 7, 8, and 9, it becomes evident that the Successive Over-relaxation (SOR) technique outperforms the Refined Successive Over-relaxation (RSOR) technique.

3.1.3. Comparison of Computational Time (in Seconds), Number of Iterations (NIT) and Error for MP3

Numerical simulations for Model Problem 3 were executed using MATLAB software, with a maximum of 100 iterations and a tolerance of 10^{-5}. The results are as presented in Table 3.

Table 3. *Comparison of SOR and RSOR methods for MP3.*

SOR				RSOR			
ω	NIT	Time (s)	Error	ω	NIT	Time (s)	Error
0.1	101	1.39×10^{-1}	4.11×10^{-3}				
0.3	101	2.08×10^{-4}	3.43×10^{-3}				
0.4	101	4.49×10^{-4}	2.08×10^{-3}				
1	82	3.71×10^{-4}	9.62×10^{-6}				
1.1	68	1.06×10^{-4}	9.42×10^{-6}				
1.2	56	1.51×10^{-4}	8.86×10^{-6}	1.0623	73	0.00208	9.55×10^{-6}
1.3	45	2.03×10^{-4}	8.64×10^{-6}				
1.4	35	2.25×10^{-4}	7.36×10^{-6}				
1.5	22	1.78×10^{-4}	8.43×10^{-6}				
1.6	27	5.92×10^{-5}	5.02×10^{-6}				
1.7	36	6.64×10^{-5}	7.55×10^{-6}				
1.8	55	9.41×10^{-5}	4.21×10^{-6}				

The comparison between SOR and RSOR is visually depicted in Figures 10, 11, and 12, as derived from the data presented in Table 3.

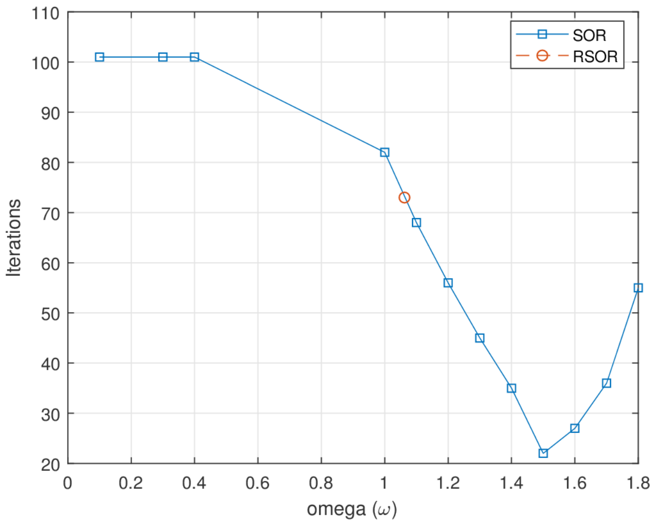

Figure 10. *Variations of omega against number of Iterations for MP3.*

Examining Figure 10, the findings indicate a constant number of iterations for the Successive Over-relaxation (SOR) technique across the range $0.1 \leq \omega \leq 0.4$. Notably, the quantity of iterations for the SOR technique is less than that required for the Refined Successive Over-relaxation (RSOR), establishing the superiority of the SOR technique in terms of quantity of iterations. Specifically, the SOR technique achieves faster convergence when $\omega = 1.5$, with 22 iterations, further emphasizing its stability compared to RSOR at the desirable solvent.

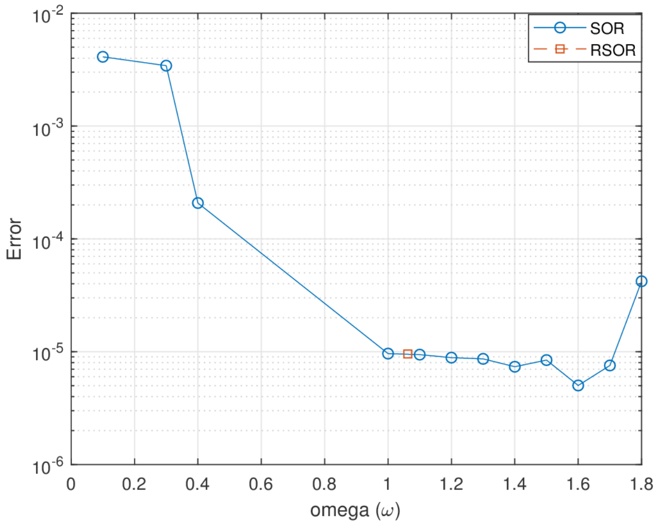

Figure 11. *Variations of omega against Error for MP3.*

Referencing Figure 11 , the Refined Successive Over-relaxation (RSOR) method exhibits superior performance over the Successive Over-relaxation (SOR) method in terms of magnitude error, particularly within the range $0.1 \leq \omega \leq 1.0$. More interestingly, the trial-and-error approach employed by SOR proves to be more effective than RSOR in terms of convergence when $\omega > 1.0$. Additionally, the figure above illustrates that the magnitude error of the SOR method at the optimal solvent is less than that of RSOR, substantiating the superiority of the SOR method over RSOR method.

