AN EXTENDED PARAMETERIZED ACCELERATED OVER-RELAXATION ITERATIVE METHOD WITH REFINEMENT FOR LINEAR SYSTEMS

BY

AUDU, Khadeejah James

(PhD/SPS/2017/913)

DEPARTMENT OF MATHEMATICS,

FEDERAL UNIVERSITY OF TECHNOLOGY,

MINNA, NIGERIA.

OCTOBER, 2021

ABSTRACT

Most largely sparse linear systems encountered in applied sciences and engineering requires efficient iterative methods to obtain an approximate solution. In this thesis, an Extended Accelerated Over-relaxation (EAOR) iterative method for solving linear systems was developed by introducing a new acceleration parameter to improve convergence rate of family of Accelerated Over-relaxation (AOR) methods. The method was developed through decomposition of the coefficient matrix A by the usual splitting approach and interpolation procedure on the sub-matrices. Analysis of convergence of the method were examined on some special matrices like $L-$, $M-$, and Irreducible weak diagonally dominant matrices. In addition, the Refinement of Extended Accelerated Over-relaxation (REAOR) iterative method was also developed coupled with its convergence properties for the special matrices. To validate the effectiveness of the proposed methods, some numerical tests including problems from fuzzy linear systems and heat transfer were conducted to verify the theoretical results. The results obtained indicated that the spectral radii of the proposed methods are smaller than the compared methods reviewed in the work. Based on the spectral radii results and the convergence results produced, it was concluded that the developed methods converge with lower number of iterations and computational time than the compared methods. This reveals that the introduction of a parameter to the general two-parameter AOR family methods has improved the convergence results. From the results obtained, the EAOR iterative method converges approximately 1.4 times faster than the KAOR iterative method and 1.8 times faster than the Quasi Accelerated Over-Relaxation (QAOR) iterative method. While the REAOR method converges 1.2 times faster than the Refinement of Accelerated Over-relaxation (RAOR) iterative method.

TABLE OF CONTENTS

Content **Page**

LIST OF FIGURES

Figures **Pages**

LIST OF TABLES

Tables **Page**

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

The problem of solving the linear (mostly sparse) systems

$$Az = b \tag{1.1}$$

Where A is the coefficient matrix, b is the right hand side vector, and z is vector of unknowns, appears as a final stage in solving many problems in different areas of science and engineering. It is the result of discretization techniques of the mathematical models representing realistic problems. In most cases, the number of these equations is generally large and for this reason, their solution is a major problem itself. Accordingly, techniques employed for their solutions are essentially the direct methods or that of the iterative methods (indirect methods). If the coefficient matrix of the linear system is large and sparse (most of the elements are zero), then iterative methods are recommended against the direct methods (Youssef & Farid, 2015). Iterative methods are more attractive since they are very effective, requires less memory and arithmetic operations (Fiseha, 2020).

Linear systems $(Az = b)$ are among the most important and common problem encountered in scientific computing. The existence of solution of a linear system is distinguished into three situations. From theoretical point of view, it is well understood when a solution exists, when it does not and when there are infinitely many solutions. In addition, explicit expression of the solution using determinants exists. However, from the numerical point of view, it is far more complex. Approximations may be available but it may be difficult to estimate how accurate they are. This clearly will depend on the data at hand, primarily on the coefficient matrix (Saad, 2003).

Undoubtedly, Partial Differential Equations (PDE) constitute the major source of linear systems or sparse matrix problems. One possible way to obtain solutions to such equations is to discretize them that is by approximating them with finite number of unknowns. Several different ways of discretizing a PDE is through some discretization procedures such as finite difference methods, finite element methods and finite volume method (Kiusalass, 2005).

According to Behzadi (2019), a general iterative method involves a process that converts the system of linear equations (1.1), by splitting A into $M - N$ and the matrix splitting M is required to be easily invertible such that

$$z^{(k+1)} = M^{-1}Nz^{(k)} + M^{-1}b \ , \qquad (k = 0, 1 ,2, 3, \cdots, n) \tag{1.2}$$

This can be equivalently written as a system of the form

$$z^{(k+1)} = Jz^{(k)} + f \tag{1.3}$$

Where $J = M^{-1}N,$ $f = M^{-1}b,$ for some matrix J and vector $f.$ After an arbitrary initial guess $z^{(0)}$ is selected, the sequence of approximate solution vectors is generated by the iteration and the sequence $(z^{(k)}: k = 0,1 ,2, \cdots, n)$ is required to converge to z^* where z^* is the solution of $Az = b$. In order to solve the linear system in equation (1.1) more effectively by using the iterative methods, usually, efficient splitting of the coefficient matrix A is required.

Large and sparse linear systems often occur in scientific or engineering application when finding solutions to partial differential equations. Sparse linear solvers are mainly the iterative methods such as Jacobi, Gauss Seidel, Successive Over-Relaxation and Accelerated Over-Relaxation methods, this is due to their fast computations of matrix splitting techniques. The technique of iterative method in obtaining solution for linear systems involves one in which an initial estimation is utilized in computing a second estimation, the second estimation is equally used to compute a third estimation and it goes on continuously until a desired result is achieved (Fadugba, 2015).

As the difference between the exact solution and successive estimation tends to zero, the iteration procedure becomes convergent. The most important aspect of an iterative method is that the set of iterates from the iteration should converge fast to a desired accuracy (Anton & Rorres, 2013).

When designing an iterative method for solving systems of linear equations, the major question is how to achieve rapid convergence, or to put this in other words, how to construct an iteration matrix with the smallest spectral radius possible. To do this, one should be able to exert control over the properties of iterative matrix A, and this can be realized by the use of a special procedure called the method of relaxation. A technique that involves the process of speeding up the convergence rate of virtually any iterative method is known as relaxation method. This method tends to converge under general conditions although it usually progresses slowly than competing methods, which implies that its major setback is that of slow convergence. For example, assuming we have an initial estimate say $z^{\text{INITIAL}}(\boldsymbol{z}^{\mathbf{I}})$, of a quantity and we desire to advance towards a target estimate say z^{TARGET} $(\boldsymbol{z}^{\mathbf{T}})$ by a particular method. Let the application of the particular method change the estimate from z^{INITIAL} to $z^{\text{APPROXIMATE}}(\boldsymbol{z}^{\mathbf{A}})$. If $\boldsymbol{z}^{\mathbf{A}}$ is in the middle of $\boldsymbol{z}^{\mathbf{I}}$ and $z^{\text{RELAXED}}(\boldsymbol{z}^{\mathbf{R}})$, which is nearer to $\boldsymbol{z}^{\mathbf{T}}$ than $\boldsymbol{z}^{\mathbf{A}}$, then we can advance towards $\boldsymbol{z}^{\mathbf{T}}$ faster by magnifying the change $(\boldsymbol{z}^{\mathbf{A}} - \boldsymbol{z}^{\mathbf{I}})$. In order to achieve the above aim, we ought to apply a magnifying factor $\omega > 1$ to obtain

$$z^{\text{R}} - z^{\text{I}} = \omega(z^{\text{A}} - z^{\text{I}}) \tag{1.4}$$

$$z^{\text{R}} - z^{\text{I}} = \omega z^{\text{A}} - \omega z^{\text{I}} \tag{1.5}$$

$$z^{\text{R}} = \omega z^{\text{A}} - \omega z^{\text{I}} + z^{\text{I}} \tag{1.6}$$

$$z^{\text{R}} = \omega z^{\text{A}} + (1 - \omega) z^{\text{I}} \tag{1.7}$$

This amplification process is an extrapolation and it is typically an example of *Over-relaxation*. Over-relaxation method is seen as a process of over estimating the residual/error by a factor towards the true solution of the problem in a fast manner.

Suppose the midway estimate (z^{R}) tends to exceed or surpass the target estimate (z^{T}), then one will have to apply $\omega < 1$ which is known as *Under-relaxation*. Examples of Over-relaxation methods are the methods of Successive Over-relaxation (SOR) and Symmetric Successive Over-relaxation (SSOR).

1.1.1 Sparse matrix

A sparse matrix is a matrix in which majority of its coefficients are mostly zero values. Such matrices are different from matrices that contains mostly non-zero coefficients which are termed dense matrices. These matrices are generally common and occur mostly in areas such as machine language, data representation and others. Additionally, sparse matrices are quite expensive computationally to work with compared to dense matrices. One possible way to improve their performance is through the use of representations and operations that can handle sparsity of the matrices specifically. The main interest in sparsity of a matrix stems from the fact that exploitation of such matrices leads to great computational savings and practically, various large matrix problems that occur are sparse (Duff *et al.*, 2017).

Matrix sparsity are measured by a certain computed score, this score is represented as

$$\text{Sparsity of a matrix} = \frac{\text{Number of zero values in matrix}}{\text{Total Number of entries in matrix}}$$

For example the matrix below

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 0 & 0 & 5 & 0 \\ 0 & 0 & 4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix} \tag{1.8}$$

Contains 26 zero values out of the 35 entries in the matrix, thereby giving a sparsity of 0.7428571429 or approximately 74%. A lot of memory is required by numerous large matrices. More so, representing the zero values in a 32-bit or 64-bit is a clear waste of memory because those values (zero) do not contain any vital information. It is quite wasteful to utilize some general methods on sparsity problems due to the fact that most mathematical operations of $O(n^3)$ designed to compute the linear equations constitute zero operands. This leads to an associated problem of increased in time complexity of matrix operations that increases with the size of the matrix. There are several efficient ways in storing and working with sparse matrix, although, iterative methods provides an excellent implementation that one can utilize directly with respect to sparse matrices (Strang, 2016). The survey of iterative methods will be carried out in the next chapter.

It is an undisputable fact that in real world, time is of essence that is no one wants to waste time. In regards to the solution of linear system of equations, it can sometimes be more desirable to get a close approximation of the solutions than to get the exact solution, when time is taken into consideration, this is where the proposed Extended Accelerated Over Relaxation (EAOR) method comes in to play. The Gaussian elimination method which is an exact solution techniques requires approximately $\frac{n^3}{3}$ operations to solve the system, which becomes time-consuming when n gets big. The proposed EAOR iterative method on the other hand, even though it only produces an approximation, can give us these approximations much faster than Gaussian elimination.

1.2 Statement of the Research Problem

Many physical problems in science and engineering are modelled into differential equations. The discretization of these equations in most cases results into a system of linear equations. These linear equations are usually large and sparse which necessitates the application of iterative solution method such as Jacobi and Gauss-Seidel methods. A basic requirement of such iterative method is convergence, it is not just enough for a method to converge, we are equally interested in how fast a method converges. This goes a long way in saving storage, time and reducing cost. The search for automation and increasing the efficiency of iterative methods led to the discovery of Successive Over Relaxation (SOR) method. The Accelerated Over Relaxation (AOR) iterative method is a two-parameter modification of the SOR method that results into better convergence and greater efficiency in certain cases. Several researchers have also worked on modifications of the AOR method including generalizations, extrapolation, block and refinement form. Yet, most of these methods fail for some kind of matrices, and even with very high number of iterations before convergence could be achieved.

This present work is a further attempt at developing an iterative method that would be effective and efficient, towards solving real life problems that are modelled as system of linear equations, which will in turn help to reduce the iteration number, computational time and storage capacity.

1.3 Aim and Objectives of the Study

The aim of this research work is to construct a parameterized iterative method and a Refinement version of it for solving linear systems, especially those arising from discretization of partial differential equations.

The objectives are to:

I. develop an Extended Accelerated Over Relaxation (EAOR) iterative method.
II. investigate the convergence of the proposed Extended Accelerated Over Relaxation method.

III. determine the conditions placed on the coefficient matrix with regards to the proposed method.

IV. develop a Refinement version of the Extended Accelerated Over Relaxation method.

V. investigate the convergence of the proposed Refinement of Extended Accelerated Over-Relaxation method.

VI. undertake some numerical experiments including fuzzy linear system and a real life problem for the purpose of evaluating and validating the new methods.

1.4 Justification of the Study

The proposed iterative method is required for usage in areas like computational fluid dynamics, oil and gas industry, machine learning, structural engineering, linear elasticity and others.

Given that real life problems are usually transformed into mathematical equations or linear algebraic equations, therefore finding solutions of such equations becomes paramount for researchers in the quest to obtain solutions to real world problems. Solving large systems of linear equations cannot be handled by direct methods, especially in a case where the matrix of the system is sparse, thereby requiring an iterative method for obtaining its solution.

Application of a speedy converging iterative method with regards to solution of linear systems would save computational time and considerable resources. Hence the desire to develop a speedy converging iterative method that would solve large linear systems efficiently. The basic idea behind constructing the proposed method is mainly to speed up convergence rate.

1.5 Significance of the Study

The results and conclusion of this study are important, because iterative methods have important relevance in real world applications in the fields of computational fluid dynamics, mathematical programming, linear elasticity, machine learning, among many others. In dynamics, for example, the study of heat conduction, turbulent flows, boundary layer flows or chemically reacting flows are some of the application areas where the proposed Extended Accelerated Over Relaxation iterative method is important for both researchers and policy makers. In essence, real life problems encountered in areas of science and engineering would be simplified.

1.6 Scope and Limitation of the Study

In a quest to improve the convergence speed of parameterized stationary iterative methods, the need to introduce an efficient iterative method becomes pertinent, this research study strives to do just that. A new parameter is introduced into the family of general Accelerated methods to formulate the proposed iterative method in order to improve convergence rate. Certain restrictions placed on the coefficient matrix of the linear system $Az = b$ are derived, analyzed and discussed extensively. The study will also advance convergence theorems and establish their proofs. More so, it covers the development of the Refinement version of the proposed Extended Accelerated Over Relaxation (REAOR) method. Irreducible weak diagonally dominant, L- and M-matrices are investigated. This study is limited to Extended Accelerated Over Relaxation (EAOR) method for solving linear systems of the form $Az = b$, where A is a square non-singular coefficient matrix, b is a column vector of constants and z is the solution vector to be determined.

1.7 Definition of Terms

Basic iterative method: Is a single-step method of the form $z^{(k+1)} = Jz^{(k)} + f$ for some invertible matrix P, where $J = I - P^{-1}A$ and $f = P^{-1}b$ which involves obtaining successive approximate solutions from an initial estimation to the true solution of a linear system $Az = b$.

Lower triangular matrix: Is a square matrix in which all the entries above the diagonal entries are zero that is $a_{ij} = 0$ whenever $i < j$.

Upper triangular matrix: Is a square matrix in which all the entries below the diagonal entries are zero that is $a_{ij} = 0$ whenever $i > j$.

Triangular matrix: A square matrix which is either upper triangular or lower triangular matrix is called a triangular matrix.

Strictly lower triangular matrix: Is a lower triangular matrix in which all its main diagonal elements are zero that is $a_{ij} = 0$ whenever $i \leq j$

Strictly upper triangular matrix: Is a upper triangular matrix in which all its main diagonal elements are zero that is $a_{ij} = 0$ whenever $i \geq j$

Nonsingular matrix: A square matrix $(A_{ij})_{m\times m}$ is said to be nonsingular if its determinant is not equals to zero, that is to say det $A \neq 0$.

Diagonally dominant (weak) matrix; A square matrix A is said to be diagonally dominant or weak diagonally dominant if and only if

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, \qquad i = 1, 2, \cdots, n \tag{1.9}$$

Irreducibly diagonally dominant matrix; A square matrix A is said to be irreducibly diagonally dominant if matrix A is irreducible and satisfy the condition

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, \qquad i = 1, 2, \cdots, n \tag{1.11}$$

With strict inequality for at least one row.

Directed graph: Let matrix $A = [a_{ij}]$ with any m distinct points $T_1, T_2, \cdots, T_m$ in the plane called nodes, then for every nonzero entries a_{ij} of the specific matrix, the set of connected nodes T_i to T_j through a directed path $\overrightarrow{T_i\,T_j}$ is called a directed graph denoted as $G(A)$.

Strongly connected graph: A directed graph of square matrix $A = [a_{ij}]$ is said to be strongly connected if it possess a directed path from T_i to T_j and a directed path from T_j to T_i for every pair of the nodes (i, j). For illustration, the directed graph of $A_1 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$ shown in Figure 1.1 is strongly connected since it is possible to reach any of the points starting from a specific point.

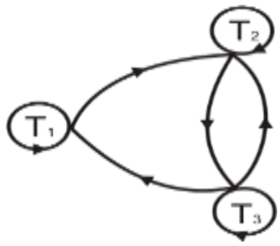

Figure 1.1: Directed graph of A_1

Similarly, the directed graph of matrix $A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 2 & 3 \\ 5 & 6 & 7 \end{pmatrix}$ shown in Figure 1.2 is not strongly connected since it is not possible to reach T_2 or T_3 from T_1.

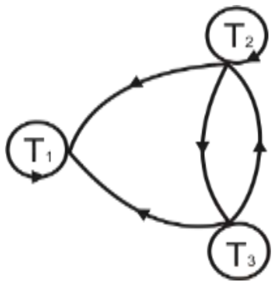

Figure 1.2: Directed graph of A_2

Irreducible matrix: A matrix $A = [a_{ij}]$ is irreducible if and only if its directed graph $G(A)$ is strongly connected. For example, from figures 1.1 and 1.2, $G(A_1)$ is irreducible because its directed graph is strongly connected and $G(A_2)$ is not irreducible since its directed graph is not strongly connected.

Refinement iterative method: A Refinement iterative method is an iteration technique for improving the estimate solutions to the true solution of the linear system $Az = b$. It involves the process of calculating the residual $r = b - Az$, solving the specified iterative method $z = Jz + f$, forming the updates $\bar{z} = z + Y(b - Az)$ for some invertible matrix Y and repeating these steps as necessary until accuracy is achieved.

Sparse matrix: Is a matrix in which most of its elements are zero with few non-zero elements. A special type of sparse matrix is the band matrix.

Nonnegative matrix: A matrix $A = [a_{ij}]$ where $a_{ij} \geq 0$ $(i, j = 1,2, \cdots, n)$

Hermitian matrix: A hermitian matrix is a complex square matrix that is equal to its own conjugate transpose. If the conjugate transpose of a given matrix A is denoted as A^H, then the hermitian property is expressed as $A = A^H$.

***L*-matrix:** Is a square matrix $A = [a_{ij}] \in R^{nxn}$ where $a_{ij} \leq 0$ $(i \neq j)$ with $a_{ii} > 0$, $i = 1,2, \cdots, n$.

***M*-matrix:** Is an L-matrix $A = [a_{ij}] \in R^{nxn}$ where A is nonsingular and $A^{-1} \geq 0$.

***H*-matrix:** A matrix $A = [a_{ij}]$ is said to be an H-matrix if its comparison matrix, that is, the matrix $\langle A \rangle$ with elements $\alpha_{ii} = |a_{ii}|$, $i = 1, 2, \cdots, n$ and $\alpha_{ii} = -|a_{ii}|$, $i \neq j$ is an M-matrix.

Property *A*: An $m\ x\ m$ matrix $A = [a_{ij}]$ is said to possess property A if there exists a set W from matrix A containing the union of two disjoint subsets V and U, such that if either $a_{j.i} \neq 0$ or $a_{ij} \neq 0$, then $j \in U$ and $i \in V$ or $i \in U$ and $j \in V$.

Consistently ordered matrix: A square matrix $A = [a_{ij}]$ is considered a consistently ordered matrix if it is obtained from permutation of columns and corresponding rows of a given matrix provided the given matrix possess Property A.

Splitting: The decomposition of any given matrix A into the form $A = M - N$ where M is a non singular matrix is called a splitting of A. And such splitting is:

I. Nonnegative if $M^{-1} \geq 0$

II. Regular if $M^{-1} \geq 0$ and $N \geq 0$

III. Convergent if $\rho(M^{-1}N) < 1$

Usual splitting: For any matrix A, the splitting $A = D_A - L_A - U_A$ where U_A is the strict upper component of A, L_A is the strict lower component of A and D_A is the diagonal part of A, is referred to as the usual splitting of A. Also, if matrix A is assumed to have a non-vanishing diagonal elements, then the usual splitting becomes $A = I - L - U$ where $U = {D_A}^{-1}U_A, L = {D_A}^{-1}L_A$ and $I = {D_A}^{-1}D_A$.

Spectral radius: is the greatest value among the absolute values of the eigenvalues λ_k of a square matrix, denoted as $\rho[A] = \max_{\lambda_k \in A}|\lambda_k|$.

Fuzzy Number: Fuzzy number is a generalization of a regular, real number in the sense that it does not refer to one single value but rather to a connected set of possible values, where each possible value has its own weight between 0 and 1. It is an ordered pair $(\underline{u}, \overline{u})$ on the functions $\underline{u}(\alpha), \overline{u}(\alpha)$, $0 \leq \alpha \leq 1$ that satisfies the requirement;

I. $\underline{u}(\alpha)$ is a bounded non-decreasing function over $[0,1]$

II. $\overline{u}(\alpha)$ is a bounded non-increasing function over $[0,1]$

III. $\underline{u}(\alpha) \leq \overline{u}(\alpha)$, $0 \leq \alpha \leq 1$

Fuzzy Linear System: Fuzzy linear system are systems of linear equations in which coefficients and variables are uncertain and this uncertainty is expressed using fuzzy numbers. Fuzzy linear system are used in practical situation where some of the system's parameters or variables are uncertain.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Basic Iterative Methods

Numerical methods in approximating solutions of linear system allow the possibility of obtaining values of the roots system with the desired accuracy. This procedure of developing such sequences is referred to as iteration. While the direct method tries to compute the true solution in a finite number of steps, iterative method begins with an initial guess and produces successive improved estimations in an infinite sequence whose limit is the true solution. Practically, the iterative technique has more advantage due to the fact that the direct solution is subjected to rounding errors (Karunanithi *et al*., 2018).

As discussed in chapter one, a general linear iterative scheme of the form $z = Jz + f$ is termed stationary if J and f are not dependent on the iteration count z. This implies that at every step of the iteration, J and f remain constant. Such stationary iterative methods are the Jacobi, Gauss-Seidel, Successive Over-Relaxation methods and others.

Jacobi method and the Gauss-Seidel method are well known classical iterative methods introduced in the late eighteenth century for solving linear system. Solution of linear system of small dimension usually do not require an iterative technique, simply because time needed for necessary accuracy exceeds that of the direct methods. Large linear system with high percentage of zero elements which is usually obtained when solving partial differential equations and boundary value problems, usually requires an iterative method for their solutions. An iterative procedure for solving $m\,x\,m$ linear system $Az = b$, begins with an initial guess $z^{(0)}$ to the solution z and produces successive estimations of vectors $\left[z^{(k)}\right]_{k=0}^{\infty}$ which converge to z (Kisabo *et al.*, 2016). A very basic idea that leads to effective iterative solvers is to split the matrix of a given linear system in the sum of two or three matrices that would lead to a system that can easily be solved.

For example, the classical Jacobi and Gauss-Seidel iterations are obtained by splitting the matrix A into its diagonal and off diagonal parts $A = D - L - U$ and the coefficient matrix A can be further transformed into $D^{-1}A = I - L - U$, where I is the unit matrix of order N, $-L$ is the strictly lower triangular and $-U$ is the upper triangular parts of A respectively.

2.1.1 The Jacobi method

As discussed by Markatos and Karabeks (2015), the Jacobi method is an iterative method that cycles through each of the variables $z^1, z^2, z^3, \cdots, z^k$ in turn to refine an initial guess. The main idea of the Jacobi method is to determine the kth variable of the next approximate solution in z^{k+1} in relation to the other variables. The iteration method involves solving one variable at once for a single step of the iteration process or simply, at each iteration stage. That is to say, we use the values of z^k to update the z^{k+1} at each stage for each iteration. The matrix form of the Jacobi method is formulated from the linear system in equation (1.1) based on the splitting of $A = D - U - L$, and its matrix form is given as:

$$z^{(k)} = D^{-1}(U + L)z^{(k-1)} + D^{-1}b \qquad k = 1, 2, \cdots, N \tag{2.1}$$

And the general iterative form of (2.1) is denoted as

$$z^{(k+1)} = Jz^{(k)} + V \tag{2.2}$$

Where $J = D^{-1}(U + L)$ is the Jacobi iterative matrix and $V = D^{-1}b$. The algebraic form of the Jacobi method above is expressed as;

$$z_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_{\ i} - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} z_i^{(k)}\right) \qquad i = 1\,, \cdots, n \quad k = 0, 1, 2, \cdots, N \tag{2.3}$$

Where the z_i are the elements of z, b_i are the elements of b and the a_{ij} are the elements of the coefficient matrix $A = (a_{ij})$ respectively.

Salkuyeh (2007) in an attempt to enhance the rate of convergence of the Jacobi method proposed the generalized Jacobi (GJ) method and states that the generalized Jacobi method is convergent for M −matrix, strictly diagonally dominant matrix and symmetric positive definite matrix. It was revealed that the generalized Jacobi method performed better than the conventional Jacobi method based on the outcome that the Jacobi method took a longer time to converge to the true solution than the generalized Jacobi method.

Tesfaye (2016) introduced the second degree generalized Jacobi iteration method. The convergence rate properties and the spectral radius of the method were studied and discussed. The method was validated and it proves that the method converges faster than first degree Jacobi, generalized first degree Jacobi and second degree Jacobi methods. Also, the method can be further improved by the application of extrapolating procedures.

2.1.2 The Gauss-Seidel method

The Gauss-Seidel iterative method is a modification of the Jacobi method and essentially superior to the Jacobi method. This iterative scheme is also known as successive displacement method which is based on the process of updating the kth iterative values as soon as the new estimates are available. Regarding the method of Jacobi, the estimates of z_i^k obtained in the kth iteration remains unchanged until the entire $(k+1)th$ iteration has been computed. With regards to the Gauss-Seidel iterative method, we utilize the new estimates z_i^{k+1} as soon as they are known, (Saha and Chakrabarty, 2020). For illustration, once we have calculated z_1^{k+1} from the first equation, its estimate is then utilized in the second equation of the linear system to obtain the new estimate z_2^{k+1}. The algebraic form of the method is given as

$$z_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} z_j^k - \sum_{j=i+1}^{n} c_{ij} z_j^k\right) \qquad \begin{matrix} i = 1, 2, \cdots, n \\ k = 0, 1, 2, \cdots, N \end{matrix} \tag{2.5}$$

while the matrix form formulated from splitting of (1.1) into $A = D - U - L$ is

$$z^{k+1} = [(D-L)^{-1}U]z^k + (D-L)^{-1}b \tag{2.6}$$

This is represented in the general iterative form as

$$z^{k+1} = J_{GS} z^k + V_{GS} \tag{2.7}$$

where $J_{GS} = U(D-L)^{-1}$ is the Gauss-Seidel iteration matrix and $V_{GS} = (D-L)^{-1}b$.

In a quest to enhance the rate of convergence of the Gauss-Seidel method, Salkuyeh (2007) presented a modified form of the Gauss-Seidel method called generalized Gauss-Seidel (GGS) method. The convergence properties of the GGS for $M-$matrix, strictly diagonally dominant matrix and symmetric positive definite matrix were studied. Analysis of the results indicates that the generalized Gauss-Seidel method is superior to the classical Gauss-Seidel method.

On the other hand, Tesfaye (2014) presented a method called second degree generalized Gauss-Seidel method (SDGGS) and studied the convergence of the method for symmetric positive definite matrix, strictly diagonally dominant matrix and irreducible matrix. It was revealed that the second degree generalized Gauss-Seidel method in comparison with methods of first degree Gauss-Seidel and generalized Gauss-Seidel performs better, due to the fact that the spectral radius of the second degree generalized Gauss-Seidel method is lesser than those of the methods examined in the work.

Sebro (2018) improved on the refinement of Generalized Gauss-Seidel iterative scheme through extrapolation procedures and developed the scheme called Extrapolated refinement of Generalized Gauss-Seidel (ERGGS) method. The method was validated and compared with methods of Refinement of generalized Jacobi, generalized Gauss-Seidel and refined version of generalized Gauss-Seidel. It was shown that the convergence rate of the Extraolated refinement of Generalized Gauss-Seidel (ERGGS) method is higher than other methods considered for comparison.

2.2 Development of Successive Over-relaxation (SOR) Method

Over the years, iterative methods of solving large sparse linear systems have been introduced starting from the Jacobi to Gauss-Seidel methods which are non-parameterized. In an attempt to rectify the setback (low convergence rate) associated with non-parameterized iterative methods, some researchers came up with the idea of developing parameterized methods so as to achieve greater convergence rate. Examples of such parameterized methods are the Successive Over-Relaxation and Accelerated Over-Relaxation which has proved to outperform the existing Jacobi and Gauss Seidel methods.

As discussed in Hadjidimos (2000), the method of Successive Over-relaxation (SOR) was invented with the aim of solving linear systems on digital computer (computationally). This method essentially seeks to reduce the number of iterations needed to minimize the error of an initial guess of the solution through a predetermine factor and application of extrapolation on the Gauss-Seidel method. The main idea of this method is taking an average weight of the previous iterates and the new computed iterates for each component successively. Setting $\bar{z}_j^k$ as the Gauss-Seidel kth component, the iteration of the SOR method is given by the relation below

$$z_j^k = \omega z_j^k + (1-\omega) z_j^{k-1} \tag{2.9}$$

where the variable $\omega[0 < \omega < 2]$ is the relaxation parameter. This implies that the accepted value at step k is extrapolated from the Gauss-Seidel value and the previous computed values. The concept is to select a suitable value for ω that will improve the convergence rate of the Gauss-Seidel method to the true solution. The SOR method reduces to that of Gauss-Seidel when $\omega = 1$, the algebraic form of the SOR is given below as

$$z_i^k = \omega \left\{ \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} z^k - \sum_{j=1+1}^{n} a_{ij} z_i^{k-1} \right) \right\} + (1-\omega) z_i^{k-1}$$

$$k = 1, 2, 3, \cdots, N \tag{2.10}$$

while the matrix form is represented as

$$z^k = (D - \omega L)^{-1} [(1-\omega) D + \omega U] z^{k-1} + (D - \omega L)^{-1} \omega b \tag{2.11}$$

and the general iterative form is denoted as

$$z^k = J_{SOR} z^{k-1} + C_{SOR} \tag{2.12}$$

where $J_{SOR} = (D - \omega L)^{-1} [(1-\omega) D + \omega U]$ is the iteration matrix of the SOR method and $C_{SOR} = (D - \omega L)^{-1} \omega b$. It is well known that SOR iterative method are convergent for linear systems.

Mayooran and Elliot (2016) discussed and analysed the significance of the Successive Over-relaxation iterative method for improving solutions concerning real world problems. They examined the performance of the classical SOR scheme by solving an heat equation when a steady boundary temperature is been applied to a flat plate, through finite difference approach. The result indicates a remarkable convergence rate due to the fact that the SOR method is an effective iterative solver in the case of large sparse matrices. Future work was recommended to construct an AOR algorithm that could generate better and closer result to the true values.

2.3 Variants of Successive Over-Relaxation Method

Many researchers have shown interest in magnifying the convergence rate of the SOR method by developing several modifications and versions of the SOR iterative method used for computations of the linear system $(Az = b)$.

The Symmetric Successive Over-Relaxation (SSOR) method combines two SOR processes together in such a manner that the resulting matrix becomes a symmetric matrix. It is an iteration process where one of the two iterations is that of the forward SOR and the other iteration is that of the backward SOR. The forward SOR iteration is given as

$$z^{k-\frac{1}{2}} = (D - \omega L)^{-1}[(1-\omega)D + \omega U]z^{k-1} + (D - \omega L)^{-1}\omega b \qquad (2.13)$$

while the backward SOR is given as

$$z^{k} = (D - \omega U)^{-1}[(1-\omega)D + \omega L]z^{k-\frac{1}{2}} + (D - \omega U)^{-1}\omega b \qquad (2.14)$$

Hence combining equations (2.13) and equation (2.14), gives rise to the SSOR iterative matrix form as

$$z^{k+1} = J_{SSOR}z^{k} + F_{SSOR} \qquad (2.15)$$

where we have $F = \omega(2-\omega)(D-\omega U)^{-1}D(D-\omega L)^{-1}b$ and $J_{SSOR} = (D-\omega U)^{-1}[\omega L + (1-\omega)D] \times (D-\omega L)^{-1}[\omega U + (1-\omega)D]z^k$ is the SSOR iteration matrix.

Youssef (2012) developed a version of the SOR method called modified Successive Over-Relaxation (KSOR) method. The method is based on the treatment of assumption that recent value can be utilized in the evaluation along with the most current computed values as in SOR method. Apart from the range of values of SOR method, the KSOR method is capable of taking values from $\omega \in [-2,0]$. It also possess the same structure as the SOR method and advantage of the method over some iterative methods is that of updating the first computation from the first step, thereby reflecting rapid convergence from the start of the iteration process. Also, consistency, spectral radius, theoretical conditions and convergence of the KSOR method were proved in the research work. The spectral radius of the method was examined and the outcome reveals that KSOR spectral radius appears comparable to SOR spectral radius for a certain value of the relaxation parameter (ω) which corresponds to optimal (ω).

Yousef and Taha (2013) presented some modifications of the KSOR method in three different forms called MKSOR, MKSOR1 and MKSOR2 is a subclass of consistently ordered matrix. The three schemes are improvements on the SOR and modified Successive Over-Relaxation (KSOR) methods, through the process of updating the residue simultaneously with the solution and utilizing the most current computed solution at the same time. They established the functional relationship between eigenvalues of Jacobi method and those of the MKSOR methods with restrictions on the relaxation parameters. Theoretical properties, consistency and convergence of the methods were proven. Validation of the methods were performed and compared with the MSOR method where it reveals that the MKSOR methods are more efficient. The outcome of their research agrees with the theoretical findings, leading to the suggestion that acceleration procedures combined with SOR and KSOR formulas can be more efficient, however, determination of optimal values for the relaxation parameters was not considered in their work.

Zhang *et al*. (2016a) extended the convergence of SOR method for non-Hermitian positive definite matrices in their paper. Some sufficient conditions for convergence as regards the SOR method were presented and they discovered that these conditions appear theoretically relevant but difficult in application to practical calculations. Numerical samples were verified to ascertain the convergence of the SOR method for non-Hermitian positive definite matrices, the results analysis indicates that the SOR iterative scheme converges for a non-Hermitian positive definite matrix.

Zhang *et al.* (2016b) in an attempt to verify if the SOR method is convergent for system of linear equations whose coefficient matric is a weak H $-$matrix, presented a convergence analysis of the SOR method for linear system with weak $H-$matrix in their work. They surveyed the convergence analysis of forward Successive Over-Relaxation [FSOR] method, backward Successive Over-Relaxation [BSOR] method and symmetric Successive Over-Relaxation [SSOR] method for weak $H-$matrix [whose comparison matrix is a singular M-matrix] and proposed some sufficient conditions for the methods to converge to the real solution. Evaluation and validation of the convergence of SOR methods for weak $H-$ matrices were carried out through some numerical samples and the results obtained reveals that the FSOR, BSOR and SSOR iterative methods are convergent for weak $H-$matrices having singular comparison matrices. They suggested the idea of investigation on the convergence of Accelerated Over-Relaxation method for linear system with weak $H-$matrices.

Chunping (2017) presented a generalization of the SSOR method with 3 parameters called 3 SSOR-like iteration method for solution to saddle point problems. Convergence of the method was discussed extensively. Numerical validation conducted confirms the theoretical proofs and effectiveness of the method. Although, the 3 SSOR- like parameters are not optimal and therefore further study on determination of optimal values was suggested.

Vatti *et al*. (2020a) studied the Second Degree Successive Over-Relaxation (SDSOR) Method and compared its performance with other methods such as Jacobi, Second Degree (SDJ), Gauss-Seidel, Second Degree Gauss-Seidel (SDGS) and SOR methods. It was observed that the SDSOR iterative method exhibits higher rate of convergence.

Firew *et al*. (2020) improved on the convergence rate of the Generalized Successive Over Relaxation (GSOR) method and developed the Second Degree Generalized SOR method. They discussed the convergence properties and compared the method with SOR and GSOR methods. It was reported that the SDGSOR iterative method converges faster than Successive Over Relaxation and Generalized Successive Over Relaxation methods.

2.4 Development of Accelerated Over-Relaxation (AOR) Method

The Accelerated Over-relaxation (AOR) iterative method which has been proven to be a powerful technique for solving linear systems of equation was developed by Apostolos Hadjidimos in 1978. The Accelerated Over-relaxation method which is viewed as an extrapolation of the Successive Over-relaxation method, having *over-relaxation parameter* (r) and extrapolated parameter $\left(\mathrm{s} = \frac{\omega}{\mathrm{r}}\right)$, *was* derived through the interpolation procedure with respect to the sub-matrices in application of general linear stationary schemes. It is an improvement on the Successive Over-relaxation iterative method which involves two parameters generalization of the Successive Over-relaxation scheme to accelerate the convergence of the Successive Over-relaxation method. The presence of two parameters used in speeding up the convergence of the AOR method instead of the usual one parameter in speeding up convergence of an iterative method proves the powerfulness of AOR method when compared with conventional method such as Successive Over-relaxation method. More so, exploiting the presence of these two parameters provides numerical solvers with a method that converges faster than any other equivalent method.

The AOR method is considered as *an improvement on the Successive Over-relaxation (SOR) method for the sole purpose of speeding up the convergence rate of the SOR method. He utilized the splitting* $A = D - U - L$ *to established the below splitting method with an over-relaxation and acceleration parameters* r *and* ω *of the coefficient matrix* A *of the linear system* $Az = b$.

$$A = \frac{1}{\omega}\left[\, M_{\omega,r} - N_{\omega,r} \,\right] \tag{2.17}$$

where $M_{\omega,r} = \frac{1}{\omega}(D - rL)$ and $N_{\omega,r} = \frac{1}{\omega}[(1-\omega)D + (\omega - r)L + \omega U\,]$ and the AOR method is governed by the relation

$$z^{(k+1)} = J_{\omega,r} z^{(k)} + S_{\omega,r} \tag{2.18}$$

where $J_{\omega,r} = (D - rL)^{-1}[(1-\omega)D + (\omega - r)L + \omega U\,]$ and $S_{\omega,r} = (D - rL)^{-1}\omega b$ By so doing, he introduced one more parameter into the Successive Over-relaxation method to obtain a faster convergence rate in the AOR method. He also made an effort to investigate the constraints or limitations enforced on the accelerated and relaxation (r and ω) under different notions on the original matrix A to enable the convergence of the AOR method. He obtained some convergence theorems based on the assumptions with regards to the original matrix A and equally established the proofs of the theorems in his work. His theorem on an irreducible weak diagonal matrix states that: the AOR method tends to converge for values of $0 \leq r \leq 1$ and $0 < \omega \leq 1$ whenever the original matrix is an irreducible weak diagonally dominant matrix. For $L-$ matrix, the convergence theorem states that: if the original matrix is an $L-$ matrix such that $0 \leq r \leq \omega \leq 1$, then the AOR method converges if and only if the Jacobi method converges. The main results from his findings indicates that the AOR iterative method converges for some specific values of r and ω when the coefficient matrix is $L-$ matrix, irreducible weak diagonally dominant matrix or consistently ordered matrix. Numerical experiment performed showed the superiority of the AOR method over the SOR method.

Finding an optimal acceleration parameter of the AOR method for a consistently ordered matrix is often associated with the relationship between the eigenvalue (μ) of Jacobi iteration matrix with eigenvalue λ of AOR iteration matrix which is given as

$$(\lambda - 1 + \omega)^2 = \omega\mu^2(r\lambda - r + \omega) \tag{2.19}$$

Hadjidimos (1978) considered the convergence analysis of a consistently ordered matrix with regards to the AOR method during his research work in 1978. He established a sufficient and necessary condition for the convergence of AOR method when the Jacobi method possesses real eigenvalues only. It was shown in the result that convergence domain of r relies on the estimate of ω. He also established the fact that when the over-relaxation and acceleration parameters are easily obtainable, the AOR tends to converge fast when compared to other iterative methods. Hence the matter of determining the optimal acceleration and over-relaxation factors requires further investigation.

2.5 Variants of Accelerated Over-Relaxation Method

Many researchers have been encouraged and are still interested in exploring the AOR method for solution of the linear system $(Az = b)$. This led to several modifications and versions of the AOR iterative method used in magnifying the convergence rate of the AOR method.

Darvishi *et al.* (2011) considered the improvement of the AOR scheme and proposed a numerical method known as Symmetric Modified Accelerated Over-relaxation iterative method (SMAOR) They investigated the convergence region of the scheme and carried out numerical experiment of some problems for comparison of the method with AOR method. Their finding indicated that the SMAOR performs better than the AOR method as it shows that the convergence rate of the SMAOR method is greater than the convergence rate of AOR.

Salkuyeh (2011) by applying the idea of Hadjidimos (1978), parameterized the AOR method and developed a new proposition of the AOR method called generalized Accelerated Over-Relaxation (GAOR) method. The splitting of matrix of the form $A = T_p - E_p - F_p$ was considered, where $T_p = (t_{ij})$ is a banded matrix with bandwidth $2p+1$, $-E_p$ and $-F_p$ are strictly lower and upper part of the matrix $A - T_p$. The method is formulated as;

$$z^{(k+1)} = \left(T_p - \gamma E_p\right)^{-1}\left[(1-\omega)T_p + (\omega-\gamma)E_p + \omega F_p\right]z^{(k)}$$
$$+\omega\left(T_p - \gamma E_p\right)^{-1}b \qquad k = 0, 1, 2, \cdots, N \qquad (2.32)$$

He studied the convergence property of the GAOR method for M matrices, established that the generalized Accelerated Over-relaxation method converges for M matrices and proves that the GAOR method converges faster than the classical AOR method.

On the other hand, Nasabzadeh and Toutounion (2013) utilized a block splitting of the coefficient matrix where the block matrix form is $A_1 = \begin{pmatrix} A_a & A_b \\ A_c & A_d \end{pmatrix}$ with $A_1 = V_D - V_L - V_U$, a different approach from Salkuyeh (2011), developed an improved version of the AOR method called the generalized Accelerated Over-Relaxation method which is designated as

$$z^{(k+1)} = (V_D - \gamma V_L)^{-1}[(1-\omega)V_D + (\omega - \gamma)V_L + \omega V_U]z^{(k)}$$
$$+\omega(V_D - \gamma V_L)^{-1}b, \qquad k = 0, 1, 2, \cdots, N \qquad (2.30)$$

while the GAOR method of Salkuyeh (2011) is convergent for $M-$ matrices, this GAOR method has been proved to be convergent for other matrices such as $L-$ matrix, strictly diagonally dominant matrix, Hermitian positive definite matrix and $H-$matrix. Also, the GAOR iterative method has shown to be more efficient than the basic AOR iterative method.

