SECOND REFINEMENT OF PRECONDITIONED ACCELERATED OVERRELAXATION METHOD FOR SOLUTION OF LINEAR ALGEBRAIC SYSTEM

BY

OBED, Chuks Joseph
MTech/SPS/2018/8231

DEPARTMENT OF MATHEMATICS

FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

AUGUST, 2023

SECOND REFINEMENT OF PRECONDITIONED ACCELERATED OVERRELAXATION METHOD FOR SOLUTION OF LINEAR ALGEBRAIC SYSTEM

BY

OBED, Chuks Joseph
MTech/SPS/2018/8231

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF TECHNOLOGY IN APPLIED MATHEMATICS

AUGUST, 2023

ABSTRACT

This present work concerns the numerical solution of linear system of algebraic equation $Ax = b$ by second refinement of accelerated overrelaxation (AOR) method. This technique is especially useful in solving linear system arising from discretisation of ordinary differential equations or partial differential equation where the coefficient matrix is an irreducibly diagonally dominant L- matrix. A suitable preconditioner is applied to the linear system before a second refinement algorithm is processed. As in all iterative methods for linear systems, this is aimed at minimizing the spectral radius in order to reduce the number of iterations needed for convergence. Hence, the SRPAOR method converges faster than AOR, PAOR, RAOR and RPAOR by a factor of 5.75, 3, 2.87 and 1.5 respectively. Optimum convergence is attained when $r = 1.0, \omega = 1.1$ and when $r = 0.99, \omega = 1.0$. Numerical examples proved the efficiency of second refinement of preconditioned AOR over the AOR, preconditioned AOR and first refinement of AOR methods. The techniques of preconditioning and second refinement have been exploited to introduce a new approach towards improving the rate of convergence of the AOR iterative method in solving linear system of equations. The implication of the method indicates an enhancement or modification to the original PAOR method, which led to improved accuracy in solving linear algebraic systems.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background of Study

Iterative methods are a most preferred set of methods for solution of linear systems of equation

$$Hx = c \tag{1.1}$$

where H is a nonsingular square matrix of size $n \times n$ with nonvanishing diagonal entries, x and c are unknown and known vectors respectively. A great many iterative methods abound and, of course, not every linear system can be solved by an iterative method (Adil *et al*., 2019). However, the origin of linear systems that are given to iterative methods can be traced to the discretisation of partial differential equations of elliptic type by finite difference method, finite element method, or finite volume method. In such systems, the coefficient matrix is usually sparse and large; these are traits that iterative methods exploit to full advantage in order to obtain faster convergence than direct methods such as Gaussian elimination (Eneyew *et al.*, 2020a). Although direct methods accommodate larger classes of linear systems, they suffer from disadvantage of consuming large amount of time and storage, as the sparse structure of the system gradually accommodates fill-in as computation progresses (Adsuara *et al*., 2016).

An iterative method for solving the linear system (1.1) consists of a process whereby the system $Hx = c$ is converted not an equivalent system of the form

$$x = \mathcal{L}x + k \tag{1.2}$$

Once the system is in the form (1.2), the sequence of solution vectors can be obtained through the general linear iteration formula

$$x^{(n+1)} = \mathcal{L}_n x^{(n)} + k_n \tag{1.3}$$

Where $\mathcal{L}_n$ referred to as the iteration matrix, is a matrix depending upon A and x, and k_n is a column vector. At each step of the iteration a solution vector, x^n that is more accurate approximates the solution to the linear system than its predecessor and its procedure.

1.2 Statement of the Research Problem

When large sparse linear systems are to be solved, the method of choice is obviously iterative methods (Assefa and Teklehaymanot, 2021). However, the number of iterations needed for such methods to attain convergence could be relatively large, which could impact negatively on computer storage and computational efficiency (Faruk and Ndanusa, 2019). When such is the case, the need arises to remodel or redesign the existing methods so as to obtain approximate solutions that attain faster convergence (Eneyew *et al.*, 2019; Abdullahi and Ndanusa, 2020). Hence, the present study sought to develop a preconditioning and second refinement method for the solution of the linear system.

1.3 Aim and Objectives of the Study

The aim of this research is to investigate the successive application of two acceleration techniques, preconditioning and second refinement, to the solution of linear systems with a view to reducing the spectral radius of the iteration matrix to the barest minimum so as to attain convergence in a few number of iterations.

The objectives are to:

1. formulate or identify a suitable preconditioner for the AOR method
2. formulate a second refinement for the preconditioned AOR method
3. Identify restrictions imposed on coefficient matrix of the resulting linear system
4. establish convergence of the proposed technique
5. Validate the convergence results through numerical experiments
6. Conduct comparative convergence analysis of the various methods studied
7. establish the rates of convergence of the various methods studied

8. compare the rates of convergence of the second refinement of AOR method with those of existing methods.

1.4 Justification for the Study

Considerable amount of storage is required to store intermediate results when solving very large linear systems by direct methods. However, with iterative solution methods especially where the coefficient matrix is sparse, the presence of large number of zero entries can be taken into advantage in order to minimise the time and amount of storage space used (Mayaki and Ndanusa, 2019). Therefore, iterative methods are more desirable for solving large and sparse linear systems. Employing the dual techniques of preconditioning and second refinement will go a long a way in increasing the rates of convergence of iterative methods.

1.5 Significance of the Study

A great many real-life problems in engineering and social sciences are modelled as partial differential equations. When a partial differential equation is solved, the corresponding physical problems it represents is in effect solved as well. The resort to the solution of many partial differential equations that defy analytical solution is obviously by discretisation. The discretisation procedure eventually ends up as linear system of equations, the solution of which is obtained by iteration techniques. This research seeks to improve the convergence rate of one of such iterative techniques.

1.6 Scope and Limitations of Study

This research work involves the formulation, convergence investigation and the implementation of a second refinement method to accelerate the convergence of the preconditioned AOR method for solving linear algebraic systems.’