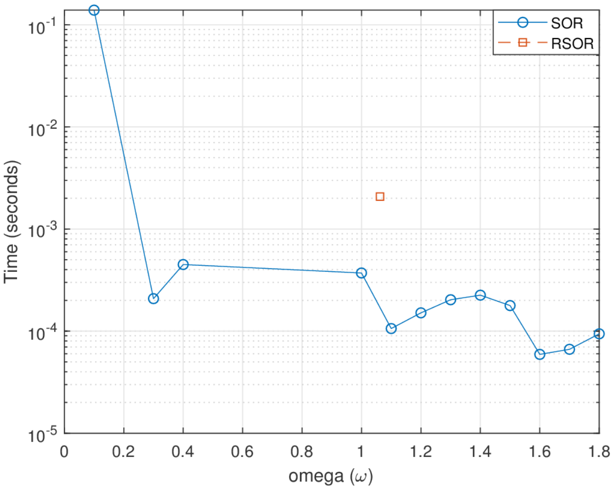

Figure 12. *Variations of omega against computational time for MP3.*

The Successive Over-relaxation (SOR) method exhibits better computational efficiency than the Refined Successive Over-relaxation (RSOR) method, except in cases where $\omega = 0.1$ and $\omega = 1.0$, as illustrated in Figure 12. Notably, the graph emphasizes that the SOR method achieves faster convergence, specifically when $\omega = 1.6$, with a computational time of 5.92×10^{-5} seconds, whereas the RSOR method exhibits comparatively slower convergence towards the optimal solvent. In summary, when employed for Model Problem 3, the computational time, quantity of iterations, and magnitude error favor the Successive Over-relaxation (SOR) technique over the Refined Successive Over-relaxation (RSOR) technique. The range of values for the ideal relaxation parameter of the SOR technique is $1.5 < \omega < 1.7$.

3.1.4. Comparison of Computational Time (in Seconds), Number of Iterations (NIT) and Error for MP4

Numerical simulations for Model Problem 4 were executed using MATLAB software, with a maximum of 100 iterations and a tolerance of 10^{-5}. The results are as presented in Table 4.

Table 4. *Comparison of SOR and RSOR methods for MP4.*

SOR				RSOR			
ω	NIT	Time (s)	Error	ω	NIT	Time (s)	Error
0.1	101	4.71×10^{-1}	1.19×10^{-1}	1.05664	28	0.00197	9.90×10^{-6}
0.3	101	4.69×10^{-4}	7.22×10^{-4}				
0.4	101	3.58×10^{-4}	7.61×10^{-5}				
0.5	94	5.84×10^{-4}	8.80×10^{-6}				
0.6	74	1.13×10^{-4}	8.92×10^{-6}				
0.7	59	3.92×10^{-4}	9.81×10^{-6}				
0.8	48	2.24×10^{-4}	9.34×10^{-6}				
0.9	39	1.79×10^{-4}	9.22×10^{-6}				
1	32	1.70×10^{-4}	7.56×10^{-6}				
1.1	26	4.69×10^{-5}	7.57×10^{-6}				
1.2	20	3.64×10^{-5}	4.59×10^{-6}				
1.3	21	3.77×10^{-5}	4.49×10^{-6}				
1.4	27	4.64×10^{-5}	5.67×10^{-6}				

The comparison between SOR and RSOR is visually depicted in Figures 13, 14, and 15, as derived from the data presented in Table 4.

Analyzing Figure 13, it becomes evident that the quantity of iterations for the Successive Over-relaxation (SOR) technique remains constant within the range $0.1 \le \omega \le 0.4$. Notably, SOR diverges within the range $0.1 \le \omega \le 0.4$. Despite this, the optimal number of iterations for achieving desirable solvents with SOR is still lower than that of the Refined Successive Over-relaxation (RSOR). Notably, the RSOR method converges faster tha SOR method within the range $0.1 \le \omega \le 1.04$. Consequently, the SOR technique proves to be superior to the RSOR method when considering the quantity of iterations required at the targeted solvent. The illustration reveals that, for $0.1 \le \omega \le 1.04$ there is no convergence because the tolerance is not met, that is the tolerance is greater than 10^{-5}.