Shaikh *et al*. (2013) further improved on the AOR method and presented a generalized one-parameter reduction of the AOR method called Critical Accelerated method. They applied a splitting of the form $A = A_L^U + D$, D is the diagonal part and A_L^U is combination of both the strictly lower and strictly upper triangular matrices of matrix A, unlike the usual splitting of $A = D - U - L$ in Gauss-Seidel, Jacobi, SOR and AOR methods. The Critical Accelerated method is denoted as

$$z^{(k+1)} = \{(1-r)I - rA'\}z^{(k)} + \omega P \qquad (2.27)$$

where $A' = D^{-1}A_L^U$ and $P = D^{-1}b$, The method converges to the true solution for system of linear equations when $0 < r < 1$. They investigated and discussed the restrictions on the acceleration parameter in the method to ascertain convergence of the method for irreducible diagonally dominant and positive definite matrices. It was observed from the outcome of the study that the critical accelerated method is more efficient and superior to SOR and AOR Gauss-Seidel. They recommended more research on determination of the optimal values of the acceleration parameter provided it will provide the Critical Accelerated method a fast convergence than its present form.

Wu and Liu (2014) presented a new version of the AOR method called Quasi Accelerated Over-Relaxation (QAOR) iterative method. Based on induced splitting of the AOR method, the Quasi Accelerated Over-relaxation iterative method is established as

$$z^{(k+1)} = [(1+\omega)D - rL_A]^{-1}[D + (\omega - r)L_A + \omega U_A]z^k$$
$$+[(1+\omega)D - rL_A]^{-1}\omega b, \qquad k = 0,1,2,\cdots, \quad (2.25)$$

They examined some special matrices and obtained sufficient conditions with regards to the convergence analysis of the matrices. Findings from the convergence analysis indicates that QAOR method converges for an irreducible matrix with weak diagonal dominance with values $-1 \leq r \leq 1$ and $\omega > 0$, for $H-$matrix with $0 \leq r \leq \omega$ $(\omega \neq 0)$, for $L-$ matrix with $0 \leq r \leq \omega$, $(\omega \neq 0)$ and for symmetric positive definite matrix. The effectiveness and attainability of the QAOR iterative was examined and compared with the modified Successive Over-relaxation (KSOR) method by Yousef (2012) and AOR methods. Although the QAOR method is effective but not as efficient as the KSOR, that is to say the KSOR method performs better than the QAOR method under some conditions.

Youssef and Farid (2015) through the use of extrapolation approach on the modified successive over-relaxation method (KSOR) method by Youssef (2012) from the AOR point of view, formulated the KAOR iterative method represented as;

$$z^{(k+1)} = ((r+1)I - rL)^{-1}[(r+1-\omega)\mathrm{I} + (\omega - r)L + \omega U]z^{(k)}$$
$$+\omega((r+1)I - rL)^{-1}D^{-1}b \qquad k = 0,1,2,\cdots, \qquad (2.26)$$

Discussion and investigation for the convergence of the method were carried out in the study for matrices such as irreducible weak diagonally dominant matrix, $L-$ matrix and consistently ordered matrix. The study indicates that the method converges faster than the AOR iterative method however, this was possible due to the fact that they considered the negative value of the acceleration and relaxation parameters which was ignored in QAOR method by Wu and Liu (2014). Otherwise, the AOR method converges faster than the KAOR iterative method.

By avoiding the computation of the spectral radius of the AOR method and employing the minimization of the residual properties of AOR method, (Luna *et al*., 2016) presented a method known as Asymptotically Optimum Accelerated Over-Relaxation (AOAOR) method. The method is an optimization procedure in finding optimum parameters of the AOR method. Optimum acceleration parameters for symmetric positive definite matrix and non-symmetric matrix were investigated, analyzed and discussed extensively. They were able to prove the efficiency of the AOAOR method over the AOR method and established the fact that the method is more robust than the AOR method in terms of larger intervals of the parameters.

Akhir and Suleiman (2017) applied the idea of the classical Accelerated Over-Relaxation scheme and considered a combination of triangle element approximation with the AOR method to produce an excellent iterative solver for a 2D Helmotz equations. Performance of the method was clarified through a numerical test. The result reveals that the associated Accelerated Over-Relaxation method as regards the 2D Helmotz equations exhibited greater convergence improvements as compared to the Successive Over-Relaxation method.

Dahalan *et al*. (2018) proposed a method named Quarter- sweep Accelerated Over-Relaxation (QSAOR) method, a family of AOR method for solving robotic problem such as free collision path from an initial location to a specific end within their environment. By application of finite difference procedures on the Laplace equation which was modelled from the problem, the numerical test conducted indicates that it is able to generate smooth path between starting positions to specified destinations. Also. Based on their simulation result, performance of the QSAOR methods is far better and gives smooth path in comparison to previous research in literature.

In an attempt to ensure high rate of convergence, increasing the number of parameters has proven to accelerate convergence.

Recently, in a quest for improvement on existing two-parameter stationary iterative methods (Vatti *et al*., 2019a) modified the AOR method and developed a three-parameterized method called Parametric Accelerated Over-Relaxation (PAOR) method. The method is a modified version of the AOR method through introduction of a new acceleration parameter α and it is represented as;

$$z^{(k+1)} = \left((\alpha+1)I-\omega L\right)^{-1}[(\alpha+1-r)\mathrm{I}+(r-\omega)L+rU]z^{(k)}$$

$$+ \quad \left((\alpha+1)I-\omega L\right)^{-1^{-1}} rD^{-1}b \qquad \alpha \neq -1, \qquad k=0,1,2,\cdots, \qquad (2.28)$$

Convergence condition with regards to consistently ordered matrices for the method was studied and choices of the PAOR parameters were obtained in respect to the eigenvalues of the Jacobi iteration matrix. Efficiency of the method was verified and the result shows that the PAOR method although reduced to AOR method with $\alpha = 0$, its spectral radius is smaller compared to spectral radii of AOR, SOR, Gauss Seidel and Jacobi methods. This confirms that the PAOR method for any $\alpha \neq -1$, performs better than the other methods examined in the research work.

Furthermore, Vatti *et al*. (2019b) generalizes the Parametric Accelerated Over-Relaxation (PAOR) method for solving non-square linear systems. It was reported that the generalized PAOR method converges faster compared to AOR method.

Again, Vatti *et al*. (2020b) embarked on modification of the AOR method and developed an iterative method called Reaccelerated Over-Relaxation (ROR) method.

Eigenvalues of the ROR method were obtained and choices of its parameters were equally established. The method is represented in the form;

$$z^{(k+1)} = (I - \omega L)^{-1}[(r\omega + 1 - r)\mathrm{I} + (r - r\omega - \omega)L + (r - r\omega)U]z^{(k)}$$
$$+ \quad (I - \omega L)^{-1}(r - r\omega)D^{-1}b, \qquad k = 0, 1, 2, \cdots, \tag{2.29}$$

Convergence of the method was only focused on linear systems with consistently ordered matrices, some theorems were proposed and proved with respect to such matrices. They compared the method with some existing methods such as SOR, AOR, Gauss-Seidel and Jacobi methods through some numerical tests and the results obtained indicates that the convergence rate of the ROR method is faster than the methods of Jacobi, AOR, Gauss-Seidel and SOR examined in the study.

Zhang *et al.* (2020) extended the AOR splitting scheme and proposed two iterative methods called Newton-Successive Over- Relaxation (NSOR) and Newton-Accelerated Over- Relaxation (NAOR) methods for solving multilinear systems such as the tensor equations. Convergence conditions with regards the two methods were established. The methods were validated and it was shown that the NAOR method outperform the other methods compared in the study.

2. 6 Convergence of Stationary Iterative Methods

First of all, convergence in terms of stationary iterative methods is a different concept from convergence of numerical schemes. In numerical schemes, convergence is focused on the analytical solution minus the numerical solution while convergence in terms of iterative methods is mainly concerned with the difference between the approximate solution z at step k minus the exact solution $e^k = z^k - z^{(exact)}$, and the exact in $z^{(exact)}$ is not the exact of the differential equation, it is the exact solution of the linear system $Az = b$. Two questions which are also very important to the choice of whether or not to use the iterative methods (Jacobi, Gauss-Seidel, SOR and AOR) are;

I. What are the conditions under which these methods converge?

II. What is the rate of convergence?

Sufficient conditions for convergence of a particular iteration method can also be derived.

Theorem 2.1 (Saad, 2003)

The standard iterative scheme $z^{(k+1)} = Jz^{(k)} + f$ converges for any f or $z^{(0)}$ provided the spectral radius of J is less than one $(\rho(J) < 1)$. The spectral radius of an iteration matrix J, represented as $\rho(J)$, is given as

$$\rho(J) = \max_k |\lambda_k| \tag{2.12}$$

The iteration matrix J determines the rate of not only the rate of convergence but also whether or not it would converge. So the iteration matrix (J) is the key to the behavior of this iterative scheme. So, once the matrix J is constructed even before going through iteration procedure, one can find out or can predict whether or not the method will converge by finding out the eigenvalue of the method to check if the spectral radius is less than one. It is a known fact that convergence rate of stationary iterative procedure lies greatly on the spectral radius of the iteration matrix.

Theorem 2.2 (Varga, 2000)

If matrix A of the linear system $Az = b$, is irreducibly diagonally dominant (diagonally dominant), then the spectral radii of Gauss-Seidel and Jacobi iteration matrices are less than 1, and both Gauss-Seidel and Jacobi methods converges.

Theorem 2.3 (Young, 2014)

If A is symmetric with positive diagonal elements, then spectral radius of the SOR iteration matrix is less than 1, that is $\rho(J_{SOR}) < 1$ provided A is positive definite and the range $0 < \omega < 2$.

Theorem 2.3 (Hadjidimos, 1978)

If A is an $L-$ matrix, then spectral radius of the AOR iteration matrix is less than 1 $[\rho(J_{AOR}) < 1]$ provided A is an L matrix and $0 \leq r \leq \omega \leq 1$, the AOR method converges if and only if the Jacobi method converges.

2.7 Refinement of Iterative Methods

Sometimes an estimation to real solution of linear systems deviates from the real solution of the system and is usually described as the residual vector which simply means the left-over of the solution after approximations. In procedures of iterative methods such as Gauss-Seidel, Jacobi, Successive Over-relaxation (SOR) and so on, each computation component to the solution vector has an associated residual vector to it (Vatti *et al*., 2015).

Suppose A is an invertible matrix, the refinement of iterative method is mainly concerned with generating better estimations successively for obtaining solutions to linear systems $Az = b$. For the solution $z = A^{-1}b$, suppose there is an invertible matrix Y such that

$z = Yb \approx A^{-1}b$, where the application of Y is cheaper compared to application of solving the system matrix A. This estimate inverse Y can come from either any of the direct methods or from carrying out few steps of a particular stationary iterative methods used in solving $Az = b$. Now the question is, is there a possibility of improving the accuracy of the estimate solutions that is obtained from any of the stationary iterative methods for a system of linear equations, $Az = b$? The basic idea associated with this method is to examine the iteration

$$z^{(0)} = Yb \tag{2.44}$$

$$z^{(k+1)} = z^{(k)} + Y\left(b - Az^{(k)}\right) \tag{2.45}$$

And if $z^{(k+1)}$ converges, then it means that the limit must satisfy the expression:

$$z = z + Y(b - Az) \tag{2.46}$$

Where $r = b - A$ is the residual vector, likewise if convergence of the method is attained, it converges to the true solution of the linear system. Also, the refinement of iterative method $z^{(k+1)} = z^{(k)} + Y\left(b - Az^{(k)}\right)$ produces iterates in the form

$$z^{(n)} = Y \sum_{k=0}^{n} (I - AY)^{k}, \quad n \geq 0 \tag{2.47}$$

And the method converges to the real solution of $Az = b$ as long as $\|I - AY\| < 1$ that is to say if Y is sufficiently close to inverse of A. A method that utilizes this postulation is referred to as iterative refinement or iterative enhancement, consisting of carrying out iterations on the linear system having the residual vector at the right hand side, for successive estimations until accuracy of the results are satisfied. Iterative refinement is seen as an iterative technique which is used in improving the estimate solution $\bar{z}$ to the linear system $Az = b$, (Burden and Faires, 2011)

Refinement of iterative methods was introduced by in fifteenth century to enhance the accuracy of numerical estimations to systems of linear equations of the form $Az = b$. Once an estimation to the solution $\bar{z}$, has been made with any of the iterative methods like methods of AOR, Jacobi, SOR and Gauss-Seidel, then the following steps needs to be carried out to achieve the refinement of the specific method;

1. Commence with an initial estimate $z^{(0)} = \left(z_1^{(0)}, z_2^{(0)}, z_3^{(0)}, \cdots, z_n^{(0)}\right)$

2. Substitute the estimate $z^{(0)}$ into the desired iterative method $z^{(k+1)} = Jz^{(k)} + f$.

3. Insert the result of 2 into the refinement formula and

4. Formulate the update using $\bar{z}^{(k+1)} = z^{(k+1)} + Y\left(b - Az^{(k+1)}\right)$

5. Return to step 2 if convergence is not achieved.

Then one checks if the solution that has been obtained is correct to the desire degree of accuracy, if so, then the iteration is stopped and if not, then the computations continues until the desired degree of accuracy is achieved. An important objective of the refinement of iterative methods is to produce sequence of estimations which will make the residual vector converge very quickly to zero, thereby enhancing the convergence rate of iterative methods. The iterations are usually terminated whenever norm of the residue $\|r^k\| = \|b - Az^k\|$ becomes sufficiently small, (Guan and Chandio, 2017).

It is in the light of the above that some researchers decided to research into the refinement of the various iterative methods in order to accelerate the convergence rate of the specified iterative method for solving linear systems. Jacobi method remains one of the iterative methods with fewer calculations and low convergence.

In an attempt to enhance the convergence rate of the Jacobi method, Dafchahi (2008) modified the Jacobi method and developed a refinement of Jacobi (RJ) method. Convergence of the method was investigated and it was proved that the refinement of Jacobi method is convergent for a strictly diagonal dominant matrix and a consistently ordered matrix. The RJ method is more efficient when compared with methods of Jacobi and Gauss-Seidel and it is as fast as the SOR method. Also, the RJ method seems easier when comparing with method of SOR since finding optimal parameter is not required during iteration process.

Vatti and Gonfa (2011) improved on the generalized Jacobi method by Salkuyeh (2007) and developed Refinement of generalized Jacobi (RGJ) method. The convergence of the method for strictly diagonal dominant and M -matrices was discussed. Also, it was confirmed that the RGJ method exhibits faster convergence in comparison with generalized Jacobi method and this gives rise to the conclusion of their study that the refinement of generalized Jacobi is superior to the generalized Jacobi method and should be used in place of generalized Jacobi when solving linear system.

Vatti and Tesfaye (2011) developed refinement of Gauss-Seidel (RGS) iterative method in order to enhance the convergence rate of Gauss-Seidel iterative method. They discussed the convergence of the method for strictly diagonally dominant matrix and positive definite matrix. It was proven that RGS method converges twice as fast as the GS iterative method when compared with Gauss-Seidel method, thereby confirms that the refinement of Gauss-Seidel method is superior to Gauss-Seidel method.

Kyurkhiev and Iliev (2013) made some improvements on the SOR and SSOR schemes and proposed the methods called Refinement of Successive Over- Relaxation (RSOR) and Refinement of Symmetric Successive Over-Relaxation (RSSOR) methods based on the reverse of Gauss-Seidel method. The methods are convergent for strictly diagonally dominant and $M-$ matrices and the two methods yield reasonable improvements in convergence rate compared to SOR and SSOR iterative methods.

While surveying the refinement of Jacobi (RJ) method, refinement of generalized Jacobi (RGJ) method and refinement of Gauss-Seidel (RGS) method, Laskar and Behera (2014), discovered that the RJ method takes longer time to converge to the true solution than RGS and RGJ methods. Based on the outcome of their findings in terms of number of iterations, level of accuracy and performances of the three refinement methods, it was observed that the RGJ method is highly efficient than the RGS and RJ iterative methods.

Genanew (2016) improved on the convergence rate of the generalized Gauss-Seidel (GGS) method and developed the Refinement of generalized Gauss-Seidel (RGGS) method. He observed that the method is quite efficient than SOR and Refinement of generalized Jacobi methods.

Vatti *et al*. (2018) refined the AOR method in order to speed up its convergence rate and proposed a two-parameterized method called Refinement of Accelerated Over-Relaxation (RAOR) method. The residual vector of the AOR method was obtained and refined to achieve the RAOR method. The matrix splitting $A = I - L - U$ with the formula of the AOR method $z^{(k+1)} = [I - rL]^{-1}[(1-\omega)I + (\omega - r)L + \omega U]z^{(k)} + [I - rL]^{-1}\omega\dot{b}$, were utilized to formulate the RAOR iterative method, which results into the form;

$$\bar{z}^{(k+1)} = ([I - rL]^{-1}[(1-\omega)I + (\omega - r)L + \omega U])^2 \omega z^{(k)} + (I + \omega[I - rL]^{-1}[(1-\omega)I + (\omega - r)L + \omega U])[I - rL]^{-1}\dot{b} \qquad (2.55)$$

They discussed and investigated the convergence of the method for irreducible weak diagonally dominant and consistently ordered matrices. The RAOR method was compared with AOR method and it shows that the convergence rate of the RAOR method is faster than the AOR method.

Muleta and Gofe (2018) in the quest to accelerate the convergence rate of the generalized Accelerated Over-relaxation method, developed the refinement of generalized Accelerated Over-relaxation (RGAOR) method. Convergence analysis of the RGAOR method for M −matrix and strictly diagonally dominant matrix were studied. In comparison with the generalized AOR method, the refinement of the generalized Accelerated Over-relaxation method converges to the true solution quicker than the generalized AOR method. Also, they arrived at a conclusion that the error of the refinement of generalization of AOR method at any predefined error of tolerance is lesser when compared to other methods examined in their study.

Tesfaye *et al*. (2019), developed the second refinement of Jacobi (SRJ) method aimed at speeding up the convergence rate of the Refinement of Jacobi method. Convergence of the SRJ method for $M -$matrix, strictly diagonally dominant matrix and symmetric positive definite matrix was investigated and discussed extensively. The second refinement of Jacobi method minimized the number of iterations by 1/3 (one third) of Jacobi and 2/3 (two-third) of refined Jacobi methods and thus proves that the SRJ method converges faster than refined Jacobi and Jacobi methods.

Recently, Tesfaye *et al*. (2020) proposed a method called second Refinement of Gauss-Seidel (SRGS) method and proves that the method is more efficient than Refinement of Gauss-Seidel (RGS) method.

Assefa and Teklehaymanot (2021) carried out a study to increase the convergence speed of the Refinement of Accelerated Over Relaxation (RAOR) Method. They introduced the second Refinement of Accelerated Over Relaxation (SRAOR) method and minimized the spectral radius of the RAOR iteration matrix. The two-parameter SRAOR method was equally refined to the third, fourth up to the m^{th} refinement versions of the Accelerated iterative methods. Numerical findings indicates that the second Refinement of Accelerated Over Relaxation method and other higher refinement versions of Accelerated iterative methods surpasses the Refined AOR method.

2.8 Application of Stationary Iterative Methods

Linear systems play a vital role in several applications. Although, most of the applications make use of fuzzy numbers instead of crisp numbers for precision reason. This necessitate the need for numerical techniques that would treat and solve fuzzy linear systems appropriately (Kargar *et al.*, 2014).

Gebregiorgis and Gofe (2018) employed the splitting technique of an $M-$ matrix and embedding procedure with refinement process to solve some fuzzy linear systems through the use of Refinement of Generalized Jacobi (RGJ) and Refinement of Generalized Gauss-Seidel (RGGS) iterative methods. In this study, we apply the newly developed numerical iterative methods to solve a pair of fuzzy linear equations.

Many scientists and engineers are usually interested in solving practical life problems or realistic problems, where such problems are often discretized into linear systems and then solved using iterative methods. Mayooran and Elliot (2016) applied the SOR iterative method in solving heat transfer problem on a flat plate with a steady boundary temperature. This study examined the performance of the newly developed EAOR and REAOR iterative methods for a real life problem by solving the heat transfer problem.

CHAPTER THREE

3.0 MATERIALS AND METHOD

3.1 Derivation of the Proposed Method

Considering a numerical solution of the linear system in the form

$$Az = b \tag{3.1}$$

Expressed in the matrix form;

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1(m-1)} & a_{1m} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2(m-1)} & a_{2m} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & \cdots & a_{(n-1)(m-1)} & a_{(n-1)m} \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{n(m-1)} & a_{nm} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_{m-1} \\ z_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} \tag{3.2}$$

Such that

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1(m-1)} & a_{1m} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2(m-1)} & a_{2m} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & \cdots & a_{(n-1)(m-1)} & a_{(n-1)m} \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{n(m-1)} & a_{nm} \end{pmatrix}, \qquad z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_{m-1} \\ z_m \end{pmatrix},$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} \tag{3.3}$$

where A is a non-singular $m \times m$ matrix having a unique solution expressed in the form;

$$z = A^{-1}b \tag{3.4}$$

If A has a non-vanishing diagonal elements, then a usual splitting of A is obtained thus:

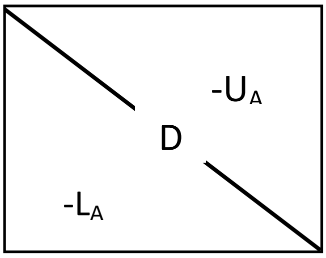

Figure 3.1: Usual Splitting of Matrix A

$$A \equiv D - L_A - U_A \tag{3.5}$$

Where the components of A are

$$D = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{(n-1)(m-1)} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{nm} \end{pmatrix} \tag{3.6}$$

$$-L_A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & 0 & 0 \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & a_{(n-1)(m-2)} & 0 & 0 \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{n(m-1)} & 0 \end{pmatrix} \tag{3.7}$$

$$-U_A = \begin{pmatrix} 0 & a_{12} & \cdots & \cdots & a_{1(m-1)} & a_{1m} \\ 0 & 0 & a_{23} & \cdots & \cdots & a_{2m} \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & a_{(n-1)m} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.8}$$

With the splitting in equation (3.5), equation (3.1) can be written as $(D - L_A - U_A)z = b$. A regular splitting of the square matrix A into

$$A = M - N \tag{3.9}$$

is required for the iterative solution of equation (3.1), such that substituting $A = M - N$ into $Az = b$, leads to the following expressions:

$$\begin{aligned} (M - N)z &= b \\ Mz &= Nz + b \\ Mz^{(k+1)} &= Nz^{(k)} + b \\ z^{(k+1)} &= M^{-1}Nz^{(k)} + M^{-1}b \\ z^{(k+1)} &= Jz^{(k)} + p \end{aligned} \tag{3.10}$$

where $J = M^{-1}N$ is the iteration matrix and $p = M^{-1}b$ is the corresponding column vector of the iterative method. This study seeks to derive a stationary linear iterative method in the same form as $z^{(k+1)} = M^{-1}Nz^{(k)} + M^{-1}b$, where the choices for M and N are given as

$$\begin{aligned} M &= \beta_1 D + \beta_2 L_A \\ N &= \beta_3 D + \beta_4 L_A + \beta_5 U_A \end{aligned} \tag{3.11}$$

Next, we apply a general linear stationary iterative method whose matrix coefficients are linear functions of the components of matrix A and the new iterate is at most a lower triangular matrix. This proposed method is in the form;

$$(\beta_1 D + \beta_2 L_A)z^{(k+1)} = (\beta_3 D + \beta_4 L_A + \beta_5 U_A)z^{(k)} + \beta_6 b, \quad k = 0,1,2,\cdots, \tag{3.12}$$

where $\beta_i, i = 1, 2, \cdots, 6$ are constants to be determined ($\beta_1 \neq 0$) and $z^{(0)}$ an arbitrary initial estimation to the solution z in (3.4). Equation (3.12) is divided by β_1 to obtain

$$\left(D + \frac{\beta_2}{\beta_1} L_A\right) z^{(k+1)} = \left(\frac{\beta_3}{\beta_1} D + \frac{\beta_4}{\beta_1} L_A + \frac{\beta_5}{\beta_1} U_A\right) z^{(k)} + \frac{\beta_6}{\beta_1} b \tag{3.13}$$

Let $\frac{\beta_i}{\beta_1} = \beta_i', i = 2, 3, \cdots 6$ and $\frac{\beta_1}{\beta_1} = 1$, then the above equation becomes

$$(D + \beta_2' L_A) z^{(k+1)} = (\beta_3' D + \beta_4' L_A + \beta_5' U_A) z^{(k)} + \beta_6' b \quad k = 0,1,2, \cdots, \tag{3.14}$$

Also, since $z^{(k+1)}$ and $z^{(k)}$ are the iteration counts, then equation (3.14) can be written as

$$[D + \beta_2' L_A - \beta_3' D - \beta_4' L_A - \beta_5' U_A] z = \beta_6' b \tag{3.15}$$

And sufficient conditions for the method (3.14) to be consistent with $Az = b$ are:

$$[(1 - \beta_3') D + (\beta_2' - \beta_4') L_A - \beta_5' U_A]\, z \equiv \beta_6' b \tag{3.16}$$

Substituting the value of $z = A^{-1} b$ into (3.16), results into

$$[(1 - \beta_3') D + (\beta_2' - \beta_4') L_A - \beta_5' U_A] A^{-1} b \equiv \beta_6' b \tag{3.17}$$

Multiply through by A to obtain

$$(1 - \beta_3') D + (\beta_2' - \beta_4') L_A - \beta_5' U_A \equiv \beta_6' A\,, \qquad \beta_6' \neq 0 \tag{3.18}$$

Or equivalently

$$A \equiv \left(\frac{1 - \beta_3'}{\beta_6'}\right) D + \left(\frac{\beta_2' - \beta_4'}{\beta_6'}\right) L_A - \left(\frac{\beta_5'}{\beta_6'}\right) U_A \tag{3.19}$$

In view of equation (3.5), the first relationship of equation (3.18) gives

$$\begin{matrix} \frac{1 - \beta_3'}{\beta_6'} = 1 \\ \frac{\beta_2' - \beta_4'}{\beta_6'} = -1 \\ \frac{-\beta_5'}{\beta_6'} = -1 \end{matrix} \quad \xrightarrow{yields} \quad \begin{matrix} 1 - \beta_3' = \beta_6' \\ \beta_2' - \beta_4' = -\beta_6' \\ -\beta_5' = -\beta_6' \end{matrix} \tag{3.20}$$

At this point, we have three linear equations with five variables (unknowns). This is a consistent linear system that has infinitely many solutions since the number of variables exceeds the number of equations. The degree of freedom of equation (3.20) is 2 and in this case, we observe that there are two free variables (β_2' and β_6') shown below

$$\begin{aligned} \beta_3' &= 1 - \beta_6' \\ \beta_4' &= \beta_2' + \beta_6' \\ \beta_5' &= \beta_6' \end{aligned} \tag{3.21}$$

Therefore, the solution to the linear system in (3.20) takes the form;

$$(\beta_2', \beta_3', \beta_4', \beta_5', \beta_6') = (\beta_2', 1 - \beta_6', \beta_2' + \beta_6', \beta_6', \beta_6') \qquad (3.22a)$$

The choice we make for the two free variables are;

$$\begin{aligned} \beta_2' &= -(v + r) = -v - r \\ \beta_6' &= \omega \end{aligned} \qquad (3.22b)$$

(3.22b) is substituted into our solution (3.22a) to obtain

$$(\beta_2', \beta_3', \beta_4', \beta_5', \beta_6') = (-v - r, 1 - \omega, \omega - v - r, \omega, \omega) \qquad (3.22c)$$

Thus, it gives the following three-parameter solution of the method:

$$\beta_2' = -v - r, \qquad \beta_3' = 1 - \omega, \qquad \beta_4' = \omega - v - r, \qquad \beta_5' = \omega, \quad \beta_6' = \omega \qquad (3.23)$$

where r, v and $\omega \neq 0$ are any 3 fixed parameters, consequently, substituting (3.23) into (3.11) results into

$$\begin{aligned} M &= D - (v + r)L_A \\ N &= (1 - \omega)D + [\omega - v - r]L_A + \omega U_A \end{aligned} \tag{3.24}$$

And (3.23) into (3.14) gives the proposed iterative method

$$[D - (v + r)L_A]z^{(k+1)} = [(1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A]z^{(k)} + \omega b \tag{3.25}$$

After multiplying the above equation by D^{-1} and setting $L = D^{-1}L_A$, $U = D^{-1}U_A$, $I = D^{-1}D$ and $\dot{b} = D^{-1}b$, it results into

$$[I - (v + r)L]z^{(k+1)} = [(1 - \omega)I + [\omega - (v + r)]L + \omega U]z^{(k)} + \omega \dot{b} \tag{3.26}$$

Or

$$\begin{aligned} z^{(k+1)} = {} & [I - (v + r)L]^{-1}[(1 - \omega)I + [\omega - (v + r)]L + \omega U]z^{(k)} \\ & + [I - (v + r)L]^{-1}\omega \dot{b} \end{aligned} \tag{3.27}$$

Method (3.25) or its equivalent (3.27) is now the proposed Extended Parameterized Accelerated Over Relaxation (EAOR) iterative method. The new parameter $\boldsymbol{v}$ will be called extended acceleration parameter, $\boldsymbol{r}$ the acceleration parameter and $\boldsymbol{\omega}$ the relaxation parameter. The proposed EAOR method can also be written in a general linear stationary method as

$$z^{(k+1)} = E_{v,\omega,r}z^{(k)} + [I-(v+r)L]^{-1}\omega\dot{b} \tag{3.28}$$

The notation $E_{v,\omega,r}$ is utilized to represent the new Extended Parameterized Accelerated Over Relaxation (EAOR) iteration matrix which is represented as

$$E_{v,\omega,r} = [I-(v+r)L]^{-1}[(1-\omega)I + [\omega-(v+r)]L + \omega U] \tag{3.29}$$

The spectral radius of the proposed Extended Accelerated Over-Relaxation iterative scheme (EAOR) is the largest eigenvalue of its iteration matrix denoted as $\rho\left(E_{v,\omega,r}\right)$. It is observed that for certain values of the parameters v, r and ω, the proposed EAOR method reduces to well-known iteration methods, which is shown in the following analysis.

$E_{0,0,1}$- method is the Jacobi method:

Considering the linear system $Az = b$, where the coefficient A matrix is decomposed into $A = D - L - U$, then the Jacobi method in matrix form is given by

$$\begin{aligned}(D-L-U)z &= b\\ Dz^{(k+1)} &= (L+U)z^{(k)} + b\\ z^{(k+1)} &= D^{-1}(L+U)z^{[k]} + D^{-1}b\end{aligned} \tag{3.30}$$

The proposed EAOR method $E_{v,r,\omega}$ is given by

$$z^{(k+1)} = [D-(v+r)L]^{-1}[(1-\omega)\mathrm{D} + [\omega-(v+r)]L + \omega U]z^{(k)}$$

$$+[D-(v+r)L]^{-1}\omega b \tag{3.31}$$

Substituting the values of $v = 0,\ r = 0$ and $\omega = 1$ for $E_{0,0,1}$ into (3.31), the EAOR reduces to the Jacobi method as:

$$z^{(k+1)} = [D-(0+0)L]^{-1}[(1-1)\mathrm{D} + (1-0-0)L + 1.U]z^{(k)}$$

$$+[D-(0+0)L]^{-1}1.b \tag{3.32}$$

$$z^{[k+1]}=D^{-1}[(L+U)]z+D^{-1}b \tag{3.33}$$

The $E_{v,\omega,r}$ in (3.31) with the reduced method of $E_{0,0,1}$ in equation (3.33) yields $z^{(k+1)}=D^{-1}[L+U]z^{(k)}+D^{-1}b$ which is equivalent to the Jacobi method in equation (3.37).

$E_{0,1,1}$- method is the Gauss-Seidel method:

The matrix form of the Gauss-Seidel method is as follows:

$$\begin{aligned}(D-L-U)z&=b\\ Dz^{(k+1)}&=Lz^{(k)}+Uz+b\\ (D-L)z^{(k+1)}&=Uz^{(k)}+b\\ x^{(k+1)}&=[D-L]^{-1}U+[D-L]^{-1}b\end{aligned} \tag{3.34}$$

Substituting the values of $v=0$, $r=1$ and $\omega=1$ for $E_{0,1,1}$ in equation (3.31), the proposed EAOR reduces to the Gauss-Seidel method as follows:

$$z^{(k+1)}=[D-(0+1)L]^{-1}[(1-1)\mathrm{I}+(1-0-1)L+1.U]z^{(k)}$$

$$+[D-(0+1)L]^{-1}1.b \tag{3.35}$$

$$z^{(k+1)}=[D-L]^{-1}U+[D-L]^{-1}b \tag{3.36}$$

$E_{v,r,\omega}$ in equation (3.31) with the reduced method $E_{0,1,1}$ in (3.35) yields $z^{(k+1)}=[D-L]^{-1}Uz^{(k)}+[D-L]^{-1}b$ which is equivalent to the Gauss-Seidel method in (3.34).

$E_{0,\omega,\omega}$ method is the SOR method:

The correction or displacement vector for the Gauss-Seidel iteration is represented by the second equation in equation (3.34)

$$Dz^{(k+1)} = Lz^{(k+1)} + Uz + b$$
$$z^{(k+1)} = D^{-1}\left(L + Uz^{(k)} + b\right) \tag{3.37}$$
$$\dot{z}^{(k+1)} = D^{-1}(Lz^{(k+1)} + Uz^{(k)} + b\ )$$

But the actual components of $z^{(k+1)}$ of SOR method is expressed as:

$$z^{(k+1)} = z^{(k)} - \ \omega\left(\dot{z}^{(k+1)} - z^{[k]}\right) = (1-\omega)z^{(k)} - \omega\left(\dot{z}^{(k)}\right) \tag{3.38}$$

Combining the last equation in (3.37) with (3.38) to a single equation gives

$$z^{(k+1)} = (1-\omega)z^{(k)} + \omega D^{-1}\left(Lz^{(k+1)} + Uz^{(k)} + b\ \right) \tag{3.39}$$

The above equation is multiplied by D to obtain

$$Dz^{(k+1)} = D(1-\omega)z^{(k)} + \omega\left(Lz^{(k+1)} + Uz^{(k)} + b\right) \tag{3.40}$$

Hence the SOR iterative method is given as:

$$z^{(k+1)} = (D-\omega L)^{-1}[(1-\omega)D + \omega U]z^{(k)} + (D-\omega L)^{-1}\omega b \tag{3.41}$$

Substituting the values of $v = 0,\ r = \omega$ and $\omega = \omega$ for $E_{0,\omega,\omega}$ into (3.31), the proposed EAOR reduces to the SOR iterative method as:

$$z^{(k+1)} = [D-(0+\omega)L]^{-1}[(1-\omega)\mathrm{D} + (\omega - 0 - \omega)L + \omega U]z^{(k)}$$
$$+[D-(0+\omega)L]^{-1}\omega b \tag{3.42}$$

$$z^{(k+1)} = [D-\omega L]^{-1}[(1-\omega)D + U]z^{(k)} + \omega[D-\omega L]^{-1}b \tag{3.43}$$

Method (3.31) with the reduced method (3.50), yields $z^{(k+1)} = [D-\omega L]^{-1}[(1-\omega)D + U]z^{[k]} + \omega[D-\omega L]^{-1}b$ which is equivalent to the SOR method in (3.43).

$E_{0,r,\omega} -$ method is the AOR method: The Accelerated Over-relaxation iteration method is given as:

$$z^{(k+1)} = [D-rL]^{-1}[(1-\omega)D + (\omega - r)L + \omega U]z^{(k)} + [D-rL]^{-1}\omega b \tag{3.44}$$

Substituting the values of $v = 0,\ r = r$ and $\omega = \omega$ for $E_{0,r,\omega}$ into (3.31) of the proposed EAOR iterative method, reduces to the AOR method as:

$$z^{(k+1)} = [D-(0+r)L]^{-1}[(1-\omega)D + (\omega - 0 - r)L + \omega U]z^{(k)}$$
$$+[D-(0+r)L]^{-1}\omega b \tag{3.45}$$

$$z^{(k+1)} = [D - rL]^{-1}[(1-\omega)D + (\omega - r)L + \omega U]z^{(k)} + [D - rL]^{-1}\omega b \quad (3.46)$$

The substitution of the specified values for $E_{0,r,\omega}$ results into $z^{(k+1)} = [D - rL]^{-1}[(1-\omega)D + (\omega - r)L + \omega U]z^{(k)} + [D - rL]^{-1}\omega b$, which is equivalent to the AOR method in equation (3.44).

Obviously, the iteration matrix of the proposed EAOR method is similar to that of the AOR method; based on this fact, the EAOR method may conserve all advantages of the AOR method. Also, the criteria for obtaining the approximate solutions of the desired linear system is $z^{(k+1)} = z^{(k)}$ which will be called a convergent solution. More so, computation of the error will be obtained through application of the formula $E = z - z^{(k)}$, where z is the true solution.

3.2 Convergence Theorems of the Proposed EAOR Method

Given that an iterative method converges whenever the spectral radius is less than one, hence convergence of the new EAOR method is established by showing that the spectral radius of the proposed EAOR method is less than one, that is $\rho(E_{v,r,\omega}) < 1$. The study shall employ the use of the following lemmas to establish the convergence of the EAOR method for certain class of matrices:

Lemma 3.1 (Yun, 2011):

Let $A \geq 0$ be an irreducible matrix. Then

i. A has a positive real eigenvalue equal to its spectral radius.

ii. To the spectral radius of A, $\rho(A)$, there corresponds an eigenvector $z > 0$.

iii. $\rho(A)$ increases when any entry of A increases.

iv. $\rho(A)$ is considered a simple eigenvalue of A.

Lemma 3.2 (Aijuan, 2011):

Let $A = [a_{ij}]$ and $C = [c_{ij}]$, be two matrices such that $A \leq C$, where $c_{ij} \leq 0$ for all $i \neq j$, then if A is an M −matrix so also is matrix C.

Lemma 3.3 (Wang and Song, 2009):

Suppose matrix A is an M −matrix and the splitting $A = M - N$ is a weak regular or regular splitting of A, then $\rho(M^{-1}N) < 1$.

3.2.1 Convergence of L −matrix

Suppose matrix A is an L −matrix, which implies a matrix whose element (a_{ij}) satisfies the relationships below

$$\begin{array}{c} a_{ii} > 0\,, \quad i = 1, 2, \cdots, N \\ a_{ij} \leq 0\,, \quad i \neq j \quad \text{for all} \quad i, j = 1, 2, \cdots, N \end{array} \tag{3.46}$$

Then the below theorem with regards to the proposed EAOR iterative method is proposed.

Theorem 3.1: If matrix A is an L −matrix, then for all v, r and ω such that $0 < v + r \leq \omega \leq 1$ and $\omega \neq 0$, the new EAOR method $(E_{v,r,\omega})$ converges if and only if the Jacobi method $(E_{0,0,1})$ converges.

Let;

The given matrix A be an L −matrix, $D^{-1}A = I - U - L$, such that $U \geq 0$ and $L \geq 0$.

The spectral radius of the Jacobi method $(L + U)$ be $\rho(E_{0,0,1})$.

The EAOR iteration matrix $E_{v,r,\omega} = [I - (v + r)L]^{-1}[(1 - \omega)I + [\omega - (v + r)]L + \omega U]$ and the spectral radius of the EAOR iteration matrix be $\rho(E_{v,r,\omega})$.

$\dot{\lambda}$ be an eigenvalue of $\rho(E_{v,r,\omega})$

Proof:

Assume that $\dot{\lambda} = \rho(E_{v,r,\omega}) \geq 1$. Due to our assumption, we can easily obtain

$$[(1-\omega)I+(\omega-v-r)L+\omega U\,]\geq\ 0 \tag{3.47}$$

The proposed EAOR matrices are examined to check if they are nonnegative. The $E_{v,r,\omega}$ becomes an identity matrix (I) when $v+r=0,\ \omega=0$ and positive with $v+r=1, \omega=1$. But when $v+r<0,\ \omega<0$ and $v+r>1,\ \omega>1$, negative values appears in the matrix. Hence, the range of values which ensures that the EAOR matrix is non-negative is within $0<v+r\leq\omega\leq 1$. In the $E_{v,r,\omega}$ matrix, for $0<v+r\leq\omega\leq 1$, $(1-\omega)I+[\omega-(v+r)]L+\omega U\geq 0$ since $U\geq 0$ and $L\geq 0$. However, $\omega\neq 0$ is considered so that U does not vanish in the matrix, therefore, range of values to ensure non-negativity of $E_{v,r,\omega}$ matrix are $0<v+r\leq\omega\leq 1$ and $\omega\neq 0$.