The limitations of this study include:

1. The second refinement of AOR method is limited to linear systems whose coefficient matrix is an irreducibly diagonally dominant L-matrix.
2. The third refinement and subsequently n*th* (*n*> 2) refinement of AOR method is yet to be undertaken.

1.7 Definition of Terms

***L*-matrix** A *Z*-matrix $A = (a_{ij}) \in R^{n \times n}$ with $a_{ii} > 0,\ i = 1(1)n$

***M*-matrix** An L-matrix $A = (a_{ij}) \in R^{n \times n}$ where A is nonsingular and $A^{-1} \geq 0$.

Negative matrix A matrix $A = (a_{ij})$ where $a_{ij} < 0,\ i,j = 1(1)n$.

Nonnegative matrix A matrix $A = (a_{ij})$ where $a_{ij} \geq 0,\ i,j = 1(1)n$.

Nonpositive matrix A matrix $A = (a_{ij})$ where $a_{ij} \leq 0,\ i,j = 1(1)n$.

Positive matrix A matrix $A = (a_{ij})$ where $a_{ij} > 0,\ i,j = 1(1)n$.

Property A A square matrix $A = (a_{ij})$ is said to have property A if there exists a set W as the union of two disjoint subsets U and V such that if either $a_{ij} \neq 0$ or $a_{ji} \neq 0$ then $i \in U$ and $j \in V$ or $i \in V$ and $j \in U$.

Spectral norm The spectral norm of an $n -$ square matrix A, denoted by $||A||_2$, is the square root of the maximum eigenvalue of A^*A, i.e.,

$$||A||_2 = (\text{maximum eigenvalue of } A^*A)^{1/2} \quad (1.5)$$

where A^* is the transpose conjugate of A. According to Saad (2000), the spectral norm of A is equal to the spectral radius of A when the matrix is Hermitian.

Spectral radius The maximum among the absolute values of the eigenvalues of an $n -$ square matrix A is called the spectral radius of A. It is denoted by

$$\rho(A) = \max_i |\lambda_i| \quad (1.6)$$

where λ_i $(i = 1(1)n)$ is an eigenvalue of A.

Splitting The decomposition of a real matrix $A \in R^{n\times n}$ into the form $A = M - N$, where M is a nonsingular matrix is called a splitting of A. Such splitting is called

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

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

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

iv. M −splitting if M is a nonsingular M −matrix and $N \geq 0$.

Usual splitting For any matrix B, the decomposition $B = D - L_B - U_B$ in which D is the diagonal of B, $-L_B$ its strict lower part, and $-U_B$ its strict upper part, is called the usual splitting of B. Moreover, with the assumption that B has non-vanishing diagonal entries, we consider the usual splitting $A = I - L - U$, where $I = D^{-1}D$, $L = D^{-1}L_B$ and $U = D^{-1}U_B$.

***Z*-matrix** A matrix $A = (a_{ij}) \in R^{n\times n}$ where $a_{ij} \leq 0$ $(i \neq j)$

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Basic Iterative Methods

Consider the solution of system of linear equations

$$Hx = c, \quad H \in R^{n \times n}, \qquad c \in \text{span (A)} \tag{2.1}$$

We also consider a usual splitting of H into its diagonal, strictly lower and strictly upper parts thus,

$$H = D - E - F \tag{2.2}$$

where D is a diagonal matrix, $-E$ and $-F$ are the strictly lower and strictly upper parts of H respectively. Iterative methods are methods that employ successive approximations in order to arrive at more accurate solutions to a system of linear equations at each step (Naumov, 2011). A basic iterative method is a one – step method of the form $x^{(n+1)} = \mathcal{L}x^{(n)} + k$ where for some nonsingular matrix Q we have $\mathcal{L} = I - Q^{-1}H$ and $k = Q^{-1}c$. The Jacobi, Gauss-Seidel and successive overrelaxation (SOR) are some of the basic iterative methods.

2.2 Jacobi Method

The Jacobi method is based on solving for every variable locally with respect to the other variables. One iteration of the method corresponds to solving for every variable once. The resulting method is easy to understand and implement, but convergence is slow (Barrett *et al.*, 1994). It is constructed from the linear system $Hx = c$ based on the splitting $H = D - E - F$. Saad (2000) stated that the basic single step of the iteration consists in replacing the current value, $x^{(n+1)}$, by the improved value, $x^{(n)}$, obtained from the matrix operations,

$$x^{(n+1)} = D^{-1}(E + F)x^{(n)} + D^{-1}c \tag{2.3}$$

which characterizes the Jacobi method. The matrix equation (2.1) is now in the general iterative form

$$x^{(n+1)} = \mathcal{L}_J x^{(n)} + k_J \qquad n = 0,1,2,\cdots \tag{2.4}$$

where $\mathcal{L}_J = D^{-1}(E+F),\ k_J = D^{-1}c,$

In equation (2.4), the subscripts on $\mathcal{L}$ and k are just to emphasize the Jacobi method. According to Ames (1977) the algebraic form of (2.3) is expressed as

$$x_i^{(n+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{k} a_{ij}x_j^{(n)} \right) \quad i = 1,\dots,k \tag{2.5}$$

where the a_{ij} denote the elements of the coefficient matrix $H = (a_{ij})$, the x_i the elements of x and the b_i the elements of b. The order in which one solves for the components $x_i^{(n)}$ is irrelevant, since the Jacobi method treats them independently. It is for this reason that the Jacobi method is known as the method of simultaneous displacements, since the updates could in principle be done simultaneously.

Numerical Algorithm of Jacobi Method

Input: $H := (a_{ij}), cb, XO = x^{(0)}$, *tolerance* TOL, *maximum number of iterations* N.

Step 1 *Set* $k = 1$

Step 2 while $(k \leq N)$ *do Steps 3-6*

Step 3 for $i = 1,2,\cdots,n$

$$x_i = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1 \\ j\neq i}}^{n}\left(-a_{ij}\boldsymbol{XO}_j\right) + b_i\right],$$

Step 4 *If* $\|\boldsymbol{x} - \boldsymbol{XO}\| < TOL$, *then OUTPUT* $(x_1,\ x_2,\ x_3,\cdots,x_n)$*;*

STOP.