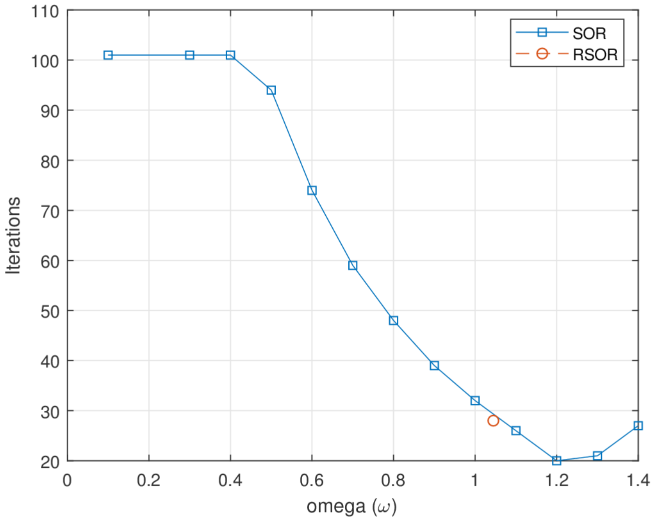

Figure 13. *Variations of omega against number of Iterations for MP4.*

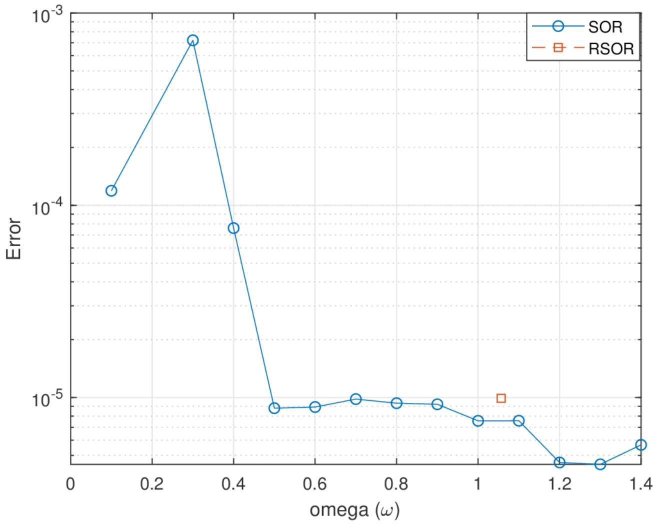

Figure 14. *Variations of omega against Error for MP4.*

The data presented in Figure 14 highlights a decrease in the magnitude error of the Successive Over-relaxation (SOR) technique with an increase in the relaxation parameter (ω). Notably, the figure illustrates that the magnitude error for the Successive Over-relaxation (SOR) method consistently outperforms that of RSOR especially when $0.1 \leq \omega \leq 0.4$. Consequently, the RSOR technique demonstrates superior efficiency over the SOR method in terms of magnitude error when $0.1 \leq \omega \leq 0.4$. Also, the graph depicts that the SOR method converges faster when the relaxation parameter ($\omega = 1.3$).

The Successive Over-relaxation (SOR) technique exhibits superior computational efficiency compared to the Refined Successive Over-relaxation (RSOR) technique, with an exception occurring when $\omega = 0.1$ and $\omega = 0.2$, as depicted in Figure 15. Additionally, the graph above clearly demonstrates that the RSOR technique experiences divergence throughout the experiment, suggesting that the SOR technique holds superiority over the SOR technique. In summary, the analyses of Figure 13, Figure 14, and Figure 15 Unquestionably, establish the superiority of the Successive Over-relaxation (SOR) technique over the Refined Successive Over-relaxation (RSOR) technique.

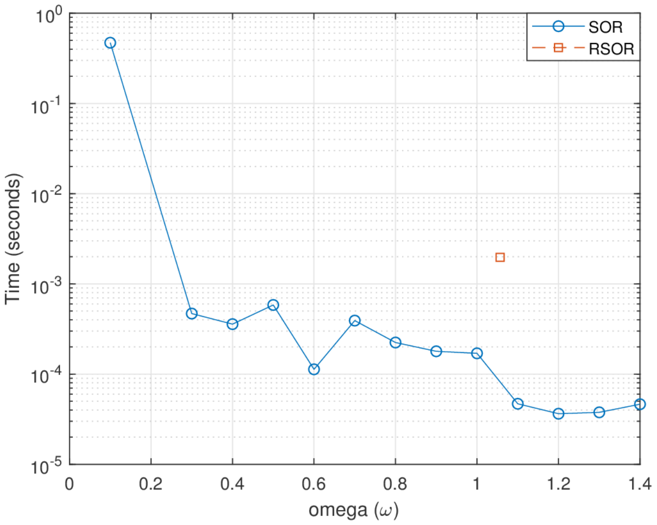

Figure 15. *Variations of omega against computational time for MP4.*

3.1.5. Comparison of Computational Time (in Seconds), Number of Iterations (NIT) and Error for MP5

Numerical simulations for Model Problem 5 were executed using MATLAB software, with a maximum of 100 iterations and a tolerance of 10^{-5}. The results are as presented in Table 5.

Table 5. *Comparison of SOR and RSOR methods for MP5.*

SOR				RSOR			
ω	**NIT**	**Time (s)**	**Error**	**ω**	**NIT**	**Time (s)**	**Error**
0.1	101	2.97×10^{-1}	1.41×10^{-2}				
0.3	101	3.52×10^{-1}	1.54×10^{-5}				
0.4	76	1.00×10^{-4}	8.75×10^{-6}				
0.5	58	1.63×10^{-4}	8.82×10^{-6}				
0.6	46	4.90×10^{-4}	7.96×10^{-6}				
0.7	37	9.54×10^{-4}	7.37×10^{-6}				
0.8	30	1.58×10^{-4}	6.70×10^{-6}	1.0640	16	0.00022	7.31×10^{-6}
0.9	24	1.04×10^{-4}	7.21×10^{-6}				
1.0	19	3.47×10^{-5}	7.45×10^{-6}				
1.1	14	1.32×10^{-5}	9.38×10^{-6}				
1.2	12	1.25×10^{-5}	2.43×10^{-6}				
1.3	14	1.37×10^{-5}	8.52×10^{-6}				
1.4	18	1.90×10^{-5}	3.90×10^{-6}				

The comparison between SOR and RSOR is visually depicted in Figures 16, 17, and 18, as derived from the data presented in Table 5.