Likewise, binomial expansion of $[I-(v+r)L]^{-1}$ gives:

$$[I-(v+r)L]^{-1}=I+(v+r)L+(v+r)^2L^2+\cdots+(v+r)^{N-1}L^{N-1}\geq 0 \tag{3.47}$$

Given that L is nonnegative and $(v+r)\geq 0$ hence, matrix $[I-(v+r)L]^{-1}$ is nonnegative. Next is to check if matrix $[I-(v+r)L]^{-1}[(1-\omega)I+[\omega-(v+r)]L+\omega U\,]$ is nonnegative, we obtain

$$\begin{aligned}
E_{v,r,\omega} &= [I-(v+r)L]^{-1}[(1-\omega)I+(\omega-v-r)L+\omega U\,]\\
&= I+(v+r)L+(v+r)^2L^2+\cdots+(v+r)^{N-1}L^{N-1}\\
&\qquad\times[(1-\omega)I+(\omega-v-r)L+\omega U\ \,]\\
&= (1-\omega)I+(v+r)(1-\omega)L+(v+r)^2(1-\omega)L^2+(\omega-v-r)L\\
&\qquad+(v+r)(\omega-v-r)L^2+(v+r)^2(\omega-v-r)L^3+\omega U\\
&\qquad+\omega(v+r)LU+\omega(v+r)^2L^2U+\cdots\\
&= (1-\omega)I+(v+r)(1-\omega)L+\omega U+P
\end{aligned} \tag{3.48}$$

where P represents non-negative terms. And finally,

$$E_{v,r,\omega}=[I-vl-rL]^{-1}[(1-\omega)I+[\omega-(v+r)]L+\omega U\,]\geq 0 \tag{3.49}$$

Since $E_{v,r,\omega}$ is a non-negative matrix as shown above, it implies that $\dot{\lambda}$ is an eigenvalue of $E_{v,r,\omega}$. If $z \neq 0$ is the corresponding eigenvector to $\dot{\lambda}$, then we have $E_{v,r,\omega}\, z = \dot{\lambda} z$ which yields the following equations below;

$$\begin{gathered} E_{v,r,\omega} z = \dot{\lambda} z \\ [I-(v+r)L]^{-1}[(1-\omega)I + [\omega-(v+r)]L + \omega U\,]z = \dot{\lambda} z \\ [(1-\omega)I + [\omega-(v+r)]L + \omega U\,]z = \dot{\lambda}[I-(v+r)L]z \\ \left[\omega - vL - rL + \dot{\lambda} rL + \dot{\lambda} vL + \omega U\,\right]z = \left[\omega I + \dot{\lambda} I - I\right]z \\ \left[(\dot{\lambda} r + \dot{\lambda} v + \omega - v - r)L + \omega U\,\right]z = \left(\dot{\lambda} + \omega - 1\right)Iz \\ \left[\frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}L + U\,\right]z = \left(\frac{\dot{\lambda}+\omega-1}{\omega}\right)Iz \end{gathered} \tag{3.50}$$

The last equation in (3.50) indicates that $\frac{\dot{\lambda}+\omega-1}{\dot{\omega}}$ is an eigenvalue of $\frac{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}{\omega}L + U$ by definition of an eigenvalue $(Az = \dot{\lambda} z)$. If $\dot{\lambda} \geq 1$ is an eigenvalue, then by definition

$$E_{v,r,\omega} z - \dot{\lambda} z \geq 0 \;\;\Rightarrow\;\; E_{v,r,\omega} z \geq \dot{\lambda} z \;\;\Rightarrow\;\; \dot{\lambda} z \;\leq E_{v,r,\omega} z \tag{3.51}$$

Which gives $\dot{\lambda} \leq \rho\left(E_{v,r,\omega}\right)$ and consequently

$$\frac{\dot{\lambda}+\omega-1}{\omega} \;\leq\; \rho\left(\frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}L + U\right) \tag{3.52}$$

It is easily seen that $\frac{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}{\omega} \geq 1$ so that

$$\begin{aligned} 0 &\leq \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}L + U \\ &\leq \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}L + \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}U \\ &\leq \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}(L+U) \\ &= \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\omega}E_{0,0,1} \end{aligned} \tag{3.53}$$

where $E_{0,0,1}$ is the Jacobi matrix and combining equations (3.52) with (3.53), yields

$$\dot{\lambda} + \omega - 1 \le \omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)\rho(E_{0,0,1})$$

$$\frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} \le \rho(E_{0,0,1}) \tag{3.54}$$

$$\rho(E_{0,0,1}) \ge \frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)}$$

bbut

$$\frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} - 1 = \frac{(r+v)(1-\dot{\lambda}) + (\dot{\lambda} - 1)}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)}$$

such that

$$\frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} = 1 + \frac{(r+v)(1-\dot{\lambda}) + (\dot{\lambda} - 1)}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)}$$

and this implies

$$1 + \frac{(r+v)(1-\dot{\lambda}) + (\dot{\lambda} - 1)}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} \ge 1$$

Then we can deduce that

$$\rho(E_{0,0,1}) \ge \frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} = 1 + \frac{(r+v)(1-\dot{\lambda}) + (\dot{\lambda} - 1)}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} \ge 1$$

$$\rho(E_{0,0,1}) \ge \frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} \ge 1 \tag{3.55}$$

and thus

$$\rho(E_{0,0,1}) \ge 1 \tag{3.56}$$

The above analysis shows that if $\dot{\lambda} \ge 1$, then the spectral radius of the Jacobi scheme is equally greater than or equal to 1. Similarly, suppose $\rho(E_{0,0,1}) < 1$, then

$$\rho(E_{0,0,1}) < \frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} \tag{3.57}$$

but

$$1 - \frac{\dot{\lambda} + \omega - 1}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)} = \frac{r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1) + (1-\dot{\lambda})}{\omega + r(\dot{\lambda} - 1) + v(\dot{\lambda} - 1)}$$

such that

$$\frac{\dot{\lambda}+\omega-1}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}=1-\frac{r(\dot{\lambda}-1)+v(\dot{\lambda}-1)+(1-\dot{\lambda})}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}$$

and this implies

$$1-\frac{r(\dot{\lambda}-1)+v(\dot{\lambda}-1)+(1-\dot{\lambda})}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}<1$$

Then we can deduce that

$$\rho(E_{0,0,1})<\frac{\dot{\lambda}+\omega-1}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}=1-\frac{r(\dot{\lambda}-1)+v(\dot{\lambda}-1)+(1-\dot{\lambda})}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}<1$$

$$\rho(E_{0,0,1})<\frac{\dot{\lambda}+\omega-1}{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}<1 \tag{3.55}$$

Hence

$$\rho(E_{0,0,1})<1 \tag{3.58}$$

which implies $\dot{\lambda}<1$, so that if $\rho(E_{0,0,1})<1$ then the proposed EAOR scheme equally converges $\{\rho(E_{v,r,\omega})<1\}$ since the spectral radius of the Jacobi matrix $\{\rho(E_{0,0,1})\}$ is incorporated inside the proposed EAOR iterative method and this completes the proof.

3.2.2 Convergence of irreducible matrix with weak diagonal dominance

If a matrix $A=(a_{ij})$ is irreducible and weakly diagonally dominant, then the matrix will be non-singular with non-vanishing diagonal elements and thus the following theorem is proposed;

Theorem 3.2: If A is an irreducible matrix with weak diagonal dominance and $0<v+r\leq 1$ and $0<\omega\leq 1$, then the proposed EAOR iterative method converges to the true solution for any initial estimation $z^{(0)}$.

Proof:

The theorem can be proved by contradiction. Let A be an irreducible matrix with

$|a_{ii}| \geq \sum_{j=1,j\neq i}^{n} |a_{ij}|$. From lemma 3.1, it is assumed that there exists an eigenvalue $\dot{\lambda}$ of the EAOR iteration matrix $E_{v,r,\omega}$ such that

$$
\begin{aligned}
&\det\left(E_{v,r,\omega} - \dot{\lambda}I\right) = 0 \\
&\det([(I-(v+r)L]^{-1}[(1-\omega)I + [\omega-(v+r)]L + \omega U] - \lambda I) = 0 \\
&\det[I-(v+r)L]^{-1}\det[(1-\omega-\lambda)I + (\lambda r + \lambda v + \omega - v - r)L + \omega U] = 0 \\
&\det[I-(v+r)L]^{-1}\left[(\lambda+\omega-1)I - \left(\omega + v(\lambda-1) + r(\lambda-1)\right)L - \omega U\right] = 0 \\
&\frac{\det[(\lambda+\omega-1)I - [\omega + v(\lambda-1) + r(\lambda-1)]L - \omega U]}{det[I-(v+r)L]} = 0
\end{aligned}
\tag{3.59}
$$

But $det[(I-(v+r)L]$ is a determinant of a unit lower triangular matrix which is equals to one, hence the eigenvalue of the proposed EAOR iterative method are the $\dot{\lambda}$ roots of

$$\det[(\lambda+\omega-1)I - [\omega + v(\lambda-1) + r(\lambda-1)]L - \omega U] = 0 \tag{3.60}$$

and it is transformed into the expression;

$$\left(\dot{\lambda}+\omega-1\right)\det\left(I - \frac{\omega + r(\dot{\lambda}-1) + v(\dot{\lambda}-1)}{\lambda+\omega-1}L - \frac{\omega}{\dot{\lambda}+\omega-1}U\right) = 0 \tag{3.61}$$

From the assumption that $|\dot{\lambda}| \geq 1$, implying that $\dot{\lambda}+\omega-1 \neq 0$, hence

$$\det(R) = 0 \tag{3.62}$$

where R is given as

$$R = I - \frac{\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)}{\dot{\lambda}+\omega-1}L - \frac{\omega}{\dot{\lambda}+\omega-1}U \tag{3.63}$$

The modulus of the coefficients of L and U in (3.63) are less than one. To prove this, it is sufficient and necessary to prove that

$$\left|\dot{\lambda}+\omega-1\right| \geq \left|\omega + r\left(\dot{\lambda}-1\right) + v\left(\dot{\lambda}-1\right)\right| \quad \text{and} \quad \left|\dot{\lambda}+\omega-1\right| \geq |\omega| \tag{3.64}$$

Let $\dot{\lambda}^{-1} = qe^{i\theta}$, $\dot{\lambda} = \frac{1}{qe^{\iota\theta}} = q^{-1}e^{-\iota\theta}$ with $e^{-\iota\theta} = cos\theta - \iota s\iota n\theta$, then $\dot{\lambda} = q^{-1}[cos\theta - i\ sin\theta]$ where q and θ are real with $0 < q \leq 1$, then the first inequality in (3.64) is analyzed as follows;

$$\left.\begin{array}{c}
|\dot{\lambda}+\omega-1| \geq |\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)| \\
|\, q^{-1}\, e^{-i\theta}-1+\omega| \geq |\omega+\ (q^{-1}\, e^{-i\theta})r-r+(q^{-1}\, e^{-i\theta})v-v| \\
|\, q^{-1}\, e^{-i\theta}-1+\omega| \geq |\omega+\ (q^{-1}\, e^{-i\theta})r-r+(q^{-1}\, e^{-i\theta})v-v| \\
\left|\frac{cos\theta}{q}+\omega-1-i\left[\frac{sin\theta}{q}\right]\right| \geq \left|\omega+\ \frac{r}{q}cos\theta-i\left[\frac{r}{q}sin\theta\right]-r+\frac{v}{q}cos\theta-i\left[\frac{v}{q}sin\theta\right]-v\right| \\
\left[\left(\frac{cos\theta}{q}+\omega-1\right)^2+\left(\frac{sin\theta}{q}\right)^2\right]^{\frac{1}{2}} \geq \left[\left(\frac{r}{q}cos\theta+\frac{v}{q}cos\theta+\omega-r-v\right)^2+\left(\frac{r}{q}sin\theta+\frac{v}{q}sin\theta\right)^2\right]^{\frac{1}{2}} \\
|\dot{\lambda}+\omega-1|^2 \geq |\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)|^2 \\
=\left(\frac{cos\theta}{q}+\omega-1\right)^2+\left(\frac{sin\theta}{q}\right)^2-\left[\left(\frac{(v+r)}{q}cos\theta+\omega-r-v\right)^2+\left(\frac{(v+r)}{q}sin\theta\right)^2\right] \geq 0
\end{array}\right\} \quad (3.65)$$

which is simplified to give

$$1-2qcos\theta+2q\omega cos\theta-2q^2\dot{\omega}+q^2-r^2-v^2-2vq\omega cos\theta$$
$$-2rq\omega cos\theta-2vr+2v^2qcos\theta+4vrqcos\theta+2vq^2\omega$$
$$+2r\,q^2\omega+2qr^2cos\theta-2vrq^2-v^2q^2-q^2r^2 \geq 0 \qquad (3.66)$$

Rearrange to get

$$(1-v^2-r^2-2vr)+(1-v^2-r^2-2vr)q^2-(1-v^2-r^2-2vr)2qcos\theta$$
$$+(1-v-r)2q\omega cos\theta-(1-v-r)2q^2\omega \geq 0 \qquad (3.67)$$

$$=[1-(v+r)^2]+[1-(v+r)^2]q^2-[1-(v+r)^2]2qcos\theta$$
$$+[1-(v+r)]2q\omega cos\theta \quad -[1-(v+r)]2q^2\omega \geq 0 \qquad (3.68)$$

which holds for $v+r=1$; factorizing $[1-(v+r)]$ in the above equation {note: $[1-(v+r)^2]=[1-(v+r)][1+(v+r)]$} gives

$$\big[[1-(v+r)][1+(v+r)]+[1-(v+r)][1+(v+r)]q^2$$
$$-[1-(v+r)][1+(v+r)]2qcos\theta+[1-(v+r)]2q\omega cos\theta$$
$$-[1-(v+r)]2q^2\omega\big] \geq 0 \qquad (3.69)$$

$$[1-(v+r)][(1+v+r)+(1+v+r)q^2-(1+v+r)2qcos\theta+2q\omega cos\theta$$
$$-2q^2\omega] \geq 0 \qquad (3.70)$$

If $(1-v-r)=0$, then it is equivalent to

$$(1+v+r)+(1+v+r)q^2-[(1+v+r)-\omega]2qcos\theta-2q^2\omega \geq 0 \qquad (3.71)$$

Since the expression in the brackets above is nonnegative, (3.71) holds for all θ if and only if it holds for $cos\theta = 1$, where for $\theta = 2n\pi; n = 0,1,2,\ldots$ also due to the fact that $[(1+v+r)-\omega]2qcos\theta \leq [(1+v+r)-\omega]$, hence (3.71) is equivalent to

$$\begin{aligned} (1+v+r)+(1+v+r)q^2-\omega 2q-2q^2\omega &\geq 0 \\ (1+v+r)[1+q^2-2q]+2q\omega(1-q) &\geq 0 \\ (1+v+r)(1-q)^2+2q\omega(1-q) &\geq 0 \\ (1-q)[(1+v+r)(1-q)]+2q\omega &\geq 0 \end{aligned} \tag{3.72}$$

which is true. Similarly, the second inequality $|\dot{\lambda}+\omega-1| \geq |\omega|$ is analyzed as follows;

$$\begin{aligned} \left| q^{-1}e^{-i\theta}-1+\omega\right| &\geq |\omega| \\ \left|\frac{cos\theta}{q}+\omega-1-i\left[\frac{sin\theta}{q}\right]\right| &\geq |\omega| \\ |\dot{\lambda}+\omega-1| \geq |\omega| = \left[\left(\frac{cos\theta}{q}+\omega-1\right)^2+\left(\frac{sin\theta}{q}\right)^2\right]^{\frac{1}{2}} &\geq [\omega^2]^{\frac{1}{2}} \end{aligned} \tag{3.73}$$

If $|\dot{\lambda}+\omega-1| \geq |\omega|$, then it implies that $|\dot{\lambda}+\omega-1|^2 \geq |\omega|^2$, then

$$\begin{aligned} |\dot{\lambda}+\omega-1|^2 \geq |\omega|^2 = \left[\left(\frac{cos\theta}{q}+\omega-1\right)^2+\left(\frac{sin\theta}{q}\right)^2\right]-\omega^2 &\geq 0 \\ 1+2\omega q cos\theta-2qcos\theta-2q^2\omega+q^2 &\geq 0 \\ 1+q^2-2q(1-\omega)cos\theta-2\omega q^2 &\geq 0 \end{aligned} \tag{3.74}$$

which for same reason, must be satisfied for $cos\,\theta = 1$, for $\theta = 2n\pi; n = 0,1,2,\ldots$ and due to the fact that $2q(1-\omega)cos\theta \leq 2q(1-\omega)$, then one arrives at

$$\begin{aligned} 1+q^2-2q-2\omega q-2\omega q^2 &\geq 0 \\ (1-q)^2+2\omega q(1-q) &\geq 0 \\ (1-q)[1-q+2\omega q] &\geq 0 \end{aligned} \tag{3.75}$$

which holds for $q = 1$, thus the above analysis shows that

$$\left|\frac{\omega+r(\dot{\lambda}-1)+v(\dot{\lambda}-1)}{\dot{\lambda}+\omega-1}\right| < 1 \quad \text{and} \quad \left|\frac{\omega}{\dot{\lambda}+\omega-1}\right| < 1 \tag{3.76}$$

Given that A is irreducible and it has a weak diagonal dominance, it therefore means that $D^{-1}A = I-U-L$ equally contains the same properties too. Similarly, it is also true for the matrix R considering that 1 is greater than the modulus of the coefficients of L and U and that they are different from zero. Hence it implies that $\det(\mathrm{R}) \neq 0$ and thereby R is nonsingular, which contradicts the equation $[\det(R) = 0]$ and consequently, $\det\left(E_{v,r,\omega} - \dot{\lambda} I\right) = 0$. This implies that $|\dot{\lambda}| \geq 1$ does not hold which indicates that $|\dot{\lambda}| \leq 1$ and thereby $\rho(E_{v,r,\omega}) < 1$ which indicates that the EAOR method is convergent.

3.2.3 Convergence of $\mathbf{M}$ −matrix

Theorem 3.3: If matrix A is an M −matrix with $0 < v + r \leq \omega \leq 1$ and $\omega \neq 0$, then the Extended Parameterized Accelerated Over-Relaxation method converges to the true solution $z = A^{-1}b$ of the system $Az = b$ or simply $\rho\left(E_{\omega,v,r}\right) < 1$.

Proof:

Given that a square matrix A is an M matrix, let A be decomposed into $A = D - L_A - U_A$ with the regular splitting of $A = M - N$. In the EAOR iterative method, we have the splitting $A = M - N$ with the following choices of

$$M = \frac{1}{\omega}(D - vL_A - rL_A), \quad N = \frac{1}{\omega}\left((1 - \omega)D + (\omega - v - r)L_A + \omega U_A\right) \quad (3.77)$$

Such that

$$\omega A = M - N = [D - (v + r)L_A] - [(1 - \omega)D + (\omega - v - r)L_A + \omega U_A] \quad (3.78)$$

Obviously, it is observed that $A \leq M$ and as such, by implication of lemma 3.2, it suffices to say that matrix M is an M matrix too. Consequently, one obtains $M^{-1} \geq 0$.

Similarly, letting $D - (v + r)L_A$ to be a splitting of the matrix M, it is easily seen that D is a matrix that is nonsingular (det $D \neq 0$), implying that D is an M matrix and thus satisfy the condition $D^{-1} \geq 0$. Also, since $L_A \geq 0$, then it means that $(v + r)L_A \geq 0$ for $v + r \geq 0$ which signifies that the matrix $M = D - (v + r)L_A$ is considered an $M-$ splitting. Now, it is observed that $(v + r)L_A$ is a strict lower triangular matrix, with the fact that eigenvalues of a strict lower triangular matrices are the diagonal entries, then by implication, the eigenvalues of $(v + r)L_A$ are on the main diagonals in which they are all zeros.

Therefore, its spectral radius (largest moduli of its eigenvalues), $\rho\big((v + r)L_A\big) = 0$ and since zero is less than one, then it becomes $\rho\big((v + r)L_A\big) < 1$. Considering the fact that $\rho\big((v + r)L_A\big) < 1$, $M = D - (v + r)L_A$ is an $M-$ splitting and as such $\rho\big((v + r)D^{-1}L_A\big) < 1$. Lemma 3.3 is then employed to establish the fact that M is an $M-$ matrix and by definition of an $M-$ matrix, it follows that

$$M^{-1} = (D - (r + v)L_A)^{-1} \geq 0 \tag{3.79}$$

On the contrary, the matrix $N = [(1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A\,]$, with $L_A \geq 0$ and $U_A \geq 0$ and considering the following inequalities; $(1 - \omega) \geq 0$, $[\omega - (v + r)] \geq 0$, $\omega \geq 0$, gives the range of values $0 \leq v + r \leq \omega \leq 1$, $\omega \neq 0$ for the matrix N to be nonnegative, so we have

$$N = [(1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A\,] \geq 0 \qquad 3.80)$$

Then the matrix $M^{-1}N$ is analyzed as follows;

$$\begin{aligned} M^{-1}N &= [D - (v + r)L_A]^{-1} \times [(1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A\,] \\ &= \big(D + (v + r)L_A + (v + r)^2{L_A}^2 + (v + r)^3{L_A}^3 + \cdots + (v + r)^{N-1}{L_A}^{N-1}\big) \\ &\quad \times [(1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A\,] \end{aligned} \tag{3.81}$$

Which gives

$$(1-\omega)D+(1-\omega)(v+r)L_A+(1-\omega)(v+r)^2{L_A}^2+\cdots$$
$$+(1-\omega)(v+r)^{N-1}{L_A}^{N-1}+(\omega-v-r)L_A+(v+r)(\omega-v-r){L_A}^2$$
$$+(v+r)^2(\omega-v-r){L_A}^3+(v+r)^3(\omega-v-r){L_A}^4+\cdots$$
$$+(v+r)^{N-1}(\omega-v-r){L_A}^N+\omega U_A+\omega(v+r)L_AU_A$$
$$+\omega(v+r)^2{L_A}^2U_A+\omega(v+r)^3{L_A}^3U_A+\cdots+\omega(v+r)^{N-1}{L_A}^{N-1}$$
$$\geq 0 \qquad (3.82)$$

Multiply through equation (3.82) by D^{-1} and after letting $D^{-1}L_A=L,\ I=D^{-1}D$ and $D^{-1}U_A=U$, it becomes

$$(1-\omega)I+(1-\omega)(v+r)L+(1-\omega)(v+r)^2L^2+\cdots+(1-\omega)(v+r)^{N-1}L^{N-1}$$
$$+(\omega-v-r)L+(v+r)(\omega-v-r)L^2+(v+r)^2(\omega-v-r)L^3$$
$$+(v+r)^3(\omega-v-r)L^4+\cdots+(v+r)^{N-1}(\omega-v-r)L^N+\omega U$$
$$\geq 0 \qquad (3.83a)$$

Therefore the iteration matrix becomes

$$M^{-1}N=\sum_{k=0}^{\infty}(I-(r+v)L)^{-1}\left[(1-\omega)I+\big(\omega-(r+v)\big)L+\omega U\right]\geq 0 \qquad (3.83b)$$

The matrix $M^{-1}N$ is nonnegative, therefore $\omega A=M-N$ is obviously a weak regular splitting of matrix ωA. And in view of lemma 3.3, then it means $\rho(M^{-1}N)<1$ or equivalently

$\rho((I-(r+v)L)^{-1}[(1-\omega)\mathrm{I}+(\omega-r-v)L+\omega U])<1$, which completes the proof.

3.3: Conditions on the Coefficient Matrix for the Proposed Method

From the convergence analysis in the previous chapter, the conditions placed on the coefficient matrices with regards to the proposed EAOR method before convergence can be achieved are as follows;

I. For $L-$ matrix, the condition is $0 < v + r \leq \omega \leq 1,\ \omega \neq 0$ and $v \neq 0.$

II. For $M-$ matrix, the conditions placed on it are given as $0 < v + r \leq 1$, and $0 < \omega \leq 1,\ \ v \neq 0$ and $\omega \neq 0.$

III. For Irreducible weak diagonally dominant matrix, the condition placed on it is given as $0 < v + r \leq 1$ and $0 < \omega \leq 1,\ v \neq 0.$

3.4 Derivation of Refinement of EAOR Method

Having in mind that significant improvements of any iteration matrix will decrease the spectral radius and enhances the rate of convergence of a stationary iterative method, therefore, this section seeks to derive the Refinement of the proposed EAOR method so as to accelerate the convergence rate of the EAOR iterative method. Derivation of the Refinement method in matrix form will be described below:

Considering the linear system $Az = b$ in m linear equations and n unknowns, where matrix A is nonsingular. Its solution is $z = A^{-1}b$ and for a vector z, the residual represented by $r = r(z)$ of $Az = b$ is given as;

$$r = b - Az \tag{3.84}$$

By the usual splitting $A = D - L_A - U_A$ in (3.5) with the EAOR regular splitting;

$$A = \frac{1}{\omega}[M - N] \quad \xrightarrow{yields} \quad \omega A = M - N \quad \text{or} \quad N = M - \omega A \tag{3.85}$$

Where the choices for M and N of the proposed EAOR method are represented as

$$M = [D - (v + r)L_A], \quad N = (1 - \omega)D + [\omega - (v + r)]L_A + \omega U_A \tag{3.86}$$

And multiplying the linear system $Az = b$ by the parameter ω to obtain

$$\omega(Az) = \omega(b) \tag{3.87}$$

Then the proposed Refinement method is derived as follows;

$$\begin{gathered}
Az = b \\
\omega Az = \omega b \\
[M - N]z = \omega b \\
\big[[D - (v+r)L_A] - [(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]\big]z = \omega b \\
[D - (v+r)L_A]z = [(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z + \omega b \\
[D - (v+r)L_A]z = [D - (v+r)L_A]z + \omega(b - Az) \\
z = [D - (v+r)L_A]^{-1}[D - (v+r)L_A]z + [D - (v+r)L_A]^{-1}\omega(b - Az) \\
z = z + \omega[D - (v+r)L_A]^{-1}(b - Az)
\end{gathered} \tag{3.88}$$

The proposed EAOR Refinement formula takes the form

$$\bar{z}^{(k+1)} = z^{(k+1)} + \omega[D - (v+r)L_A]^{-1}\big(b - Az^{(k+1)}\big) \tag{3.89}$$

where $z^{(k+1)}$ appearing in the right hand side, is $(k+1)^{th}$ estimation of the proposed EAOR iterative method. The EAOR method in (3.27) is inserted into (3.89) to obtain

$$\begin{aligned}
\bar{z}^{(k+1)} = {} & [D - (v+r)L_A]^{-1}[(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z^{(k)} \\
& + [D - (v+r)L_A]^{-1}\omega b \ + [D - (v+r)L_A]^{-1}\omega b \\
& - [D - (v+r)L_A]^{-1}\omega A \\
& \times \big[[D - (v+r)L_A]^{-1}[(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z^{(k)} \\
& + [[D - (v+r)L_A]^{-1}\omega b \,\big]
\end{aligned} \tag{3.90}$$

The value of ωA is substituted into (3.90) to get

$$\begin{aligned}
\bar{z}^{(k+1)} = {} & [D - (v+r)L_A]^{-1}[(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z^{(k)} \\
& + [[D - (v+r)L_A]^{-1}\omega b \ + [D - (v+r)L_A]^{-1}\omega b \\
& - [D - (v+r)L_A]^{-1} \\
& \times \big([D - (v+r)L_A] - [(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A] \\
& \times \big[[D - (v+r)L_A]^{-1}[(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z^{(k)} \\
& + [D - (v+r)L_A]^{-1}\omega b \ + [D - (v+r)L_A]^{-1}\omega b\big]\big)
\end{aligned} \tag{3.91}$$

$$\bar{z}^{(k+1)} = [D - (v+r)L_A]^{-1}[(1-\omega)D + [\omega - (v+r)]L_A + \omega U_A]z^{(k)}$$

$$+2[D-(v+r)L_A]^{-1}\omega b-[D-(v+r)L_A]^{-1}\times H \qquad (3.92)$$

where

$$H=([D-(v+r)L_A]-[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A])$$

$$\times\big([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$+[[D-(v+r)L_A]^{-1}\omega b\big) \qquad (3.93)$$

Simplifying H gives

$$H=[D-(v+r)L_A]$$

$$\times\big([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$+[D-(v+r)L_A]^{-1}\omega b\big)-[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]$$

$$\times[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$+[D-(v+r)L_A]^{-1}\omega b \qquad (3.94)$$

$$=[D-(v+r)L_A]\times[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$+[D-(v+r)L_A]\times[D-(v+r)L_A]^{-1}\omega b$$

$$-[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]$$

$$\times[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$-[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]$$

$$\times[D-(v+r)L_A]^{-1}\omega b \qquad (3.95)$$

$$H=[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}+\omega b$$

$$-([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A])^2z^{[k]}$$

$$-[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]\omega b \qquad (3.96)$$

Substituting (3.96) into (3.92) gives

$$\bar{z}^{(k+1)} = [D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}$$

$$+2[D-(v+r)L_A]^{-1}\omega b-[D-(v+r)L_A]^{-1}$$

$$\times[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}+\omega b$$

$$-([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A])^2 z^{(k)}$$

$$-[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]\omega b \quad (3.97)$$

$$= [D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)} +2[D-(v+r)L_A]^{-1}\omega b-[D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]z^{(k)}-[D-(v+r)L_A]^{-1}\omega b+[D-(v+r)L_A]^{-1}\big([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]^2 z^{(k)}\big)+[D-(v+r)L_A]^{-1}\times([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A]\omega b) \quad (3.98)$$

Rearrange (3.98) to get

$$\bar{z}^{(k+1)} = \Big((D-(v+r)L_A)^{-1}\big((1-\omega)D+[\omega-(v+r)]L_A+\omega U_A\big)\Big)^2 z^{(k)} + \Big(I+(D-(v+r)L_A)^{-1}\big((1-\omega)D+[\omega-(v+r)]L_A+\omega U_A\big)\Big)(D-(v+r)L_A)^{-1}\omega b \quad (3.99)$$

Hence the method (3.99) shall be called Refinement of Extended Accelerated Over-Relaxation (REAOR) method. Alternatively, the Refinement of proposed EAOR iterative method can be derived as follows;

$$\begin{aligned}
\bar{z}^{(k+1)} &= z^{(k+1)}+\omega[D-(v+r)L_A]^{-1}\big(b-Az^{(k+1)}\big)\\
&= z^{(k+1)}+\omega M^{-1}\big(b-Az^{(k+1)}\big)\\
&= z^{(k+1)}+\omega M^{-1}b-M^{-1}\omega A z^{(k+1)}\\
&= z^{(k+1)}+M^{-1}\omega b-M^{-1}(M-N)z^{(k+1)}\\
&= z^{(k+1)}+M^{-1}\omega b-M^{-1}Mz^{(k+1)}+M^{-1}Nz^{(k+1)}\\
&= z^{(k+1)}+M^{-1}\omega b-z^{(k+1)}+M^{-1}Nz^{(k+1)}\\
&= M^{-1}\omega b+M^{-1}Nz^{(k+1)}
\end{aligned}$$

$$\bar{z}^{(k+1)} = M^{-1}\omega b+M^{-1}Nz^{(k+1)} \quad (3.100)$$

But $z^{(k+1)} = M^{-1}Nz^{(k+1)}+M^{-1}\omega b$, so substituting it in (3.100) gives

$$\begin{aligned}\bar{z}^{(k+1)} &= M^{-1}\omega b + M^{-1}N\big(M^{-1}Nz^{(k)} + M^{-1}\omega b\big)\\ &= M^{-1}\omega b + (M^{-1}N)^2 z^{(k)} + (M^{-1})^2 N\omega b\\ &= (M^{-1}N)^2 z^{(k)} + (I + M^{-1}N)M^{-1}\omega b\end{aligned} \tag{3.101}$$

Substituting the values of M and N into (3.101) to obtain

$$\bar{z}^{(k+1)} = \Big((D-(v+r)L_A)^{-1}\big((1-\omega)D + [\omega-(v+r)]L_A + \omega U_A\big)\Big)^2 z^{(k)} + \Big(I + (D-(v+r)L_A)^{-1}\big((1-\omega)D + [\omega-(v+r)]L_A + \omega U_A\big)\Big)(D - (v+r)L_A)^{-1}\omega b \quad k = 0,1,2,\cdots \tag{3.102}$$

Or equivalently,

$$\bar{z}^{(k+1)} = \Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2 z^{(k)} + \Big(I + (I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)(I - (v+r)L)^{-1}\omega\dot{b}, \qquad k = 0,1,2,\cdots \tag{3.103}$$

By setting $L = D^{-1}L_A,\ U = D^{-1}U_A,\ I = D^{-1}D$ and $\dot{b} = D^{-1}b$, or in a more compact form;

$$\bar{z}^{(k+1)} = J_{REAOR}z^{(k)} + F, \qquad (\,k = 0,1,2,\cdots) \tag{3.104}$$

where $J_{REAOR} = \Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2 = E_{v,\omega,r}^2$ is called the REAOR iteration matrix and $F = \omega\big(I + E_{v,\omega,r}\big)[I-(v+r)L]^{-1}\dot{b}$ is the corresponding vector of the refined EAOR method. Comparing methods of REAOR and EAOR, there is a relationship between REAOR iteration matrix and EAOR iteration matrix (3.29). If $E_{v,\omega,r}$ represents the iteration matrix of EAOR method and $RE_{v,\omega,r}$ denotes the iteration matrix of REAOR method, then the iteration matrix of REAOR method is the square of the iteration matrix of EAOR method, that is to say $RE_{v,r,\omega} = \big(E_{v,\omega,r}\big)^2$. The spectral radius of the refinement of proposed Extended Accelerated Over-relaxation (REAOR) iterative method is the largest eigenvalue of its iteration matrix denoted as $\rho\big(RE_{v,\omega,r}\big)$ with the relationship

$$\rho(RE_{v,r,\omega}) = [\rho(E_{v,\omega,r})]^2 \quad (3.105)$$

It is observed that for some specific values of the parameters v, r and ω, the proposed REAOR method produces some refined iterative methods and they are shown in the following analysis;

$RE_{0,0,1} -$ method is the Refinement of Jacobi (RJ) method: Suppose the matrix A of the linear system $Az = b$ is splitted into $A = D - L - U$, then the Refinement of Jacobi (RJ) method is denoted as

$$\bar{z}^{(k+1)} = \left(D^{-1}(L+U)\right)^2 z^{(k)} + \left(I + D^{-1}(L+U)\right)D^{-1}b \quad (3.106)$$

With the proposed Refinement of EAOR method $RE_{v,r,\omega}$ expressed as

$$\bar{z}^{(k+1)} = \left((D-(v+r)L)^{-1}\left((1-\omega)D + [\omega-(v+r)]L + \omega U\right)\right)^2 z^{(k)} + \left(I + (D-(v+r)L)^{-1}\left((1-\omega)D + [\omega-(v+r)]L + \omega U\right)\right)(D - (v+r)L)^{-1}\omega b \quad (3.107)$$

Inserting values of $v = 0$, $r = 0$ and $\omega = 1$ for $RE_{0,0,1}$ into the above equation gives

$$\bar{z}^{(k+1)} = \left((D-(0+0)L)^{-1}\left((1-1)D + [1-(0+0)]L + 1.U\right)\right)^2 z^{(k)} + \left(I + (D-(0+0)L)^{-1}\left((1-1)D + [1-(0+0)]L + 1.U\right)\right)(D - (0+0)L)^{-1}1.b \quad (3.108)$$

$\bar{z}^{(k+1)} = \left((D)^{-1}(L+U)\right)^2 z^{(k)} + \left(I + (D)^{-1}(L+U)\right)(D)^{-1}\omega b$ is obtained from the above substitution. This indicates that REAOR method reduces to RJ iterative method.

$RE_{0,1,1} -$ method is the Refinement of Gauss-Seidel (RGS) method: The Refinement of the Gauss-Seidel is represented as

$$\bar{z}^{(k+1)} = ((D-L)^{-1}U)^2 z^{(k)} + (I + (D-L)^{-1}U)(D-L)^{-1}b \quad (3.109)$$

Values of $v = 0$, $r = 1$ and $\omega = 1$ for $RE_{0,1,1}$ are inserted in (3.107) to obtain

$$\bar{z}^{(k+1)} = \left((D-(0+1)L)^{-1}\left((1-1)D + [1-(0+1)]L + 1.U\right)\right)^2 z^{(k)} +$$

$$\left(I+(D-(0+1)L)^{-1}\left((1-1)D+[1-(0+1)]L+1.U\right)\right)(D$$

$$-(0+1)L)^{-1}1.b \qquad (3.110)$$

which results into $\bar{z}^{(k+1)}=\left((D-L)^{-1}(U)\right)^2 z^{[k]}+\left(I+(D-L)^{-1}(U)\right)(D-L)^{-1}b$, indicating that the new REAOR iterative method reduces to the RSOR method.

$RE_{0,\omega,\omega}-$ method is the Refinement of Successive Over-Relaxation (RSOR) method: Method of the RSOR is expressed as

$$\bar{z}^{(k+1)}=\left((D-\omega L)^{-1}\left((1-\omega)D+\omega U\right)\right)^2 z^{(k)}$$

$$+\left(I+(D-\omega L)^{-1}\left((1-\omega)D+\omega U\right)\right)(D-\omega L)^{-1}\omega b \qquad (3.111)$$

With values of $v=0,\ r=\omega$ and $\omega=\omega$ for $RE_{0,\omega,\omega}$ substituted into (3.107), it gives

$$\bar{z}^{(k+1)}=\left((D-(0+\omega)L)^{-1}\left((1-\omega)D+[\omega-(0+\omega)]L+\omega U\right)\right)^2 z^{(k)}+$$

$$\left(I+(D-(0+\omega)L)^{-1}\left((1-\omega)D+[\omega-(0+\omega)]L+\omega U\right)\right)(D-(0+$$

$$\omega)L)^{-1}\omega b \qquad (3.112)$$

Resulting into the RSOR $\bar{z}^{(k+1)}=\left((D-(\omega)L)^{-1}\left((1-\omega)D+\omega U\right)\right)^2 z^{[k]}+\left(I+(D-(\omega)L)^{-1}\left((1-\omega)D+\omega U\right)\right)(D-(\omega)L)^{-1}\omega b$, which signifies that the proposed REAOR iterative method can be reduce to RSOR method.

$RE_{0,r,\omega}-$method is the Refinement of Accelerated Over-Relaxation (RAOR) method: And the RAOR iterative method is given as

$$\bar{z}^{(k+1)}=\left((D-rL)^{-1}\left((1-\omega)D+(\omega-r)L+\omega U\right)\right)^2 z^{(k)}$$

$$+\left(I+(D-rL)^{-1}\left((1-\omega)D+(\omega-r)L+\omega U\right)\right)(D-rL)^{-1}\omega b \quad (3.113)$$

Substituting $v=0,\ r=r$ and $\omega=\omega$ for $RE_{0,\omega,\omega}$ in (3.107) to obtain

$$\bar{z}^{(k+1)}=\left((D-(0+r)L)^{-1}\left((1-\omega)D+[\omega-(0+r)]L+\omega U\right)\right)^2 z^{(k)}+$$

$$\left(I + (D - (0+r)L)^{-1}\left((1-\omega)D + [\omega - (0+r)]L + \omega U\right)\right)(D - (0+r)L)^{-1}\omega b \qquad (3.114)$$

After the substitution, one arrives at $\bar{z}^{(k+1)} = \left((D-(r)L)^{-1}\left((1-\omega)D + [\omega-(r)]L + \omega U\right)\right)^2 z^{(k)} + \left(I + (D-(r)L)^{-1}\left((1-\omega)D + [\omega-(r)]L + \omega U\right)\right)(D - (r)L)^{-1}\omega b$ which shows that the proposed REAOR method can reduce to RAOR iterative method.

3.5 Convergence of Refinement of EAOR method

Lemma 3.4 (Varga, 2000): Let z be a vector in a set of $m-$ dimensional column vectors $\mathbb{R}^m$, with real number components, then the sequence $\left[z^{(k)}\right]_{k=0}^{\infty}$ converges to z with respect to the infinity norm $\|.\|_\infty$ if and only if $\lim k \to \infty\ z_i^k = z_i$ for each $i = 1, 2, \cdots k$ or for any norm $\|.\|_\infty$, then $\lim_{k\to\infty}\|z^k - z\|_\infty = 0$

Lemma 3.5 (Edalatpanam and Najafa, 2013): Let z and y be vectors of $\mathbb{R}^m$ in a set of real numbers $\mathbb{R}$, then

I. $\|z\| > 0$ for all $z \in \mathbb{R}^m$

II. $\|z\| = 0$ if and only if $z = 0$

III. If β is a scalar, then $\|\beta z\| = |\beta|\|z\|$

IV. $\|z+y\| \leq \|z\| + \|y\|$

Lemma 3.6 (Martins *et al.*, 2012): Let J be an iteration matrix of any iterative method, if the norm of matrix J is less than one, that is $\|J\| < 1$, then the sequence z^k converges to z for any initial estimation $z^{(0)}$ and $\|z - z^k\| \leq \|J\|^k\|z - z^0\|$.

Theorem 3.4: If matrix A is an $L-$matrix, then the Refinement of the proposed Extended Accelerated Over-relaxation (REAOR) method converges to the exact solution for any initial guess $z^{(0)}$.