Step 5 *Set* $k = k+1$.

Step 6 *For* $i = 1,2,\cdots,n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 *OUTPUT* $(x_1,\ x_2,\ x_3, \cdots, x_n)$;

STOP.

The matrix $\mathcal{L}_J = D^{-1}(E+F)$ is known as the Jacobi iteration matrix and its spectral radius is defined by

$$\rho(\mathcal{L}_J) = \bar{\mu} \tag{2.6}$$

2.3 Gauss-Seidel Method

The Gauss-Seidel method is like the Jacobi method, except that it uses updated values as soon as they are available. In general, if the Jacobi method converges, the Gauss-Seidel method will converge faster than the Jacobi method, though still relatively slowly (Barrett *et al.*, 1994). The order in which one solves for the components of the nth approximation $x^{(n)}$ must be established beforehand. Such a sequential arrangement is called an ordering of the mesh points. For an arbitrary but fixed ordering, which we designate by x_i $(i = 1,2, \ldots, k)$, where k is the number of mesh points. The algebraic form of the Gauss-Seidel method is given by

$$x_i^{(n+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{k} a_{ij}x_j^{(n)}\right) \quad i = 1, \ldots, k \tag{2.7}$$

The matrix form of the Gauss-Seidel iteration derived from the decomposition of the linear system $Hx = c$ is as follows.

$$x^{(n)} = (D-E)^{-1}Fx^{(n-1)} + (D-E)^{-1}b \tag{2.8}$$

Equation (2.8) is now in the general iterative form

$$x^{(n+1)} = \mathcal{L}_G x^{(n)} + k_G \qquad n = 0,1,2, \cdots \tag{2.9}$$

where $\mathcal{L}_G = (D-E)^{-1}F,\ k_G = (D-E)^{-1}b$,

Numerical Algorithm of Gauss-Seidel Method

Input: $H \coloneqq (a_{ij}), c, XO = x^{(0)}$, *tolerance* TOL, *maximum number of iterations* N.

Step 1 Set $k = 1$

Step 2 while $(k \leq N)$ *do Steps 3-6*

Step 3 for $i = 1,2,\cdots,n$

$$x_i = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j) - \sum_{j=i+1}^{n}(a_{ij}\boldsymbol{XO}_j) + b_i\right],$$

Step 4 If $\|\boldsymbol{x} - \boldsymbol{XO}\| < TOL$, *then OUTPUT* $(x_1,\ x_2,\ x_3,\cdots,x_n)$*;*

STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1,2,\cdots,n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 *OUTPUT* $(x_1,\ x_2,\ x_3,\cdots,x_n)$*;*

STOP.

For the Gauss-Seidel iteration matrix $\mathcal{L}_G = (D - E)^{-1}F$, its spectral radius is found to be the square of spectral radius of Jacobi iteration matrix, that is,

$$\rho(\mathcal{L}_G) = \bar{\mu}^2 \tag{2.10}$$

2.4 Successive Overrelaxation (SOR) Method

Successive Overrelaxation (SOR) can be derived from the Gauss-Seidel method by introducing an extrapolation parameter ω. For the optimal choice of ω, SOR may converge faster than Gauss-Seidel by an order of magnitude. The SOR, seeks to substantially reduce the number of iterations needed to reduce the error of an initial estimate of the solution by a predetermined factor by applying extrapolation to the Gauss-Seidel method. This extrapolation takes the form of a weighted average between the previous iterate and the computed Gauss-Seidel iterate successively for each component.

Letting $\overline{x_i}^{(n+1)}$ be the components of the nth Gauss-Seidel iteration, the SOR iteration is defined by means of the relation

$$x_i^{(n+1)} = \omega\overline{x_i}^{(n+1)} + (1-\omega)x_i^{(n)} \qquad (2.11)$$

where the quantity $0 < \omega < 2$ is the relaxation factor. That is, the accepted value at step $n+1$ is extrapolated from the Gauss-Seidel value and the previous accepted value. The idea is to choose a value for ω that will accelerate the rate of convergence of the iterates to the solution. If $\omega = 1$ the SOR method reduces to that of Gauss-Seidel.

The algebraic form of SOR iteration takes the form

$$x_i^{(n+1)} = \omega\left\{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{k} a_{ij}x_j^{(n)}\right)\right\} + (1-\omega)x_i^{(n)} \qquad (2.12)$$

While the matrix form of the SOR is

$$x^{(n+1)} = (D-\omega E)^{-1}\{(1-\omega)D + \omega F\}x^{(n)} + (D-\omega E)^{-1}\omega b \qquad (2.13)$$

That is,

$$x^{(n+1)} = \mathcal{L}_\omega x^{(n)} + k_\omega \qquad n = 0,1,2,\cdots \qquad (2.14)$$

where $\mathcal{L}_\omega = (D-\omega E)^{-1}\{(1-\omega)D + \omega F\}, k_\omega = (D-\omega E)^{-1}\omega b,$ for $0 < \omega < 2$.

Numerical Algorithm of SOR Method

Input: the number of equations and unknown n; the entries $a_{ij}, 1 \leq i,j \leq n,$ *of the matrix H; the entries* c_i, $1 \leq i \leq n$, *of c; the entries* ***XOᵢ*** $1 \leq i \leq n$, *of* $\boldsymbol{XO} = \boldsymbol{x}^{(\mathbf{0})}$*; the parameter* ω, *tolerance* TOL; *maximum number of iterations* N.