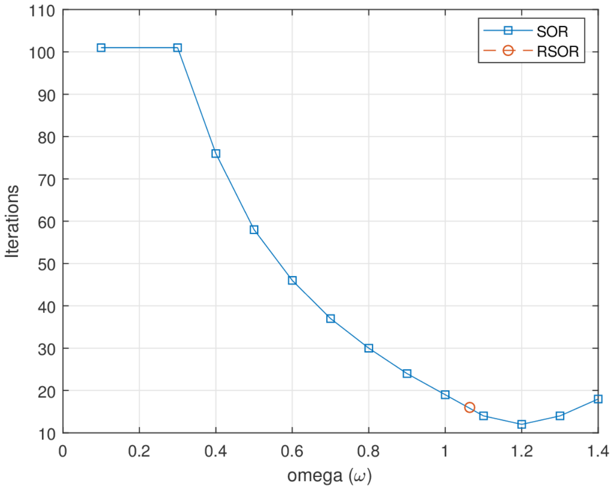

Figure 16. *Variations of omega against number of Iterations for MP5.*

Examining Figure 16, it is observed that the quantity of iterations for Successive Over-relaxation (SOR) technique remains constant within the range $\omega = 0.1$ to $\omega = 0.3$. Additionally, the results indicate a decrease in the quantity of iterations for the SOR technique with an increase in the relaxation parameter (ω). However, beyond $\omega = 1.2$, the number of iterations for SOR diverges immediately upon reaching the desirable solvent, occurring at $\omega = 1.2$, where the number of iterations is 12, this is notably fewer than the corresponding number for the Refined Successive Over-relaxation (RSOR) method. Consequently, the SOR technique demonstrates superior efficiency over the RSOR technique in terms of required quantity of iterations to reach the targeted solvent.

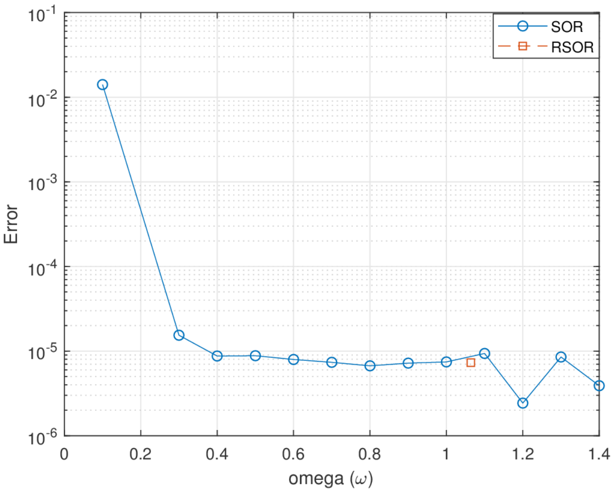

Figure 17. *Variations of omega against Error for MP5.*

The error reduction in the Successive Over-relaxation (SOR) technique is evident with an increase of relaxation parameter (ω), as illustrated in Figure 17. Moreover, the findings reveal that the magnitude error for the Refined Successive Over-relaxation (RSOR) technique consistently surpasses that of the SOR technique, except at the targeted solvent. This observation suggests that RSOR technique converges more rapidly than SOR. Consequently, it is evident that the RSOR technique excels over the conventional SOR technique when considering magnitude error. Notably, the SOR method shows optimal convergence especially at $\omega = 1.2$.

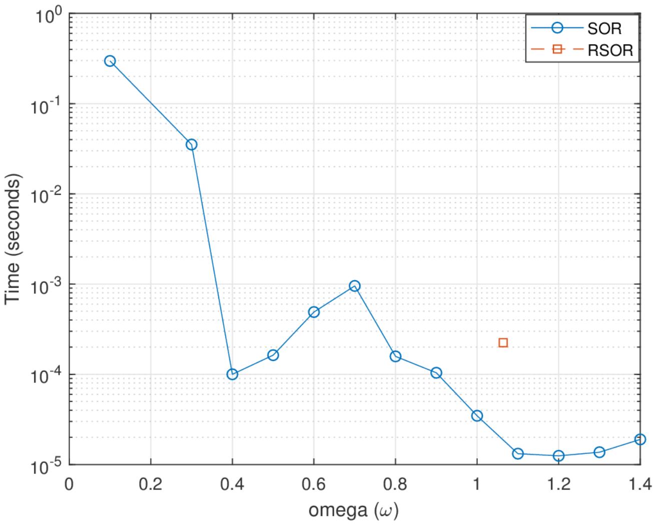

Figure 18. *Variations of omega against computational time for MP5.*

The findings presented in Figure 18 underscore the Model Problem 5 is sensitive to variations in the relaxation parameter. The results indicate that the computational time for SOR decreases as the increase of relaxation parameter throughout the experiment. Additionally, the results suggest that, at the optimal solvent, the Refined Successive Over-relaxation (RSOR) technique exhibits lower computational efficiency compared to the Successive Over-relaxation (SOR) technique. To summarize, the insights drawn from Figure 16, Figure 17, and Figure 18 collectively suggest a preference for the Successive Over-relaxation (SOR) technique over the Refined Successive Over-relaxation (RSOR) technique.

3.2. Condition Number and Sensitivity Analysis on Systems of Linear Equations

According to Rice (1966), a condition number for systems of linear equations $Ax = b$ is given by $k(A) = ||A^{-1}|||A||$. It is very important to note that, the perturbations may be applied to the system of equations with a non-singular matrix otherwise the perturbations cannot be done. It means that the condition number for the system of equations with singular matrix is infinite, i.e. $k(A) = \infty$. In this subsection we investigate the nature of condition number before perturbation and after perturbation and how does it affect the exact solvent of MP1 to MP5. The investigations can be done using Figures 6 & 7.

The effects of perturbations for Model Problem 1-5 were simulated and executed using the MATLAB software, and the results were careful organized and presented in Tables 6 & 7.