Proof: Assuming Z is the true solution of $Az = b$, since the coefficient matrix A is an $L-$matrix, it follows from theorem 3.1 that the EAOR method is convergent, thereby one can close $z^{(k+1)}$ to Z. Let $z^{(k+1)} \to Z$ and suppose $\bar{z}^{(k+1)}$ is the $(k+1)^{th}$ estimation to solution of $Az = b$ by REAOR method $\bar{z}^{(k+1)} = z^{(k+1)} + \omega[D-(v+r)L_A]^{-1}\left(b - Az^{(k+1)}\right)$, then applying lemma 3.5 to the REAOR approximations gives;

$$\left.\begin{aligned} \left\|\bar{z}^{(k+1)} - Z\right\|_\infty &= \left\|z^{(k+1)} + \omega[D-(v+r)L_A]^{-1}\left(b - Az^{(k+1)}\right) - Z\right\|_\infty \\ &= \left\|z^{(k+1)} - Z + \omega[D-(v+r)L_A]^{-1}\left(b - Az^{(k+1)}\right)\right\|_\infty \\ &\le \left\|z^{(k+1)} - Z\right\|_\infty + \|\omega[D-(v+r)L_A]^{-1}\|_\infty \left\|\left(b - Az^{(k+1)}\right)\right\|_\infty \\ &\le \left\|z^{(k+1)} - Z\right\|_\infty + \|\omega[D-(v+r)L_A]^{-1}\|_\infty \left\|\left(b - Az^{(k+1)}\right)\right\|_\infty \end{aligned}\right\} \quad (3.115)$$

Next is to analyze each terms of the right hand side. From theorem 3.1, it is observed that $z^{(k+1)} \to Z$ describes the convergence condition of the EAOR method to the real solution and taking the infinity norm of it as k tends to zero guarantees convergence. Therefore, this implies that the following expression holds

$$\left\|z^{(k+1)} - Z\right\|_\infty = \|Z - Z\|_\infty = 0 \qquad (3.116)$$

That is to say for every improvement in $z^{(k+1)}$, there is probable value of it to become z. Likewise for the case of $\left\|\left(b - Az^{(k+1)}\right)\right\|$, there will be an equivalent corresponding improvement in $Az^{(k+1)}$ to become Az as k tends to zero and since $Az \equiv b$, then

$$\left\|\left(b - Az^{(k+1)}\right)\right\|_\infty = \|(b - Az)\|_\infty = \|(b - b)\|_\infty = 0 \qquad (3.117)$$

Evidently, this implies that $\left\|\left(b - Az^{(k+1)}\right)\right\|_\infty \to 0$. It is clearly seen that $|\omega|\|[D-(v+r)L_A]^{-1}\|_\infty$ vanishes since $\left\|\left(b - Az^{(k+1)}\right)\right\|$ tends to zero as k tends to infinity. Also, since $\left\|z^{(k+1)} - Z\right\|_\infty$ and $\left\|\left(b - Az^{(k+1)}\right)\right\|_\infty$ tends to zero, such that

$$\left\|\bar{z}^{[k+1]} - Z\right\|_\infty = 0 + |\omega|\|[D-(v+r)L_A]^{-1}\|_\infty \times 0 = 0 + 0 = 0 \qquad (3.118)$$

Hence by lemma 3.4, it is sufficient to deduce that $\bar{z}^{(k+1)} \to Z$ tends to zero as $k \to \infty$, which can be written as

$$\left\|\bar{z}^{(k+1)} - Z\right\|_{\infty} \to 0 \tag{3.119}$$

From (3.116) and (3.117), it is obvious that (3.119) holds and this implies that $\bar{z}^{[k+1]} \to z$ and thus $\rho(RT_{v,r,\omega}) = \left[\rho(T_{v,\omega,r})\right]^2 < 1$ or equivalently

$$\rho\left((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\right)^2$$

$$= \left[\rho\left((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\right)\right]^2 < 1 \tag{3.120}$$

The REAOR iterative method converges to the solution of the linear system $Az = b$. As a result, the new REAOR method is convergent for L matrix, which completes the proof.

Theorem 3.5: If the coefficient matrix A is an irreducible diagonally dominant, then for any choice of initial guess $z^{(0)}$, the proposed REAOR converges to the true solution .

Proof:

Let Z be the real exact solution of $Az = b$. Given that $A = (a_{ij})$ is irreducible diagonally dominant. Then the proposed EAOR method is convergent by theorem 3.2 and so let $z^{(k+1)}$ converges to Z when

$z^{(k+1)} = [D-(v+r)L_A]^{-1}[(1-\omega)D + [\omega-(v+r)]L_A + \omega U_A]z^{(k)} + [D-(v+r)L_A]^{-1}\omega b$. Then, $\bar{z}^{(k+1)} = z^{(k+1)} + \omega[D-(v+r)L_A]^{-1}\big(b - Az^{(k+1)}\big)$ or

$$\bar{z}^{(k+1)} - Z = z^{(k+1)} + \omega[D-(v+r)L_A]^{-1}\big(b - Az^{(k+1)}\big) - Z \tag{3.121}$$

Hence taking norm of both sides gives

$$\left.\begin{aligned}\left\|\bar{z}^{(k+1)} - Z\right\|_{\infty} &= \left\|z^{(k+1)} - Z + \omega[D-(v+r)L_A]^{-1}\big(b - Az^{(k+1)}\big)\right\|_{\infty} \\ &\le \left\|z^{(k+1)} - Z\right\|_{\infty} + \|\omega[D-(v+r)L_A]^{-1}\|_{\infty}\left\|\big(b - Az^{(k+1)}\big)\right\|_{\infty} \\ &\le \left\|z^{(k+1)} - Z\right\|_{\infty} + \|\omega[D-(v+r)L_A]^{-1}\|_{\infty}\left\|\big(b - Az^{(k+1)}\big)\right\|_{\infty} \\ &= 0 + \|\omega[D-(v+r)L_A]^{-1}\|_{\infty}\,\|(b-b)\|_{\infty} = 0 + 0 = 0\end{aligned}\right\} \tag{3.122}$$

$\left\|\bar{z}^{(k+1)} - Z\right\| = 0$ Hence, $\bar{z}^{(k+1)}$ converges to Z and this implies that

$$\rho([D-(v+r)L_A]^{-1}[(1-\omega)D+[\omega-(v+r)]L_A+\omega U_A])^2 < 1 \qquad (3.123)$$

Therefore, the proposed REAOR iterative method is convergent for matrices that are irreducible diagonally dominant and this completes the proof.

Theorem 3.6: Let a square matrix $A = (a_{ij})$ be an $M-$ matrix, a matrix A whose off diagonal entries are non-positive having positive diagonal entries such that A is a non-singular matrix with $A^{-1} \geq 0$. Then for any arbitrary initial approximation $z^{(0)}$, the proposed REAOR method is convergent.

Proof:

If Z is the real solution of $Az = b$ and since A is an $M-$ matrix, the EAOR iterative method is convergent as obtained in theorem 3.3. Similar procedure of theorem 3.4 is employed to prove that the proposed REAOR iterative method is convergent for M matrices. Next is to use the spectral radius of REAOR iterative method to show convergence of the method. Suppose A is an $M-$matrix, then the spectral radius of the proposed EAOR method is less than 1. It is observed from theorem 3.3, that the spectral radius of the EAOR method is $\rho\Big((I-(v+r)L)^{-1}\big((1-\omega)I+[\omega-(v+r)]L+\omega U\big)\Big) < 1$. Now, since the spectral radius of the proposed REAOR method is the square of the EAOR spectral radius, this implies that the spectral radius of the REAOR will also be less than one by the relation

$$\rho\Big((I-(v+r)L)^{-1}\big((1-\omega)I+[\omega-(v+r)]L+\omega U\big)\Big)^2$$

$$= \left[\rho\Big((I-(v+r)L)^{-1}\big((1-\omega)I+[\omega-(v+r)]L+\omega U\big)\Big)\right]^2 < 1 \qquad (3.124)$$

The spectral radius of the proposed Refinement of EAOR is less than 1 and this shows that the proposed REAOR method is convergent.

Theorem 3.7: For any initial guess $z^{(0)}$, the refinement of Extended Accelerated Over-relaxation (REAOR) method converges faster than the EAOR method to the real solution, whenever both methods converges.

Proof:

Let the EAOR method be represented as $z^{(k+1)} = Jz^{(k)} + C$ and the REAOR method be denoted as $z^{(k+1)} = J^2 z^{(k)} + D$ where

$$\left.\begin{array}{l} J = [I - vL - rL]^{-1}[(1-\omega)I + (\omega - v - r)L + \omega U] \\ C = [I - (v+r)L]^{-1}\omega\dot{b} \\ D = (I+J)[I - (v+r)L]^{-1}\omega\dot{b} \end{array}\right\} \tag{3.125}$$

Given that the norm of J is less than one ($\|J\| < 1$) for convergence, suppose Z is the real solution of $Az = b$ which satisfies $z^{(k+1)} = Jz^{(k)} + C$, then it implies that $Z = JZ + C$ with respect to EAOR method and similarly $Z = J^2 Z + D$ also satisfies the equation $z^{(k+1)} = J^2 z^{(k)} + D$ with respect to REAOR method. And let $k = 0,1,2,3, \dots$ be nonnegative integer.

If we consider the Proposed EAOR method, then

$z^{(k+1)} = Jz^{(k)} + C \Longrightarrow z^{(k+1)} - Z = Jz^{(k)} + C - Z$, which is analyzed as follows

$$\begin{aligned} z^{(k+1)} - Z &= Jz^{(k)} - JZ + C - Z + JZ \\ &= J\left(z^{(k)} - Z\right) + Z - Z \\ z^{(k+1)} - Z &= J(z^{(k)} - Z) \end{aligned} \tag{3.126}$$

Taking the norm of the expression $z^{(k+1)} - Z = J\left(z^{(k)} - Z\right)$ and applying lemma 3.5 to the expression results into

$$
\begin{aligned}
\left\|z^{(k+1)} - Z\right\| &= \left\|J\left(z^{(k)} - Z\right)\right\| \\
&\leq \|J\| \|(z^{(k)} - Z)\| \\
&\leq \|J^2\| \|(z^{(k-1)} - Z)\| \\
&\leq \|J^3\| \|(z^{(k-2)} - Z)\| \\
&\vdots \\
&\leq \|J^k\| \|(z^{(1)} - Z)\| \\
\|z^{(k+1)} - Z\| &\leq \|J\|^k \|(z^{(1)} - Z)\|
\end{aligned} \tag{3.127}
$$

From the inequality $\left\|z^{(k+1)} - Z\right\| \leq \|J\|^k \left\|\left(z^{(1)} - Z\right)\right\|$, if $\|J\|^k < 1$, then it results into $z^{(k+1)} \to Z$ as $k \to \infty$ by lemma 3.6. Next, let us consider the proposed Refinement of EAOR method:

$z^{(k+1)} = J^2 z^{(k)} + D \Longrightarrow z^{(k+1)} - Z = J^2 z^{(k)} - Z + D$, analyze as

$$
\begin{aligned}
z^{(k+1)} - Z &= J^2 z^{(k)} + D - J^2 Z - Z + J^2 Z \\
&= J^2\left(z^{(k)} - Z\right) - Z + Z \\
z^{(k+1)} - Z &= J^2 (z^{(k)} - Z)
\end{aligned} \tag{3.128}
$$

Again, taking the norm of $z^{(k+1)} - Z = J^2\left(z^{(k)} - Z\right)$ and applying lemma 3.5 to the expression gives;

$$
\begin{aligned}
\left\|z^{(k+1)} - Z\right\| &= \left\|J^2\left(z^{(k)} - Z\right)\right\| \\
&\leq \|J^2\| \|(z^{(k)} - Z)\| \\
&\leq \|J^4\| \|(z^{(k-1)} - Z)\| \\
&\leq \|J^6\| \|(z^{(k-2)} - Z)\| \\
&\vdots \\
&\leq \|J^{2k}\| \|(z^{(1)} - Z)\| \\
\|z^{(k+1)} - Z\| &\leq \|J\|^{2k} \|(z^{(1)} - Z)\|
\end{aligned} \tag{3.129}
$$

If $\|J\|^{2k} < 1$, then $z^{(k+1)} \to Z$ as $k \to \infty$. According to the coefficients of the above inequalities, we have

$$
\|J\|^{2k} < \|J^k\| \quad \text{since} \quad \|J\| < 1 \tag{3.130}
$$

This indicates that $\|J\|^{2k} < 1$ which also implies convergence of the proposed REAOR. Therefore, the above analysis indicates that the REAOR method converges faster than EAOR method and the proof is completed.

3.6 Algorithms for Numerical Computations

3.6.1 Algorithm for EAOR method

To solve

$$Az = b \quad \text{or} \quad (I - L - U)z = b$$

Step 0: Input the entries a_{ij} and b_i; $1 \le i, j \le n$ of the matrix A and b respectively

Input $v,\ r,\ \omega,\ L,\ U,$ and I

Step 1: Choose an initial guess $z_i^{(0)} = 0$ for $k = 0, 1, 2, \dots, k_{max}$ and for $i =$ $1,2,3, \dots, N$

Where $i = 1,2,3, \dots, N$ refers to number of unknowns,

$k = 0,1,2, \dots, k_{max}$ refers to the number of iterations

Step 2: Set $S = (I - vL - rL)^{-1}\big((1 - \omega)I + (\omega - v - r)L + \omega U\big)$

Set $P = (I - vL - rL)^{-1}(\omega b)$

Step 3: for $i = 0,1,2, \dots, n$, then,

compute $z_i^{(k+1)} = Sz_i^{(k)} + P$

If $\left\|z - z_i^{(0)}\right\| <$ TOL, output $(z_1, z_2, \dots, z_n)$

Step 4: Update $k = k + 1$

Step 5: for $k = 0, 1, 2, \dots, n$, then,

output ("maximum number of iterations exceeded")

STOP

3.6.2 Algorithm for REAOR method

To solve

$$Az = b \quad \text{or} \quad (I - L - U)z = b$$

Step 0: Input the entries a_{ij} and b_i; $1 \le i, j \le n$ of the matrix A and b respectively

Input $v,\ r,\ \omega,\ L,\ U,$ and I

Step 1: Choose an initial guess $Z_i^{(0)} = 0$ for $k = 0,1,2,\dots,k_{max}$ and for $i = 1,2,3,\dots,N$

Where $i = 1,2,3,\dots,N$ refers to number of unknowns,

$k = 0, 1, 2, \dots, k_{max}$ ref ers to the number of iterations

Step 2: Set $S1 = (I-(v+r)L)^{-1}\Big((1-\omega)I + \big(\omega-(v+r)\big)L + \omega U\Big), S = (S1)^2$

Set $P = (I+S1)(I-(v+r)L)^{-1}(\omega b)$

Step 3: for $i = 0,1,2,\dots,n$, then,

compute $z_i^{(k+1)} = S z_i^{(k)} + P$

If $\left\|z - z_i^{(0)}\right\| <$ TOL, output $(z_1 z_2, \dots, z_n)$

Step 4: Update $k = k+1$

Step 5: for $k = 0, 1, 2, \dots, n$, then,

output (“maximum number of iterations exceeded”)

STOP

3.7 Numerical Experiments

3.7.1 Problem 1

Consider the second-order partial differential equation (Laplace equation) in the form

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \qquad (3.131)$$

From Vatti (2016) for the square mesh with the boundary values shown in figure (3.1)

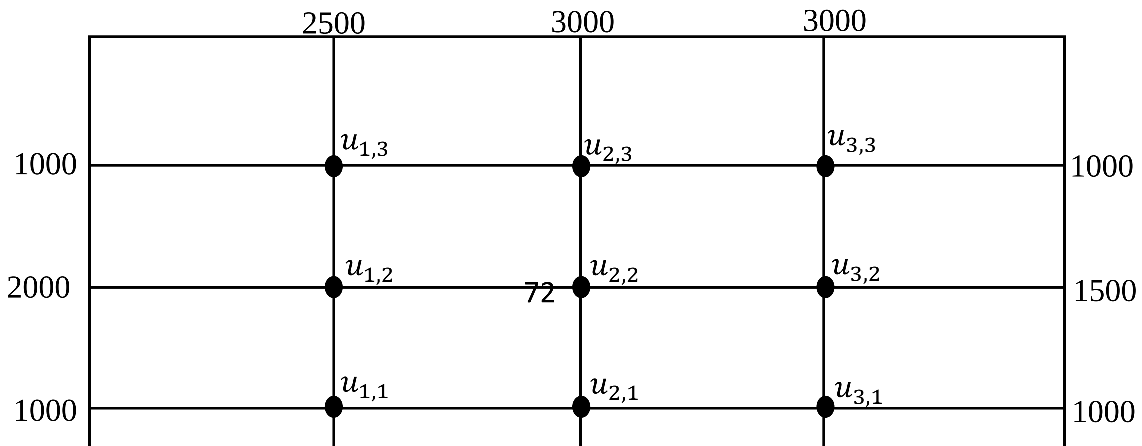

Figure 3.2: Discretization of square mesh with boundary values

From figure 3.2, it is observed that there are nine internal points, $u_{1,1}, u_{2,1}, u_{3,1}$, $u_{1,2}, u_{2,2}, u_{3,2}$, $u_{3,1}, u_{3,2}, u_{3,3}$ and all the boundary points are known. Application of method of finite differences is used to solve the second order partial differential equation in (3.131), At each interior point of the region, the partial derivatives $\frac{\partial^2 U}{\partial x^2}$ and $\frac{\partial^2 U}{\partial y^2}$ appearing in equation (3.131) are substituted by the standard second order three-point central difference quotients represented as

$$\frac{\partial^2 U}{\partial x^2} = \frac{U_{n+1,m} - 2U_{n,m} + U_{n-1,m}}{h^2}$$
$$\frac{\partial^2 U}{\partial y^2} = \frac{U_{n,m+1} - 2U_{n,m} + U_{n,m-1}}{k^2} \qquad (3.132)$$

Thus the central finite difference approximation to equation (3.131) at each interior grid point is denoted as

$$\frac{U_{n+1,m} - 2U_{n,m} + U_{n-1,m}}{h^2} + \frac{U_{n,m+1} - 2U_{n,m} + U_{n,m-1}}{k^2} = 0 \qquad (3.133)$$

For $k = h$ and simplification of (3.133) results into

$$4U_{n,m} - U_{n+1,m} - U_{n-1,m} - U_{n,m+1} - U_{n,m-1} = 0 \qquad (3.134)$$

Application of (3.134) to every interior point produces the following linear algebraic equations

$$\left.\begin{aligned}
4u_{1,1} - u_{2,1} - 1000 - u_{1,2} - 2500 = 0 \\
4u_{2,1} - u_{3,1} - u_{1,1} - u_{2,2} - 3000 = 0 \\
4u_{3,1} - 1000 - u_{2,1} - u_{3,2} - 3000 = 0 \\
4u_{1,2} - u_{2,2} - 2000 - u_{1,3} - u_{1,1} = 0 \\
4u_{2,2} - u_{3,2} - u_{1,2} - u_{2,3} - u_{2,1} = 0 \\
4u_{3,2} - 1500 - u_{2,2} - u_{3,3} - u_{3,1} = 0 \\
4u_{1,3} - u_{2,3} - 1000 - 2500 - u_{1,2} = 0 \\
4u_{2,3} - u_{3,3} - u_{1,3} - 3000 - u_{2,2} = 0 \\
4u_{3,3} - 1000 - u_{2,3} - 3000 - u_{3,2} = 0
\end{aligned}\right\} \qquad (3.135)$$

And the above algebraic linear equations is rearranged to obtain the following

$$\left.\begin{aligned}
4u_{1,1} - u_{2,1} - u_{1,2} = 3500 \\
4u_{2,1} - u_{3,1} - u_{1,1} - u_{2,2} = 3000 \\
4u_{3,1} - u_{2,1} - u_{3,2} = 4000 \\
4u_{1,2} - u_{1,3} - u_{1,1} - u_{2,2} = 2000 \\
4u_{2,2} - u_{1,2} - u_{2,3} - u_{3,2} - u_{2,1} = 0 \\
4u_{3,2} - u_{2,2} - u_{3,1} - u_{3,2} = 1500 \\
4u_{1,3} - u_{1,2} - u_{2,3} = 3500 \\
4u_{2,3} - u_{1,3} - u_{2,2} - u_{3,3} = 3000 \\
4u_{3,3} - u_{2,3} - u_{3,2} = 4000
\end{aligned}\right\} \qquad (3.136)$$

Representing the linear algebraic equations of (3.131) in a matrix form results into the linear system below.

$$\begin{pmatrix}
4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4
\end{pmatrix}
\begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix}
=
\begin{pmatrix} 3500 \\ 3000 \\ 4000 \\ 2000 \\ 0 \\ 1500 \\ 3500 \\ 3000 \\ 4000 \end{pmatrix} \qquad (3.137)$$

The coefficient matrix is an M matrix and system (3.137) is represented in the form $Bz = f$ where,

$$B = \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}, \quad z = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix} \quad f = \begin{pmatrix} 3500 \\ 3000 \\ 4000 \\ 2000 \\ 0 \\ 1500 \\ 3500 \\ 3000 \\ 4000 \end{pmatrix} \tag{3.138}$$

and from (3.138), the diagonal component of matrix B is obtained as

$$D = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}, D^{-1} = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{pmatrix} \tag{3.139}$$

Now, the linear system in (3.138) is multiplied by D^{-1} to obtain

$$\begin{pmatrix} 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 1 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix} = \begin{pmatrix} 875 \\ 750 \\ 1000 \\ 500 \\ 0 \\ 375 \\ 875 \\ 750 \\ 1000 \end{pmatrix} \quad (3.140)$$

The transformed linear system in (3.140) can be expressed in the form $Az = b$ where

$$A = \begin{pmatrix} 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 1 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 \end{pmatrix}$$

$$z = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 875 \\ 750 \\ 1000 \\ 500 \\ 0 \\ 375 \\ 875 \\ 750 \\ 1000 \end{pmatrix} \quad (3.142)$$

The coefficient matrix A of the above equation is decomposed into $A = I - L - U$ where

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.143}$$

$$-U = \begin{pmatrix} 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.144}$$

$$-L=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \end{pmatrix} \tag{3.145}$$

The proposed EAOR method $z^{(k+1)}=[I-(v+r)L]^{-1}[(1-\omega)I+[\omega-(v+r)]L+\omega U]z^{(k)}+[I-(v+r)L]^{-1}\omega\dot{b}$ is applied to the linear system (3. 137) as follows.

By letting $r=0.04,\ \omega=0.1$ and $v=0.05$, then

$[I-(v+r)L]^{-1}=$

$$
\left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - (v+r)\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \end{bmatrix}\right)^{-1}
$$

$$
= \begin{bmatrix} 1.0000 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -0.0225 & 1.0000 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.0005 & -0.0225 & 1.0000 & 0. & 0. & 0. & 0. & 0. & 0. \\ -0.0225 & 0. & 0. & 1.0000 & 0. & 0. & 0. & 0. & 0. \\ 0.0010 & -0.0225 & 0. & -0.0225 & 1.0000 & 0. & 0. & 0. & 0. \\ 0. & 0.0010 & -0.0225 & 0.0005 & -0.0225 & 1.0000 & 0. & 0. & 0. \\ 0.0005 & 0. & 0. & -0.0225 & 0. & 0. & 1.0000 & 0. & 0. \\ 0. & 0.0005 & 0. & 0.0010 & -0.0225 & 0. & -0.0225 & 1.0000 & 0. \\ 0. & 0. & 0.0005 & 0. & 0.0010 & -0.0225 & 0.0005 & -0.0225 & 1.0000 \end{bmatrix} \tag{3.146}
$$

$(1-\omega)I + [\omega - (v+r)]L + \omega U =$

$$
(1-0.1)\times\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.147}
$$

$$+\left[0.1-(0.04+0.05)\right]\times\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \end{pmatrix} \tag{3.148}$$

$$+0.1\times\begin{pmatrix} 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.149}$$

$$=\begin{bmatrix} 0.900 & -0.025 & 0. & -0.025 & 0. & 0. & 0. & 0. & 0. \\ -0.002 & 0.900 & -0.025 & 0. & -0.025 & 0. & 0. & 0. & 0. \\ 0. & -0.002 & 0.900 & 0. & 0. & -0.025 & 0. & 0. & 0. \\ -0.002 & 0. & 0. & 0.900 & -0.025 & 0. & -0.025 & 0. & 0. \\ 0. & -0.002 & 0. & -0.002 & 0.900 & -0.025 & 0. & -0.025 & 0. \\ 0. & 0. & -0.002 & 0. & -0.002 & 0.900 & 0. & 0. & -0.025 \\ 0. & 0. & 0. & -0.002 & 0. & 0. & 0.900 & -0.025 & 0. \\ 0. & 0. & 0. & 0. & -0.002 & 0. & -0.002 & 0.900 & -0.025 \\ 0. & 0. & 0. & 0. & 0. & -0.002 & 0. & -0.002 & 0.900 \end{bmatrix} \tag{3.150}$$

$$[I-(v+r)L]^{-1}[(1-\omega)I+[\omega-(v+r)]L+\omega U]=$$

$$\begin{bmatrix} 0.9000 & -0.0250 & 0. & -0.0250 & 0. & 0. & 0. & 0. & 0. \\ -0.0228 & 0.9006 & -0.0250 & 0.0006 & -0.0250 & 0. & 0. & 0. & 0. \\ 0.0005 & -0.0228 & 0.9006 & -0. & 0.0006 & -0.0250 & 0. & 0. & 0. \\ -0.0228 & 0.0006 & 0. & 0.9006 & -0.0250 & 0. & -0.0250 & 0. & 0. \\ 0.0010 & -0.0228 & 0.0006 & -0.0228 & 0.9011 & -0.0250 & 0.0006 & -0.0250 & 0. \\ -0. & 0.0010 & -0.0228 & 0.0005 & -0.0228 & 0.9011 & -0. & 0.0006 & -0.0250 \\ 0.0005 & -0. & 0. & -0.0228 & 0.0006 & 0. & 0.9006 & -0.0250 & 0. \\ -0. & 0.0005 & -0. & 0.0010 & -0.0228 & 0.0006 & -0.0228 & 0.9011 & -0.0250 \\ 0. & -0. & 0.0005 & -0. & 0.0010 & -0.0228 & 0.0005 & -0.0228 & 0.9011 \end{bmatrix} \quad (3.151)$$

Using Maple 2017 software the eigenvalues computed from the iteration matrix in (3.151) are given by

$$\begin{bmatrix} 0.9697412031 \\ 0.9342940343 \\ 0.8347587969 \\ 0.9000000000 \\ 0.8668309657 \\ 0.9342940343 \\ 0.8668309657 \\ 0.9000000000 \\ 0.9000000000 \end{bmatrix} \quad (3.152)$$

And the spectral radius of the EAOR iteration matrix, which is the absolute largest value of the eigenvalues obtained in (3.152) is 0.9697412031

$$\omega\dot{b}=(0.1)\times\begin{pmatrix} 875 \\ 750 \\ 1000 \\ 500 \\ 0 \\ 375 \\ 875 \\ 750 \\ 1000 \end{pmatrix}=\begin{pmatrix} 87.5 \\ 75.0 \\ 100.0 \\ 50.0 \\ 0 \\ 37.5 \\ 87.5 \\ 75.0 \\ 100.0 \end{pmatrix},\ [I-(v+r)L]^{-1}\omega\dot{b}=\begin{bmatrix} 87.50000000 \\ 76.96875000 \\ 101.7317969 \\ 52.28896543 \\ 1.176501722 \\ 37.52647129 \\ 88.34434560 \\ 76.98774778 \\ 101.7322243 \end{bmatrix} \quad (3.153)$$

Applying the initial estimation $z^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, the first 3 iterations of the proposed EAOR method are as follows.

$$z^{(1)} = \begin{bmatrix} 875.0000000 \\ 946.8750000 \\ 1213.046875 \\ 772.9355469 \\ 173.9104980 \\ 414.1298621 \\ 968.1792190 \\ 967.8403243 \\ 1217.764073 \end{bmatrix} \quad (3.154)$$

$$z^{(2)} = \begin{bmatrix} 1304.952637 \\ 1412.228687 \\ 1638.189682 \\ 1184.441280 \\ 631.3152231 \\ 1067.879511 \\ 1367.586218 \\ 1386.352398 \\ 1336.125298 \end{bmatrix} \quad (3.155)$$

$$z^{(3)} = \begin{bmatrix} 1524.167492 \\ 1692.937728 \\ 1979.296903 \\ 1486.021905 \\ 977.5239378 \\ 1286.653645 \\ 1537.782157 \\ 1464.221965 \\ 1364.108752 \end{bmatrix} \quad (3.156)$$

Next, we apply algorithm 3.2 for the proposed Refinement of Extended Accelerated Over Relaxation method in similar manner.

The proposed REAOR represented as $\bar{z}^{(k+1)} = \Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2 z^{(k)} + \Big(I + (I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)(I-(v+r)L)^{-1}\omega\dot{b}$ is applied to the linear system (3. 136). By letting $\omega = 0.1$, $r = 0.04$ and $v = 0.05$, we have $\Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2 =$

$$\begin{bmatrix} 0.81114 & 0.04503 & 0.00062 & 0.04503 & 0.00125 & 0. & 0.00063 & 0. & 0. \\ 0.04101 & 0.81272 & 0.04504 & 0.00215 & 0.04507 & 0.00125 & 0.00003 & 0.00062 & 0. \\ 0.00144 & 0.04105 & 0.81215 & 0.00007 & 0.00215 & 0.04506 & 0. & 0.00003 & 0.00062 \\ 0.04101 & 0.00215 & 0.00003 & 0.81272 & 0.04507 & 0.00062 & 0.04504 & 0.00125 & 0. \\ 0.00288 & 0.04112 & 0.00215 & 0.04112 & 0.81431 & 0.04508 & 0.00215 & 0.04508 & 0.00125 \\ 0.00013 & 0.00289 & 0.04108 & 0.00145 & 0.04116 & 0.81374 & 0.00007 & 0.00215 & 0.04507 \\ 0.00144 & 0.00007 & 0. & 0.04105 & 0.00215 & 0.00003 & 0.81215 & 0.04506 & 0.00062 \\ 0.00013 & 0.00145 & 0.00007 & 0.00289 & 0.04116 & 0.00215 & 0.04108 & 0.81374 & 0.04507 \\ 0.00001 & 0.00013 & 0.00144 & 0.00013 & 0.00289 & 0.04112 & 0.00144 & 0.04112 & 0.81317 \end{bmatrix} \quad (3.157)$$

Using Maple 2017 software, the eigenvalues computed from the iteration matrix in (3.157) are given by

$$\begin{bmatrix} 0.940398001048955 \\ 0.872905342529066 \\ 0.696822248951045 \\ 0.751395923095934 \\ 0.872905342529066 \\ 0.751395923095935 \\ 0.810000000000000 \\ 0.810000000000000 \\ 0.810000000000000 \end{bmatrix} \quad (3.158)$$

And the spectral radius of the new REAOR iteration matrix, which is the largest value of the eigenvalues obtained in (3.158) is 0.940398001048955. The spectral radius of the REAOR iteration matrix indicates that the method is convergent. Therefore, we compute the approximate solutions by carrying out several iterations until convergence is attained.

$$\left(I + (I - (v + r)L)^{-1}\left((1-\omega)I + [\omega - (v+r)]L + \omega U\right)\right) =$$

$$\begin{bmatrix} 61.901 & 0.0250 & 0. & 0.0250 & 0. & 0. & 0. & 0. & 0. \\ 0.0228 & 61.901 & 0.0250 & 0.0006 & 0.0250 & 0. & 0. & 0. & 0. \\ 0.0005 & 0.0228 & 61.901 & 0. & 0.0006 & 0.0250 & 0. & 0. & 0. \\ 0.0228 & 0.0006 & 0. & 61.901 & 0.0250 & 0. & 0.0250 & 0. & 0. \\ 0.0010 & 0.0228 & 0.0006 & 0.0228 & 61.901 & 0.0250 & 0.0006 & 0.0250 & 0. \\ 0. & 0.0010 & 0.0228 & 0.0005 & 0.0228 & 61.901 & 0. & 0.0006 & 0.0250 \\ 0.0005 & 0. & 0. & 0.0228 & 0.0006 & 0. & 61.901 & 0.0250 & 0. \\ 0. & 0.0005 & 0. & 0.0010 & 0.0228 & 0.0006 & 0.0228 & 61.901 & 0.0250 \\ 0. & 0. & 0.0005 & 0. & 0.0010 & 0.0228 & 0.0005 & 0.0228 & 61.901 \end{bmatrix} \quad (3.159)$$

$$\omega\dot{b} = (0.1) \times \begin{pmatrix} 875 \\ 750 \\ 1000 \\ 500 \\ 0 \\ 375 \\ 875 \\ 750 \\ 1000 \end{pmatrix} = \begin{pmatrix} 87.5 \\ 75.0 \\ 100.0 \\ 50.0 \\ 0 \\ 37.5 \\ 87.5 \\ 75.0 \\ 100.0 \end{pmatrix}, [I - (v+r)L]^{-1}\omega\dot{b} = \begin{bmatrix} 87.50000000 \\ 76.96875000 \\ 101.7317969 \\ 52.28896543 \\ 1.176501722 \\ 37.52647129 \\ 88.34434560 \\ 76.98774778 \\ 101.7322243 \end{bmatrix} \quad (3.160)$$

Then,

$$[I + [I - (v+r)L]^{-1}[(1-\omega)I + [\omega - (v+r)]L + \omega U](1 - (v+r)L]^{-1}\omega\dot{b} =$$

$$\begin{bmatrix} 1304.952637 \\ 1412.228687 \\ 1638.189682 \\ 1184.441280 \\ 631.3152231 \\ 1067.879511 \\ 1367.586218 \\ 1386.352398 \\ 1336.125298 \end{bmatrix} \quad (3.161)$$

Applying the initial estimation $z^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, the first 3 iterations of the proposed REAOR method are as follows;

$$z^{(1)} = \begin{bmatrix} 1304.9526367188 \\ 1412.2286865234 \\ 1638.1896817017 \\ 1184.4412795105 \\ 631.3152231436 \\ 1067.8795106394 \\ 1367.5862175121 \\ 1386.3523976543 \\ 1336.1252975789 \end{bmatrix} \tag{3.162}$$

$$z^{(2)} = \begin{bmatrix} 1669.7399082996 \\ 1903.0008769630 \\ 2163.6675282184 \\ 1665.1341402297 \\ 1099.5246318384 \\ 1372.3038679849 \\ 1581.9902027415 \\ 1485.4195375818 \\ 1370.8249450870 \end{bmatrix} \tag{3.163}$$

$$z^{(3)} = \begin{bmatrix} 1809.4160515972 \\ 2052.2864992950 \\ 2294.6563504146 \\ 1758.0521382573 \\ 1161.3052179206 \\ 1406.9369310711 \\ 1599.3249048078 \\ 1492.8954591653 \\ 1373.1816001811 \end{bmatrix} \tag{3.164}$$

3.7.2 Problem 2

In this experiment, we consider the system of linear equations from Mohammed and Rivaie (2017), whose coefficient matrix is an M matrix in the form $Bz = f$.

$$\begin{pmatrix} 7 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 7 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 7 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\ -1 & 0 & -1 & 7 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 7 & -1 & 0 & -1 & 0 & -1 \\ -1 & 0 & -1 & 0 & -1 & 7 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 & 7 & -1 & 0 & -1 \\ -1 & 0 & -1 & 0 & -1 & 0 & -1 & 7 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 7 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 7 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \tag{3.165}$$

And from (3.165), the diagonal component of matrix B is obtained as

$$D = \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \end{pmatrix} \tag{3.166}$$

Then we obtain

$$D^{-1} = \begin{pmatrix} \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{7} & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \end{pmatrix} \quad (3.167)$$

Now, the linear system in (3.165) is multiplied by D^{-1} to obtain

$$A = \begin{pmatrix} 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & 0 \\ -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 \\ 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} \\ -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 \\ 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} \\ -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 \\ 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 & -\frac{1}{7} \\ -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} & 0 \\ 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 & -\frac{1}{7} \\ 0 & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{pmatrix}$$

$$= \begin{pmatrix} 11.90 \\ 9.32 \\ 8.09 \\ 9.32 \\ 8.09 \\ 8.32 \\ 8.09 \\ 8.32 \\ 8.09 \\ 8.32 \end{pmatrix} \qquad (3.168)$$

The coefficient matrix A of the above equation is decomposed into $A = I - L - U$ and then solved with the proposed Extended Accelerated Over Relaxation method, Refinement of Extended Accelerated Over Relaxation method, the classical AOR method, QAOR method by Wu and Liu (2014), KAOR method by Youssef and Farid (2015) and RAOR by Vatti *et al*., (2018). The results are tabulated and discussed in chapter four.

3.7.3 Problem 3

We consider a linear system of (3.1) whose coefficient matrix is an irreducible weak diagonally dominant matrix, expressed in $Az = b$

$$A = \begin{pmatrix} 1 & 0.1 & 0.2 & 0.0 & 0.2 & -0.5 \\ 0.2 & 1 & 0.3 & 0.0 & -0.4 & 0.1 \\ 0.0 & 0.2 & 1 & -0.6 & 0.2 & 0.0 \\ 0.2 & -0.3 & 0.1 & 1 & 0.1 & 0.3 \\ 0.0 & 0.3 & 0.2 & 0.1 & 1 & 0.2 \\ 0.2 & 0.3 & 0.0 & -0.3 & 0.1 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.2 \\ 0.8 \\ 1.4 \\ 1.8 \\ 0.7 \end{pmatrix} \tag{3.169}$$

The coefficient matrix A of (3.169) is in the form $A = I - L - U$, where

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, -L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.0 & 0.2 & 0 & 0 & 0 & 0 \\ 0.2 & -0.3 & 0.1 & 0 & 0 & 0 \\ 0.0 & 0.3 & 0.2 & 0.1 & 0 & 0 \\ 0.2 & 0.3 & 0.0 & -0.3 & 0.1 & 0 \end{pmatrix}$$

$$-U = \begin{pmatrix} 0 & 0.1 & 0.2 & 0.0 & 0.2 & -0.5 \\ 0 & 0 & 0.3 & 0.0 & -0.4 & 0.1 \\ 0 & 0 & 0 & -0.6 & 0.2 & 0.0 \\ 0 & 0 & 0 & 0 & 0.1 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.170}$$

The spectral radii and convergence rate of the two proposed methods (EAOR and REAOR methods) are computed along with some existing methods such as the AOR method and some variants of AOR method with the aid of maple 2017 software.

3.5.4 Problem 4

We consider the second-order partial differential equation from Ndanusa (2012)

$$(x+1)\frac{\partial^2 U}{\partial x^2} + (y^2+1)\frac{\partial^2 U}{\partial y^2} + U = -1 \tag{3.171}$$

to be solved in the region

$$\begin{aligned} 0 \le y \le 1 \\ 0 \le x \le 1 \end{aligned} \tag{3.172}$$

For $h = \frac{1}{6}$ and boundary conditions;

$$\begin{aligned} U(x,0) = 0\,, \quad U(0,y) = y \\ U(x,1) = 1\,, \; U(1,y) = y^2 \end{aligned} \tag{3.173}$$

To evaluate the partial differential equation in (3.171), the method of finite difference is used, where a square mesh of horizontal and vertical lines with mesh spacing of $h = \frac{1}{6}$ in both x and y directions is applied to the region $0 \le x \le 1,\ 0 \le y \le 1$. This generates twenty-five internal points as follows; $u_{1,1}, u_{2,1}, u_{3,1}, u_{4,1}, u_{5,1}$, $u_{1,2}, u_{2,2}, u_{3,2}, u_{4,2}, u_{5,2}$, $u_{1,3}, u_{2,3}, u_{3,3}, u_{4,3}, u_{5,3}$, $u_{1,4}, u_{2,4}, u_{3,4}, u_{4,4}, u_{5,4}$, $u_{1,5}, u_{2,5}$, $u_{3,5}, u_{4,5}, u_{5,5}$ and 24 boundary points, $u_{0,0}, u_{1,0}, u_{2,0}, u_{3,0}, u_{0,1}, u_{3,1}, u_{0,2}, u_{3,2}, u_{0,3}, u_{1,3}, u_{2,3}$ and $u_{3,3}$.