Output: the approximate solution $x_1, x_2, \cdots, x_n$ *or a message that the number of iterations was exceeded.*

Step 1 Set $k = 1$

Step 2 while $(k \leq N)$ *do Steps 3-6*

Step 3 For for $i = 1,2,\cdots,n$

$$x_i = (1-\omega)\boldsymbol{XO}_i + \frac{1}{a_{ii}}\left[\omega\left(-\sum_{j=1}^{i-1}\left(a_{ij}x_j\right) - \sum_{j=i+1}^{n}\left(a_{ij}\boldsymbol{XO}_j\right) + b_i\right)\right],$$

Step 4 *If* $\|\boldsymbol{x} - \boldsymbol{XO}\| < TOL$, *then OUTPUT* $(x_1,\ x_2,\ x_3, \cdots, x_n)$;

STOP.

Step 5 *Set* $k = k + 1$.

Step 6 *For* $i = 1,2,\cdots,n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 *OUTPUT* ('Maximum number of iterations exceeded);

(The procedure was successful)

STOP.

For matrices with certain properties, an optimum value for the relaxation parameter ω that appears in the SOR iteration matrix $\mathcal{L}_\omega = (I - \omega E)^{-1}\{(1-\omega)I + \omega F\}$ is governed by the relation

$$\omega = \frac{2}{1+\sqrt{1-\bar{\mu}^2}} \tag{2.15}$$

And for this choice of ω the spectral radius of the SOR iteration matrix is obtained as

$$\rho(\mathcal{L}_\omega) = \omega - 1 = \frac{1-\sqrt{1-\bar{\mu}^2}}{1+\sqrt{1-\bar{\mu}^2}} \tag{2.16}$$

2.5 AOR Method

The accelerated overrelaxation (AOR) method introduced by Hadjidimos (1978) is a two-parameter generalization of the SOR method. Judicious exploitation of the two parameters involved leads to methods that will converge in minimal number of iterations than other methods of the same type. It has the representation

$$x^{(n+1)} = \mathcal{L}_{r,\omega}x^{(n)} + k_{r,\omega} \qquad n = 0,1,2,\cdots \tag{2.17}$$

where $\mathcal{L}_{r,\omega} = (I - rE)^{-1}[(1-\omega)I + (\omega - r)E + \omega F], k_{r,\omega} = (I - rE)^{-1}\omega b$, for $0 \leq r \leq \omega < 1$.

The matrix defined by $\mathcal{L}_{r,\omega} = (I - rE)^{-1}[(1-\omega)I + (\omega - r)E + \omega F]$ is the AOR iteration matrix whose spectral radius can be computed from the formula

$$\rho(\mathcal{L}_{r,\omega}) = \frac{\underline{\mu}\sqrt{\bar{\mu}^2 - \underline{\mu}^2}}{\sqrt{1 - \underline{\mu}^2}(1 + \sqrt{1 - \bar{\mu}^2})} \tag{2.18}$$

provided

$$0 < \underline{\mu} < \bar{\mu} \text{ and } 1 - \underline{\mu}^2 < \sqrt{1 - \bar{\mu}^2} \tag{2.19}$$

for the optimum values of ω and r

$$\omega = \frac{2}{1 + \sqrt{1 - \bar{\mu}^2}} \text{ and } r = \frac{\left(1 - \underline{\mu}^2\right) - \sqrt{1 - \bar{\mu}^2}}{\left(1 - \underline{\mu}^2\right)(1 + \sqrt{1 - \bar{\mu}^2})} \tag{2.20}$$

respectively; where $\underline{\mu} = \min_i |\mu_i|, = \bar{\mu} = \max_i |\mu_i|$ and $\mu_i | i = 1(1)n$ define the eigenvalues of Jacobi iteration matrix $\mathcal{L}_J = D^{-1}(E + F)$.

2.6 Fundamental Theorem of Iterative Methods

An iterative method is said to converge if, for any given iteration count n, each component of the successive iterants $x^{(n)}$ tends to the corresponding component of the solution vector x for all initial vectors $x^{(0)}$. A necessary condition for all stationary methods to attain convergence is contained in the following theorems.

Theorem 2.1 (Byrne (2008))

The stationary linear iteration $x^{(n+1)} = \mathcal{L}x^{(n)} + k$ converges if and only if the spectral radius of $\mathcal{L}$ is less than 1.

The spectral radius of an iterative matrix $\mathcal{L}$, denoted by $\rho(\mathcal{L})$, is defined as

$$\rho(\mathcal{L}) = \max_{\mu \in \sigma(\mathcal{L})} |\mu| \tag{2.21}$$

where $\sigma(\mathcal{L})$, known as the spectrum of $\mathcal{L}$, is the set of all the eigenvalues of $\mathcal{L}$. The computational effectiveness of a convergent iterative method is directly related to the magnitude of the spectral radius of the matrix $\mathcal{L}$ of the iterative method. The rate of convergence is best when the spectral radius is near zero and poorest when it is near 1. Sufficient conditions for convergence of specific iterative methods could also be derived.

Theorem 2.2 (Byrne (2008))

If the matrix H, of the linear system $Hx = c$, is diagonally dominant (or irreducibly diagonally dominant), then the spectral radii of the Jacobi and Gauss-Seidel matrices are less than 1, and the Jacobi and Gauss-Seidel methods converge.

Theorem 2.3 (Saad (2000))

If the system $Hx = c$ has a symmetric positive definite matrix A, the spectral radius of the Gauss-Seidel iteration matrix is less than 1, and the Gauss-Seidel method always converges, without further restrictions on H.

Theorem 2.4 (Noor *et al.* (2012))

If H is symmetric with positive diagonal elements, then $\rho(\mathcal{L}_\omega) < 1$ if and only if H is positive definite and $0 < \omega < 2$.

2.7 Preconditioned Iterative Methods

The rate at which an iterative method converges depends greatly on the spectrum of the coefficient matrix. Hence, iterative methods usually involve a second matrix that transforms the coefficient matrix into one with a more favourable spectrum. The transformation matrix is called a preconditioner. A good preconditioner improves the convergence of the iterative method sufficiently to overcome the extra cost of constructing and applying the preconditioner. Indeed, without a preconditioner the iterative method may even fail to converge. Thus, preconditioning aims at reducing the spectral radius of the corresponding iterative matrix so as to accelerate the convergence of the classical iterative methods.