Table 6. *Comparison of condition number before and after perturbation for MP1-MP3.*

$\triangle A$	$k(A_1)$	$\lvert k(A_1^0) - k(A_1)\rvert$	$k(A_2)$	$\lvert k(A_2^0) - k(A_2)\rvert$	$k(A_3)$	$\lvert k(A_3^0) - k(A_3)\rvert$
100	18.9614	12.6699	499.6005	485.6723	864.8330	831.4150
1	11.1857	4.8942	4.6351	9.2931	8.3616	25.0564
0.5	8.0312	1.7397	3.7253	10.2029	8.3623	25.0557
0.1	6.5735	0.2820	18.3070	4.3788	15.3031	18.1149
0.01	6.3185	0.0270	16.8531	2.9249	48.4452	15.0272
0.001	6.2942	0.0027	14.9524	1.0242	34.4838	1.0658
0.0001	6.2918	0.0003	13.9524	0.0242	33.5215	0.1035
0.00001	6.2915	0.0000	13.9306	0.0024	33.4283	0.0103
0.00000	6.2915	0.0000	13.9282	0.0000	33.4180	0.0000

Table 7. *Comparison of condition number before and after perturbation for MP4-MP5.*

$\triangle A$	$k(A_4)$	$\lvert k(A_4^0) - k(A_4)\rvert$	$k(A_5)$	$\lvert k(A_5^0) - k(A_5)\rvert$
100	213.8952	186.2222	376.7932	371.6119
1	14.1745	13.4985	4.4602	0.7211
0.5	16.4285	11.2445	2.8442	2.3371
0.1	23.2383	4.4347	3.2609	1.9204
0.01	27.1170	0.5560	4.8774	0.3039
0.001	27.6160	0.0570	5.1491	0.0322
0.0001	27.6673	0.0057	5.1781	0.0032
0.00001	27.6725	0.0005	5.1810	0.0003
0.00000	27.6730	0.0000	5.1813	0.0000

The effects of perturbations on MP1-MP5 is visually depicted in Figures 19, 20, 21, 22 and 23, as derived from the data presented in Tables 6 & 7.

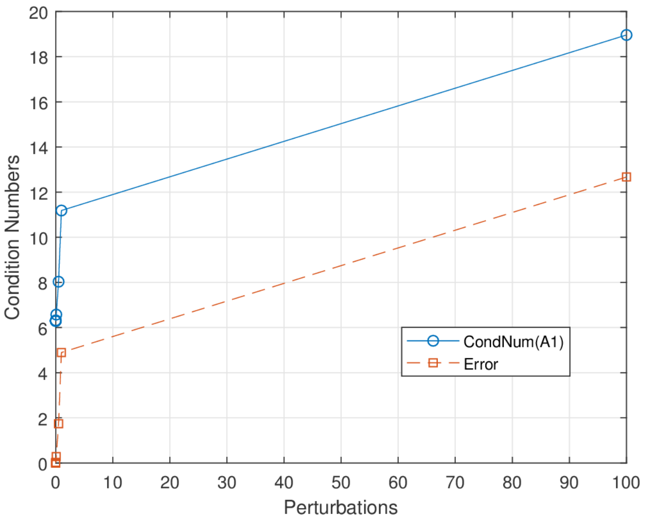

Figure 19. *Effects of perturbations for MP1.*

In Figure 19, the observed results reveal a noteworthy trend: the condition number of Model Problem 1 (MP1) escalates with the augmentation of perturbations on MP1, and conversely, it diminishes as perturbations decrease. The disparity between the condition number of the original matrix's system and that of the modified system contracts proportionally with the reduction in perturbations on MP1. The discernible gap between these two plots underscores the sensitivity of MP1 to even minor perturbations. Consequently, it is evident that the system constitutes an ill-posed problem.

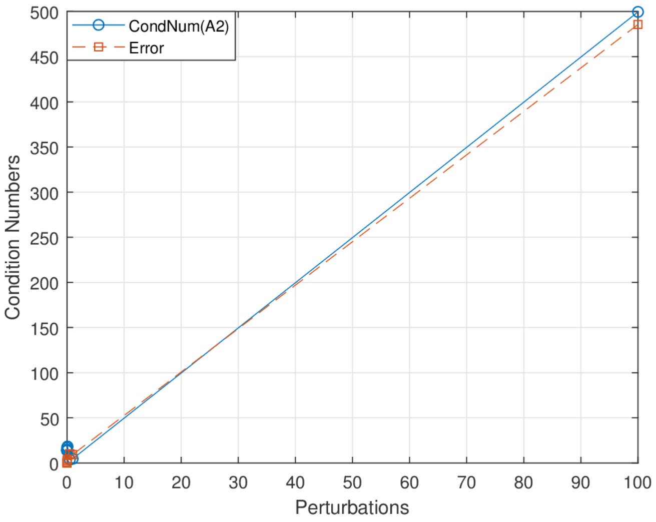

Figure 20. *Effects of perturbations for MP2.*

Referring to Figure 20, the analysis demonstrates a clear correlation: the condition number of Model Problem 2 (MP2) rises with the amplification of perturbations on MP2 and decreases conversely. The graphical representation also illustrates that the discrepancy in magnitude between the original and new condition numbers for MP2 contracts as perturbations in matrix entries decrease. Additionally, the findings indicate that Model Problem 2 transforms into a well-posed problem when subjected to minor perturbations in matrix entries, but becomes an ill-structured problem otherwise.