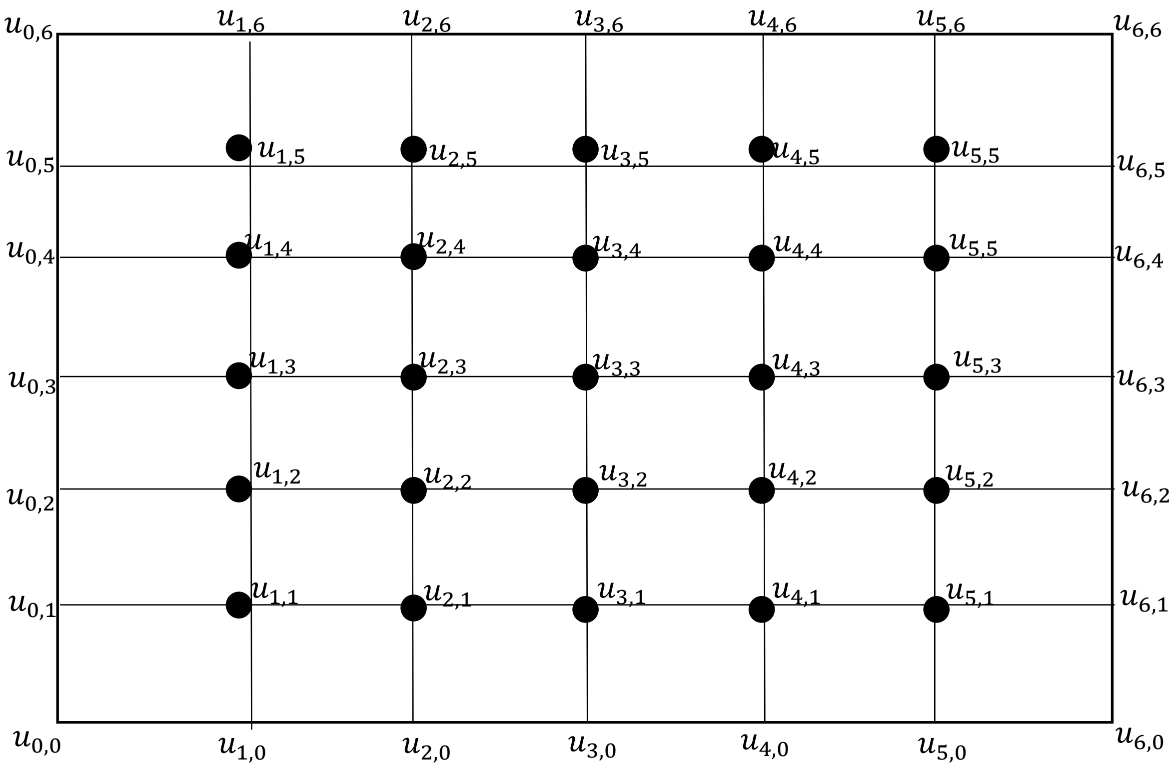

Figure 3.3: Discretization of square mesh with 25 inner grids

At each interior point of the region, the partial derivatives $\frac{\partial^2 U}{\partial x^2}$ and $\frac{\partial^2 U}{\partial y^2}$ appearing in (3.171) are substituted by its quotients (3.132) and thus the central finite difference approximation to (3.171) at each interior grid point is given by

$$(x+1)\frac{U_{n+1,m}-2U_{n,m}+U_{n-1,m}}{h^2}+(y^2+1)\frac{U_{n,m+1}-2U_{n,m}+U_{n,m-1}}{k^2}$$

$$=-\left(U_{n,m}+1\right) \quad , \quad k=h \qquad (3.174)$$

which is simplified to obtain

$$(x+1)U_{n+1,m}+(y^2+1)U_{n,m+1}+(x+1)U_{n-1,m}+(y^2+1)U_{n,m-1}$$

$$+[-2(x+1)-2(y^2+1)+h^2]U_{n,m}=-h^2 \qquad (3.175)$$

$$[2(x+y^2+2)-h^2]U_{n,m}-(x+1)U_{n+1,m}-(y^2+1)U_{n,m+1}-(x+1)U_{n-1,m}$$

$$-(y^2+1)U_{n,m-1}=h^2 \qquad (3.176)$$

At $h=\frac{1}{6}$, the above equation becomes

$$\left(2x+2y^2+\frac{143}{36}\right)U_{n,m}-(x+1)U_{n+1,m}-(y^2+1)U_{n,m+1}-(x+1)U_{n-1,m}$$

$$-(y^2+1)U_{n,m-1}=\frac{1}{36} \qquad (3.177)$$

Equation (3.177) is applied to each of the interior points to get $u_{1,1}=(1/6\,,1/6)$, $u_{2,1}=(2/6\,,1/6)$, $u_{3,1}=(3/6\,,1/6)$, $u_{4,1}=(4/6\,,1/6)$, $u_{5,1}=(5/6\,,1/6)$, $u_{1,2}=(1/6\,,2/6)$, $u_{2,2}=(2/6\,,2/6)$, $u_{3,2}=(3/6\,,2/6)$, $u_{4,2}=(4/6\,,2/6)$, $u_{5,2}=(5/6\,,2/6)$, $u_{1,3}=(1/6\,,3/6)$, $u_{2,3}=(2/6\,,3/6)$, $u_{3,3}=(3/6\,,3/6)$, $u_{4,3}=(4/6\,,3/6)$, $u_{5,3}=(5/6\,,3/6)$, $u_{1,4}=(1/6\,,4/6)$, $u_{2,4}=(2/6\,,4/6)$, $u_{3,4}=(3/6\,,4/6)$ $u_{4,4}=(4/6\,,4/6)$, $u_{5,4}=(5/6\,,4/6)$, $u_{1,5}=(1/6\,,5/6)$, $u_{2,5}=(2/6\,,5/6)$, $u_{3,5}=(3/6\,,5/6)$, $u_{4,5}=(4/6\,,5/6)$ and $u_{5,5}=(5/6\,,5/6)$. Application of the transformed partial differential equation by finite differences in (3.177) to each of the twenty-five interior points generated the following system of linear equations:

$$\left.\begin{aligned}
&\frac{157}{36}u_{1,1}-\frac{7}{6}u_{2,1}-\frac{37}{36}u_{1,2}-\frac{7}{6}u_{0,1}-\frac{37}{36}u_{1,0}=\frac{1}{36}\\
&\frac{169}{36}u_{2,1}-\frac{4}{3}u_{3,1}-\frac{37}{36}u_{2,2}-\frac{4}{3}u_{1,1}-\frac{37}{36}u_{2,0}=\frac{1}{36}\\
&\frac{181}{36}u_{3,1}-\frac{3}{2}u_{4,1}-\frac{37}{36}u_{3,2}-\frac{3}{2}u_{2,1}-\frac{37}{36}u_{3,0}=\frac{1}{36}\\
&\frac{193}{36}u_{4,1}-\frac{5}{3}u_{5,1}-\frac{37}{36}u_{4,2}-\frac{5}{3}u_{3,1}-\frac{37}{36}u_{4,0}=\frac{1}{36}\\
&\frac{205}{36}u_{5,1}-\frac{11}{6}u_{6,1}-\frac{37}{36}u_{5,2}-\frac{11}{6}u_{4,1}-\frac{37}{36}u_{5,0}=\frac{1}{36}\\
&\frac{163}{36}u_{1,2}-\frac{7}{6}u_{2,2}-\frac{10}{9}u_{1,3}-\frac{7}{6}u_{0,2}-\frac{10}{9}u_{1,1}=\frac{1}{36}\\
&\frac{175}{36}u_{2,2}-\frac{4}{3}u_{3,2}-\frac{10}{9}u_{2,3}-\frac{4}{3}u_{1,2}-\frac{10}{9}u_{2,1}=\frac{1}{36}\\
&\frac{187}{36}u_{3,2}-\frac{3}{2}u_{4,2}-\frac{10}{9}u_{3,3}-\frac{3}{2}u_{2,2}-\frac{10}{9}u_{3,1}=\frac{1}{36}\\
&\frac{199}{36}u_{4,2}-\frac{5}{3}u_{5,2}-\frac{10}{9}u_{4,3}-\frac{5}{3}u_{3,2}-\frac{10}{9}u_{4,1}=\frac{1}{36}\\
&\frac{211}{36}u_{5,2}-\frac{11}{6}u_{6,2}-\frac{10}{9}u_{5,3}-\frac{5}{3}u_{4,2}-\frac{10}{9}u_{5,1}=\frac{1}{36}\\
&\frac{173}{36}u_{1,3}-\frac{7}{6}u_{2,3}-\frac{5}{4}u_{1,4}-\frac{7}{6}u_{0,3}-\frac{5}{4}u_{1,2}=\frac{1}{36}\\
*\quad&\frac{185}{36}u_{2,3}-\frac{4}{3}u_{3,3}-\frac{5}{4}u_{2,4}-\frac{4}{3}u_{1,3}-\frac{5}{4}u_{2,2}=\frac{1}{36}\\
&\frac{197}{36}u_{3,3}-\frac{3}{2}u_{4,3}-\frac{5}{4}u_{3,4}-\frac{3}{2}u_{2,3}-\frac{5}{4}u_{3,2}=\frac{1}{36}\\
&\frac{209}{36}u_{4,3}-\frac{5}{3}u_{5,3}-\frac{5}{4}u_{4,4}-\frac{5}{3}u_{3,3}-\frac{5}{4}u_{4,2}=\frac{1}{36}\\
&\frac{221}{36}u_{5,3}-\frac{11}{6}u_{6,3}-\frac{5}{4}u_{5,4}-\frac{11}{6}u_{4,3}-\frac{5}{4}u_{5,2}=\frac{1}{36}\\
&\frac{187}{36}u_{1,4}-\frac{7}{6}u_{2,4}-\frac{5}{4}u_{1,5}-\frac{7}{6}u_{0,4}-\frac{13}{9}u_{1,3}=\frac{1}{36}\\
&\frac{199}{36}u_{2,4}-\frac{4}{3}u_{3,4}-\frac{13}{9}u_{2,5}-\frac{4}{3}u_{1,4}-\frac{13}{9}u_{2,3}=\frac{1}{36}\\
&\frac{211}{36}u_{3,4}-\frac{3}{2}u_{4,4}-\frac{13}{9}u_{3,5}-\frac{3}{2}u_{2,4}-\frac{13}{9}u_{3,3}=\frac{1}{36}\\
&\frac{223}{36}u_{4,4}-\frac{5}{3}u_{5,4}-\frac{13}{9}u_{4,5}-\frac{5}{3}u_{3,4}-\frac{13}{9}u_{4,3}=\frac{1}{36}\\
&\frac{235}{36}u_{5,4}-\frac{11}{6}u_{6,4}-\frac{13}{9}u_{5,5}-\frac{11}{6}u_{4,4}-\frac{13}{9}u_{5,3}=\frac{1}{36}
\end{aligned}\right\}$$

$$\left.\begin{aligned}
\frac{205}{36}u_{1,5}-\frac{7}{6}u_{2,5}-\frac{61}{36}u_{1,6}-\frac{7}{6}u_{0,5}-\frac{61}{36}u_{1,4}&=\frac{1}{36}\\
\frac{217}{36}u_{2,5}-\frac{4}{3}u_{3,5}-\frac{61}{36}u_{2,6}-\frac{4}{3}u_{1,5}-\frac{61}{36}u_{2,4}&=\frac{1}{36}\\
\frac{229}{36}u_{3,5}-\frac{3}{2}u_{4,5}-\frac{61}{36}u_{3,6}-\frac{3}{2}u_{2,5}-\frac{61}{36}u_{3,4}&=\frac{1}{36}\\
\frac{241}{36}u_{4,5}-\frac{5}{3}u_{5,5}-\frac{61}{36}u_{4,6}-\frac{5}{3}u_{3,5}-\frac{61}{36}u_{4,4}&=\frac{1}{36}\\
\frac{253}{36}u_{5,5}-\frac{11}{6}u_{6,5}-\frac{61}{36}u_{5,6}-\frac{11}{6}u_{4,5}-\frac{61}{36}u_{5,4}&=\frac{1}{36}
\end{aligned}\right\} \tag{3.178}$$

The known boundary values are;

$$\left.\begin{matrix}
\begin{aligned} u_{0,0}&=0\\ u_{1,0}&=0\\ u_{2,0}&=0\\ u_{3,0}&=0\\ u_{4,0}&=0\\ u_{5,0}&=0 \end{aligned}, &
\begin{aligned} u_{0,1}&=\frac{1}{6}\\ u_{0,2}&=\frac{2}{6}\\ u_{0,3}&=\frac{3}{6}\\ u_{0,4}&=\frac{4}{6}\\ u_{0,5}&=\frac{5}{6}\\ u_{0,6}&=1 \end{aligned}, &
\begin{aligned} u_{1,6}&=1\\ u_{2,6}&=1\\ u_{3,6}&=1\\ u_{4,6}&=1\\ u_{5,6}&=1\\ u_{6,6}&=1 \end{aligned}, &
\begin{aligned} u_{6,0}&=0\\ u_{6,1}&=\frac{1}{36}\\ u_{6,2}&=\frac{4}{36}\\ u_{6,3}&=\frac{9}{36}\\ u_{6,4}&=\frac{16}{36}\\ u_{6,5}&=\frac{25}{16} \end{aligned}
\end{matrix}\right\} \tag{3.179}$$

And they are substituted into (3.178) to generate the following linear system;

$$\frac{157}{36}u_{1,1}-\frac{7}{6}u_{2,1}-\frac{37}{36}u_{1,2}=\frac{2}{9}$$
$$\frac{169}{36}u_{2,1}-\frac{4}{3}u_{3,1}-\frac{37}{36}u_{2,2}-\frac{4}{3}u_{1,1}=\frac{1}{36}$$
$$\frac{181}{36}u_{3,1}-\frac{3}{2}u_{4,1}-\frac{37}{36}u_{3,2}-\frac{3}{2}u_{2,1}=\frac{1}{36}$$
$$\frac{193}{36}u_{4,1}-\frac{5}{3}u_{5,1}-\frac{37}{36}u_{4,2}-\frac{5}{3}u_{3,1}=\frac{1}{36}$$
$$\frac{205}{36}u_{5,1}-\frac{37}{36}u_{5,2}-\frac{11}{6}u_{4,1}=\frac{17}{216}$$
$$\frac{163}{36}u_{1,2}-\frac{7}{6}u_{2,2}-\frac{10}{9}u_{1,3}-\frac{10}{9}u_{1,1}=\frac{5}{12}$$
$$\frac{175}{36}u_{2,2}-\frac{4}{3}u_{3,2}-\frac{10}{9}u_{2,3}-\frac{4}{3}u_{1,2}-\frac{10}{9}u_{2,1}=\frac{1}{36}$$
$$\frac{187}{36}u_{3,2}-\frac{3}{2}u_{4,2}-\frac{10}{9}u_{3,3}-\frac{3}{2}u_{2,2}-\frac{10}{9}u_{3,1}=\frac{1}{36}$$
$$\frac{199}{36}u_{4,2}-\frac{5}{3}u_{5,2}-\frac{10}{9}u_{4,3}-\frac{5}{3}u_{3,2}-\frac{10}{9}u_{4,1}=\frac{1}{36}$$
$$\frac{211}{36}u_{5,2}-\frac{10}{9}u_{5,3}-\frac{5}{3}u_{4,2}-\frac{10}{9}u_{5,1}=\frac{25}{108}$$
$$\frac{173}{36}u_{1,3}-\frac{7}{6}u_{2,3}-\frac{5}{4}u_{1,4}-\frac{5}{4}u_{1,2}=\frac{11}{18}$$
$$\frac{185}{36}u_{2,3}-\frac{4}{3}u_{3,3}-\frac{5}{4}u_{2,4}-\frac{4}{3}u_{1,3}-\frac{5}{4}u_{2,2}=\frac{1}{36}$$
$$\frac{197}{36}u_{3,3}-\frac{3}{2}u_{4,3}-\frac{5}{4}u_{3,4}-\frac{3}{2}u_{2,3}-\frac{5}{4}u_{3,2}=\frac{1}{36}$$
$$\frac{209}{36}u_{4,3}-\frac{5}{3}u_{5,3}-\frac{5}{4}u_{4,4}-\frac{5}{3}u_{3,3}-\frac{5}{4}u_{4,2}=\frac{1}{36}$$
$$\frac{221}{36}u_{5,3}-\frac{5}{4}u_{5,4}-\frac{11}{6}u_{4,3}-\frac{5}{4}u_{5,2}=\frac{35}{72}$$
$$\frac{187}{36}u_{1,4}-\frac{7}{6}u_{2,4}-\frac{5}{4}u_{1,5}-\frac{13}{9}u_{1,3}=\frac{29}{36}$$
$$\frac{199}{36}u_{2,4}-\frac{4}{3}u_{3,4}-\frac{13}{9}u_{2,5}-\frac{4}{3}u_{1,4}-\frac{13}{9}u_{2,3}=\frac{1}{36}$$
$$\frac{211}{36}u_{3,4}-\frac{3}{2}u_{4,4}-\frac{13}{9}u_{3,5}-\frac{3}{2}u_{2,4}-\frac{13}{9}u_{3,3}=\frac{1}{36}$$
$$\frac{223}{36}u_{4,4}-\frac{5}{3}u_{5,4}-\frac{13}{9}u_{4,5}-\frac{5}{3}u_{3,4}-\frac{13}{9}u_{4,3}=\frac{1}{36}$$
$$\frac{235}{36}u_{5,4}-\frac{13}{9}u_{5,5}-\frac{11}{6}u_{4,4}-\frac{13}{9}u_{5,3}=\frac{91}{108}$$

$$\left.\begin{array}{r}\frac{205}{36}u_{1,5}-\frac{7}{6}u_{2,5}-\frac{61}{36}u_{1,4}=\frac{97}{36}\\ \frac{217}{36}u_{2,5}-\frac{4}{3}u_{3,5}-\frac{4}{3}u_{1,5}-\frac{61}{36}u_{2,4}=\frac{31}{18}\\ \frac{229}{36}u_{3,5}-\frac{3}{2}u_{4,5}-\frac{3}{2}u_{2,5}-\frac{61}{36}u_{3,4}=\frac{31}{18}\\ \frac{241}{36}u_{4,5}-\frac{5}{3}u_{5,5}-\frac{5}{3}u_{3,5}-\frac{61}{36}u_{4,4}=\frac{31}{18}\\ \frac{253}{36}u_{5,5}-\frac{11}{6}u_{4,5}-\frac{61}{36}u_{5,4}=\frac{647}{216}\end{array}\right\} \tag{3.180}$$

The above linear equations are represented in the matrix form $Az = b$, where;

$$A=\begin{pmatrix}
\frac{157}{36} & -\frac{7}{6} & 0 & 0 & 0 & -\frac{37}{36} & 0 & 0 & 0 & \cdots & 0 & 0\\
-\frac{4}{3} & \frac{169}{36} & -\frac{4}{3} & 0 & 0 & 0 & -\frac{37}{36} & 0 & 0 & \cdots & 0 & 0\\
0 & -\frac{3}{2} & \frac{181}{36} & -\frac{3}{2} & 0 & 0 & 0 & -\frac{37}{36} & 0 & \cdots & 0 & 0\\
0 & 0 & -\frac{5}{3} & \frac{193}{36} & -\frac{5}{3} & 0 & 0 & 0 & -\frac{37}{36} & \cdots & 0 & 0\\
0 & 0 & 0 & -\frac{11}{6} & \frac{205}{36} & 0 & 0 & 0 & 0 & \cdots & 0 & 0\\
-\frac{10}{9} & 0 & 0 & 0 & 0 & \frac{163}{36} & -\frac{7}{6} & 0 & 0 & \cdots & 0 & 0\\
0 & -\frac{10}{9} & 0 & 0 & 0 & -\frac{4}{3} & \frac{175}{36} & -\frac{4}{3} & 0 & \cdots & 0 & 0\\
0 & 0 & -\frac{10}{9} & 0 & 0 & 0 & -\frac{3}{2} & \frac{187}{36} & -\frac{3}{2} & \cdots & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{-3}{2} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{241}{36} & -\frac{5}{3}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -\frac{11}{6} & \frac{253}{36}
\end{pmatrix}$$

Matrix A is 25×25 dimension while b and z are 25×1 dimensions, the above linear system is then solved with the new EAOR iterative method, the proposed REAOR iterative method, the classical AOR method, QAOR method by Wu and Liu (2014), KAOR method by Youssef and Farid (2015) and RAOR by Vatti *et al*., (2018). The results are tabulated and discussed in chapter four.

3.5.5: **Problem 5** (Application Problem 1)

In this section, we shall apply the proposed iterative methods to a fuzzy linear equations. Specifically solving a square 6×6 fuzzy linear problem using the new EAOR and REAOR methods. Solve the square fuzzy linear equations from Lubna and Naji (2018) with the proposed EAOR and REAOR iterative methods

$$\left.\begin{aligned} 9x_1 + 2x_2 - x_3 + x_4 + x_5 - 2x_6 &= (-53 + 8\alpha, -25 - 20\alpha) \\ -x_1 + 10x_2 + 2x_3 + x_4 - x_5 - x_6 &= (-13 + 9\alpha, 18 - 22\alpha) \\ x_1 + 3x_2 - x_3 + x_4 + x_5 - 2x_6 &= (18 + 17\alpha, 73 - 38\alpha) \\ 2x_1 - x_2 + x_3 + 10x_4 - 2x_5 + 3x_6 &= (31 + 16\alpha, 61 - 14\alpha) \\ x_1 + x_2 - x_3 + 2x_4 + 7x_5 - x_6 &= (34 + 8\alpha, 58 - 16\alpha) \\ 3x_1 + 2x_2 + x_3 + x_4 - x_5 + 10x_6 &= (51 + 26\alpha, 99 - 22\alpha) \end{aligned}\right\} \quad (3.181)$$

Fuzzy linear systems cannot be solved directly, so the standard approach is to reduce it to crisp linear system so as to make it easier for computation using the proposed iterative methods. The fuzzy linear equations in (3.181) is then transformed into matrix notations through the use of embedding method (EM). The EM helps in extending the fuzzy linear system into crisp linear system.

To define a solution $(x_1, x_2, \cdots, x_m)^T$ to the fuzzy system in (3.181), some arithmetic operations needs to be performed on the fuzzy numbers $x = \left(\underline{x}(\alpha), \overline{x}(\alpha)\right)$, $z = \left(\underline{z}(\alpha), \overline{z}(\alpha)\right)$ and $K \in \Re$. The arithmetic operators on the fuzzy numbers are defined as

I. $x = z$ if and only if $\underline{x}(\alpha) = \underline{z}(\alpha)$ and $\overline{x}(\alpha) = \overline{z}(\alpha)$

II. $x + z = \left(\underline{x}(\alpha) + \underline{z}(\alpha);\ \overline{x}(\alpha) + \overline{z}(\alpha)\right)$

III. $Kx = \begin{cases} \left(K\underline{x}, K\overline{x}\right), & \text{if } k \geq 0 \\ \left(K\overline{x}, K\underline{x}\right), & \text{if } k < 0 \end{cases}$

From the arithmetic operations on the fuzzy numbers in the system (3.181), we obtain the 6×6 system of fuzzy linear equations

$$(9\underline{x}_1, 9\overline{x}_1) + (2\underline{x}_2, 2\overline{x}_2) + (-\overline{x}_3, -\underline{x}_3,) + (\underline{x}_4, \overline{x}_4) + (\underline{x}_5, \overline{x}_5) + (-2\overline{x}_6, -2\underline{x}_6)$$

$$= (-53 + 8\alpha, -25 - 20\alpha) \tag{3.182}$$

$$(-\overline{x}_1, -\underline{x}_1) + (10\underline{x}_2, 10\overline{x}_2) + (2\underline{x}_3, 2\overline{x}_3) + (\underline{x}_4, \overline{x}_4) + (-\overline{x}_5, -\underline{x}_5) + (-\overline{x}_6, -\underline{x}_6)$$

$$= (-13 + 9\alpha, 18 - 22\alpha) \tag{3.183}$$

$$(\underline{x}_1, \overline{x}_1) + (3\underline{x}_2, 3\overline{x}_2) + (-\overline{x}_3, -\underline{x}_3,) + (\underline{x}_4, \overline{x}_4) + (\underline{x_5}, \overline{x_5}) + (-2\overline{x}_6, -2\underline{x}_6)$$

$$= (18 + 17\alpha, 73 - 38\alpha) \tag{3.184}$$

$$(2\underline{x}_1, 2\overline{x}_1) + (-\overline{x}_2, -\underline{x}_2) + (\underline{x}_3, \overline{x}_3) + (10\underline{x}_4, 10\overline{x}_4) + (-2\overline{x}_5, -2\underline{x}_5) + (3\underline{x}_6, 3\overline{x}_6)$$

$$= (31 + 16\alpha, 61 - 14\alpha) \tag{3.185}$$

$$(\underline{x}_1, \overline{x}_1) + (\underline{x}_2, \overline{x}_2) + (-\overline{x}_3, -\underline{x}_3) + (2\underline{x}_4, 2\overline{x}_4) + (7\underline{x}_5, 7\overline{x}_5) + (-\overline{x}_6, -\underline{x}_6)$$

$$= (34 + 8\alpha, 58 - 16\alpha) \tag{3.186}$$

$$(3\underline{x}_1, 3\overline{x}_1) + (2\underline{x}_2, 2\overline{x}_2) + (\underline{x}_3, \overline{x}_3) + (\underline{x}_4, \overline{x}_4) + (-\overline{x}_5, -\underline{x}_5) + (10\underline{x}_6, 10\overline{x}_6)$$

$$= (51 + 26\alpha, 99 - 22\alpha) \tag{3.187}$$

Picking out each of the first pairs for the linear equations in (3.182 to 3.187) and followed by the second pairs in each equations gives the following equations

$$\left.\begin{array}{r}
9\underline{x}_1 + 2\underline{x}_2 - \overline{x}_3 + \underline{x}_4 + \underline{x}_5 - 2\overline{x}_6 = -53 + 8\alpha \\
-\overline{x}_1 + 10\underline{x}_2 + 2\underline{x}_3 + \underline{x}_4 - \overline{x}_5 - \overline{x}_6 = -13 + 9\alpha \\
\underline{x}_1 + 3\underline{x}_2 - \overline{x}_3 + \underline{x}_4 + \underline{x}_5 - 2\overline{x}_6 = 18 + 17\alpha \\
2\underline{x}_1 - \overline{x}_2 + \underline{x}_3 + 10\underline{x}_4 - 2\overline{x}_5 + 3\underline{x}_6 = 31 + 16\alpha \\
\underline{x}_1 + \underline{x}_2 - \overline{x}_3 + \underline{x}_4 - \overline{x}_5 - \overline{x}_6 = 34 + 8\alpha \\
3\underline{x}_1 + 2\underline{x}_2 + \underline{x}_3 + \underline{x}_4 - \overline{x}_5 + 10\underline{x}_6 = 51 + 26\alpha \\
9\overline{x}_1 + 2\overline{x}_2 - \underline{x}_3 + \overline{x}_4 + \overline{x}_5 - 2\underline{x}_6 = -25 - 20\alpha \\
-\underline{x}_1 + 10\overline{x}_2 + 2\overline{x}_3 + \overline{x}_4 - \underline{x}_5 - \underline{x}_6 = 18 - 22\alpha \\
\overline{x}_1 + 3\overline{x}_2 - \overline{x}_3 + \overline{x}_4 + \overline{x}_5 - 2\underline{x}_6 = 73 - 38\alpha \\
2\overline{x}_1 - \underline{x}_2 + \overline{x}_3 + 10\overline{x}_4 - 2\underline{x}_5 + 3\overline{x}_6 = 61 - 14\alpha \\
\overline{x}_1 + \overline{x}_2 - \underline{x}_3 + 2\overline{x}_4 + 7\overline{x}_5 - \underline{x}_6 = 58 - 16\alpha \\
3\overline{x}_1 + 2\overline{x}_2 + \overline{x}_3 + \overline{x}_4 - \underline{x}_5 + 10\overline{x}_6 = 99 - 22\alpha
\end{array}\right\} \tag{3.188}$$

Where the size of the generated matrix is twice the size of the original fuzzy linear system in equation (3.181). The extended 12×12 matrix from (3.188) is

$$\begin{pmatrix} 9 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -2 \\ 0 & 10 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 3 & 9 & 0 & 1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 \\ 2 & 0 & 1 & 10 & 0 & 3 & 0 & -1 & 0 & 0 & -2 & 0 \\ 1 & 1 & 0 & 2 & 7 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 3 & 2 & 1 & 1 & 0 & 10 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -2 & 9 & 2 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & -1 & 0 & 10 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 3 & 9 & 0 & 1 & 2 \\ 0 & -1 & 0 & 0 & -2 & 0 & 2 & 0 & 1 & 10 & 0 & 3 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 0 & 2 & 7 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 3 & 2 & 1 & 1 & 0 & 10 \end{pmatrix} \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \\ \underline{x}_4 \\ \underline{x}_5 \\ \underline{x}_6 \\ \overline{x}_1 \\ \overline{x}_2 \\ \overline{x}_3 \\ \overline{x}_4 \\ \overline{x}_5 \\ \overline{x}_6 \end{pmatrix}$$

$$= \begin{pmatrix} -53 + 8\alpha \\ 13 + 9\alpha \\ 18 + 17\alpha \\ 31 + 16\alpha \\ 34 + 8\alpha \\ 51 + 26\alpha \\ -25 - 20\alpha \\ 18 - 22\alpha \\ 73 - 38\alpha \\ 61 - 14\alpha \\ 58 - 16\alpha \\ 99 - 22\alpha \end{pmatrix} \qquad (3.189)$$

Which is expressed in $Bz = f$, where

$$B = \begin{pmatrix} 9 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -2 \\ 0 & 10 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 3 & 9 & 0 & 1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 \\ 2 & 0 & 1 & 10 & 0 & 3 & 0 & -1 & 0 & 0 & -2 & 0 \\ 1 & 1 & 0 & 2 & 7 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 3 & 2 & 1 & 1 & 0 & 10 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -2 & 9 & 2 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & -1 & 0 & 10 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 3 & 9 & 0 & 1 & 2 \\ 0 & -1 & 0 & 0 & -2 & 0 & 2 & 0 & 1 & 10 & 0 & 3 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 0 & 2 & 7 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 3 & 2 & 1 & 1 & 0 & 10 \end{pmatrix}, z = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \\ \underline{x}_4 \\ \underline{x}_5 \\ \underline{x}_6 \\ \overline{x}_1 \\ \overline{x}_2 \\ \overline{x}_3 \\ \overline{x}_4 \\ \overline{x}_5 \\ \overline{x}_6 \end{pmatrix}$$

$$\text{and } f = \begin{pmatrix} -53+8\alpha \\ 13+9\alpha \\ 18+17\alpha \\ 31+16\alpha \\ 34+8\alpha \\ 51+26\alpha \\ -25-20r \\ 18-22\alpha \\ 73-38\alpha \\ 61-14\alpha \\ 58-16\alpha \\ 99-22\alpha \end{pmatrix} \tag{3.190}$$

The above linear system is multiplied by D^{-1} to obtain $A = D^{-1}B$. Furthermore, matrix A is decomposed into $A = I - L - U$ and then solved with the proposed EAOR method, Proposed REAOR method, the classical AOR method, QAOR method by Wu and Liu (2014), KAOR method by Youssef and Farid (2015) and RAOR by Vatti *et al*., (2018). The results are tabulated and discussed in chapter four.

3.5.6: Problem 6 (Application Problem 2)

In this section, we shall apply the proposed iterative methods to a real life problem. Specifically solving a two dimensional heat transfer problem using the new EAOR and REAOR methods. We consider a plate (metal) of size $0.9m \times 0.9m$ with its edges held at constant temperature shown in figure 3.4. What is the field's temperature developed within the plate while attaining steady state conditions? (Mayooran and Elliot, 2016).

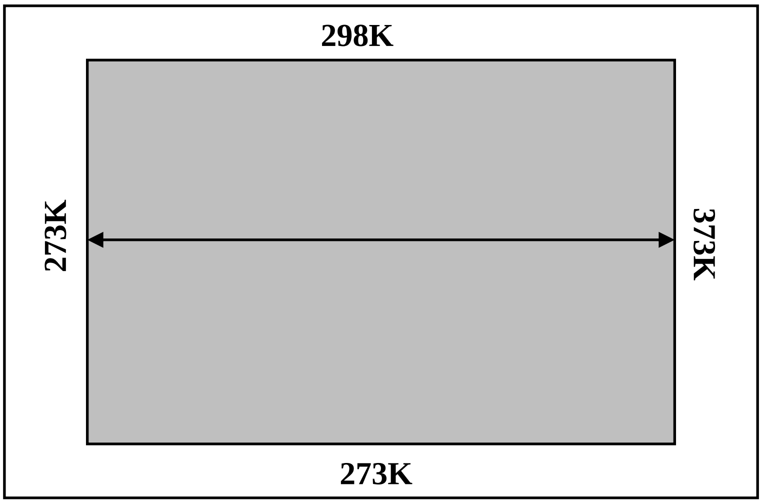

Figure 3.4: Metal plate with constant Temperature

Solution: Using a step-size $h = 0.1$, we divide the metal plate as shown in the figure 3.5 below to obtain 64 unknowns. Each cell in the figure represents the nodal temperature of each 0.1m x 0.1m element in the plate. It should be noted that the mesh points are not at the boundaries of the plate but only at the center of each element.

298	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**
273	$u_{1,8}$	$u_{2,8}$	$u_{3,8}$	$u_{4,8}$	$u_{5,8}$	$u_{6,8}$	$u_{7,8}$	$u_{8,8}$	**373**
273	$u_{1,7}$	$u_{2,7}$	$u_{3,7}$	$u_{4,7}$	$u_{5,7}$	$u_{6,7}$	$u_{7,7}$	$u_{8,7}$	**373**
273	$u_{1,6}$	$u_{2,6}$	$u_{3,6}$	$u_{4,6}$	$u_{5,6}$	$u_{6,6}$	$u_{7,6}$	$u_{8,6}$	**373**
273	$u_{1,5}$	$u_{2,5}$	$u_{3,5}$	$u_{4,5}$	$u_{5,5}$	$u_{6,5}$	$u_{7,5}$	$u_{8,5}$	**373**
273	$u_{1,4}$	$u_{2,4}$	$u_{3,4}$	$u_{4,4}$	$u_{5,4}$	$u_{6,4}$	$u_{7,4}$	$u_{8,4}$	**373**
273	$u_{1,3}$	$u_{2,3}$	$u_{3,3}$	$u_{4,3}$	$u_{5,3}$	$u_{6,3}$	$u_{7,3}$	$u_{8,3}$	**373**
273	$u_{1,2}$	$u_{2,2}$	$u_{3,2}$	$u_{4,2}$	$u_{5,2}$	$u_{6,2}$	$u_{7,2}$	$u_{8,2}$	**373**
273	$u_{1,1}$	$u_{2,1}$	$u_{3,1}$	$u_{4,1}$	$u_{5,1}$	$u_{6,1}$	$u_{7,1}$	$u_{8,1}$	**373**
273	**273**	**273**	**273**	**273**	**273**	**273**	**273**	**273**	**273**

Figure 3.5: Discretization of square mesh with boundary conditions for the metal plate

The partial differential equation governing the two dimensional steady state heat transfer problem is Laplace's equation of the form;

$$\frac{\partial^2}{\partial x^2}U(x,y) + \frac{\partial^2}{\partial y^2}U(x,y) = 0 \tag{3.191}$$

for

$$\begin{aligned} 0 \le x \le 0.9 \\ 0 \le y \le 0.9 \end{aligned} \tag{3.192}$$

From the metal plate, we can deduce the boundary conditions as

$$\begin{aligned} U(x,0) = 273, \qquad & U(0,y) = 273 \\ U(x,0.9) = 298, \qquad & U(0.9,y) = 373 \end{aligned} \tag{3.193}$$

In the discretization of the governing equation, we apply the central finite difference approximation and substitute the appropriate partial derivatives of $\frac{\partial^2 U}{\partial x^2}$ and $\frac{\partial^2 U}{\partial y^2}$ into the governing equation.

$$\frac{\partial^2 U}{\partial x^2} = \frac{U_{n+1,m} - 2U_{n,m} + U_{n-1,m}}{h^2}$$
$$\frac{\partial^2 U}{\partial y^2} = \frac{U_{n,m+1} - 2U_{n,m} + U_{n,m-1}}{k^2} \tag{3.194}$$

where the integer subscripts (n, m) is the position on $x - y$ axis respectively with a discrete value range. Then, the governing equation gives

$$\frac{U_{n+1,m} - 2U_{n,m} + U_{n-1,m}}{h^2} + \frac{U_{n,m+1} - 2U_{n,m} + U_{n,m-1}}{k^2} = 0 \tag{3.195}$$

For $k = h$ and simplifying further to obtain;

$$4U_{n,m} - U_{n+1,m} - U_{n-1,m} - U_{n,m+1} - U_{n,m-1} = 0 \tag{3.196}$$

We then apply the discretized form of the model equation of the metal plate to each of the internal mesh points of the plate to obtain the following set of 64 system of linear equations below;

$$4u_{1,1} - u_{2.1} - u_{1,2} = 546$$
$$4u_{2,1} - u_{3,1} - u_{1,1} - u_{2,2} = 273$$
$$4u_{3,1} - u_{4,1} - u_{2,1} - u_{3,2} = 273$$
$$4u_{4,1} - u_{5,1} - u_{3,1} - u_{4,2} = 273$$
$$4u_{5,1} - u_{6,1} - u_{4,1} - u_{5,2} = 273$$
$$4u_{6,1} - u_{7,1} - u_{5,1} - u_{6,2} = 273$$
$$4u_{7,1} - u_{8,1} - u_{6,1} - u_{7,2} = 273$$
$$4u_{8,1} - u_{7,1} - u_{8,2} = 646$$
$$4u_{1,2} - u_{2,2} - u_{1,3} - u_{1,1} = 273$$
$$4u_{2,2} - u_{3,2} - u_{1,2} - u_{2,3} - u_{2,1} = 0$$
$$4u_{3,2} - u_{4,2} - u_{2,2} - u_{3,3} - u_{3,1} = 0$$
$$4u_{4,2} - u_{5,2} - u_{3,2} - u_{4,3} - u_{4,1} = 0$$
$$4u_{5,2} - u_{6,2} - u_{4,2} - u_{5,3} - u_{5,1} = 0$$
$$4u_{6,2} - u_{7,2} - u_{5,2} - u_{6,3} - u_{6,1} = 0$$
$$4u_{7,2} - u_{8,2} - u_{6,2} - u_{7,3} - u_{7,1} = 0$$
$$4u_{8,2} - u_{7,2} - u_{8,3} - u_{8,1} = 373$$
$$4u_{1,3} - u_{2,3} - u_{1,4} - u_{1,2} = 273$$
$$4u_{2,3} - u_{3,3} - u_{1,3} - u_{2,4} - u_{2,2} = 0$$
$$4u_{3,3} - u_{4,3} - u_{2,3} - u_{3,4} - u_{3,2} = 0$$
$$4u_{4,3} - u_{5,3} - u_{3,3} - u_{4.4} - u_{4,2} = 0$$
$$4u_{5,3} - u_{6,3} - u_{4,3} - u_{5,4} - u_{5,2} = 0$$
$$4u_{6,3} - u_{7,3} - u_{5,3} - u_{6,4} - u_{6,2} = 0$$
$$4u_{7,3} - u_{8,3} - u_{6,3} - u_{7,4} - u_{7,2} = 0$$
$$4u_{8,3} - u_{7,3} - u_{8,4} - u_{8,2} = 373$$
$$4u_{1,4} - u_{2,4} - u_{1,5} - u_{1,3} = 273$$
$$4u_{2,4} - u_{3,4} - u_{1,4} - u_{2,5} - u_{2,3} = 0$$
$$4u_{3,4} - u_{4,4} - u_{2,4} - u_{3,5} - u_{3,3} = 0$$
$$4u_{4,4} - u_{5,4} - u_{3,4} - u_{4,5} - u_{4,3} = 0$$
$$4u_{5,4} - u_{6,4} - u_{4,4} - u_{5,5} - u_{5,3} = 0$$
$$4u_{6,4} - u_{7,4} - u_{5,4} - u_{6,5} - u_{6,3} = 0$$
$$4u_{7,4} - u_{8,4} - u_{6,4} - u_{7,5} - u_{7,3} = 0$$
$$4u_{8,4} - u_{7,4} - u_{8,5} - u_{8,3} = 373$$
$$4u_{1,5} - u_{2,5} - u_{1,6} - u_{1,4} = 273$$
$$4u_{2,5} - u_{3,5} - u_{1,5} - u_{2,6} - u_{2,4} = 0$$
$$4u_{3,5} - u_{4,5} - u_{2,5} - u_{3,6} - u_{3,4} = 0$$
$$4u_{4,5} - u_{5,5} - u_{3,5} - u_{4,6} - u_{4,4} = 0$$
$$4u_{5,5} - u_{6,5} - u_{4,5} - u_{5,6} - u_{5,4} = 0$$
$$4u_{6,5} - u_{7,5} - u_{5,5} - u_{6,6} - u_{6,4} = 0$$
$$4u_{7,5} - u_{8,5} - u_{6,5} - u_{7,6} - u_{7,4} = 0$$
$$4u_{8,5} - u_{7,5} - u_{8,6} - u_{8,4} = 373$$
$$4u_{1,6} - u_{2,6} - u_{1,7} - u_{1,5} = 273$$
$$4u_{2,6} - u_{3,6} - u_{1,6} - u_{2,7} - u_{2,5} = 0$$
$$4u_{3,6} - u_{4,6} - u_{2,6} - u_{3,7} - u_{35,} = 0$$
$$4u_{4,6} - u_{5,6} - u_{3,6} - u_{4,7} - u_{4,5} = 0$$
$$4u_{5,6} - u_{6,6} - u_{4,6} - u_{5,7} - u_{5,5} = 0$$
$$4u_{6,6} - u_{7,6} - u_{5,6} - u_{6,7} - u_{6,5} = 0$$
$$4u_{7,6} - u_{8,6} - u_{6,6} - u_{7,7} - u_{7,5} = 0$$
$$4u_{8,6} - u_{7,6} - u_{8,7} - u_{8,5} = 373$$

$$
\begin{gathered}
4u_{1,7} -, u_{2,7} -, u_{1,8} -, u_{1,6} = 273 \\
4u_{2,7} - u_{3,7} - u_{1,7} - u_{2,8} - u_{2,6} = 0 \\
4u_{3,7} - u_{4,7} - u_{2,7} - u_{3,8} - u_{3,6} = 0 \\
4u_{4,7} - u_{5,7} - u_{3,7} - u_{4,8} - u_{4,6} = 0 \\
4u_{5,7} - u_{6,7} - u_{4,7} - u_{5,8} - u_{5,6} = 0 \\
4u_{6,7} - u_{7,7} - u_{5,7} - u_{6,8} - u_{6,6} = 0 \\
4u_{7,7} - u_{8,7} - u_{6,7} - u_{7,8} - u_{7,6} = 0 \\
4u_{8,7} - u_{7,7} - u_{8,8} - u_{8,6} = 373 \\
4u_{1,8} - u_{2,8} - u_{1,7} = 571 \\
4u_{2,8} - u_{3,8} - u_{1,8} - u_{2,7} = 298 \\
4u_{3,8} - u_{4,8} - u_{2,8} - u_{3,7} = 298 \\
4u_{4,8} - u_{5,8} - u_{3,8} - u_{4,7} = 298 \\
4u_{5,8} - u_{6,8} - u_{4,8} - u_{5,7} = 298 \\
4u_{6,8} - u_{7,8} - u_{5,8} - u_{6,7} = 298 \\
4u_{7,8} - u_{8,8} - u_{6,8} - u_{7,7} = 298 \\
4u_{8,8} - u_{7,8} - u_{8,7} = 671
\end{gathered}
\tag{3.197}
$$

Representing the above system of linear equations in matrix form of 64×64 dimension, a large sparse 64×64 linear system is obtained.

$$
A = \begin{pmatrix}
4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 4
\end{pmatrix}
$$

$$z = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \\ u_{5,1} \\ u_{6,1} \\ u_{7,1} \\ u_{8,1} \\ u_{1,2} \\ u_{2,2} \\ u_{2,3} \\ \vdots \\ u_{5,8} \\ u_{6,8} \\ u_{7,8} \\ u_{8,8} \end{pmatrix}, \qquad b = \begin{pmatrix} 546 \\ 273 \\ 273 \\ 273 \\ 273 \\ 273 \\ 273 \\ 646 \\ 273 \\ 0 \\ 0 \\ \vdots \\ 298 \\ 298 \\ 298 \\ 671 \end{pmatrix}$$

The spectral radii and convergence results using the newly developed iterative methods, QAOR method by Wu and Liu (2014), KAOR method by Youssef and Farid (2015) and RAOR method by Vatti *et al*. (2018) are computed. The numerical results are presented and discussed in chapter four.

3.8 Convergence Rate of the New Methods

It is essentially important to know that an iteration method converges as well as the quest to know how fast it converges. Young (2014), introduced the expression

$$R(J) = -\log \rho(J) \tag{3.198}$$

This is used to compute the rate of convergence of linear iteration methods of the form $z^{(k+1)} = Jz^{(k)} + C$, where $\rho(J)$ represents the spectral radius of a particular iteration method and J converges as long as $\rho(J) < 1$. The newly developed parameterized iterative methods, AOR methods and its variants tend to converge rapidly based on the choice of the over relaxation parameter, acceleration parameter and extended acceleration parameter involved in them.