In 1987, the preconditioner P introduced by Milaszewicz (1987) assumes the form $P = I + S$, where

$$S = (s_{ij}) = \begin{cases} -a_{i1}, & \text{for } i = 2,\cdots,n \\ 0, & \text{otherwise} \end{cases} \tag{2.22}$$

with the condition that the coefficient matrix A is an L −matrix with $a_{i,i+1}a_{i+1,i} > 0$ and $0 < a_{1i}a_{i1} < 1$ for $i = 2,3,\cdots,n$. Gunawardena *et al*. (1991) proposed the preconditioned Gauss-Seidel method with $P = I + S$, where

$$S = (s_{ij}) = \begin{cases} -a_{ii+1}, & \text{for } i = 1,2,\cdots,n-1, \quad j = i+1 \\ 0, & \text{otherwise} \end{cases} \tag{2.23}$$

Similar preconditioners were proposed by Kotakemori *et al.* (1996), Kohno *et al.* (1997), Kotakemori *et al.* (2002), Morimoto *et al.* (2003) and Byrne (2008). The preconditioned effect of these preconditioners is seldom observed on the last row of A, because they are formed from a part of upper triangular part of A. The preconditioner of Morimoto *et al.* (2003) was an attempt at providing the preconditioned effect on the last row of A. It takes the form $P_{R_1} = I + R$, where R is defined as

$$R = (r_{nj}) = \begin{cases} -a_{nj}, & 1 \le j \le n-1 \\ 0, & \end{cases} \tag{2.24}$$

The preconditioned matrix PA, denoted by A_{R_1}, is defined by

$$A_{R_1} = (I + R)A = (a_{ij}^{R_1}), \qquad a_{ij}^{R_1} = \begin{cases} a_{ij}, & 1 \le i < n-1, 1 \le j \le n, \\ a_{nj} - \sum_{k=1}^{n-1} a_{nk}a_{kj}, & 1 \le j \le n. \end{cases} \tag{2.25}$$

Then, a splitting of the preconditioned matrix $A_{R_1}$is obtained thus

$A_{R_1} = M_{R_1} - N_{R_1} = (I - L + R - RL - RU) - U = (I - L - D_R + R - RL - E_R) - U$, where D_R, E_R are the diagonal and strictly lower triangular parts of RU, respectively. if $\sum_{k=1}^{n-1} a_{nk}a_{ki} \neq 1$, then $M_{R_1}^{-1}$ exists, and the Gauss-Seidel iterative matrix T_{R_1} is defined by $T_{R_1} = (I - D_R - L + R - RL - E_R)^{-1}U$. Niki *et al.* (2004) built on Morimoto *et al.* (2003) to propose the preconditioner $P_R = I + S + R$, arising from which the preconditioned matrix A_R assumes the structure

$$A_R = (I + S + R)A = \left(a_{ij}^R\right), \qquad a_{ij}^R = \begin{cases} a_{ij} - a_{ii+1}a_{i+1j}, & 1 \le i < n, \\ a_{nj} - \sum_{k=1}^{n-1} a_{nk}a_{kj}\,, & 1 \le j \le n. \end{cases} \quad (2.26)$$

with the corresponding splitting

$$A_R = M_R - N_R = (I - D - D_R) - (L - R + RL + E + E_R) - (U - S + SU). \quad (2.27)$$

In a quest to address the shortcomings of the preconditioner (3), Dehghan and Hajarian (2009) introduced two new preconditioners $\bar{P} = I + \bar{S}$ and $\tilde{P} = I + \tilde{S}$, with

$$\bar{S} = \begin{cases} -(a_{i1} + \gamma_i), & \text{for } i = 2, \cdots, n \\ 0, & \text{otherwise } 0 \end{cases} \quad (2.28)$$

$$\tilde{S} = \begin{cases} -(a_{in} + \delta_i), & \text{for } i = 1, \cdots, n-1 \\ 0, & \text{otherwise } 0 \end{cases} \quad (2.29)$$

where $\gamma_2, \gamma_3, \cdots, \gamma_n$ and $\delta_1, \delta_2, \cdots, \delta_{n-1}$ are real parameters. These preconditioners were applied to accelerate the convergence of the successive overrelaxation (SOR) iterative method under mild conditions on the coefficient matrix A. In furtherance of the search for fast converging iterative methods, Ndanusa and Adeboye (2012) attempted an improvement on the SOR method by proposing a preconditioner $P = I + S$, with S having the structure

$$S = \begin{cases} -a_{i1}, & i = 2, \cdots, n \\ -a_{i,i+1}, & i = 1, \cdots, n-1 \\ 0, & \text{otherwise} \end{cases} \quad (2.30)$$

For Mayaki and Ndanusa (2019), the S of the preconditioner $P = I + S$ takes the form

$$S = \begin{cases} -a_{ij}, & j = i + 1, \text{ for } i < 2 \text{ and } j = i - 1 \text{ for } i > j \\ 0, & \text{otherwise} \end{cases} \quad (2.31)$$

Faruk and Ndanusa (2019) proposed the preconditioner $P = I + S$ where,

$$S = \begin{cases} -a_{ij}\,, & (i,j) = (1,2),(2,1),(n-1,n),(n,n-1) \\ 0\,, & \text{otherwise} \end{cases} \tag{2.32}$$

Similarly, the preconditioner of Abdullahi and Ndanusa (2020) is defined by $P = I + \hat{S}$ where,

$$\hat{S} = \bar{S} + S' = \begin{cases} -a_{1n} & \forall\, n > 0 \\ -a_{i1}\,, & i = 2,\cdots,n \\ -a_{i,i+1}\,, & i = 1,\cdots,n-1 \\ 0\,, & \text{otherwise} \end{cases}, \tag{2.33}$$

$$\bar{S} = \left(s_{ij}\right) = \begin{cases} -a_{i1}, & i = 2,\cdots,n \\ -a_{i,i+1}, & i = 1,\cdots,n-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.34}$$

and

$$S' = \left(s_{ij}\right) = \begin{cases} -a_{1n} & \forall\, n > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

Other preconditioned iterative techniques include those of Ndanusa (2020) and Ndanusa *et al.* (2020).