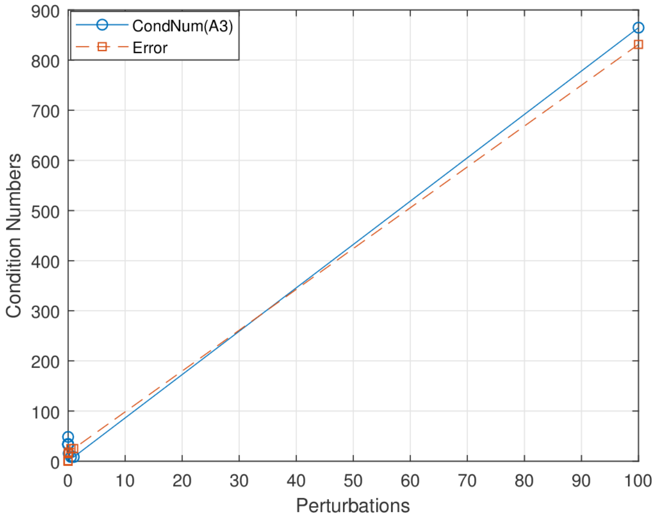

Figure 21. *Effects of perturbations for MP3.*

Upon scrutinizing Figure 21, it becomes evident that alterations in the perturbations of Model Problem 3 (MP3) have a discernible impact on its condition number. The results further highlight that the disparity in magnitude between the original condition number and the new condition number for MP3 increases with escalating perturbations and diminishes vice versa. The sensitivity of the MP3 system to perturbations is clearly illustrated in Figure 21. Consequently, it is unequivocal that Model Problem 3 constitutes an ill-posed problem.

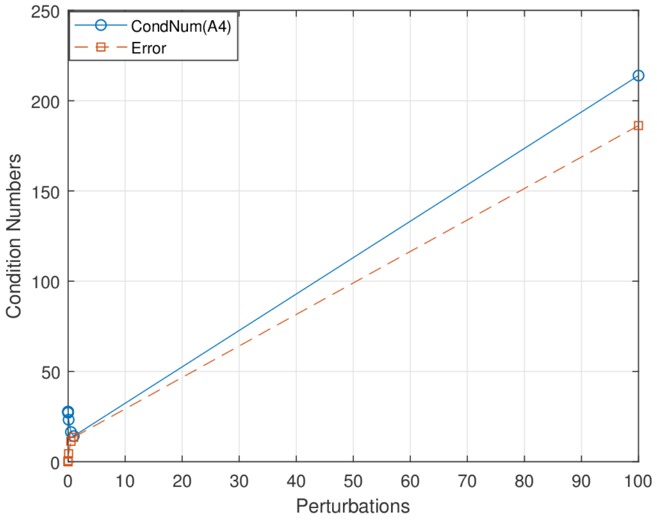

Figure 22. *Effects of perturbations for MP4.*

Examining Figure 22, the findings underscore a noteworthy trend: the condition number of Model Problem 4 (MP4) escalates with the augmentation of perturbations on MP4 and decreases conversely. Furthermore, the contrast between the condition number of the original matrix system and that of the new system contracts with the reduction of perturbation on MP4. The perceptible gap between these two plots emphasizes the heightened sensitivity of MP4 to even modest perturbations. Consequently, it becomes evident that MP4 represents an ill-structured problem.

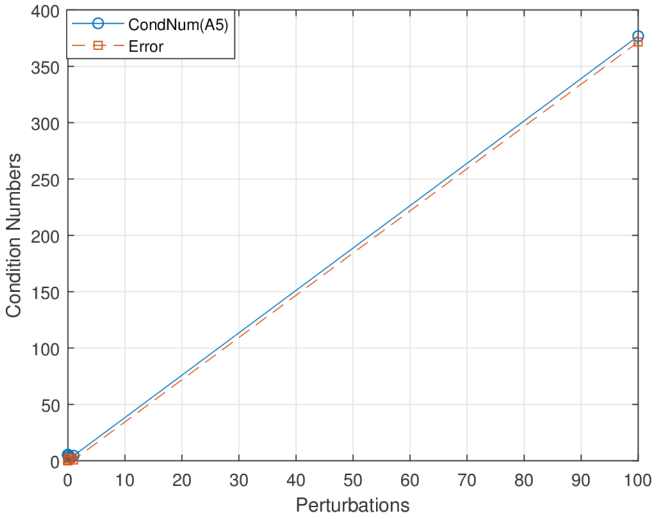

Figure 23. *Effects of perturbations for MP5.*

Referring to Figure 22, the analysis reveals a discernible pattern: the condition number of Model Problem 5 (MP5) rises with escalating perturbations on MP5 and diminishes in vice versa. The graphical representation also conveys that the magnitude difference between the original condition number and the new condition number for MP5 diminishes with the reduction of perturbations in matrix entries. The subtle gap between these two plots suggests that Model Problem 5 is a well-posed problem.

In summary, the condition number of the new matrix with a smaller perturbation ($\triangle A$) appears to be similar to the condition number of the original matrix. This similarity indicates that the system is well-perturbed, implying that small changes imposed on the original system do not significantly affect the desirable solvent. However, when larger changes are introduced to the original system, it leads to a deviation from the desirable solvent, rendering the new system ill-posed.

Furthermore, a notable observation is that the smaller the increment to the original matrix, the smaller the magnitude difference between the condition number of the original matrix and the new matrix, and vice versa. This observation leads to the understanding that if the condition number before and after perturbation remains the same, the problem is termed well-structured. Conversely, if there is a significant difference, it indicates an ill-structured problem.

Adding to this observations, it is worth noting that if the original matrix contains many zero entries, the analysis reveals that any perturbation imposed on the system will have a substantial impact on the solvent, regardless of the condition number. This underscores the sensitivity of systems with numerous zero entries to perturbations, highlighting an important aspect to consider in the context of the matrix's composition and its implications for the behavior of the system of linear equations.