With the aid of Maple 2017 software, the convergence rates of the proposed EAOR and REAOR iterative methods derived in this study are calculated using equation (3.198) with AOR methods for comparison purpose. The computational results of the comparisons are presented in the next chapter.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Results of Problem 1 (see page 71)

The spectral radii for the iteration matrices of the proposed EAOR and REAOR methods were calculated using Maple 2017 software with methods of the classical AOR, Wu and Liu (2014), Youssef and Farid (2015) and Vatti *et al.*, (2018). The following notations are used for the comparison and clarity of the results obtained for the numerical experiments in chapter three;

J_{AOR} = Iteration matrix of the AOR method

J_{EAOR} = Iteration matrix of the proposed EAOR method

J_{REAOR} = Iteration matrix of the proposed Refinement of EAOR method

$J_{\text{Wu \& Liu,2014}}$ = Iteration matrix of the QAOR method of Wu and Liu (2014)

$J_{\text{Youssef \& Farid,2015}}$ = Iteration matrix of the KAOR method of Youssef and Farid (2015)

$J_{\text{Vatti } et\ al.,\ 2018}$ = Iteration matrix of the Refinement of AOR method of Vatti *et al.* (2018)

$\rho(J_{AOR})$ = Spectral radius of J_{AOR}.

$\rho(J_{EAOR})$ = Spectral radius of J_{EAOR}.

$\rho(J_{REAOR})$ = Spectral radius of J_{REAOR}.

$\rho(J_{\text{Wu \& Liu,2014}})$ = Spectral radius of $J_{\text{Wu \& Liu,2014}}$.

$\rho(J_{\text{Youssef \& Farid,2015}})$ = Spectral radius of $J_{\text{Youssef \& Farid,2015}}$.

$\rho(J_{\text{Vatti } et\ al.,2018})$ = Spectral radius of $J_{\text{Vatti } et\ al.,2018}$.

In addition to the above, their methods and iteration matrices are represented as:

AOR

$$z^{(k+1)} = (I - rL)^{-1}[(1-\omega)I + (\omega - r)L + \omega U]z^{(k)} + (I - rL)^{-1}\omega\dot{b}$$

$$\boldsymbol{J_{AOR}} = (I - rL)^{-1}[(1-\omega)I + (\omega - r)L + \omega U]$$

Proposed EAOR:

$$z^{(k+1)} = [I - vL - rL]^{-1}[(1-\omega)I + [\omega - (v+r)]L + \omega U]z^{(k)}$$
$$+ [I - vL - rL]^{-1}\omega\dot{b}$$

$$\boldsymbol{J_{EAOR}} = [I - vL - rL]^{-1}[(1-\omega)I + [\omega - (v+r)]L + \omega U]$$

Proposed REAOR

$$\bar{z}^{(k+1)} = \Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2 z^{(k)} +$$
$$\Big(I + (I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)(I-(v+r)L)^{-1}\omega\dot{b}$$

$$\boldsymbol{J_{REAOR}} = \Big((I-(v+r)L)^{-1}\big((1-\omega)I + [\omega-(v+r)]L + \omega U\big)\Big)^2$$

QAOR:

$$z^{(k+1)} = ([(1+\omega)]I - rL)^{-1}[I + (\omega - r)L + \omega U]z^{(k)} + ([(1+\omega)]I - rL)^{-1}\omega\dot{b}$$

$$\boldsymbol{J}_{\textbf{Wu \& Liu,2014}} = ([(1+\omega)]I - rL)^{-1}[I + (\omega - r)L + \omega U]$$

KAOR:

$$z^{(k+1)} = ([(1+r)]I - rL)^{-1}[(1+r-\omega)I + (\omega - r)L + \omega U]z^{(k)}$$
$$+([(1+r)]I - rL)^{-1}\omega\dot{b}$$

$$\boldsymbol{J}_{\textbf{Youssef \& Farid,2015}} = ([(1+r)]I - rL)^{-1}[(1+r-\omega)I + (\omega - r)L + \omega U]$$

RAOR:

$$\bar{z}^{(k+1)} = \Big((I-(r)L)^{-1}\big((1-\omega)I + [\omega - r]L + \omega U\big)\Big)^2 z^{(k)}$$
$$+(I + [I - rL]^{-1}[(1-\omega)I + [\omega - r]L + \omega U])[I - rL]^{-1}\omega\dot{b}$$

$$\boldsymbol{J}_{\textbf{Vatti }\textbf{\textit{et al.}}\textbf{,2018}} = \Big((I-(r)L)^{-1}\big((1-\omega)I + [\omega - r]L + \omega U\big)\Big)^2$$

$R(J_{QAOR})$ = Rate of convergence of the QAOR linear iteration matrix $J_{\text{Wu \& Liu,2014}}$

$R(J_{\text{Vatti } et\ al.\text{,2018}}) =$ Rate of convergence of the RAOR linear iteration matrix $J_{\text{Vatti } et\ al.\text{,2018}}$

$R(J_{KAOR}) =$ Rate of convergence of the KAOR linear iteration matrix $J_{\text{Youssef and Farid,2015}}$

$R(J_{REAOR})$ = Rate of convergence of the REAOR linear iteration matrix J_{REAOR}

$R(J_{EAOR})$ = Rate of convergence of the EAOR linear iteration matrix J_{EAOR}

Also, the conditions placed on the coefficient matrix for the various iterative methods considered for comparison are stated as follows;

AOR

$L -$ matrix: $0 \leq r \leq \omega \leq 1$,

Irreducible weak diagonally dominant: $0 \leq r \leq 1$ and $0 < \omega \leq 1$

QAOR

$L -$ matrix: $0 \leq r \leq \omega$, $[\omega \neq 0]$

Irreducible weak diagonally dominant: $-1 \leq r \leq 1$ and $\omega > 0$

KAOR

$L -$ matrix: $0 < r < \omega < r + 1$, $[\omega \neq 0]$

Irreducible weak diagonally dominant: $0 < k$, and $0 < \omega$

EAOR

$L -$ matrix: $0 < v + r \leq \omega \leq 1$, $v \neq 0$ and $\omega \neq 0$

Irreducible weak diagonally dominant: $0 < v + r \leq 1$, $0 < \omega \leq 1$ and $v \neq 0$

RAOR

$L -$ matrix: $0 \leq r \leq \omega \leq 1$,

Irreducible weak diagonally dominant: $0 \leq r \leq 1$ and $0 < \omega \leq 1$

REAOR

$L-$ matrix: $0 < v + r \leq \omega \leq 1, \omega \neq 0$ and $v \neq 0$

Irreducible weak diagonally dominant: $0 < v + r \leq 1,\ 0 < \omega \leq 1,\ v \neq 0$ and $\omega \neq 0$

A point to note is that the rate of convergence of the various iterative methods is best when the spectral radius is near zero and poorest when is near one.

4.1.1 Comparison of the Proposed Methods

Table 4.1: Results of spectral radii of EAOR and REAOR iteration matrices for problem 1

$\boldsymbol{\omega}$	$\boldsymbol{r}$	$\boldsymbol{v}$	$\boldsymbol{\rho(J_{EAOR})}$	$\boldsymbol{\rho(J_{REAOR})}$
0.1	0.04	0.05	0.9549337380	0.9118984439
0.2	0.08	0.10	0.9068702250	0.8224136050
0.3	0.12	0.15	0.8554124670	0.7317304887
0.4	0.16	0.20	0.8000680959	0.6401089581
0.5	0.20	0.25	0.7402111025	0.5479124763
0.6	0.24	0.30	0.6750196533	0.4556515324
0.7	0.28	0.35	0.6033676553	0.3640525275
0.8	0.32	0.40	0.5236168516	0.2741746073
0.9	0.36	0.45	0.4331605794	0.1876280875

Table 4.1 shows the performance of the new Extended Accelerated Over Relaxation and Refinement of Extended Accelerated Over Relaxation methods for problem 1. For values of the relaxation parameter ω, acceleration parameter r and extended acceleration parameter v, the outcome clearly reveals that the spectral radius of the proposed Refined EAOR is lesser than that of EAOR method $[\rho(J_{REAOR}) < \rho(J_{EAOR}) < 1]$.

4.1.2 Comparison of the EAOR Method with variants of AOR Methods

Table 4.2: Results of spectral radii of AOR, its variants and EAOR iteration matrices for problem 1

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu,2014}})$	$\rho(J_{\text{Youssef \& Farid,2015}})$	v	$\rho(EAOR)$
0.1	0.04	0.9557161302	0.9597924888	0.9574419974	0.05	0.9549337380
0.2	0.08	0.9101858613	0.9255050783	0.9170122503	0.10	0.9068702250
0.3	0.12	0.8633429922	0.8959156995	0.8785470488	0.15	0.8554124670
0.4	0.16	0.8151154107	0.8701178584	0.8418995087	0.20	0.8000680959
0.5	0.20	0.7654242883	0.8474247347	0.8069374273	0.25	0.7402111025
0.6	0.24	0.7141831211	0.8273063106	0.7735414626	0.30	0.6750196533
0.7	0.28	0.6612965781	0.8093469574	0.7416035804	0.35	0.6033676553
0.8	0.32	0.6066591050	0.7932161041	0.7110257234	0.40	0.5236168516
0.9	0.36	0.5501532158	0.7786475031	0.6817186663	0.45	0.4331605794

Table 4.2 shows the various spectral radii of AOR, some of its variants and proposed EAOR methods for problem 1 with values of the parameters $\omega, v,$ and r. Obviously, spectral radius of the proposed EAOR method is smaller than those of the KAOR, QAOR and AOR methods which inform us that the method of the EAOR has the tendency to converge faster than the other methods compared.

4.1.3 Comparison of Refinement of AOR and Refinement of EAOR Methods

Table 4.3: Results of spectral radii of RAOR and REAOR iteration matrices for problem 1

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.1	0.04	0.9133933215	0.05	0.9118984439
0.2	0.08	0.8284383021	0.10	0.8224136050
0.3	0.12	0.7453611222	0.15	0.7317304887
0.4	0.16	0.6644131328	0.20	0.6401089581
0.5	0.20	0.5858743411	0.25	0.5479124763
0.6	0.24	0.5100575304	0.30	0.4556515324
0.7	0.28	0.4373131642	0.35	0.3640525275
0.8	0.32	0.3680352697	0.40	0.2741746073
0.9	0.36	0.3026685609	0.45	0.1876280875

Table 4.3 displays the comparison of the spectral radii of refinement of AOR and proposed refinement of EAOR schemes for problem 1. The spectral radius of the proposed REAOR iterative method is smaller in comparison with Vatti *et al*. (2018) $[\rho(J_{REAOR}) < \rho(J_{\text{Vatti } et\ al.,2018}) < 1]$ by checking how close their spectrums are to 1 with different values of the parameters.

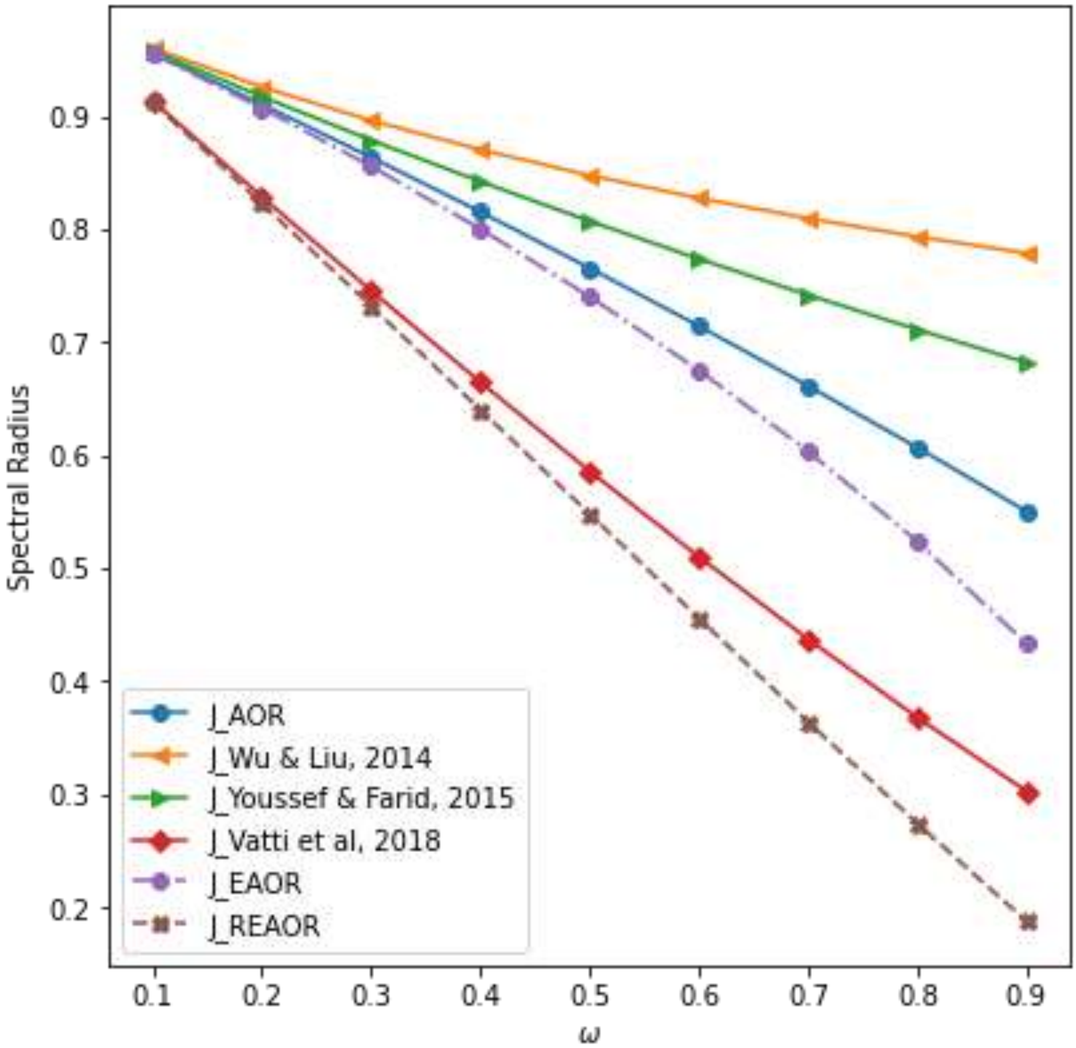

Figure 4.1: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.,2018}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu},2014}$ and $J_{\text{Youssef \& Farid},2015}$ Iteration Matrices for Problem 1

The above figure depicts the performance of the spectral radii of problem 1 and comparison between the newly developed schemes and the methods from existing literature. It is observed that J_{REAOR} has the least spectral radius, which shows that J_{REAOR} will outperform all the compared methods.

4.1.4 Comparison of Rates of convergence of EAOR and Existing Methods

Table 4.4: Results of convergence rate of EAOR and existing methods for problem 1

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.01782265312	0.01888752647	0.020026763	1.123669011	1.060316873
0.03362119391	0.03762486258	0.042454857	1.262740910	1.128372413
0.04773285302	0.05623497579	0.067824425	1.420917056	1.206089699
0.06042191774	0.07473974392	0.096873047	1.603276603	1.296138331
0.07189886393	0.09316014066	0.130644405	1.817057987	1.402363759
0.08233366291	0.1115164027	0.170683582	2.073071645	1.530569296
0.09186526160	0.1298281821	0.219417975	2.388476026	1.690064295
0.1006084772	0.1481146871	0.280986385	2.792869874	1.897086579
0.1086591047	0.1663948143	0.363351074	3.343954241	2.183668256

The above table presents rates of convergence of the new EAOR method in relation to some variants of AOR iterative method concerning Problem 1. Apparently, with all values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$ in table 4.4, the proposed EAOR iterative method will converge quicker by a factor of approximately 1.488296611 than the KAOR method and 1.980670372 faster than the QAOR method.

4.1.5 Comparison of Rates of convergence of RAOR and the REAOR Method

Table 4.5: Results of convergence rate of RAOR and REAOR methods for problem 1

$R(J_{\text{Vatti } et\ al.,2018})$	$R(J_{REAOR})$	$R(J_{REAOR})/R(J_{\text{Vatti } et\ al.,2018})$
0.039342168	0.040053525	1.018081292
0.08173983	0.084909714	1.038780161
0.127633264	0.135648849	1.062801696
0.177561792	0.193746095	1.091147442
0.232195522	0.26128881	1.12529651
0.292380836	0.341367165	1.167542885
0.359207449	0.438835949	1.22167831
0.43411056	0.56197277	1.294538355
0.519032688	0.726702148	1.400108634

The table above shows rates of convergence of refinement of the proposed Extended Accelerated Over relaxation method in relation to refinement of AOR method concerning Problem 1. Evidently, for all values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$ the proposed REAOR method is likely to converge quicker than the RAOR by a factor of approximately 1.2 times.

The True Solution of problem 1 by method of finite difference is:

$$\begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix} = \begin{pmatrix} \frac{33639875}{18302} \\ \frac{19040500}{9151} \\ \frac{42383125}{18302} \\ \frac{16210750}{9151} \\ \frac{10697500}{9151} \\ \frac{12911000}{9151} \\ \frac{29303875}{18302} \\ \frac{13668250}{9151} \\ \frac{25136125}{18302} \end{pmatrix} \xrightarrow{\text{at 10 decimal places}} \begin{pmatrix} 1838.0436564310 \\ 2080.7015626707 \\ 2315.7646705278 \\ 1771.4730630532 \\ 1168.9979237242 \\ 1410.8840563873 \\ 1601.1296579609 \\ 1493.6345754562 \\ 1373.4086438641 \end{pmatrix}$$

An accuracy of 10 decimal places was utilize to verify the convergence result of problem 1

4.1.6 Convergence Results Comparison for Problem 1 (see Appendix A)

Table 4.6: Summary of convergence result for problem 1

ITERATIVE	NUMBER OF	CPU TIME

METHODS	ITERATIONS	(seconds)
AOR	46	0.672
QAOR	117	0.734
KAOR	75	0.703
EAOR	32	0.515
RAOR	24	0.500
REAOR	17	0.453

The above table displays the summary of the convergence results of the various methods examined. Analysis of the results reveals that the EAOR method takes a shorter time (0.515 secs) to compute the 9 x 9 linear system to the desired accuracy compared to the other AOR-type methods. The REAOR method takes 0.453 secs as compared to 0.500 secs of RAOR method. This indicates that the new methods demonstrate efficiency as compared to their counterparts.

4.2: Results of Problem 2 (see page 84)

4.2.1 Comparison of the Two Proposed Methods

Table 4.7: Results of spectral radii of EAOR and REAOR iteration matrices for Problem 2

ω	r	v	$\rho(J_{EAOR})$	$\rho(J_{REAOR})$
0.1	0.04	0.05	0.9690220033	0.9390036428
0.2	0.08	0.10	0.9359663473	0.8760330033
0.3	0.12	0.15	0.9005738634	0.8110332834
0.4	0.16	0.20	0.8625309526	0.7439596442
0.5	0.20	0.25	0.8214516534	0.6747828189
0.6	0.24	0.30	0.7768509636	0.6034974196
0.7	0.28	0.35	0.7281033573	0.5301344989
0.8	0.32	0.40	0.6743746009	0.4547811023
0.9	0.36	0.45	0.6145013535	0.3776119135

Table 4.7 shows the performance of the proposed Extended AOR and refinement of the Extended AOR methods for problem 2. The results shows that the spectral radius of the proposed refined EAOR scheme is smaller than the spectral radius of the new EAOR method.

4.2.2 Comparison of the new Method with variants of AOR Methods

Table 4.8: Results of spectral radii of AOR, its variants and EAOR iteration matrices for problem 2

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu,2014}})$	$\rho(J_{\text{Youssef \& Farid,2015}})$	v	$\rho(EAOR)$
0.1	0.04	0.9695659299	0.9723678417	0.9707522261	0.05	0.9690220033
0.2	0.08	0.9382651880	0.9487977104	0.9429586969	0.10	0.9359663473
0.3	0.12	0.9060538059	0.9284531175	0.9165095988	0.15	0.9005738634
0.4	0.16	0.8728842724	0.9107129453	0.8913061678	0.20	0.8625309526
0.5	0.20	0.8387051160	0.8951062120	0.8672593113	0.25	0.8214516534
0.6	0.24	0.8034604202	0.8823532346	0.8407111735	0.30	0.7768509636
0.7	0.28	0.7670892570	0.8700519883	0.8183365049	0.35	0.7281033573
0.8	0.32	0.7295250213	0.8578213961	0.8012888150	0.40	0.6743746009
0.9	0.36	0.6906946434	0.8478005714	0.7811329984	0.45	0.6145013535

Table 4.8 shows the various spectral radii of AOR, its variants and proposed EAOR methods for problem 2 with different values of all the parameters $\omega, v,$ and r. Obviously, spectral radius of the proposed EAOR method is smaller than those of the existing methods compared which indicates that the proposed EAOR performs better.

4.2.3 Comparison of Refined AOR with the Proposed Refined EAOR Methods

Table 4.9: Results of spectral radii of RAOR and REAOR iteration matrices for problem 2

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.1	0.04	0.9400580924	0.05	0.9390036428
0.2	0.08	0.8803415630	0.10	0.8760330033
0.3	0.12	0.8209334992	0.15	0.8110332834
0.4	0.16	0.7619269530	0.20	0.7439596442
0.5	0.20	0.7034262716	0.25	0.6747828189
0.6	0.24	0.6455486468	0.30	0.6034974196
0.7	0.28	0.5884259282	0.35	0.5301344989
0.8	0.32	0.5322067567	0.40	0.4547811023
0.9	0.36	0.4770590904	0.45	0.3776119135

Table 4.9 reveals the comparison of the spectral radii of refinement of AOR and Refinement of EAOR methods for problem 2. The spectral radius of the proposed REAOR method is lesser in comparison with Vatti *et al.* (2018) that is to say $\rho(J_{REAOR}) < \rho(J_{\text{Vatti } et\ al.,\ 2018}) < 1$ and this indicates that the proposed REAOR method performs better than the RAOR method.

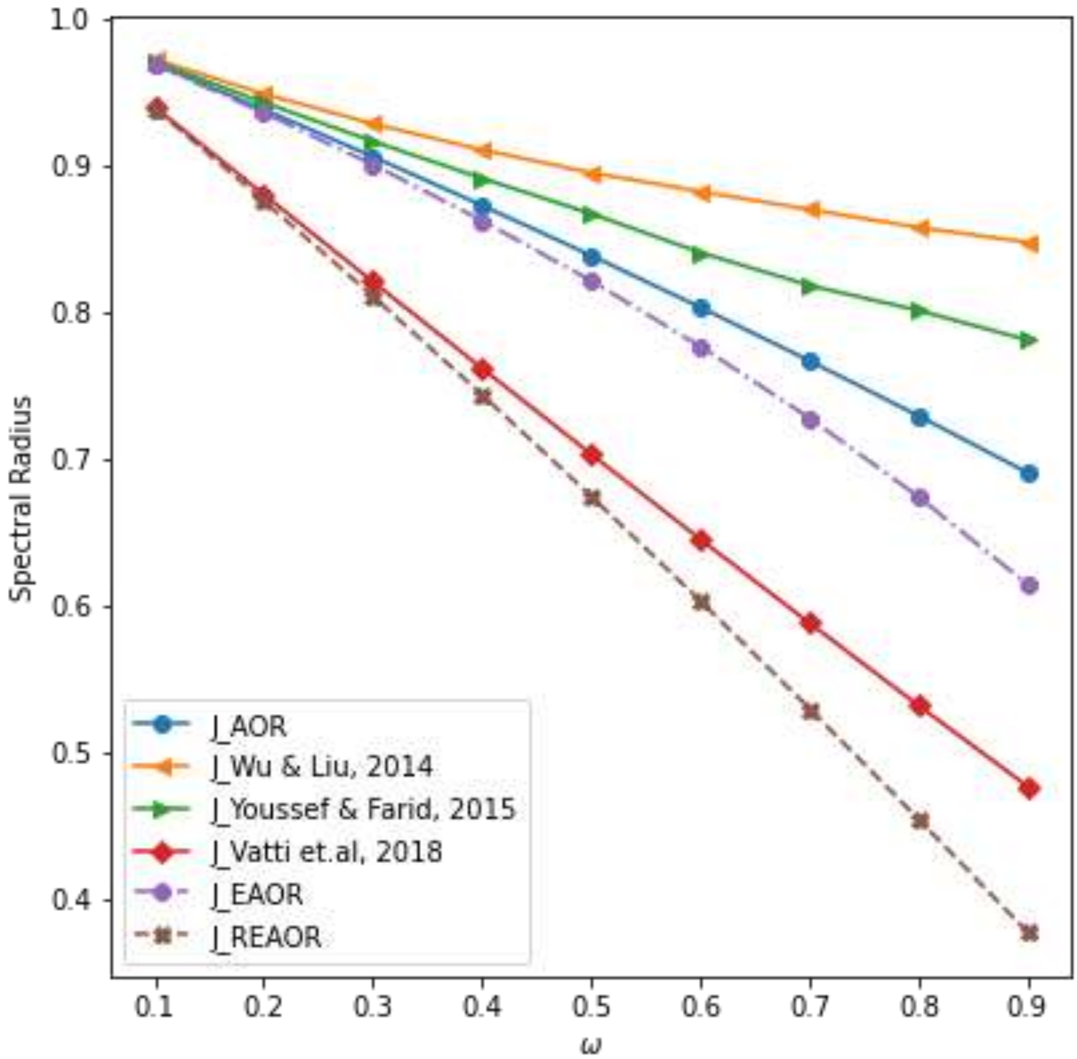

Figure 4.2: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.,2018}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu},2014}$ and $J_{\text{Youssef \& Farid},2015}$ Iteration Matrices for Problem 2

Figure 4.2 illustrates the performance of the proposed methods and some variants of AOR methods with respect to their spectral radii for problem 2. Clearly, the spectral radius of the J_{EAOR} is lesser than its counterpart. Likewise, the J_{REAOR} outperformed the $J_{\text{Vatti } et\ al.,2018}$ in terms of their spectral radii.

4.2.4 Comparison of Rates of convergence of EAOR and Existing Methods

Table 4.10: Results of convergence rate of EAOR and existing methods for problem 2

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.01216941265	0.01289160487	0.020026763	1.123009124	1.060097760
0.02282637200	0.02550732964	0.042454857	1.259059742	1.126725787
0.03224002119	0.03786298222	0.067824425	1.410689561	1.201190680
0.04061849012	0.04997308817	0.096873047	1.581184106	1.285197960
0.04812542888	0.06185102867	0.130644405	1.774903495	1.381027830
0.05435751790	0.07535318054	0.170683582	2.017426391	1.455310717
0.06045479617	0.08706807509	0.219417975	2.279504276	1.582749662
0.06660312563	0.09621091943	0.280986385	2.568930412	1.778371895
0.07170629506	0.1072750154	0.363351074	2.949213259	1.971355170

The above table presents rates of convergence of the new EAOR method in relation to variants of AOR iterative method concerning Problem 2. Apparently, with different values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$, the proposed EAOR iterative method converges quicker by a factor of approximately 1.426891940 than the KAOR method and 1.884880041 than the QAOR method.

4.2.5 Comparison of Rates of convergence of the RAOR and REAOR Methods

Table 4.11: Results of convergence rate of RAOR and REAOR methods for problem 2

$R(J_{\text{Vatti } et\ al., 2018})$	$R(J_{REAOR})$	$\frac{R(J_{REAOR})}{R(J_{\text{Vatti } et\ al., 2018})}$
0.02684530765	0.02733272292	1.018156442
0.05534879359	0.05747953210	1.038496566
0.08569202205	0.09096132273	1.061491147
0.1180866631	0.1284506220	1.087765702
0.1527814157	0.1708359838	1.118172541
0.1900710249	0.2193245825	1.153908559
0.2303081989	0.2756139328	1.196717851
0.2739196161	0.3421975900	1.249262812
0.3214278242	0.4229543123	1.315860920

The table above shows rates of convergence of the proposed REAOR method, $R(J_{REAOR})$ with RAOR method, $R(J_{RAOR})$ concerning Problem 2. Evidently, for all values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$ used in table 4.11, the proposed REAOR method converges quicker than the RAOR method, by a factor of approximately 1.2 times.

The True Solution of problem 2 by $\boldsymbol{z = A^{-1}b}$ is:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{pmatrix} = \xrightarrow{\text{at 10 decimal places}} \begin{pmatrix} 1.6995918367 \\ 1.3314285714 \\ 1.1551020408 \\ 1.1885714286 \\ 1.1551020408 \\ 1.1885714286 \\ 1.1551020408 \\ 1.1885714286 \\ 1.1551020408 \\ 1.1885714286 \end{pmatrix}$$

4.2.6 Convergence Results Comparison for Problem 2 (see Appendix B)

Table 4.12: Summary of convergence result for problem 2

ITERATIVE METHODS	NO OF ITERATIONS	CPU TIME (seconds)
AOR	60	0.109
QAOR	144	0.375
KAOR	94	0.141
EAOR	43	0.031
RAOR	31	0.062
REAOR	22	0.016

The table above reveals the number of iterations reached for convergence by the various methods for the linear system consisdered in problem 2. It was observed that the proposed REAOR has 22 iterations, RAOR method has 31 iterations and the proposed EAOR has 43 iterations. AOR methods obtained 60 iterations, KAOR obtained 94 iterations while QAOR has 144 iterations.

4.3: Results of Problem 3 (see page 87)

4.3.1 Comparison of the Proposed Methods

Table 4.13: Results of spectral radii of EAOR and REAOR iteration matrices for problem 3

ω	r	v	$\rho(J_{EAOR})$	$\rho(J_{REAOR})$
0.1	0.04	0.05	0.9351778995	0.8745577038
0.2	0.08	0.10	0.8687339784	0.7546987253
0.3	0.12	0.15	0.8007555404	0.6412094354
0.4	0.16	0.20	0.7314112725	0.5349624495
0.5	0.20	0.25	0.6610142142	0.4369397913
0.6	0.24	0.30	0.5901486651	0.3482754469
0.7	0.28	0.35	0.5199440922	0.2703418590
0.8	0.32	0.40	0.4527087726	0.2049452328
0.9	0.36	0.45	0.3934697438	0.1548184393

The above table showing the comparison results of the spectral radii of the new methods for different values of the parameters (v, r and ω), reveals that the refinement method has a lower spectral radius compared to the Extended Accelerated Over Relaxation method.

4.3.2 Comparison of the EAOR Method with variants of AOR Methods

Table 4.14: Results of spectral radii of AOR, its variants and EAOR iteration matrices for problem 3

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu},2014})$	$\rho(J_{\text{Youssef \& Farid},2015})$	v	$\rho(EAOR)$
0.1	0.04	0.9362345989	0.9420086521	0.9386768056	0.05	0.9351778995
0.2	0.08	0.8731492477	0.8940791034	0.8824375732	0.10	0.8687339784
0.3	0.12	0.8111584708	0.8539358667	0.8309339876	0.15	0.8007555404
0.4	0.16	0.7508224216	0.8199279315	0.7838666115	0.20	0.7314112725
0.5	0.20	0.6929017621	0.7908297727	0.7409771518	0.25	0.6610142142
0.6	0.24	0.6384289029	0.7657137443	0.7020414568	0.30	0.5901486651
0.7	0.28	0.5887905079	0.7438656759	0.6668630202	0.35	0.5199440922
0.8	0.32	0.5457981563	0.7247275290	0.6352668421	0.40	0.4527087726
0.9	0.36	0.5116886038	0.7078575123	0.6070935758	0.45	0.3934697438

The comparison results of $\rho(J_{AOR})$, $\rho(J_{\text{Wu \& Liu, 2014}})$, $\rho(J_{\text{Youssef \& Farid, 2015}})$ and $\rho(J_{EAOR})$ displayed in table 4.14 for different values of the parameters (v, r and ω), shows that the EAOR method has a lower spectral radius compared to AOR, QAOR and KAOR methods.

4.3.3 Comparison of Refinement of AOR and Refinement of EAOR Methods

Table 4.15: Results of spectral radii of RAOR and REAOR iteration matrices for problem 3

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.1	0.04	0.8765352241	0.05	0.8745577038
0.2	0.08	0.7623896088	0.10	0.7546987253
0.3	0.12	0.6579780647	0.15	0.6412094354
0.4	0.16	0.5637343087	0.20	0.5349624495
0.5	0.20	0.4801128520	0.25	0.4369397913
0.6	0.24	0.4075914641	0.30	0.3482754469
0.7	0.28	0.3466742622	0.35	0.2703418590
0.8	0.32	0.2978956275	0.40	0.2049452328
0.9	0.36	0.2618252273	0.45	0.1548184393

Table 4.15 displays the spectral radii of the two refinement methods of AOR method and the proposed REAOR method. The spectral radius of the REAOR method is smaller compared to that of the RAOR method, that is to say $\rho(J_{REAOR}) < \rho(J_{\text{Vatti } et\ al.,\ 2018}) < 1$, for values of $v,\ r$ and ω. This means that convergence rate of the REAOR iterative method is faster than the RAOR iterative method.

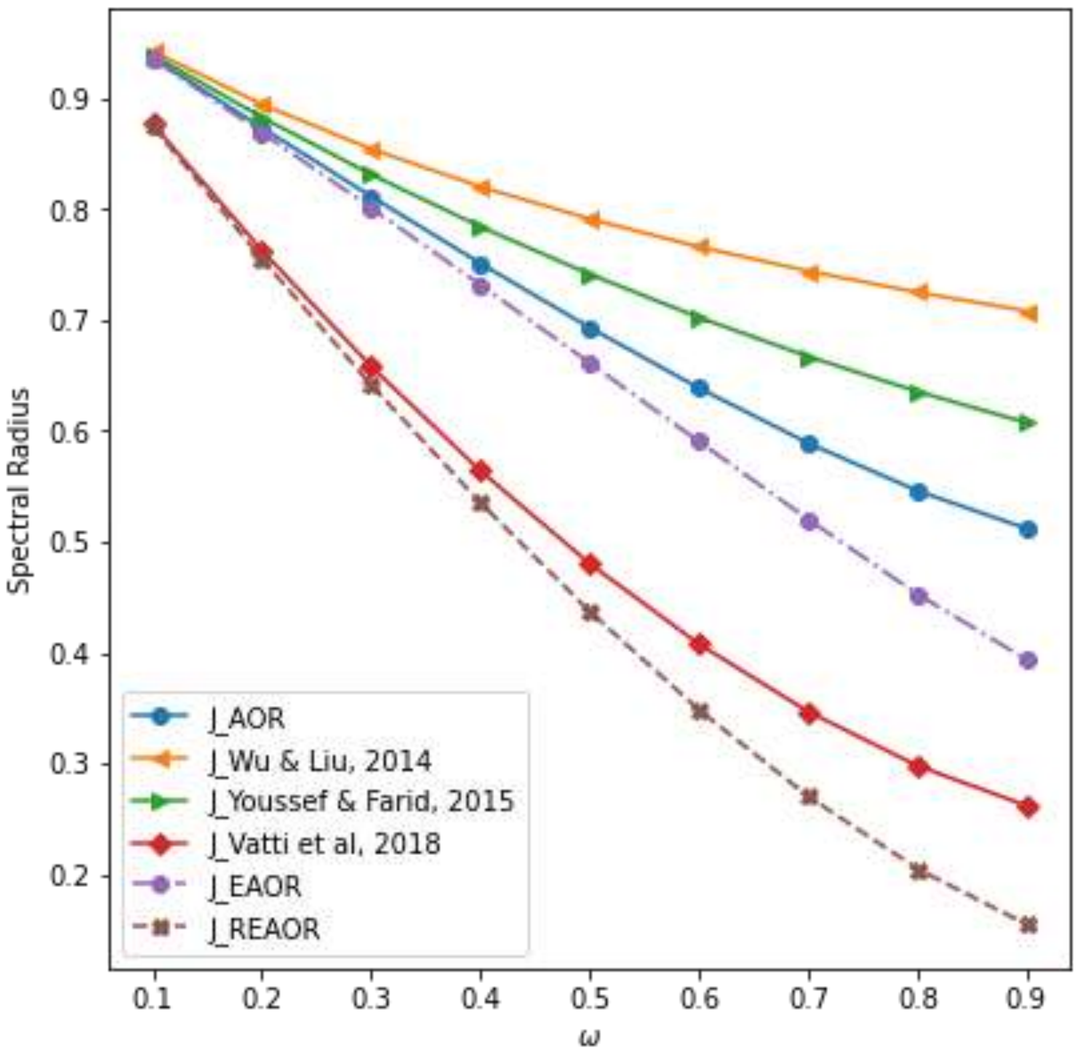

Figure 4.3: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.,2018}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu},2014}$ and $J_{\text{Youssef \& Farid},2015}$ Iteration Matrices for Problem 3

The above figure 4.3 shows the rate of convergence of J_{REAOR}, $J_{\text{Vatti } et\ al.,2018}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu},2014}$ and $J_{\text{Youssef \& Farid},2015}$ in terms of their spectral radii for problem 3. Due to its minimized spectral radius, the J_{REAOR} will converge faster than the compared methods.

4.3.4. Comparison of Rates of convergence of EAOR Method and Existing Methods

Table 4.16: Results of convergence rate of EAOR Method and existing methods for problem 3

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.02594510831	0.02748391328	0.0291057651	1.121820913	1.059010950
0.4862405542	0.0543160084	0.0611131918	1.256850982	1.125141438
0.06857474498	0.0804334768	0.0965000478	1.407224304	1.199749801
0.08622431875	0.1057578338	0.1358383506	1.575406481	1.199749801
0.1019169890	0.1301951834	0.1797892015	1.764074893	1.380920529
0.1159335574	0.1536372412	0.2290385709	1.975602026	1.490775082
0.1285054804	0.1759633650	0.2840433520	2.210359831	1.614218687
0.1398252416	0.1970438122	0.3441810897	2.461508994	1.746723664
0.1500541545	0.2167443628	0.4050886575	2.699616408	1.868969750

The rate of convergence results with respect to variants of Accelerated Over Relaxation method and the Extended Accelerated Over Relaxation method for problem 3 is shown in the table above. Clearly, it is seen that the rate of convergence of the EAOR iterative method is quicker than QAOR method by 1.830273870 times and 1.418882018 times than KAOR method.

4.3.5. Comparison of Rates of convergence of RAOR and the REAOR Method

Table 4.17: Results of convergence rate of RAOR and REAOR methods for problem 3

$R(J_{\text{Vatti } et\ al.,\ 2018})$	$R(J_{REAOR})$	$R(J_{REAOR}) / R(J_{\text{Vatti } et\ al.,\ 2018})$
0.0572306268	0.0582115303	1.0171394841
0.1178230317	0.1222263835	1.0373725902
0.1817885844	0.1930000956	1.0616733512
0.2489255333	0.2716767012	1.0913974858
0.3186566684	0.3595784031	1.1284195147
0.3897749197	0.4580771418	1.1752350361
0.4600784000	0.5680867041	1.2347606496
0.5259358713	0.6883621793	1.3088329145
0.5819885108	0.8101773149	1.3920847232

Results concerning the rate of convergence of the Refinement of Accelerated Over Relaxation (RAOR) method and the Refinement of Extended Accelerated Over Relaxation (REAOR) method for problem 3 is displayed in the table above. Clearly, the rate of convergence of the REAOR iterative method is quicker by approximately 1.2 times quicker than the RAOR method.

The True Solution of problem 3 by $z = A^{-1}b$ is

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} = \xrightarrow{\text{at 10 decimal places}} \begin{pmatrix} 0.7107095747 \\ 1.0878085301 \\ 1.1093417628 \\ 1.2217138647 \\ 1.0306242498 \\ 0.4949672605 \end{pmatrix}$$

4.3.6 Convergence Results Comparison for Problem 3 (see Appendix C)

Table 4.18: Summary of convergence result for problem 3

ITERATIVE METHODS	NO OF ITERATIONS	CPU TIME (seconds)
AOR	34	0.391
QAOR	62	0.562
KAOR	43	0.407
EAOR	24	0.375
RAOR	18	0.344
REAOR	13	0.312

The above table displays the summary of the convergence results of the new methods and some existing methods. The new EAOR method takes a shorter time (0.531 secs) to compute the 6×6 linear system of problem 3 to the desired accuracy compared to the other AOR-type methods. The REAOR method takes 0.312 secs as compared to 0.344 secs of RAOR method.

4.4: Results of Problem 4 (see page 87)

4.4.1 Comparison of the Two Proposed Methods

Table 4.19: Results of spectral radii of EAOR & REAOR iteration matrices for problem 4

ω	r	v	$\rho(J_{EAOR})$	$\rho(J_{REAOR})$
0.10	0.04	0.05	0.9863632718	0.9729125040
0.20	0.08	0.10	0.9715574788	0.9439239346
0.30	0.12	0.15	0.9554099845	0.9128082385
0.40	0.16	0.20	0.9377087689	0.8792977353

0.50	0.20	0.25	0.9181893627	0.8430717057
0.60	0.24	0.30	0.8965157199	0.8037404360
0.70	0.28	0.35	0.8722511973	0.7608221512
0.80	0.32	0.40	0.8448125547	0.7137082526
0.90	0.36	0.45	0.8133929469	0.6616080861

Table 4.19 shows the performance of the proposed EAOR and Refinement of the Extended AOR methods for problem 4. The results reveals that the spectral radius of the proposed refined EAOR scheme is lesser than that of the new EAOR method.

4.4.2 Comparison of the EAOR Method with Some Variants of AOR Method

Table 4.20: Results of spectral radii of AOR, its variant and EAOR iteration matrices for problem 4

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu,2014}})$	$\rho(J_{\text{Youssef \& Farid,2015}})$	v	$\rho(EAOR)$
0.10	0.04	0.9866661234	0.9878978038	0.9871877039	0.05	0.9863632718
0.20	0.08	0.9728500538	0.9775107211	0.9749284036	0.10	0.9715574788
0.30	0.12	0.9585228978	0.9684975404	0.9631856562	0.15	0.9554099845
0.40	0.16	0.9436532142	0.9606020877	0.9519262454	0.20	0.9377087689
0.50	0.20	0.9282067049	0.9536282492	0.9411198245	0.25	0.9181893627
0.60	0.24	0.9121458583	0.9474233207	0.9307386063	0.30	0.8965157199
0.70	0.28	0.8954295331	0.9418666115	0.9207570932	0.35	0.8722511973
0.80	0.32	0.8780124723	0.9368614631	0.9111518412	0.40	0.8448125547
0.90	0.36	0.8598447300	0.9323295455	0.9019012527	0.45	0.8133929469

Table 4.20 shows the various spectral radii of AOR, proposed EAOR and some variants of AOR methods for problem 4 with values of the parameters $\omega, v,$ and r. The table clearly shows that the spectral radius of the proposed EAOR method is smaller than those of other methods compared.