2.8 Refinement of Iterative Methods

The rate of convergence of an iterative technique depends on the spectral radius of the matrix associated with the method. One way to select a procedure to accelerate convergence is to choose a method whose associated matrix has minimal spectral radius. Thus, there is the need to introduce a new means of measuring the amount by which an approximation to the solution to a linear system differs from the true solution to the system (Faruk and Ndanusa, 2019).

Suppose $\bar{x} \in R^n$ is an approximation to the solution of the linear system defined by $Ax = b$. The residual vector for $\bar{x}$ with respect to this system is $r = b - A\bar{x}$. In procedures such as the Jacobi, Gauss-Seidel or SOR methods, a residual vector is associated with each calculation of an approximate component to the solution vector. The true objective is to generate a sequence of approximations that will cause the residual vectors to converge rapidly to zero. Consider the linear system

$$Ay = r \tag{2.36}$$

The approximate solution $\bar{y}$ of the above system satisfies

$$\bar{y} \approx A^{-1}r = A^{-1}(b - A\bar{x}) = A^{-1}b - A^{-1}A\bar{x} = x - \bar{x} \tag{2.37}$$

and

$$x \approx \bar{x} + \bar{y} \tag{2.38}$$

So $\bar{y}$ is an estimate of the error produced when $\bar{x}$ approximates the solution x to the original system. In general, $\bar{x} + \bar{y}$ is a more accurate approximation to the solution of the linear system $Ax = b$ than the original approximation $\bar{x}$. The method using this assumption is called *iterative refinement*, or *iterative improvement*, and consists of performing iterations on the system whose right-hand side is the residual vector for successive approximations until satisfactory accuracy results.

Refinement of iterative methods entail performing iterations on the linear system whose right-hand side is the residual vector for successive approximations until satisfactory accuracy results. Refinement of AOR method, introduced by Vatti *et al.* (2018) is described by the relation

$$x^{(n+1)} = {\mathcal{L}_{r,\omega}}^2 x^{(n)} + d \tag{2.39}$$

where ${\mathcal{L}_{r,\omega}}^2 = [(I - rE)^{-1}\{(1-\omega)I + (\omega - r)E + \omega F\}]^2$, $d = \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c$. This research discusses a refinement of refined accelerated overrelaxation method for solving the linear system (1.1), which is named second refinement of accelerated overrelaxation method. some pioneering studies in this field include the works of Kebede (2017), who proposed a new method for solving the linear system $Ax = b$ that often arise in engineering and scientific applications; this method, which is known as second-degree refinement of Jacobi iterative method, is based on the second-degree Jacobi stationary iterative method. The relationships between the spectral radius of second-degree refinement of Jacobi method and spectral radii of first-degree Jacobi, first-degree refinement of Jacobi and second-degree Jacobi methods were established. Numerical results demonstrated that for a coefficient matrix that is strictly diagonally dominant and positive definite, the second-degree refinement of Jacobi iterative method proved to be very effective and efficient as it converges faster than the existing first-degree Jacobi, first-degree refinement of Jacobi and second-degree Jacobi methods.

Eneyew *et al.* (2019) focused on a second refinement of Jacobi (SRJ) method for the solution of system of linear equations obtained from ordinary differential equation and partial differential equation problems, where the coefficient matrix is strictly diagonally dominant or symmetric positive definite or $M-\text{matrix}$. In such cases, there occurs a significant reduction in spectral radius of iteration matrix of the proposed method, with attendant reduction in number of iterations, which translates to increased convergence. Some numerical examples were presented to validate the theoretical analysis which further established the superiority of the second refinement of Jacobi method over Jacobi and refinement of Jacobi methods. Eneyew *et al.* (2020a) modified the Gauss-Seidel method to obtain a second-refinement of Gauss-Seidel method for solution of system of linear equations, in order to enhance convergence rate, minimize the spectral radius, and by implication, reduce the number of iterations needed for convergence. This method is equally applicable to solution of differential equations that are transformed into linear systems by application of finite differences. Such systems are characterized by coefficient matrices that are strictly diagonally dominant, symmetric positive definite, or M-matrices. Theoretical analysis established that the method converges for these types of matrices. Results of numerical experiments further demonstrated the efficiency of second-refinement of Gauss-Seidel method over the Gauss-Seidel and refinement of Gauss-Seidel methods. In Assefa and Teklehaymanot (2021), a second refinement of accelerated over relaxation method was introduced; which is just a refinement of first-degree refinement of accelerated over relaxation method, whereby the spectral radius of iteration matrix of the method was observed to be significantly reduced in comparison to the spectral radii of accelerated over relaxation (AOR) method and first-degree refinement of accelerated over relaxation methods. In addition, the optimal value of each parameter involved in the method was derived. Derivation of the third-degree, fourth-degree and in general the $kth-$degree refinement of accelerated methods were also obtained. The spectral radius of the iteration matrix and convergence criteria of the second refinement of accelerated over relaxation (SRAOR) are discussed. Finally, a numerical experiment was undertaken to demonstrate the efficiency of the proposed method over other existing methods. Eneyew *et al.* (2020b) proposed a second refinement of generalized Jacobi method for solution of linear systems. This method proved to be the fastest method to converge to the exact solution when compared to Jacobi, refinement of Jacobi, generalized Jacobi and refinement of generalized Jacobi methods for strictly diagonally dominant, symmetric positive definite and M-matrices.

2.9 Rate of Convergence

It is not just sufficient to know that an iterative method converges. Of equal importance is the desirability of knowing how fast it converges. Thus Young (1954) introduced the number

$$R(G) = -\log \rho(G)$$

as the rate of convergence of the linear iteration $x^{(n+1)} = Gx^{(n)} + k,$ where $\rho(G)$ is the spectral radius for that iterative method.