4. Conclusions and Recommendations

4.1. Conclusion

Findings showed that Refined Succession Over-relaxation method is very efficient especially when $1.0 < \omega < 1.2$ where ω is constant for speeding up the convergence and reduces the number of iterations. More interestingly it was observed that trial and error method for finding ω using SOR gives appropriate solvent than the ideal parameter for relaxation which is computed using formula. The successive ω at desirable solvent with tolerance of 10^{-5} via SOR method becomes more efficient with rapidly convergence than the Refined Succession-Over Relaxation method.

In most cases trial and error method of finding best ω via the SOR method has noticed to be the best than using the ω of the RSOR that is obtained by using the specified formula. Further findings may be done to generate an optimal algorithm for the refined SOR that would yield suitable, desirable solvent with intermittent iterations to the true values. This is due to the fact that, the existing formula for finding optimal relaxation parameter (ω) fails in some cases.

Moreover, the results show that small inputs to the coefficients of the original matrix does not affect the desirable solvent because, the condition number of the original matrix is the same as that of perturbed matrix hence the problem becomes well-structured. If the increment or decrement to the coefficients of matrix exceeds 10^{-4} then the condition number of the new system will be greater than that of the original system hence the problem becomes ill-conditioned.

In a nutshell, the system with many 0's entries seem to be sensible to any perturbation imposed to the system and the new system will be unstable. If the tolerance $|k(A_i^0) - k(A_i)| > 10^{-5}$ for all $i \geq 1$ where i is any positive integer then the problem becomes badly structured.

4.2. Recommendations

The findings underscore the versatility of the SOR method, particularly in scenarios where fine-tuning the relaxation parameter is feasible, and precision in convergence is utmost importance. However, it is essential to consider that the method's efficiency may be highly dependent on the phenomena's characteristics and the specific criteria set for convergence, making it crucial to tailor the choice of the method to the unique requirements of each problem.

The use of a trial and error approach for determining the parameter for relaxation (ω) in the Successive Over Relaxation (SOR) technique is encouraged. The SOR approach was observed to yield more suitable solvents compared to obtaining ω through the prescribed formula. For specific cases where successive iterations toward a desirable solution with a tolerance of 10^{-5} are essential, the SOR approach converges rapidly compared to the RSOR method. It is advisable to assess the nature of the problem and select the method accordingly.

Care should be taken regarding the inputs to the coefficients of the original matrix. If small adjustments are made and the condition number remains consistent, the problem is considered well-structured. However, exceeding a certain threshold of adjustment can result in an ill-conditioned system, potentially leading to instability. Systems with a significant number of zero entries are particularly sensitive to perturbations. Any imposed perturbation can render the system unstable. It is advisable to exercise caution and consider the impact of perturbations on the stability of the system.

ORCID

0009-0004-4866-900X (Odeli John Kigodi)
0000-0001-8378-9219 (Chacha Stephen Chacha)
0009-0006-9436-6207 (Adeline Peter Mtunya)

Acknowledgments

We express sincere gratitude to everyone who played a significant role in completing this project. Special thanks to our advisors for their invaluable guidance in shaping our simulations and sensitivity analyses. We appreciate those who participated in discussions, provided feedback, and shared their expertise, which greatly enriched the paper. This collaborative endeavor stands as a testament to their support.

Conflicts of Interest

The authors assert that they have no conflicts of interest related to the publication of this paper.

References

[1] Abdullahi, I., & Ndanusa, A. (2020). A new modified preconditioned Accelerated Over-relaxation (AOR) iterative method for L-Matrix linear algebraic systems. *Science World Journal*, 15(2), 45-50.

[2] Abdullahi, R., & Muhammad, R. (2021). Refinement of Preconditioned Over-relaxation algorithm for solution of the linear algebraic system Ax=b. 16(3).

[3] Allahviranloo, T. (2005). Successive over relaxation iterative method for fuzzy system of linear equations. *Applied Mathematics and Computation*, 162(1), 189-196. https://doi.org/10.1016/j.amc.2003.12.085

[4] Bai, Z.-Z., Parlett, B. N., & Wang, Z.-Q. (2005). On generalized successive overrelaxation methods for augmented linear systems. *Numerische Mathematik*, 102(1), 1-38. https://doi.org/10.1007/s00211-005-0643-0

[5] Bai, Z. Z., & Chi, X. B. (2003). Asymptotically optimal successive over-relaxation methods for systems of linear equations. *Journal of Computational Mathematics*, 603-612.

[6] Carre, B. A. (1961). The determination of the optimum accelerating factor for successive over-relaxation. *The computer journal*, 4(1), 73-78.

[7] Chicone, C. (2006). *Ordinary differential equations with applications* (Vol. 34). Springer Science & Business Media.

[8] Constantinescu, R., Poenaru, R. C., Pop, F., & Popescu, P. G. (2019). A new version of KSOR method with lower number of iterations and lower spectral radius. *Soft Computing*, 23(22), 11729-11736. https://doi.org/10.1007/s00500-018-03725-2

[9] Dehghan, M., & Hajarian, M. (2011). Improving preconditioned SOR-type iterative methods for L-matrices. *International Journal for numerical methods in biomedical engineering*, 27(5), 774-784.