4.4.3 Comparison of Refinement of AOR and Refinement of EAOR Methods

Table 4.21: Results of spectral radii of RAOR and REAOR iteration matrices for problem 4

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.10	0.04	0.9735100390	0.05	0.9729125040
0.20	0.08	0.9464372271	0.10	0.9439239346
0.30	0.12	0.9187661457	0.15	0.9128082385
0.40	0.16	0.8904813886	0.20	0.8792977353
0.50	0.20	0.8615676870	0.25	0.8430717057
0.60	0.24	0.8320100668	0.30	0.8037404360
0.70	0.28	0.8017940488	0.35	0.7608221512
0.80	0.32	0.7709059014	0.40	0.7137082526
0.90	0.36	0.7393329597	0.45	0.6616080861

Table 4.21 reveals the comparison of the spectral radii of Refinement of AOR and Refinement of Extended AOR methods for problem 4. The spectral radius of the proposed REAOR method is smaller in comparison with Vatti *et al*. (2018), that is to say $\rho(J_{REAOR}) < \rho(J_{\text{Vatti et al.,2018}}) < 1$ and this indicates that the proposed refinement of EAOR method will converge faster than the refinement of AOR method developed by Vatti *et al*. (2018).

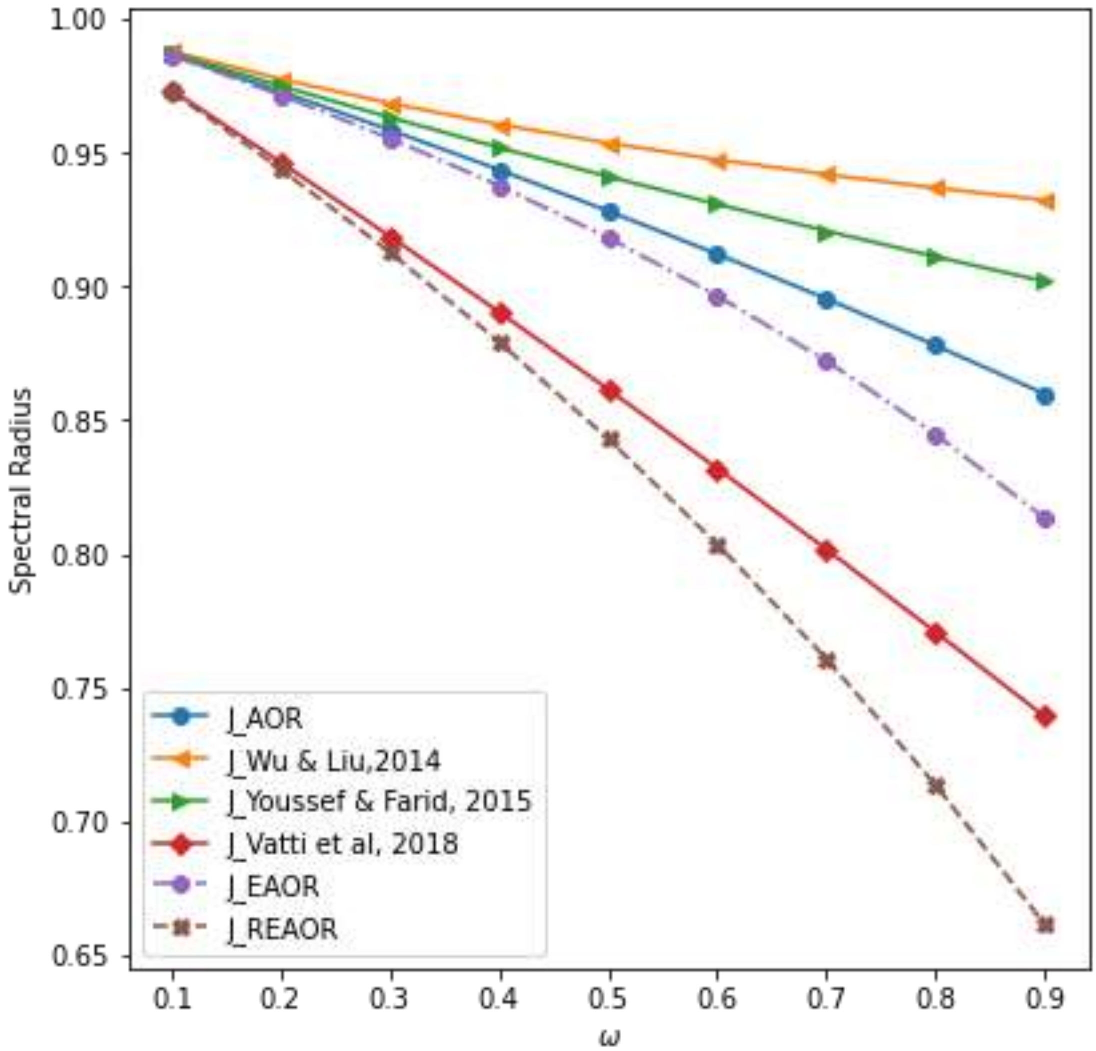

Figure 4.4: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.\text{,2018}}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu,2014}}$ and $J_{\text{Youssef \& Farid,2015}}$ Iteration Matrices for Problem 4

Figure 4.4 shows the performance of the proposed methods J_{REAOR} and J_{EAOR} with other methods in the literature in terms of the spectral radii. J_{REAOR} performs better than the refinement of AOR, $J_{\text{Vatti } et\ al.\text{,2018}}$. J_{EAOR} also performs better than J_{AOR}, $J_{\text{Wu \& Liu,2014}}$ and $J_{\text{Youssef \& Farid,2015}}$.

4.4.4 Comparison of Rates of Convergence of EAOR Method and Existing Methods

Table 4.22: Results of convergence rate of EAOR method and existing methods for problem 4

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.00528798005	0.00560026271	0.0059631075	1.127672087	1.064790672
0.00987847067	0.01102727667	0.0125315008	1.268566889	1.136409394
0.01390147788	0.01628999357	0.0198102244	1.425044486	1.216097742
0.01745647383	0.02139669915	0.0279320227	1.600095356	1.305436063
0.02062089237	0.02635507816	0.0370677429	1.797581900	1.406474406
0.02345592938	0.03117227152	0.0474420909	2.022605463	1.521932429
0.02601059826	0.03585492680	0.0593584257	2.282086137	1.655516577
0.02832462497	0.04040924290	0.0732396407	2.585723228	1.812447734
0.03043055278	0.04484100986	0.0896995978	2.947682168	2.000392009

It is shown that the EAOR iterative method converges quicker by a factor of approximately 1.457721892 quicker than the KAOR method and 1.895228636 faster than the QAOR method.

4.4.5 Comparison of Rates of convergence of RAOR and the REAOR Method

Table 4.23: Results of convergence rate of RAOR & REAOR methods for problem 4

$R(J_{\text{Vatti } et\ al.\text{,2018}})$	$R(J_{REAOR})$	$\frac{R(J_{REAOR})}{R(J_{\text{Vatti } et\ al.\text{,2018}})}$
0.0116595656	0.0119262150	1.0228695827
0.0239081855	0.0250630016	1.0483021199
0.0367950159	0.0396204489	1.0767884701

0.0503751530	0.0558640454	1.1089603124
0.0647105976	0.0741354858	1.1456467498
0.0798714190	0.0948841818	1.1879616392
0.0959371715	0.1187168515	1.2374437310
0.1129986297	0.1464792815	1.2962925468
0.1311599324	0.1793991956	1.3677896315

Table 4.23 shows the different rates of convergence of the new refinement of Extended AOR method in relation to the existing RAOR method for Problem 4. Obviously, with different values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$, it is easily seen that the new refinement of EAOR method appears to converge quicker by a ratio of 1.2328897476 times quicker than the RAOR method.

The True Solution by method of finite difference for Problem 4 is:

$$\begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \\ u_{5,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{5,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{5,3} \\ u_{1,4} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \\ u_{5,4} \\ u_{1,5} \\ u_{2,5} \\ u_{3,5} \\ u_{4,5} \\ u_{5,5} \end{pmatrix} = \xrightarrow{\text{at 10 decimal places}} \begin{bmatrix} 0.1851621840 \\ 0.1867937083 \\ 0.1743247547 \\ 0.1463767056 \\ 0.0992822439 \\ 0.3574358687 \\ 0.3598044227 \\ 0.3395021150 \\ 0.2928184937 \\ 0.2123963269 \\ 0.5185943370 \\ 0.5260250607 \\ 0.5068066957 \\ 0.4575476376 \\ 0.3735473067 \\ 0.6564256370 \\ 0.6867597253 \\ 0.6766755149 \\ 0.6362085311 \\ 0.5621772441 \\ 0.8431284475 \\ 0.8523659017 \\ 0.8458519157 \\ 0.8221319182 \\ 0.7762326159 \end{bmatrix}$$

An accuracy of 10 decimal places was utilize to verify the convergence result, refinement of EAOR reaches convergence at the 54th iteration, EAOR at 107th iteration and AOR at the 155th iteration.

4.4.6 Convergence Results Comparison for Problem 4 (see Appendix D)

4.4.6 Convergence Results Comparison for Problem 4 (see Appendix D)

Table 4.24: Summary of convergence result for problem 4

ITERATIVE METHODS	NO OF ITERATIONS	CPU TIME (seconds)
AOR	155	0.640
QAOR	350	1.281
KAOR	237	0.906
EAOR	107	0.437
RAOR	78	0.375
REAOR	29	0.360

Table 4.24 shows summary of the convergence results for all the methods compared for problem 4. Method of the proposed EAOR achieved the desired result after 0.437 secs with 107 iterations for the 25 x 25 linear system against 0.640 secs for AOR with 155 iterations,

0.906 secs for KAOR with 237 iterations and 1.281 secs for QAOR with 350 iterations. The proposed REAOR method takes a shorter time, 0.360 secs to attain the desired accuracy with 29 iterations in comparison of 0.375 secs with 78 iterations for the RAOR method.

4.5: Results of Problem 5 (Application Problem 1, see page 94)

4.5.1 Comparison of the Proposed Methods

Table 4.25: Results of spectral radii of EAOR and REAOR iteration matrices for problem 5

ω	r	v	$\rho(J_{EAOR})$	$\rho(J_{REAOR})$
0.1	0.04	0.05	0.9275147621	0.8602836340
0.2	0.08	0.10	0.8540611607	0.7294204663
0.3	0.12	0.15	0.7791350729	0.6070514619
0.4	0.16	0.20	0.7024538135	0.4934413600
0.5	0.20	0.25	0.6237494155	0.3890633334
0.6	0.24	0.30	0.5427088651	0.2945329123
0.7	0.28	0.35	0.4589180368	0.2106057645
0.8	0.32	0.40	0.3717692252	0.1382123568
0.9	0.36	0.45	0.2802590419	0.0785451306

Table 4.25 shows the performance of the proposed EAOR and REAOR methods for problem 5 with varied values of the relaxation parameter ω, acceleration parameter r and extended acceleration parameter v. Clearly, the spectral radius of the proposed REAOR is lesser than that of EAOR method $[\rho(J_{REAOR}) < \rho(J_{EAOR}) < 1]$.

4.5.2 Comparison of the EAOR Method with variants of AOR Methods

Table 4.26: Results of spectral radii of AOR, its variants and EAOR iteration matrices for problem 5

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu,2014}})$	$\rho(J_{\text{Youssef \& Farid,2015}})$	v	$\rho(EAOR)$
0.1	0.04	0.9276375964	0.9342167570	0.9304212907	0.05	0.9275147621
0.2	0.08	0.8551033965	0.8793214233	0.8658726450	0.10	0.8540611607
0.3	0.12	0.7821384274	0.8327052057	0.8056467754	0.15	0.7791350729
0.4	0.16	0.7086250476	0.7925934316	0.7492571586	0.20	0.7024538135
0.5	0.20	0.6344854568	0.7576992354	0.6963137307	0.25	0.6237494155
0.6	0.24	0.5596582442	0.7270594422	0.6464884762	0.30	0.5427088651
0.7	0.28	0.4840893261	0.6999365335	0.5994992209	0.35	0.4589180368
0.8	0.32	0.4077273884	0.6757553224	0.5550999467	0.40	0.3717692252
0.9	0.36	0.3492485491	0.6540602141	0.5130741955	0.45	0.2802590419

The comparison results of $\rho(J_{AOR})$, $\rho(J_{\text{Wu \& Liu,2014}})$, $\rho(J_{\text{Youssef \& Farid,2015}})$ and $\rho(J_{EAOR})$ displayed in table 4.26 for different values of the parameters (v, r and ω), shows that the $\rho(J_{EAOR})$ has a lower spectral radius compared to $\rho(J_{AOR})$, $\rho(J_{\text{Wu \& Liu,2014}})$ and $\rho(J_{\text{Youssef \& Farid,2015}})$ such that $(J_{\text{Wu \& Liu,2014}}) < \rho(J_{\text{Youssef \& Farid,2015}}) < \rho(J_{AOR}) < \rho(J_{EAOR}) < 1$.

4.5.3 Comparison of Refinement of AOR and Refinement of EAOR Methods

Table 4.27: Results of spectral radii of RAOR and REAOR iteration matrices for problem 5

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.1	0.04	0.8605115102	0.05	0.8602836340
0.2	0.08	0.7312018186	0.10	0.7294204663
0.3	0.12	0.6117405196	0.15	0.6070514619
0.4	0.16	0.5021494581	0.20	0.4934413600
0.5	0.20	0.4025717949	0.25	0.3890633334
0.6	0.24	0.3132173503	0.30	0.2945329123
0.7	0.28	0.2343424757	0.35	0.2106057645
0.8	0.32	0.1662416233	0.40	0.1382123568
0.9	0.36	0.1219745491	0.45	0.0785451306

Table 4.27 displays the comparison of the spectral radii of refinement of AOR and proposed refinement of EAOR schemes for problem 5. The spectral radius of the proposed REAOR iterative method is smaller in comparison with Vatti *et al.* (2018) $[\rho(J_{REAOR}) < \rho(J_{\text{Vatti } et\ al.,2018}) < 1]$ by checking how close their spectrums are to 1 with the different values of the parameters.

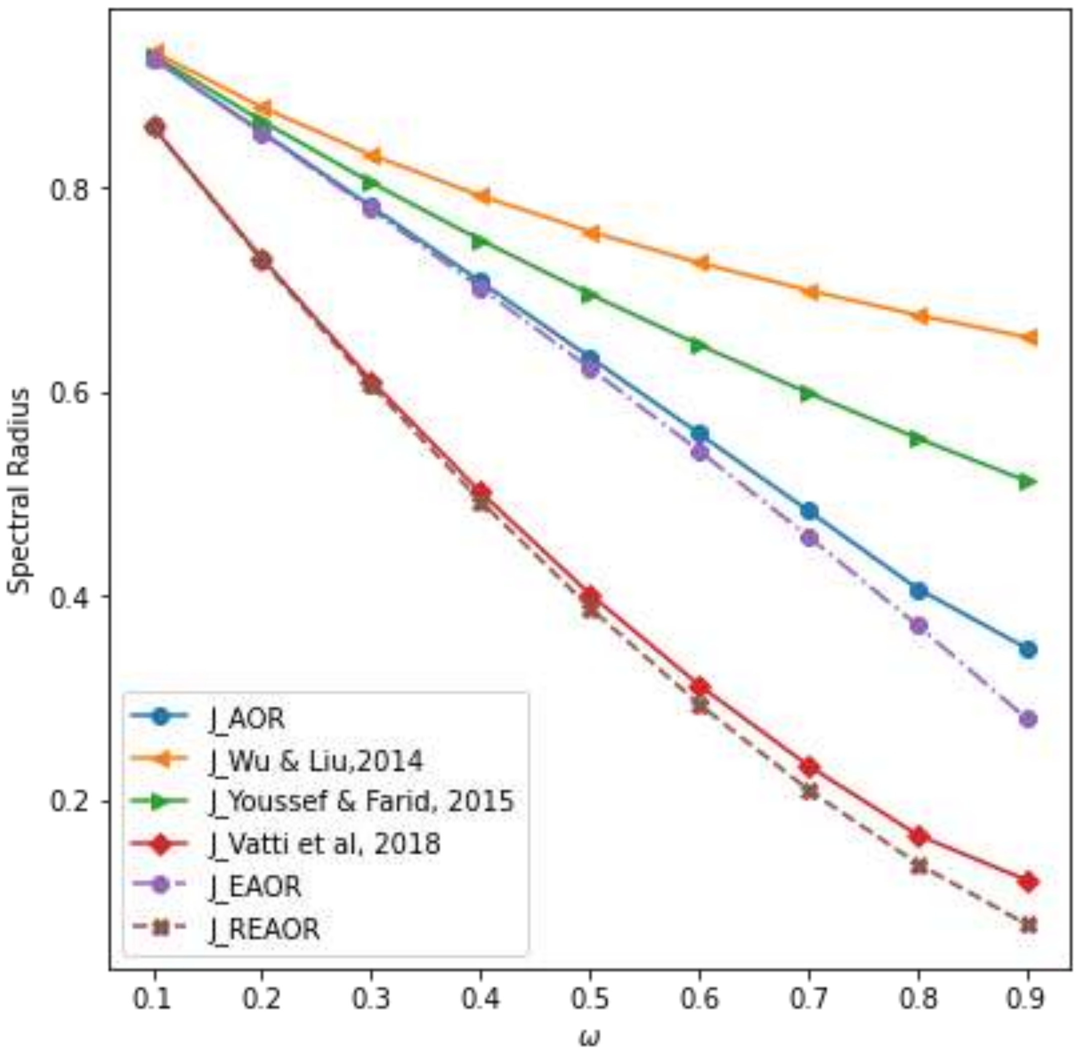

Figure 4.5: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.,2018}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu,2014}}$ and $J_{\text{Youssef \& Farid,2015}}$ Iteration Matrices for Problem 5

Figure 4.5 illustrates the performance of the proposed methods and some variants of AOR methods with respect to their spectral radii for problem 5. Clearly, the spectral radii of the J_{EAOR} is smaller than those of its counterpart. Likewise, the J_{REAOR} outperformed the $J_{\text{Vatti } et\ al.,2018}$ in terms of the spectral radii.

4.5.4 Comparison of Rates of Convergence of EAOR and Existing Methods

Table 4.28: Results of convergence rate of EAOR and existing methods for problem 5

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.02955234706	0.03132036027	0.9275147621	1.105806230	1.043384215
0.05585234580	0.06254598055	0.8540611607	1.226645483	1.095370591
0.07950872032	0.09385532683	0.7791350729	1.363212047	1.154833178
0.1009495311	0.1253690990	0.7024538135	1.519395125	1.223445224
0.1205031510	0.1571950408	0.6237494155	1.701116084	1.304047808
0.1384300811	0.1894392121	0.5427088651	1.917452349	1.401151753
0.1549413377	0.2222113770	0.4589180368	2.183180279	1.522266220
0.1702105250	0.2556288144	0.3717692252	2.524676801	1.681056827
0.1843822678	0.2898198272	0.2802590419	2.996168631	1.906151047

The above table displays the convergence rates of the new REAOR method in comparison with the classical AOR method for Problem 5. It is observed that the ratio of convergence rate of the EAOR iterative method with respect to KAOR method is 1.370189652 and 1.837517003 with respect to QAOR method.

4.5.5 Comparison of Rates of convergence of RAOR and REAOR Method

Table 4.29: Results of convergence rate of RAOR and REAOR methods for problem 5

$R(J_{\text{Vatti } et\ al.,2018})$	$R(J_{REAOR})$	$R(J_{REAOR})$ $/R(J_{\text{Vatti } et\ al.,2018})$
0.0652433162	0.0653583390	1.0017629828
0.1359627371	0.1370220553	1.0077912389
0.2134327524	0.2167744907	1.0156571017
0.2991670016	0.3067644510	1.0253953454
0.3951566560	0.4099796966	1.0375118079
0.5041541887	0.5308661683	1.0529837502
0.6301489863	0.6765297459	1.0736028473
0.7792602290	0.8594531274	1.1029090096
0.9137307787	1.1048807338	1.2091972380

The above table displays the convergence rates of the new REAOR, $R(J_{REAOR})$ method in comparison with refinement of AOR method, $R(J_{RAOR})$ for Problem 5. Obviously, the convergence rate of the proposed refinement of EAOR method is faster than the refinement of AOR method by a factor of 1.0585345913.

4.5.6 Convergence Results Comparison for Problem 5

The True Solution of problem 5 (Application Problem 1) **by** $\boldsymbol{z = A^{-1}b}$ **is**

$$\begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \\ \underline{x}_4 \\ \underline{x}_5 \\ \underline{x}_6 \\ \overline{x}_1 \\ \overline{x}_2 \\ \overline{x}_3 \\ \overline{x}_4 \\ \overline{x}_5 \\ \overline{x}_6 \end{pmatrix} = \xrightarrow{\text{at 10 decimal places}} \begin{pmatrix} -4.1235541942 + 0.1235541942\alpha \\ -0.2485771435 + 0.2485771435\alpha \\ 0.7825179488 + 1.2174820512\alpha \\ 3.6052817208 + 0.3947182792\alpha \\ 6.6566736561 + 0.3433263439\alpha \\ 6.7823343544 + 2.2176656456\alpha \\ -2.8764458058 - 1.1235541942\alpha \\ 1.2485771435 - 1.2485771435\alpha \\ 5.2174820512 - 3.2174820512\alpha \\ 4.3947182792 - 0.3947182792\alpha \\ 8.3433263439 - 1.3433263439\alpha \\ 10.2176656456 - 1.2176656456\alpha \end{pmatrix}$$

An accuracy of 10 decimal places was utilize to verify the convergence result and after some operations, the true solution for the fuzzy linear system becomes

$$x_1 = (-4.1235541942 + 0.1235541942\alpha, -2.8764458058 - 1.1235541942\alpha)$$

$$x_2 = (-0.2485771435 + 0.2485771435\alpha, 1.2485771435 - 1.2485771435\alpha)$$

$$x_3 = (0.7825179488 + 1.2174820512\alpha, 5.2174820512 - 3.2174820512\alpha)$$

$$x_4 = (3.6052817208 + 0.3947182792\alpha, 4.3947182792 - 0.3947182792\alpha)$$

$$x_5 = (6.6566736561 + 0.3433263439\alpha, 8.3433263439 - 1.3433263439\alpha)$$

$$x_6 = (6.7823343544 + 2.2176656456\alpha, 10.2176656456 - 1.2176656456\alpha)$$

4.5.6 Convergence Results Comparison for Problem 5

Table 4.30: Summary of convergence result for problem 5

ITERATIVE METHODS	NO OF ITERATIONS	CPU TIME (seconds)
AOR	32	0.516
QAOR	50	0.734
KAOR	34	0.594
EAOR	18	0.484
RAOR	16	0.469
REAOR	10	0.453

The above table displays the number of iterations and computational time of the various methods to attain convergence. It is observed that the EAOR method takes a shorter time to compute the 12×12 linear system to the desired accuracy compared to the other methods examined. Similarly, the refinement of the Extended Accelerated Over-Relaxation (REAOR) method takes 0.453 secs as compared to 0.469 secs of refinement of AOR (RAOR) method. This indicates that the new methods demonstrate efficiency as compared to their counterparts.

4. 6: Results of Problem 6 (Application Problem 2, see page 97)

4.6.1 Comparison of the Proposed Methods

Table 4.31: Results of spectral radii of EAOR and REAOR iteration matrices for roblem 6

ω	r	v	$\rho(J_{EAOR})$	$\rho(J_{REAOR})$
0.1	0.04	0.05	0.9936764826	0.9873929521
0.2	0.08	0.10	0.9867694121	0.9737138726
0.3	0.12	0.15	0.9791905790	0.9588141901
0.4	0.16	0.20	0.9708322197	0.9425151988
0.5	0.20	0.25	0.9615610144	0.9245995844
0.6	0.24	0.30	0.9512096225	0.9047997459
0.7	0.28	0.35	0.9395644282	0.8827813147
0.8	0.32	0.40	0.9263472516	0.8581192305
0.9	0.36	0.45	0.9111870362	0.8302618150

The above table shows the comparison of the spectral radius of the proposed EAOR method and the Refinement of EAOR method for problem 6, with different values of the relaxation parameter ω, acceleration parameter r and extended acceleration parameter v, It is observed that spectral radii of both J_{EAOR} and J_{REAOR} are lesser than 1, but the rate of convergence of the new REAOR method is faster than the new EAOR method since $\rho(J_{REAOR}) < \rho(J_{EAOR}) < 1$.

4.6.2 Comparison of the EAOR Method with variants of AOR Methods

Table 4.32: Results of spectral radii of AOR, its variants and EAOR iteration matrices for problem 6

ω	r	$\rho(AOR)$	$\rho(J_{\text{Wu \& Liu,2014}})$	$\rho(J_{\text{Youssef \& Farid,2015}})$	v	$\rho(EAOR)$
0.1	0.04	0.9938273704	0.9943982314	0.9940691300	0.05	0.9936764826
0.2	0.08	0.9874145348	0.9895797250	0.9883803046	0.10	0.9867694121
0.3	0.12	0.9807466872	0.9853907711	0.9829186891	0.15	0.9791905790
0.4	0.16	0.9738077346	0.9817154450	0.9776706591	0.20	0.9708322197
0.5	0.20	0.9665801497	0.9784646603	0.9726236769	0.25	0.9615610144
0.6	0.24	0.9590448011	0.9755688214	0.9677661842	0.30	0.9512096225
0.7	0.28	0.9511807589	0.9729727661	0.9630875067	0.35	0.9395644282
0.8	0.32	0.9429650693	0.9706322063	0.9585777699	0.40	0.9263472516
0.9	0.36	0.9343724943	0.9685111731	0.9542278247	0.45	0.9111870362

Table 4.32 shows the various spectral radii of AOR, some of its variants and proposed EAOR methods for problem 6 with values of the parameters $\omega, v,$ and r. Obviously, spectral radius of the proposed EAOR method is smaller than those of the KAOR, QAOR and AOR methods which reveals that the rate of convergence of the proposed EAOR is method faster than the other methods compared.

4.6.3 Comparison of Refinement of AOR and Refinement of EAOR Methods

Table 4.33: Results of spectral radii of RAOR and REAOR iteration matrices for problem 6

ω	r	$\rho(J_{\text{Vatti } et\ al.,\ 2018})$	v	$\rho(J_{REAOR})$
0.1	0.04	0.9876928421	0.05	0.9873929521
0.2	0.08	0.9749874636	0.10	0.9737138726
0.3	0.12	0.9618640645	0.15	0.9588141901
0.4	0.16	0.9483015040	0.20	0.9425151988
0.5	0.20	0.9342771858	0.25	0.9245995844
0.6	0.24	0.9197669305	0.30	0.9047997459
0.7	0.28	0.9047448361	0.35	0.8827813147
0.8	0.32	0.8891831220	0.40	0.8581192305
0.9	0.36	0.8730519581	0.45	0.8302618150

Table 4.33 displays the comparison of the spectral radii of Refinement of AOR and proposed Refinement of EAOR schemes for problem 6. The spectral radius of the proposed REAOR iterative method is smaller in comparison with Vatti *et al.* (2018) $[\rho(J_{REAOR}) < \rho(J_{\text{Vatti } et\ al.,2018}) < 1]$ by checking how close their spectrums are to 1 with different values of the parameters.

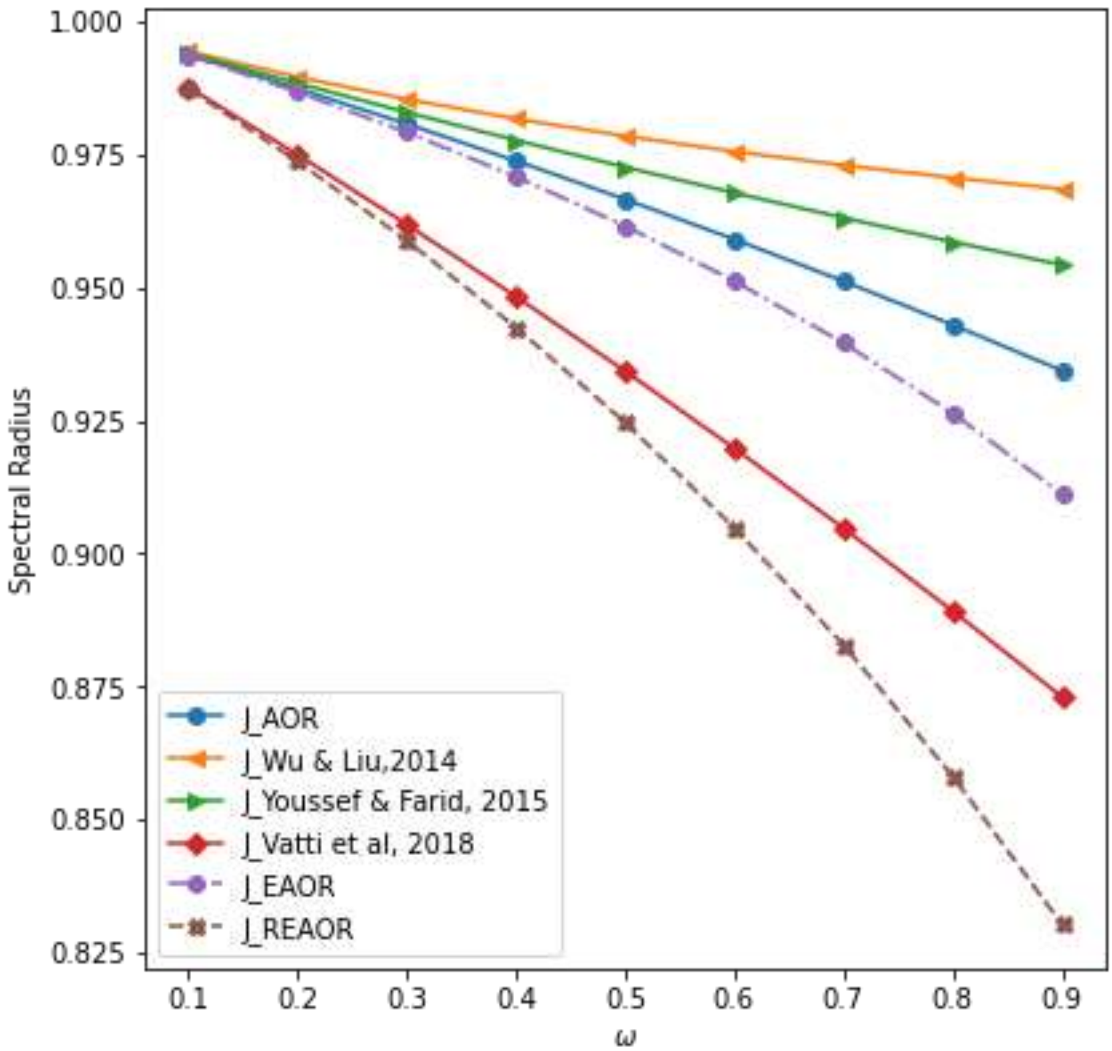

Figure 4.6: Spectral Radii of J_{REAOR}, $J_{\text{Vatti } et\ al.\text{,2018}}$, J_{EAOR}, J_{AOR}, $J_{\text{Wu \& Liu,2014}}$ and $J_{\text{Youssef \& Farid,2015}}$ Iteration Matrices for Problem 6

The above figure depicts the performance of the spectral radii of problem 6 and comparison between the newly developed schemes and the methods from existing literature. It is observed that J_{REAOR} has the least spectral radii, which shows that J_{REAOR} will outperform all the compared methods.

4.6.4 Comparison of Rates of Convergence of EAOR and Existing Methods

Table 4.34: Results of convergence rate of EAOR and existing methods for problem 6

$R(J_{QAOR})$	$R(J_{KAOR})$	$R(J_{EAOR})$	$\frac{R(J_{EAOR})}{R(J_{QAOR})}$	$\frac{R(J_{EAOR})}{R(J_{KAOR})}$
0.002439656786	0.002583412651	0.0027549885	1.129252490	1.066414419
0.004549211328	0.005075917346	0.0057843213	1.271499801	1.139561759
0.006391509525	0.007482407230	0.0091327736	0.0091327736	1.220566233
0.008014376342	0.009807418243	0.0128558189	1.604094736	1.310826008
0.009454855355	0.01205516246	0.0170231531	1.800466793	1.412104827
0.01074208793	0.01422955713	0.0217237650	2.022303777	1.526664871
0.01189931564	0.01633425083	0.0270734339	2.275209324	1.657464072
0.01294530257	0.01837264683	0.0332261827	2.566659413	1.808459228
0.01389536474	0.02034792383	0.0403924678	2.906902305	1.985090378

The above table presents rates of convergence of the new EAOR method in relation to AOR iterative method concerning Problem 6. The proposed EAOR iterative method converges 1.889475571 quicker than the QAOR method and 1.458572422 quicker than the KAOR method.

4.6.5 Comparison of Rates of convergence of AOR and the EAOR Method

Table 4.35: Results of convergence rate of RAOR & REAOR methods for problem 6

$R(J_{\text{Vatti } et\ al., 2018})$	$R(J_{REAOR})$	$\frac{R(J_{REAOR})}{R(J_{\text{Vatti } et\ al., 2018})}$
0.0053780936	0.0055099770	1.0245223441
0.0110009684	0.0115686425	1.0516021923
0.0168863003	0.0182655472	1.0816784510
0.0230535607	0.0257116377	1.1153000663
0.0295242561	0.0340463061	1.1531638926
0.0363222092	0.0434475300	1.1961698094
0.0434738870	0.0541468679	1.2455032574
0.0510087894	0.0664523654	1.3027630357
0.0589599093	0.0807849355	1.3701672292

The table above shows rates of convergence of REAOR method in relation to RAOR method concerning Problem 6. Evidently, for all values of the parameters $\boldsymbol{r}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$ the proposed REAOR method is likely to converge quick as the RAOR by a factor of approximately 1.2 times.

4.6.6 Convergence Results Comparison for Problem 6

Table 4.36: Summary of convergence result for problem 6

ITERATIVE METHODS	NUMBER OF ITERATIONS	CPU TIME (seconds)
AOR	448	0.718
QAOR	985	1.156
KAOR	724	1.094
EAOR	300	0.531
RAOR	226	0.500
REAOR	159	0.468

The above table displays the summary of the convergence results of the various methods examined. The proposed EAOR method takes a shorter time (0.531 secs) to compute the 64×64 linear system of problem 6 to the desired accuracy compared to the other AOR-type methods. Similarly, the REAOR method takes 0.468 secs as compared to 0.500 secs of RAOR method.

4.7 Discussion of Results

In Table 4.1, using different values of acceleration parameter (r) starting from 0.04 to 0.36, over-relaxation parameter (ω) starting from 0.1 to 0.95 and the extended acceleration parameter (v) ranging from 0.05 to 0.45, spectral radius of the EAOR iteration matrix is compared with that of its Refinement version. This is mainly to ascertain their performance with regards to problem 1. From the results, the REAOR method has a lower spectral radius which depicts that the Refinement method will converge to the true solution faster than the new EAOR method. Similarly, the same deductions can be made for Tables 4.7, 4.13, 4.19, 4.25 and 4.31 for problems 2 to 6 respectively. Since the results in the tables above shows that J_{EAOR} has a greater spectral radius compared to J_{REAOR}, implying that J_{REAOR} will definitely converge to the true solution faster than J_{EAOR}.

In Table 4.2 the spectral radius of the new EAOR method J_{EAOR} is compared with spectral radii of AOR method J_{AOR} and some variants of AOR method $J_{\text{Wu \& Liu, 2014}}$ and $J_{\text{Youssef \& Farid, 2015}}$. This comparison is necessary so as to verify if the proposed method has been able to achieve the aim it was developed for, implying that apart from the method been convergent, it must also converge quicker than the AOR and AOR-type methods. From observation, it is seen that the spectral radius of J_{EAOR} is smaller than 1 for all values chosen for the parameters and this implies that the new EAOR method is convergent. Furthermore, comparing spectral radius of J_{EAOR} and those of J_{AOR}, $J_{\text{Wu \& Liu, 2014}}$ and $J_{\text{Youssef \& Farid, 2015}}$ and checking how close their spectrum is to zero, reveals that the J_{EAOR} converges faster than the others. This is due to the fact that it has a lower spectral radius compared to the other methods examined. The result affirms the superiority of the proposed EAOR method over the existing methods. Result in Table 4.2 is based on a discretized linear system in problem 1. Also in Tables 4.8, 4.14, 4.20, 4.26 and 4.32, spectral radius of J_{EAOR} is compared with those of J_{AOR}, $J_{\text{Wu \& Liu,2014}}$ and $J_{\text{Youssef \& Farid,2015}}$ for problems 2 to 6. This is to further verify the efficiency of the method for linear systems with different coefficient matrices. The results shows that the rate of convergence of the J_{EAOR} is faster when compared with the other methods.

In Tables 4.3 and 4.21, the spectral radius of the REAOR iteration matrix J_{REAOR} is compared with the spectral radius of the RAOR iteration matrix $J_{\text{Vatti } et\ al.\text{,2018}}$ based on the discretized linear systems of problems 1 and 4. Also, in Tables 4.9 and 4.15 the spectral radius of J_{REAOR} is compared with spectral radius of $J_{\text{Vatti } et\ al.\text{,2018}}$ based on the linear systems in problems 2 and 3. Again, in Tables 4.27 and 4.32, comparison of the spectral radii between J_{REAOR} and $J_{\text{Vatti } et\ al.\text{,2018}}$ based on the fuzzy linear system in Problem 5 and application of a real world problem considered in problem 6 is presented. This is to ascertain the performance of the new refinement method with existing refinement method in terms of efficiency and accuracy. From the results, it is observed that the new refinement method exhibits faster convergence since its spectral radius is lower compared to that of Refinement of AOR by Vatti *et al.* (2018).
Figures 4.1 to figure 4.6 illustrate the presented spectral radius results for problems 1 to 6 for clarity of the tabulated results. From the figures, we observed that the refinement of the proposed method has the least spectral radii in all the figures. It is also noted that the EAOR method outperformed the other variants of AOR method compared in the existing literature.

Apart from establishing the fact that the EAOR method converges quicker than existing methods, it is also important to compare its rate of convergence with those of existing methods. Tables 4.4, 4.10, 4.16, 4.22, 4.28 and 4.34 display the ratio of convergence rate with respect to J_{EAOR}, J_{QAOR} and J_{KAOR} for problems 1 to 6 respectively. Despite the fact that the rate of convergence varies with different values of the parameters $(v, r \text{ and } \omega)$, the rate of the J_{EAOR} is quite faster in comparison with J_{QAOR} and J_{KAOR} which confirms the superiority of the new EAOR method over the KAOR and QAOR method. In all, EAOR method converges at 1.4 times faster than the KAOR method and 1.8 times quicker than QAOR method.

The rate of convergence of J_{REAOR} and $J_{\text{Vatti } et\, al., 2018}$ for problems 1 to 6 are shown in Tables 4.5, 4.11, 4.17, 4.23, 4.29 and 4.35. The results from the tables indicate that the rate of convergence of proposed Refinement of EAOR method is higher than that of the Refinement of AOR method. Despite the fact that the rate of convergence varies with different values of the parameters $(v, r \text{ and } \omega)$, the rate of the J_{REAOR} is quite faster in comparison with $J_{\text{Vatti } et\, al., 2018}$ which further proves the efficiency of the REAOR method against the RAOR method. The REAOR method converges at approximately 120% faster than the RAOR method.

Tables 4.6, 4.12, 4.18, 4.24, 4.30 and 4.36 presented the convergence results of the six numerical tests performed. From the tabulated results of problem 1 in Table 4.6, EAOR converges after 56 iterations, REAOR converges after 29 iterations, AOR converges after 81 iterations, RAOR converges after 41 iterations, QAOR converges after 196 iterations and KAOR converges after 127 iterations. The respective time elapsed (in seconds) by each of the methods is 0.266, 0.094, 0.296, 0.125, 0.734, and 0.360. Similarly, the convergence results were also presented for problems 2, 3, 4, 5, and 6 in Tables 4.12, 4.18, 4.24, 4.30 and 4.36. The results indicates that the proposed (EAOR and REAOR) iterative methods requires less number of iterations to reach convergence than similar methods.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this thesis, iterative solution of large and sparse $n \times n$ linear systems were studied to improve the rate of convergence of a family of Accelerated Over- Relaxation (AOR) iterative method. We have been able to develop an efficient iterative method, analyze for convergence of some special matrices and perform six numerical tests including fuzzy linear system problem and heat transfer problem. The new iterative method called Extended Accelerated Over Relaxation (EAOR) iterative method was developed by introducing a new acceleration parameter to the family of two-parameter Accelerated Over Relaxation iterative method. The developed EAOR iterative method was analyzed for convergence of L, M, and irreducible diagonally weak dominant matrices. Furthermore, Refinement of the Extended Accelerated Over-Relaxation (REAOR) iterative method was developed to reduce the residual in the iteration process. The convergence of the L, M, and irreducible diagonally weak dominant matrices was also studied and confirmed for the REAOR iterative method.

Six numerical tests of partial differential equations, fuzzy linear system, and an application problem of heat transfer were used to validate the efficiency of the developed method and its refinement. In problem 1, second-order partial differential equation was discretized with finite difference procedure to a 9×9 linear system whose coefficient matrix is an L matrix, problem 2 considered a 10×10 linear system whose coefficient matrix is an M matrix, problem 3 also considered 6×6 linear system with an irreducible weak diagonally dominant coefficient matrix, and another second-order partial differential equation discretized to 25×25 linear systems whose coefficient matrix is an M matrix. We further considered a 6×6 fuzzy linear system which was transformed to an extended 12×12 matrix, and a metal plate heat transfer problem modeled into a two-dimensional Laplace equation which was discretized to a 64×64 linear system of equations.