CHAPTER THREE

3.0 MATERIALS AND METHODS

3.1 Derivation of the Preconditioned AOR iterative method

Consider a linear system of the form

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

where $A = (a_{ii})$ is an irreducibly diagonally dominant $L -$ matrix of order n, b is a given $n -$ dimensional vector and x is an $n -$ dimensional vector to be determined. Consider the usual splitting of A as,

$$A = D_A - E_A - F_A \tag{3.1b}$$

where $D_A, -E_A$ and $- F_A$ are the diagonal, strictly lower and strictly upper triangular parts of A respectively. Transformation of diagonal entries of (3.1) is achieved by expressing (3.1b) in the form

$$D_A{}^{-1}Ax = D_A{}^{-1}b$$

$$D_A{}^{-1}(D_A - E_A - F_A)x = D_A{}^{-1}b$$

$$\left(I - D_A{}^{-1}E_A - D_A{}^{-1}F_A\right)x = D_A{}^{-1}b$$

Thus, we have obtained the equivalent system

$$Bx = f \tag{3.2}$$

with the corresponding splitting

$$(I - E_B - F_B)x = f$$

Where I is the identity matrix of order n, $-E_B$ and $-F_B$ being the strictly lower and strictly upper triangular parts of B respectively.

A transformation matrix $P = (I + S)$ is then applied to system (3.2) as

$$PBx = Pf$$

which results in the preconditioned system

$$Tx = k \tag{3.3}$$

where

where $T = (I + S)B$ and $k = (I + S)f$ with

$$S = \begin{bmatrix} 0 & -a_{12} & 0 & \cdots & 0 & 0 \\ -a_{21} & 0 & -a_{23} & \cdots & 0 & 0 \\ -a_{31} & -a_{32} & 0 & \cdots & -a_{3,n-1} & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ -a_{n-1,1} & 0 & 0 & \cdots & 0 & -a_{n-1,n} \\ -a_{n,1} & 0 & 0 & \cdots & -a_{n,n-1} & 0 \end{bmatrix}$$

A usual splitting of the preconditioned coefficient matrix T of (3.3) into its diagonal (D), strictly lower ($-L$) and strictly upper ($-U$) components is obtained thus

$$T = D_T - E_T - F_T \tag{3.4}$$

with the following resultant representations:

$$D_T = \begin{bmatrix} 1 - a_{12}a_{21} & 0 & \cdots & 0 & 0 \\ 0 & 1 - a_{21}a_{12} - a_{23}a_{32} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 - a_{n-1,1}a_{1,n-1} - a_{n-2,n-1}a_{n-1,n-2} - a_{n-1,n}a_{n,n-1} & 0 \\ 0 & 0 & \cdots & 0 & 1 - a_{n1}a_{1n} - a_{n-1,n}a_{n,n-1} \end{bmatrix}$$

$$-E_T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ -a_{23}a_{31} & 0 & \cdots & 0 & 0 \\ -a_{21}a_{32} - a_{34}a_{41} & -a_{12}a_{31} - a_{34}a_{42} & \cdots & 0 & 0 \\ \vdots & \vdots & 0 & \vdots & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots \\ \vdots & \vdots & \cdots & 0 & 0 \\ -a_{n,n-1}a_{n-1,1} & -a_{12}a_{n1} - a_{n-1,2}a_{n,n-1} + a_{n2} & \cdots & -a_{n1}a_{1,n-1} & 0 \end{bmatrix}$$

$$\text{and} -F_T = \begin{bmatrix} 0 & 0 & -a_{12}a_{23} & \cdots & -a_{12}a_{2,n-1} + a_{1,n-1} - a_{1n} & -a_{12}a_{2n} + -a_{1n} \\ 0 & 0 & -a_{12}a_{13} & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & 0 & \vdots & \vdots \\ \vdots & \vdots & & 0 & -a_{1,n-1}a_{n-2,1} - a_{n-2,3} & -a_{1n}a_{n-2,1} - a_{3,n}a_{n-2,3} - a_{n-2,n}a_{n-1,n} + a_{n-2,n} \\ \vdots & \vdots & \vdots & \cdots & 0 & -a_{1n}a_{n-1,1} - a_{n-2,n}a_{n-1,n-2} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

From (3.4) it is observed that the effect of the preconditioner P on B is elimination of just one entry, specifically a_{12}, and scaling of the remaining entries. It is further observed that:

1. $D_T = I + D_*$,
2. $E_T = E + E_S + E_*$,
3. $F_T = F + F_S + F_*$,
4. $S = -E_S - F_S$, and
5. $-SE - SF = D_* - E_* - F_*$;

where D_*, $-E_*$ and $-F_*$ are the diagonal, strictly lower and strictly upper parts of $-SE - SF$ respectively; and $-E_S$ and $-F_S$ are the strictly lower and strictly upper parts of S respectively. Application of the AOR method to the preconditioned linear system (3.3) results in the corresponding preconditioned AOR method whose iterative matrix is defined by

$$\mathcal{L}_{r,\omega_T} = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T] \tag{3.5}$$

3.2 Convergence of Preconditioned AOR Method

Some lemmas that will be used in the succeeding sections are briefly explained.

Lemma 3.1 (Varga (1962)). Let $A \geq 0$ be an irreducible $n \times n$ matrix. Then,

i. A has a positive real eigenvalue equal to its spectral radius.
ii. To $\rho(A)$ there corresponds an eigenvector $x > 0$.
iii. $\rho(A)$ increases when any entry of A increases.
iv. $\rho(A)$ is a simple eigenvalue of A.