[10] Dihoum, B. E., Flijah, L. A. A., Al-Qiblawi, S. S., & Owen, S. A. (2022). Comparison Of Jacobi Iteration Method And Gauss-Seidel Iteration Method In Solving Fuzzy Linear Equation Systems Using A Computer. *Journal of Algebraic Statistics*, 13(2), 711-722.

[11] Emmanuel, F. S. (2015). On Some Iterative Methods for Solving Systems of Linear Equations. *Computational and Applied Mathematics*. Vol. 1, No. 2, 2015, pp. 21-28.

[12] Epelman, M., & Freund, R. M. (2000). Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88(3), 451-485.

[13] Faruk, A. I., & Ndanusa, A. (2019). Improvements of Successive Over-relaxation (SOR) methods for L-matrices. *Savanna Journal of Basic and Applied Sciences*, 1(2), pp. 218-223.

[14] Gonfa, G. (2016). Refined iterative method for solving system of linear equations. *American Journal of Computational and Applied Mathematics*, 6(3), 144-147.

[15] Hadjidimos, A. (2000). Successive overrelaxation (SOR) and related methods. *Journal of Computational and Applied Mathematics*, 123(1-2), 177-199. https://doi.org/10.1016/S0377-0427(00)00403-9

[16] Kincaid, D. R., & Young, D. M. (1972). The Modified Successive Overrelaxation with Fixed Parameters.

[17] Kiusalaas, J. (2005). Numerical Methods in engineering with Matlab.

[18] Kulsrud, H. E. (1961). A practical technique for the determination of the optimum relaxation factor of the successive over-relaxation method. *Communications of the ACM*, 4(4), 184-187. https://doi.org/10.1145/355578.366504

[19] Kyurkchiev, N., & Iliev, A. (2014). A Refinement of some Overrelaxation Algorithms for Solving a System of Linear Equations. *Serdica Journal of Computing*, 7(3), 245-256. https://doi.org/10.55630/sjc.2013.7.245-256

[20] Lanczos, C. (1952). Solution of Systems of Linear Equations by Minimized Iterations. *Journal of Research of the National Bureau of Standards*, 49, 33-53. http://dx.doi.org/10.6028/jres.049.006

[21] Lisanu Assefa, W., & Woldeselassie Teklehaymanot, A. (2021). Second Refinement of Accelerated over Relaxation Method for the Solution of Linear System. *Pure and Applied Mathematics Journal*, 10(1), 32. https://doi.org/10.11648/j.pamj.20211001.13

[22] Lotfy, H. M. S., Taha, A. A., & Youssef, I. K. (2019). Fuzzy linear systems via boundary value problem. *Soft Computing*, 23(19), 9647-9655. https://doi.org/10.1007/s00500-018-3529-7

[23] Mai, T.-Z., & Wu, L. (2013). The successive over relaxation method in multi-layer grid refinement scheme.

[24] Mayaki, Z., & Ndanusa, A. (2019). Modified successive overrelaxation (SOR) type methods for M-matrices. *Science World Journal*, 14(4), 1-5.

[25] Mayooran, T., & Light, E. (2016). Applying the Successive Over-relaxation Method to a Real World Problems. *American Journal of Applied Mathematics and Statistics*.

[26] Mehtre, V. V., & Singh, A. (2019). Jacobi and Gauss-Seidel Iterative Methods for the Solution of Systems of Linear Equations Comparison. 6(7).

[27] Muleta, H., & Gofe, G. (2018). Refinement of Generalized Accelerated Over Relaxation Method for Solving System of Linear Equations Based on the Nekrassov-Mehmke1-Method. 13(2).

[28] Ndanusa, A., & Adeboye, K. R. (2012). Preconditioned SOR iterative methods for L-matrices. *American Journal of Computational and Applied Mathematics*, 2(6): 300-305.

[29] Ndanusa, A., & Al-Mustapha, K. A. (2021). The SOR iterative method for new preconditioned linear algebraic systems. 9th (Online) *International Conference on Applied Analysis and Mathematical Modeling*.

[30] Rice, J. R. (1966). A Theory of Condition. SIAM *Journal on Numerical Analysis*, 3(2), 287-310. https://doi.org/10.1137/0703023

[31] Vatti, V. K., & Gonfa, G. G. (2011). Refinement of generalized Jacobi (RGJ) method for solving system of linear equations. *International Journal of Contemporary Mathematical Sciences*, 6(3), 109-116.

[32] Xu, M. M., Sulaiman, J., & Ali, L. H. (2021). Refinement of SOR method for the rational finite difference solution of first-order Fredholm integro-differential equations. In *AIP Conference Proceedings* (Vol. 2423, No. 1). AIP Publishing.

[33] Young, D. (1954). Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76(1), 92-111.

[34] Young, D. M. (1950). Iterative Methods for Solving Partial Difference Equations of Elliptic Type. Ph. D. Thesis. 1950. 74 p.

[35] Young, D. M. (1977). On the accelerated SSOR method for solving large linear systems. *Advances in Mathematics*, 23(3), 215-271. https://doi.org/10.1016/S0001-8708(77)80029-7

[36] Audu, K. J., Yahaya, Y. A., Adeboye, K. R., & Abubakar, U. Y. (2021). Refinement of Extended Accelerated Over-Relaxation Method for Solution of Linear Systems. *Nigeria Annals of Pure and Applied Sciences*, 4(1), 50-56. https://doi.org/10.46912/napas.226

[37] Youssef, I. K., & Taha, A. A. (2013). On the modified successive overrelaxation method. *Applied Mathematics and Computation*, 219(9), 4601-4613. https://doi.org/10.1016/j.amc.2012.10.071