We computed the spectral radii of the coefficient matrices in each of the problems mentioned above with the proposed EAOR and the refined version called REAOR iterative methods at varying values of parameters $\omega,\ r,$ and v. In contrast, the results of the spectral radii of EAOR and REAOR were compared with the spectral radii of AOR and some of its variants to examine how soon the convergence will be for the compared methods. From all the numerical results, especially indications of small spectral radii of the developed methods, we proved that REAOR converges faster than EAOR, and in general, EAOR converges faster than AOR and its variants. As reported by Sebro (2018), numerical methods that register small numbers of iterations will require less computational storage. Then, we can infer that the developed EAOR and REAOR iterative methods with lower spectral radii will have less storage capacity, computational time and number of iterations, thereby converging faster than the methods considered in literature.

5. 2 Recommendation

Further research on investigation of convergence of the proposed Extended Accelerated Over-Relaxation iterative method for Hermitian and H matrices is recommended, so as to accommodate more classes of matrices.

5.3 Contributions to Knowledge

This research work has contributed the following to the body of existing knowledge;

I. An efficient iterative method that shows an indication of a small spectral radius, which enhances convergence rate was developed for finding solution to linear systems.

II. Conditions placed on the coefficient matrix that would enable convergence of the Extended Accelerated Over-Relaxation method is established. In all, the proposed EAOR method converges faster than some existing methods reviewed in the work. As indicated in the obtained result, presented in Tables 4.4, 4.10, 4.16, 4.22, 4.28 and 4.34, the method converges approximately 1.8 times or 180% faster than the iterative method of Wu and Liu (2014) and 1.4 times or 140% faster than the iterative method of Youssef and Farid (2015).

III. A Refinement form of the proposed iterative method called REAOR, that drastically reduce the number of iterations has been developed for L, M and irreducible diagonally dominant matrices. Analysis of the result proves this, in the following number of iterations reached to obtain a desired result;

The method of Wu and Liu (2014)	-	985 iterations
The method of Youssef and Farid (2015)	-	724 iterations
The proposed Refinement method	-	159 iterations

IV. Establishment of convergence theorems for L-matrix, M-matrix and weak Irreducible diagonally dominant matrix with respect to the Extended Accelerated Over-Relaxation and its Refinement.

V. Some numerical experiments for the purpose of evaluating and validating the new methods has been carried out.

REFERENCES

Aijuan, L. (2011). Improving AOR iterative method for irreducible L −matrices. *Journal of Engineering, Science and Technology,* 19, 19 - 26.

Assefa, L. W., & Teklehaymanot, A. W. (2021). Second refinement of accelerated over relaxation method for the solution of linear system. *Pure and Applied Mathematics Journal,* 10, 32-37.

Akhir, M., K. M., & Suleiman, J. (2017). Accelerated over-relaxation iterative method using triangle element approximation for solving 2D helmhotz equations. *Journal of Physic Conference Series*, 890, 1-5.

Anton, H., & Rorres, C. (2013). *Elementary Linear Algebra Application version*, 11th Edition, Wiley.

Chunping, P. (2017). On generalized SSOR-like iteration method for saddle point problems. *WSEAS Transactions on Mathematics,* 16, 239-247.

Behzadi, R. (2019). A new class of AOR preconditioned for L-matrices. *Journal of Mathematical Research with Applications, 3*9(1), 101-110.

Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis* (9th edition). Brooks/Cole Cengage Learning.

Dafchahi, F. N. (2008). A new refinement of Jacobi method for solution of linear system equations $Ax = b$. *International Journal of Contemporary Mathematics and Science,* l3 (17), 819-827.

Duff, I. S., Erisman, A. M., & Reid, J. K. (2017). *Direct Methods for Sparse Matrices* (2nd edition). Oxford University Press.

Dahalan, A. A., Saudi, A. M., & Suleiman, H. (2018). Numerical technique for autonoumous robot path planning based on QSAOR iterative method using indoor environment. *Journal of Engineering and Applied Sciences,* 13(20), 8414-8418.

Darvishi, M. T., Khan F., Godarzi, A. M., & Tian, H. (2011). Symmetric modified AOR method for solving systems of linear equations. *Journal of Applied Mathematics and Computation*, 36, 41-51

Edalatpanam, S. A., & Najafa, H. S. (2013). Convergence analysis for double splitting of matrices. *Information of Sciences and Computing*, 2013(2), 1-6.

Fadugba, E. S. (2015). On some iterative methods for solving systems of linear equations. *Computational and Applied Mathematics*, 1(2), 21-28.

Firew, H., Genanew, G. G., & Hailu, M. C. (2020). Second degree generalized successive over-relaxation method for solving system of linear equations. *Momona Ethiopia Journal of Sciences*, 12(1), 60-71.

Fiseha, A. (2020). Accelerated over relaxation method for solving a class of complex linear system of equations. Masters Thesis, Bahir Dar University, Ethiopia. Unpublished.

Gebregiorgis, S., & Gofe, G. (2018). Refined generalized iterative methods in solving fuzzy linear system. *Ethiopia Journal of Education of Science*, 13(2), 73-81.

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

Guan, J., Ahmed, Z., & Chandio, A. A. (2017). Tridiagonal iterative method for linear system. *Journal of Information and Communication Technology,* 1, 151-155.

Hadjidimos, A. (1978). Accelerated over-relaxation method. *Mathematics of Computation,* 32, 149 -157.

Hadjidimos, A. (2000). Successive over–relaxation (SOR) accelerated method, *Journal of Computational and Applied Mathematics,* 123, 177-199.

Karunanithi, S., Gajalakshmi, N., Malarvizhi, M., & Saileshwari, M. (2018). A study on comparison of Jacobi, Gauss-Seidel and SOR method for the solution in system of linear equations. *International Journal of Mathematics Trends and Technology (IJMTT)*, 56, 214-222.

Kargar, R., Allanviranloo, T., Rostami-Malkhalifeh, M., & Jahanshaloo, G., R. (2014). A proposed method for solving fuzzy system of linear equations. *The Scientific World Journal*, 782093, 1-7.

Kisabo, A. B., Funmilayo, A. A., & Okey, A. K. (2016). Comparative analysis of numerical solution to a linear system of algebraic equations. *International Journal of Systems Science and Applied Mathematics*, 4, 50-57.

Kiusalaas, J. (2005). *Numerical Methods in Engineering with Python*. Cambridge University Press.

Kyurkchiev, N., & Iliev, A. (2013). A refinement of some over-relaxation algorithms for solving a system of linear equations. *Serdica Journal of Computing*, 3, 245-246.

Lubna, I. & Naji, Q. (2018). Numerical methods for solving fuzzy linear systems. *Mathematics*, 6(2), 1-9.

Laskar, A. H., & Behera, S. (2014). Refinement of iterative methods for the solution of system of linear equations $Ax = b$. *IOSR - Journal of Mathematics,* 10, 70-73.

Luna, R., Fujiao, R., & Wen, R. (2016). A selected method for the optimal parameters of the AOR iteration. *Journal of Inequalities and Application,* 279, 1-14.

Markatos, N. C., & Karabeks, S. J. (2015). A convergence criterion of the Jacobi iteration method. *Research gate* DO1:/0/3140/RG.2.1 3518.0640.

Martins, M. M., Yousif, W., & Santos, J. L. (2012). A variant of the AOR method for augmented systems. *Journal of Mathematics of Computation,* 81, 399-417.

Mayooran, T., & Elliot, L. (2016). Applying the successive over-relaxation method to a real world problems. *American Journal of Applied Mathematics and Statistics,* 4, 113-117.

Mohammed, F. D., & Rivaie, M. (2017). Jacobi-Davisson, Gauss-Seidel and successive over-relaxation for solving systems of linear equations. *Applied Mathematics and Computational Intelligence,* 6, 41-52.

Muleta, H., & Gofe, G. G. (2018). Refinement of generalized accelerated over relaxation method for solving linear equations based on the Nekrassov-Mehmk 1 method. *Ethiopia Journal of Education and Sciences*, 13(2), 1-18.

Nasabazadeh, H., & Toutounia, F. (2013). A new generalized AOR iterative method for solving linear systems. *International Journal of Applied Mathematical Research.* 2(4), 439-451.

Ndanusa, A. (2012). Preconditioned successive over-relaxation iterative method for solving partial equations of elliptic type. PhD Thesis, Federal University of Technology, Minna, Nigeria. Unpublished.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, USA.

Saha, M. & Chakrabarty, J. (2020). Convergence of generalized Jacobi, Gauss-Seidel and SOR methods for linear systems. *Internationl Journal of Applied Computational Mathematics,* 77, 1-6.

Salkuyeh, D. K. (2007). Generalized Jacobi and Gauss-Seidel method for solving linear system of equations. Numerical Mathematics. *Journal of Chinese University (English Series),* 16, 164-170.

Salkuyeh. D. K. (2011). Generalised AOR method for solving system of linear equations. *Australian Journal of Basic and Applied Sciences,* 5(3), 351-358.

Sebro, A. (2018). Extrapolated refinement of generalized Gauss-Seidel method for solving systems of linear equation. Masters Thesis, Jimma University, Ethiopia. Published.

Shaikh, A. W., Khuhro, Z. U. A., Maree, U. R., & Soomro, A. K. (2013). Critical accelerated method. *Sindh University Research Journal (Science Series)*, 45(1), 113-118.

Strang, G. (2016). *Introduction to Linear Algebra* (5th edition). Wellesley Cambridge Press, USA.

Tesfaye, K. (2014). Second degree generalized Gauss-Seidel iteration method for solving linear system of equation. *Ethiopia Journal of Science and technology,* 7(2), 115-124.

Tesfaye, K. (2016). Second degree generalized Jacobi iteration method for solving system of linear equations. *Tamkang Journal of Mathematics,* 47, 179-192.

Tesfaye, K. E., Awgichew, G., Haile, E., & Gashaye, D. A. (2019). Second refinement of Jacobi method for solving linear system of equations. *International Journal of Computing Science and Applied Mathematics,* 5(2), 41-47.

Tesfaye, K. E., Awgichew, G., Haile, E., & Gashaye, D. A. (2020). Second refinement of Gauss-Seidel iteration method for solving linear system of equations. *Ethiopia Journal of Science and technology,* 13(1), 1-15.

Wang, L., & Song, Y. (2009). Preconditioned AOR iterative method for M −matrices. *Applied Numerical Mathematics*, 57, 672-685.

Wu, S., & Yu-Jun, L. (2014). A new version of accelerated over-relaxation iterative method. *Journal of Applied Mathematics,* 2014, 1-6.

Varga, R. S. (2000). *Matrix Iterative Analysis*. Springer Berlin, Germany.

Vatti, V. B. K. (2016). *Numerical Analysis Iterative Methods.* I. K International Publishing House, Pvt, Limited, New Delhi, India.

Vatti, V. B. K., & Gonfa, G. (2011). A refinement of generalized Jacobi method for the solution of linear systems. *International Journal of Contemporary Mathematics and Sciences,* 6(3), 109-116.

Vatti, K., Sri, R., & Mylapalli, M. S. (2018). A refinement of accelerated over relaxation method for the solution of linear systems. *International Journal of Pure and Applied Mathematics,* 118, 1571-1577.

Vatti, V. B. K., & Tesfaye, E. K. (2011). A refinement of Gauss-Seidel method for solving of linear system of equations. *International Journal of Contemporary Mathematics and Sciences,* 63, 117 -127.

Vatti, V. B. K., Dominic, S., & Sahanica, S. (2015). A refinement of successive over-relaxation (RSOR) method for solving of linear system of equations. *International Journal of Advanced Information, Science and Technology (IJAIST).* 40, 1-4.

Vatti, V. B. K., Rao, G. C., & Pai, S. S. (2019a). Parametric accelerated over-relaxation (PAOR) method. *Numerical Optimization in Engineering and Sciences, select Proceedings in NOIEAS,* 979, 283-288.

Vatti, V. B. K., Kumar, M. S., & Rao, G. C. (2019b). Generalized PAOR method for solving non-square linear systems. *Compliance Engineering Journal,* 10(10), 77-84.

Vatti, V. B. K., Rao, G. C., & Kumar, M. S. (2020a). Second degree SOR (SDSOR) method. I*nternational Journal of Research and Innovation in Applied Science,* 4(7), 41-44.

Vatti, V. B. K., Rao, G. C., & Pai, S. S. (2020b). Reaccelerated over-relaxation (ROR) method. *Bulettin of the International Mathematical Virtual Institute,* 10(2), 315-324.

Yun, J. E. (2011). Comparison results of the preconditioned AOR methods for $L-$matrices. *Applied Mathematics and Computation*, 218, 3399-3413.

Young, D. M. (2014). *Iterative Solutions of Large Linear Systems*. Elsevier.

Youssef, I. K. (2012). On the successive over-relaxation method. *Journal of Mathematics and Statistics*, 8(2), 176-184.

Youssef, I. K., & Taha, A. A. (2013). On the modified successive over-relaxation method. *Applied Mathematics Computation, Elsevier*, 219, 4601- 4613.

Youssef, I. K., & Farid, M. M. (2015). On the accelerated over- relaxation method. *Pure and Applied Mathematics.* 4, 26-31.

Zhang, C., Miao, G., & Zhu, Y. (2016a). Convergence on successive over relaxed methods for hermitian positive definite linear systems. *Journal of Inequalities and Application,* 156.

Zhang, C., Xue, Z., & Luo, S. (2016b). A convergence analysis of SOR iterative methods for linear system with weak h-matrices. *DeGruyter Open Mathematics* 14, 747-760.

Zhang, X., Wang, Q., & Li, T. (2020). The accelerated over-relaxation splitting method for solving symmetric tensor equations. *Computational and Applied Mathematics,* 39(155), 1-14.

APPENDIX A

4.1.6 Convergence Results Comparison for Problem 1

4.1.6.1 Convergence result of the refinement of AOR and EAOR iterations for problem 1

No. of Iterations	Mesh Points	RAOR	REAOR
		1236.4687500000	
1	*u11*	1291.0437500000	1304.9526367188
	u21	1502.0940625000	1412.2286865234
	u31	1056.2583750000	1638.1896817017
	u12	502.4807343750	1184.4412795105
	u22	933.6835631250	631.3152231436
	u32	1235.8586281875	1067.8795106394
	u13	1281.5848900062	1367.5862175121
	u23	1254.3551146256	1386.3523976543
	u33		1336.1252975789
2	*u11*	1606.9094213027	1669.7399082996
	u21	1804.0051930891	1903.0008769630
	u31	2043.4301701314	2163.6675282184
	u12	1556.2434457759	1665.1341402297
	u22	995.2725096422	1099.5246318384
	u32	1297.2520705184	1372.3038679849
	u13	1526.8872496598	1581.9902027415
	u23	1453.2816761476	1485.4195375818
	u33	1354.4414342180	1370.8249450870
⋮	⋮	⋮	⋮
17	*u11*	1838.0436562400	1838.0436564310
	u21	2080.7015624424	2080.7015626707
	u31	2315.7646703298	2315.7646705278
	u12	1771.4730629060	1771.4730630532
	u22	1168.9979236253	1168.9979237242
	u32	1410.8840563270	1410.8840563873
	u13	1601.1296579279	1601.1296579609
	u23	1493.6345754394	1493.6345754562
	u33	1373.4086438572	1373.4086438641

No. of Iterations	Mesh Points	RAOR	REAOR
⋮	⋮	⋮	
24	*u11*	1838.0436564310	
	u21	2080.7015626707	
	u31	2315.7646705278	
	u12	1771.4730630532	
	u22	1168.9979237242	
	u32	1410.8840563873	
	u13	1601.1296579609	
	u23	1493.6345754562	
	u33	1373.4086438641	

4.1.6.2 Convergence result of AOR, its variant and proposed EAOR iterations for problem 1

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
1	*u11*				
	u21	437.5000000000	625.000000000000	875.0000000000	875.0000000000
	u31	396.8750000000	580.357142857143	837.5000000000	946.8750000000
	u12	519.8437500000	755.739795918367	1083.7500000000	1213.0468750000
	u22	275.9921875000	411.124271137026	608.3750000000	772.9355468750
	u32	13.7996093750	29.3660193669305	60.8375000000	173.9104980469
	u13	188.1899804688	269.954715669067	381.0837500000	414.1298620605
	u23	446.9094990234	644.282479690648	913.1083750000	968.1792189636
	u33	397.3454749512	581.734462835046	841.3108375000	967.8403242668
		519.8672737476	755.838175916789	1084.1310837500	1217.7640729600
2					1304.9526367188
	u11	740.3583984375	980.621681070387	1236.4687500000	1412.2286865234
	u21	709.9733398438	978.736770765157	1291.0437500000	1638.1896817017
	u31	883.2089379883	1183.92350969568	1502.0940625000	1184.4412795105
	u12	528.7334604492	760.439395411806	1056.2583750000	631.3152231436
	u22	127.2278236999	258.843773274668	502.4807343750	1067.8795106394
	u32	409.8384487189	616.643807429994	933.6835631250	1367.5862175121
	u13	745.2291048517	985.931568341700	1235.8586281875	1386.3523976543
	u23	709.4358143634	976.349184214988	1281.5848900062	1336.1252975789
	u33	825.2063382133	1062.30739872391	1254.3551146256	
⋮	⋮	⋮	⋮	⋮	⋮
32	*u11*	1837.2271883871	1838.03762473572	1838.0436556409	1838.0436564310
	u21	2079.6788773528	2080.69415855581	2080.7015617262	2080.7015626707
	u31	2314.8313298532	2315.75807419871	2315.7646697086	2315.7646705278
	u12	1770.7396626268	1771.46802217007	1771.4730624440	1771.4730630532
	u22	1168.4750597462	1168.99444273843	1168.9979233152	1168.9979237242
	u32	1410.5448794408	1410.88187256647	1410.8840561379	1410.8840563873
	u13	1600.9326567250	1601.12843019262	1601.1296578245	1601.1296579609
	u23	1493.5270750850	1493.63392975576	1493.6345753866	1493.6345754562
	u33	1373.3619676384	1373.40837334446	1373.4086438359	1373.4086438641

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
⋮	⋮	⋮	⋮	⋮	
	u11	1838.0236163232	1838.04364069216	1838.0436564310	
	u21	2080.6765370368	2080.70154335941	2080.7015626707	
	u31	2315.7419763073	2315.76465333893	2315.7646705278	
	u12	1771.4554004222	1771.47304993529	1771.4730630532	
46	*u22*	1168.9854906433	1168.99791468168	1168.9979237242	
	u32	1410.8761028251	1410.88405072549	1410.8840563873	
	u13	1601.1251015067	1601.12965478395	1601.1296579609	
	u23	1493.6321283725	1493.63457378916	1493.6345754562	
	u33	1373.4075948658	1373.40864316685	1373.4086438641	
⋮	⋮	⋮	⋮		
	z1	1838.0436477242	1838.0436564310		
	z2	2080.7015518080	2080.7015626707		
	z3	2315.7646606965	2315.7646705278		
	z4	1771.4730554246	1771.4730630532		
75	*z5*	1168.9979183758	1168.9979237242		
	z6	1410.8840529811	1410.8840563873		
	z7	1601.1296560182	1601.1296579609		
	z8	1493.6345744183	1493.6345754562		
	z9	1373.4086434210	1373.4086438641		
	z10				
⋮	⋮	⋮			
	z1	1838.0436564310			
	z2	2080.7015626707			
	z3	2315.7646705278			
	z4	1771.4730630532			
117	*z5*	1168.9979237242			
	z6	1410.8840563873			
	z7	1601.1296579609			
	z8	1493.6345754562			
	z9	1373.4086438641			
	z10				

APPENDIX B

4.2.6 Convergence Results Comparison for Problem 2

4.2.6.1 Convergence result of refinement methods of AOR and EAOR iterations for problem 2

No. of Iterations	Mesh Points	RAOR	REAOR
1	*z1*	1.2323621947	1.3009977786
	z2	0.7821356178	0.8691362620
	z3	0.5952761031	0.7193102010
	z4	0.6553817158	0.7728172528
	z5	0.6121110766	0.7594999494
	z6	0.6712689317	0.8169147509
	z7	0.6287630019	0.7984740553
	z8	0.6868673263	0.8575300610
	z9	0.6451444972	0.8346928331
	z10	0.7020950596	0.8932832741
2	*z1*	1.5019054672	1.5769260836
	z2	1.0979781894	1.1943510602
	z3	0.9192173511	1.0237616260
	z4	0.9621591852	1.0655593249
	z5	0.9263548298	1.0368923681
	z6	0.9690113134	1.0781620034
	z7	0.9332493068	1.0486783324
	z8	0.9756791618	1.0894767813
	z9	0.9399115826	1.0592377811
	z10	0.9821653541	1.0996726248
⋮	⋮	⋮	⋮

No. of Iterations	Mesh Points	RAOR	REAOR
22	*z1*	1.6995918301	1.6995918367
	z2	1.3314285635	1.3314285714
	z3	1.1551020328	1.1551020408
	z4	1.1885714209	1.1885714286
	z5	1.1551020331	1.1551020408
	z6	1.1885714211	1.1885714286
	z7	1.1551020333	1.1551020408
	z8	1.1885714213	1.1885714286
	z9	1.1551020335	1.1551020408
	z10	1.1885714215	1.1885714286
⋮	⋮	⋮	
31	*z1*	1.6995918367	
	z2	1.3314285714	
	z3	1.1551020408	
	z4	1.1885714286	
	z5	1.1551020408	
	z6	1.1885714286	
	z7	1.1551020408	
	z8	1.1885714286	
	z9	1.1551020408	
	z10	1.1885714286	

4.2.6.2 Convergence result of AOR, its variants and EAOR iterations for problem 2

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
1	*z1*	0.5000000000	0.7142857143	1.0000000000	1.0000000000
	z2	0.2285714286	0.3352769679	0.4857142857	0.5571428571
	z3	0.1493877551	0.2177664068	0.3134693877	0.3573469388
	z4	0.1614110787	0.2421245764	0.3607696793	0.4602303207
	z5	0.1539995002	0.2276490426	0.3340847980	0.4165194086
	z6	0.1658110644	0.2514163740	0.3798602392	0.5137828161
	z7	0.1587369592	0.2379109354	0.3557910974	0.4825771992
	z8	0.1703464061	0.2611270245	0.4001911590	0.5758284560
	z9	0.1636039994	0.2485691813	0.3786591636	0.5566122864
	z10	0.1750208061	0.2712727053	0.4218288255	0.6473928928
2	*z1*	0.8018671413	1.0295862186	1.2323621947	1.3009977786
	z2	0.4176053622	0.5827629584	0.7821356178	0.8691362620
	z3	0.2873206174	0.4153018350	0.5952761031	0.7193102010
	z4	0.3165375547	0.4621698436	0.6553817158	0.7728172528
	z5	0.2940586749	0.4271068847	0.6121110766	0.7594999494
	z6	0.3227392383	0.4729657794	0.6712689317	0.8169147509
	z7	0.3009110665	0.4390816870	0.6287630019	0.7984740553
	z8	0.3290690265	0.4839513020	0.6868673263	0.8575300610
	z9	0.3078795186	0.4512217686	0.6451444972	0.8346928331
	z10	0.3355283842	0.4951216021	0.7020950596	0.8932832741
⋮	⋮	⋮	⋮	⋮	⋮
43	*z1*	1.6990443291	1.6995829427	1.6995918265	1.6995918367
	z2	1.3307659512	1.3314178713	1.3314285592	1.3314285714
	z3	1.1544313567	1.1550912552	1.1551020286	1.1551020408
	z4	1.1879179337	1.1885609487	1.1885714167	1.1885714286
	z5	1.1544402481	1.1550914692	1.1551020289	1.1551020408
	z6	1.1879269381	1.1885611646	1.1885714171	1.1885714286
	z7	1.1544490171	1.1550916788	1.1551020293	1.1551020408
	z8	1.1879358231	1.1885613762	1.1885714174	1.1885714286
	z9	1.1544576651	1.1550918841	1.1551020296	1.1551020408
	z10	1.1879445905	1.1885615836	1.1885714178	1.1885714286

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
⋮	⋮	⋮	⋮	⋮	
	z1	1.6995643104	1.6995917496	1.6995918367	
	z2	1.3313952577	1.3314284666	1.3314285714	
	z3	1.1550683217	1.1551019351	1.1551020408	
	z4	1.1885385737	1.1885713259	1.1885714286	
60	*z5*	1.1550687687	1.1551019372	1.1551020408	
	z6	1.1885390264	1.1885713280	1.1885714286	
	z7	1.1550692096	1.1551019393	1.1551020408	
	z8	1.1885394731	1.1885713300	1.1885714286	
	z9	1.1550696444	1.1551019413	1.1551020408	
	z10	1.1885399139	1.1885713321	1.1885714286	
⋮	⋮	⋮	⋮		
	z1	1.6995917672	1.6995918367		
	z2	1.3314284872	1.3314285714		
	z3	1.1551019556	1.1551020408		
	z4	1.1885713455	1.1885714286		
94	*z5*	1.1551019567	1.1551020408		
	z6	1.1885713467	1.1885714286		
	z7	1.1551019578	1.1551020408		
	z8	1.1885713478	1.1885714286		
	z9	1.1551019589	1.1551020408		
	z10	1.1885713489	1.1885714286		
⋮	⋮	⋮			
	z1	1.6995918367			
	z2	1.3314285714			
	z3	1.1551020408			
	z4	1.1885714286			
144	*z5*	1.1551020408			
	z6	1.1885714286			
	z7	1.1551020408			
	z8	1.1885714286			
	z9	1.1551020408			
	z10	1.1885714286			

APPENDIX C

4.3.6 Convergence Results Comparison for Problem 3

4.3.6.1 Convergence result of refinement methods of AOR and EAOR iterations for problem 3

No. of Iterations	Mesh Points	RAOR	REAOR
1	$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$	$\begin{bmatrix} 0.7397375985 \\ 1.3602783809 \\ 1.1123171203 \\ 1.1812484666 \\ 1.0056814497 \\ 0.4236011693 \end{bmatrix}$	$\begin{bmatrix} 0.7859665595 \\ 1.3072609215 \\ 1.1550344754 \\ 1.2418391991 \\ 0.9526024186 \\ 0.4287141109 \end{bmatrix}$
⋮	⋮	⋮	⋮
13	$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$	$\begin{bmatrix} 0.7107095752 \\ 1.0878085305 \\ 1.1093417611 \\ 1.2217138640 \\ 1.0306242506 \\ 0.4949672600 \end{bmatrix}$	$\begin{bmatrix} 0.7107095747 \\ 1.0878085301 \\ 1.1093417628 \\ 1.2217138647 \\ 1.0306242498 \\ 0.4949672604 \end{bmatrix}$
⋮	⋮	⋮	

No. of Iterations	Mesh Points	RAOR	REAOR
17	$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$	$\begin{bmatrix} 0.7107095747 \\ 1.0878085301 \\ 1.1093417628 \\ 1.2217138647 \\ 1.0306242498 \\ 0.4949672604 \end{bmatrix}$	

4.3.6.2 Convergence result of AOR, its variant and EAOR iterations for problem 3

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
1	z_1	0.5000000000	0.6711409396	1.0000000000	1.0000000000
	z_2	0.5755000000	0.7612269718	1.1020000000	1.0020000000
	z_3	0.3718005000	0.4868454743	0.6920040000	.6016040000
	z_4	0.7086901378	0.9545457212	1.4300858040	1.4400352040
	z_5	0.8221196171	1.069540965	1.5001154036	1.2407249228
	z_6	0.3151475445	0.4095560650	0.5767229584	.5092646882
⋮	⋮	⋮	⋮	⋮	⋮
24	z_1	0.7107204796	0.7107102180	0.7107095758	0.7107095747
	z_2	1.0878423851	1.087806098	1.0878085217	1.0878085301
	z_3	1.1092597110	1.109343182	1.1093417688	1.1093417628
	z_4	1.2216890914	1.221712476	1.2217138609	1.2217138647
	z_5	1.0306668101	1.030623879	1.0306242485	1.0306242498
	z_6	0.4949351488	0.4949682042	0.4949672636	0.4949672604
⋮	⋮	⋮	⋮	⋮	
32	z_1	0.7107080939	0.7107095596	0.7107095747	
	z_2	1.0878095004	1.087808559	1.0878085301	
	z_3	1.1093446835	1.109341768	1.1093417628	
	z_4	1.2217169380	1.221713895	1.2217138647	
	z_5	1.0306223761	1.030624242	1.0306242498	
	z_6	0.4949678853	0.4949672551	0.4949672604	
⋮	⋮	⋮	⋮		

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
45	$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$	$\begin{bmatrix} 0.7107095747 \\ 1.0878085513 \\ 1.1093417314 \\ 1.2217138635 \\ 1.0306242648 \\ 0.4949672462 \end{bmatrix}$	$\begin{bmatrix} 0.7107095747 \\ 1.0878085301 \\ 1.1093417628 \\ 1.2217138647 \\ 1.0306242498 \\ 0.4949672604 \end{bmatrix}$		
⋮	⋮	⋮			
61	$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$	$\begin{bmatrix} 0.7107095747 \\ 1.0878085301 \\ 1.1093417628 \\ 1.2217138647 \\ 1.0306242498 \\ 0.4949672604 \end{bmatrix}$			

APPENDIX D

4.4.6 Convergence Results Comparison for Problem 4

4.4.6.1 Convergence result of refinement methods of AOR and EAOR iterations for problem 4

No. of Iterations	Mesh Points	RAOR	REAOR
	u11	0.0769530731	0.0803623390
	u21	0.0291252282	0.0386064744
	u31	0.0145244380	0.0222635471
	u41	0.0140576204	0.0188764251
	u51	0.0242595481	0.0276869140
	u12	0.1452560864	0.1572094591
	u22	0.0489477296	0.0713299984
	u32	0.0197627301	0.0369512315
	u42	0.0244315532	0.0351662493
	u52	0.0626693690	0.0710353603
	u13	0.2069164657	0.2286066412
	u23	0.0634305347	0.0986642626
1	*u33*	0.0219781645	0.0472437231
	u43	0.0368419064	0.0533720767
	u53	0.1227856256	0.1384371429
	u14	0.3250620764	0.3558961421
	u24	0.1597035508	0.2226991368
	u34	0.1042646278	0.1609610938
	u44	0.1253621131	0.1755763622
	u54	0.2699764215	0.3226456388
	u15	0.6103016926	0.6571884999
	u25	0.4941595633	0.5662379915
	u35	0.4446331584	0.5201267926
	u45	0.4772680177	0.5573140466
	u55	0.5669762037	0.6396057343
⋮	⋮	⋮	⋮

No. of Iterations	Mesh Points	RAOR	REAOR
54	*u11*	0.1851621824	0.1851621840
	u21	0.1867937056	0.1867937083
	u31	0.1743247517	0.1743247547
	u41	0.1463767031	0.1463767056
	u51	0.0992822425	0.0992822439
	u12	0.3574358660	0.3574358687
	u22	0.3598044182	0.3598044227
	u32	0.3395021099	0.3395021150
	u42	0.2928184895	0.2928184937
	u52	0.2123963247	0.2123963269
	u13	0.5185943340	0.5185943370
	u23	0.5260250557	0.5260250607
	u33	0.5068066901	0.5068066957
	u43	0.4575476329	0.4575476376
	u53	0.3735473041	0.3735473067
	u14	0.6564256346	0.6564256370
	u24	0.6867597211	0.6867597253
	u34	0.6766755101	0.6766755149
	u44	0.6362085272	0.6362085311
	u54	0.5621772419	0.5621772441
	u15	0.8431284461	0.8431284475
	u25	0.8523658994	0.8523659017
	u35	0.8458519130	0.8458519157
	u45	0.8221319160	0.8221319182
	u55	0.7762326147	0.7762326159
⋮	⋮	⋮	

No. of Iterations	Mesh Points	RAOR	REAOR
78	*u11*	0.1851621840	
	u21	0.1867937083	
	u31	0.1743247547	
	u41	0.1463767056	
	u51	0.0992822439	
	u12	0.3574358687	
	u22	0.3598044227	
	u32	0.3395021150	
	u42	0.2928184937	
	u52	0.2123963269	
	u13	0.5185943370	
	u23	0.5260250607	
	u33	0.5068066957	
	u43	0.4575476376	
	u53	0.3735473067	
	u14	0.6564256370	
	u24	0.6867597253	
	u34	0.6766755149	
	u44	0.6362085311	
	u54	0.5621772441	
	u15	0.8431284475	
	u25	0.8523659017	
	u35	0.8458519157	
	u45	0.8221319182	
	u55	0.7762326159	

4.4.6.2 Convergence result of AOR, its variant and EAOR iterations for problem 4

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
1	*u11*				
	u21	0.0254777070	0.0363967243	0.0509554140	0.0509554140
	u31	0.0047314665	0.0071801230	0.0117061772	0.0189424490
	u41	0.0031082724	0.0045583688	0.0069218421	0.0106110664
	u51	0.0028274176	0.0041058507	0.0060420944	0.0081502466
	u12	0.0071335903	0.0102499226	0.0145992421	0.0161827218
	u22	0.0475440585	0.0682837318	0.0970262983	0.1032784963
	u32	0.0063170657	0.0099017535	0.0174297358	0.0351060526
	u42	0.0032836142	0.0049152474	0.0079531110	0.0165141847
	u52	0.0028943609	0.0042485985	0.0064700913	0.0109807781
	u13	0.0202802036	0.0291106921	0.0413374569	0.0450657504
	u23	0.0666137182	0.0959087827	0.1372628518	0.1513455439
	u33	0.0073136377	0.0116589969	0.0213469297	0.0484320142
	u43	0.0032130014	0.0048597128	0.0081433994	0.0204193928
	u53	0.0027710117	0.0040776072	0.0062770489	0.0121883671
	u14	0.0408072240	0.0586025898	0.0833022096	0.0907201443
	u24	0.0820783891	0.1183915305	0.1703479536	0.1929570666
	u34	0.0078312442	0.0126188963	0.0236919439	0.0583033344
	u44	0.0030546967	0.0046501360	0.0079674346	0.0226974864
	u54	0.0026018239	0.0038322149	0.0059272693	0.0125384746
	u15	0.0669303072	0.0962110568	0.1371170179	0.1503141340
	u25	0.2425690806	0.3480444367	0.4934462930	0.5248455754
	u35	0.1565421674	0.2270913570	0.3320380288	0.4049501471
	u45	0.1446144337	0.2090412871	0.3029101608	0.3621251928
	u55	0.1376129313	0.1989048516	0.2880268433	0.3412577704
		0.2258582656	0.3258948933	0.4694976175	0.5389576713
⋮	⋮	⋮	⋮	⋮	⋮

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
107	*u11*	0.1851040093	0.1851610487	0.1851621821	0.1851621840
	u21	0.1866950532	0.1867917898	0.1867937051	0.1867937083
	u31	0.1742135923	0.1743226007	0.1743247511	0.1743247547
	u41	0.1462832527	0.1463749012	0.1463767026	0.1463767056
	u51	0.0992303909	0.0992812463	0.0992822423	0.0992822439
	u12	0.3573372398	0.3574339507	0.3574358655	0.3574358687
	u22	0.3596369185	0.3598011770	0.3598044173	0.3598044227
	u32	0.3393133369	0.3394984700	0.3395021090	0.3395021150
	u42	0.2926604427	0.2928154528	0.2928184887	0.2928184937
	u52	0.2123119087	0.2123947085	0.2123963243	0.2123963269
	u13	0.5184833814	0.5185921870	0.5185943335	0.5185943370
	u23	0.5258355379	0.5260214014	0.5260250547	0.5260250607
	u33	0.5065922341	0.5068025696	0.5068066890	0.5068066957
	u43	0.4573664552	0.4575441641	0.4575476320	0.4575476376
	u53	0.3734459039	0.3735453696	0.3735473036	0.3735473067
	u14	0.6563336580	0.6564238611	0.6564256341	0.6564256370
	u24	0.6865992809	0.6867566385	0.6867597203	0.6867597253
	u34	0.6764931192	0.6766720182	0.6766755092	0.6766755149
	u44	0.6360537959	0.6362055753	0.6362085264	0.6362085311
	u54	0.5620899036	0.5621755816	0.5621772415	0.5621772441
	u15	0.8430756820	0.8431274323	0.8431284458	0.8431284475
	u25	0.8522749449	0.8523641580	0.8523658989	0.8523659017
	u35	0.8457485080	0.8458499403	0.8458519125	0.8458519157
	u45	0.8220440791	0.8221302462	0.8221319156	0.8221319182
	u55	0.7761829312	0.7762316735	0.7762326144	0.7762326159
⋮	⋮	⋮	⋮	⋮	

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
	u11	0.1851606853	0.1851621790	0.1851621840	
	u21	0.1867911667	0.1867936999	0.1867937083	
	u31	0.1743218909	0.1743247453	0.1743247547	
	u41	0.1463742981	0.1463766977	0.1463767056	
	u51	0.0992809080	0.0992822395	0.0992822439	
	u12	0.3574333278	0.3574358603	0.3574358687	
	u22	0.3598001075	0.3598044085	0.3598044227	
	u32	0.3394972517	0.3395020991	0.3395021150	
	u42	0.2928144220	0.2928184804	0.2928184937	
	u52	0.2123941521	0.2123963198	0.2123963269	
	u13	0.5185914786	0.5185943276	0.5185943370	
	u23	0.5260201783	0.5260250447	0.5260250607	
155	*u33*	0.5068011708	0.5068066777	0.5068066957	
	u43	0.4575429700	0.4575476224	0.4575476376	
	u53	0.3735446944	0.3735472982	0.3735473067	
	u14	0.6564232675	0.6564256293	0.6564256370	
	u24	0.6867555920	0.6867597118	0.6867597253	
	u34	0.6766708160	0.6766754996	0.6766755149	
	u44	0.6362045449	0.6362085182	0.6362085311	
	u54	0.5621749941	0.5621772368	0.5621772441	
	u15	0.8431270881	0.8431284430	0.8431284475	
	u25	0.8523635585	0.8523658941	0.8523659017	
	u35	0.8458492517	0.8458519070	0.8458519157	
	u45	0.8221296553	0.8221319109	0.8221319182	
	u55	0.7762313360	0.7762326118	0.7762326159	
⋮	⋮	⋮	⋮		

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
237	*u11*	0.1851621811	0.1851621840		
	u21	0.1867937034	0.1867937083		
	u31	0.1743247492	0.1743247547		
	u41	0.1463767010	0.1463767056		
	u51	0.0992822413	0.0992822439		
	u12	0.3574358638	0.3574358687		
	u22	0.3598044144	0.3598044227		
	u32	0.3395021056	0.3395021150		
	u42	0.2928184858	0.2928184937		
	u52	0.2123963227	0.2123963269		
	u13	0.5185943315	0.5185943370		
	u23	0.5260250513	0.5260250607		
	u33	0.5068066850	0.5068066957		
	u43	0.4575476286	0.4575476376		
	u53	0.3735473016	0.3735473067		
	u14	0.6564256325	0.6564256370		
	u24	0.6867597173	0.6867597253		
	u34	0.6766755058	0.6766755149		
	u44	0.6362085234	0.6362085311		
	u54	0.5621772398	0.5621772441		
	u15	0.8431284448	0.8431284475		
	u25	0.8523658972	0.8523659017		
	u35	0.8458519105	0.8458519157		
	u45	0.8221319139	0.8221319182		
	u55	0.7762326134	0.7762326159		
⋮	⋮	⋮			

No. of Iterations	Mesh Points	QAOR	KAOR	AOR	EAOR
	u11	0.1851621840			
	u21	0.1867937083			
	u31	0.1743247547			
	u41	0.1463767056			
	u51	0.0992822439			
	u12	0.3574358687			
	u22	0.3598044227			
	u32	0.3395021150			
	u42	0.2928184937			
	u52	0.2123963269			
	u13	0.5185943370			
	u23	0.5260250607			
350	*u33*	0.5068066957			
	u43	0.4575476376			
	u53	0.3735473067			
	u14	0.6564256370			
	u24	0.6867597253			
	u34	0.6766755149			
	u44	0.6362085311			
	u54	0.5621772441			
	u15	0.8431284475			
	u25	0.8523659017			
	u35	0.8458519157			
	u45	0.8221319182			
	u55	0.7762326159			

Maple Program for the EAOR Method

```
with(LinearAlgebra) :
with(Student[NumericalAnalysis]) :
with(ArrayTools) :
with(ExcelTools) :

B := Import( );
Diag := Diagonal(B) :
D1 := Diagonal(Diag);
A := D1^(-1) • B;
U := -UpperTriangle(A, 1);
L := -LowerTriangle(A,-1);
i :=   ; %identity matrix of the dimension of A

ω :=  ;
r :=  ;
v :=  ;

S := (i - (v + r)·L)^(-1) • ((1 - ω)·i + (ω - v - r)·L + ω·U);

C := Import( );

b := D1^(-1) • C;
H := A^(-1) • b;

x_0 :=   ; % zero matix

P := ( i - (r + v).L)^(-1).(ω·b);
N := %number of iterations

for i from 0 by 1 to N do
x_(i+1) := S • x_i + P
end do
```

Maple Program for the REAOR Method

$with(LinearAlgebra):$
$with(Student[NumericalAnalysis]):$
$with(ArrayTools):$
$with(ExcelTools):$

$B := Import(\ );$
$Diag := Diagonal(B):$
$D1 := Diagonal(Diag);$
$A := D1^{-1} \bullet B;$
$U := -UpperTriangle(A, 1);$
$L := -LowerTriangle(A, -1);$
$i := \ ;$ *%identity matrix of the dimension of A*

$\omega := \ ;$
$r := \ ;$
$v := \ ;$

$S1 := (i - (v + r) \cdot L)^{-1} \bullet ((1 - \omega) \cdot i + (\omega - v - r) \cdot L + \omega \cdot U);$

$S := S1^2;$
$C := Import(\ );$

$b := D1^{-1} \bullet C;$
$H := A^{-1} \bullet b;$

$x_0 := \ ;$ *% zero matix*

$P := (i + S1).(i - (r + v).L)^{-1}.(\omega \cdot b);$
$N :=$ *%number of iterations*

for i **from** 0 **by** 1 **to** N **do**
$x_{i+1} := S \bullet x_i + P$
end do