Lemma 3.2 (Gunawardena *et al.* (1991)). Let A be a nonnegative matrix. Then

i. If $\alpha x \leq Ax$ for some nonnegative vector $x, x \neq 0$, then $\alpha \leq \rho(A)$.
ii. If $Ax \leq \beta x$ for some positive vector x, then $\rho(A) \leq \beta$. Moreover, if A is irreducible and if $0 \neq \alpha x \leq Ax \leq \beta x$ for some nonnegative vector x, then $\alpha \leq \rho(A) \leq \beta$ and x is a positive vector.

Lemma 3.3 (Varga (1962)). Suppose $A = M - N$ is an $M-$ splitting of A. Then the splitting is convergent iff A is a nonsingular $M-$ matrix.

Lemma 3.4 **(**Varga (1962)). Let $A = M_1 - N_1 = M_2 - N_2$ be two regular splittings of A, where $A^{-1} \geq O$. If $N_2 \geq N_1 \geq O$, then

$$1 > \rho\left({M_2}^{-1}N_2\right) \geq \rho\left({M_1}^{-1}N_1\right) \geq O.$$

If moreover, $A^{-1} > O$ and if $N_2 \geq N_1 \geq O$, equality excluded (meaning that neither N_1 nor $N_2 - N_1$ is the null matrix), then

$$1 > \rho\left({M_2}^{-1}N_2\right) > \rho\left({M_1}^{-1}N_1\right) > O.$$

Theorem 3.1 Let $\mathcal{L}_{r,\omega} = (I - rE_B)^{-1}[(1-\omega)I + (\omega - r)E_B + \omega F_B]$ and $\mathcal{L}_{r,\omega_T} = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$ be the AOR and preconditioned AOR iterative matrices corresponding to systems (3.2) and (3.3) respectively. Suppose A is an irreducibly diagonally dominant $L-$matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n-1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then $\mathcal{L}_{r,\omega}$ and $\mathcal{L}_{r,\omega_T}$ are nonnegative and irreducible matrices.

PROOF: Since B is an $L-$matrix, $E_B \geq 0$ and $F_B \geq 0$. Thus $(I - rE_B)^{-1} = I + rE_B + r^2{E_B}^2 + \cdots + r^{n-1}{E_B}^{n-1} \geq 0$. And, from definition, we have

$$\mathcal{L}_{r,\omega} = (I - rE_B)^{-1}[(1-\omega)I + (\omega - r)E_B + \omega F_B] \tag{3.6}$$

$$= [I + rE_B + r^2{E_B}^2 + \cdots + r^{n-1}{E_B}^{n-1}][(1-\omega)I + (\omega - r)E_B + \omega F_B]$$

$$= (1-\omega)I + (\omega - r)E_B + \omega F_B + rE_B(1-\omega)I + rE_B[(\omega - r)E_B + \omega F_B]$$

$$+(r^2{E_B}^2 + \cdots + r^{n-1}{E_B}^{n-1})[(1-\omega)I + (\omega - r)E_B + \omega F_B]$$

$$= (1-\omega)I + [(\omega - r)E_B + rE_B(1-\omega)I] + \omega F_B + rE_B[(\omega - r)E_B + \omega F_B]$$

$$+(r^2{E_B}^2 + \cdots + r^{n-1}{E_B}^{n-1})[(1-\omega)I + (\omega - r)E_B + \omega F_B]$$

$$= (1-\omega)I + \omega(1-r)E_B + \omega F_B + T$$

where

$$T = rE_B[(\omega - r)E_B + \omega F_B]$$

$$+(r^2{E_B}^2 + \cdots + r^{n-1}{E_B}^{n-1}) \times [(1-\omega)I + (\omega - r)E_B + \omega F_B] \geq 0.$$

It is clear that $(1-\omega)I + \omega(1-r)E_B + \omega F_B \geq 0$. Consequently, $\mathcal{L}_{r,\omega} = (1-\omega)I + \omega(1-r)E_B + \omega F_B + T \geq 0$. Hence, $\mathcal{L}_{r,\omega}$ is a nonnegative matrix. Since $B = I - E_B - F_B$ is irreducible, so also is $(1-\omega)I + \omega(1-r)E_B + \omega F_B$ since the coefficients of $I, E_B,$ and F_B are different from zero and less than 1 in absolute value. Hence, $\mathcal{L}_{r,\omega}$ is an irreducible matrix.

Now, consider the preconditioned AOR iterative matrix

$$\mathcal{L}_{r,\omega_T} = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$$

Equation (3.3) ensures that the $L-$matrix structure of A is preserved in T. Since T is an $L-$matrix, it is evident that $E_T \geq 0$ and $F_T \geq 0$. Also, by the conditions of Theorem 3.1, it is easy to get that $D_T \geq 0$. Thus,

$$\mathcal{L}_{r,\omega_T} = \left[D_T(I - r{D_T}^{-1}E_T)\right]^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$$

$$= (I - r{D_T}^{-1}E_T)^{-1}{D_T}^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$$

$$= (I - r{D_T}^{-1}E_T)^{-1}\left[(1-\omega)I + (\omega - r){D_T}^{-1}E_T + \omega {D_T}^{-1}F_T\right]$$

$$= \left[I + r{D_T}^{-1}E_T + r^2\left({D_T}^{-1}E_T\right)^2 + \cdots + r^{n-1}\left({D_T}^{-1}E_T\right)^{n-1}\right]$$

$$\times \left[(1-\omega)I + (\omega - r){D_T}^{-1}E_T + \omega {D_T}^{-1}F_T\right]$$

$$= (1-\omega)I + \omega(1-r){D_T}^{-1}E_T + \omega {D_T}^{-1}F_T + G$$

where

$$G = r{D_T}^{-1}E_T\left[(\omega - r){D_T}^{-1}E_T + \omega {D_T}^{-1}F_T\right]$$

$$+ \left[r^2\left({D_T}^{-1}E_T\right)^2 + \cdots + r^{n-1}\left({D_T}^{-1}E_T\right)^{n-1}\right]$$

$$\times [(1-\omega)I + (\omega - r){D_T}^{-1}E_T + \omega {D_T}^{-1}F_T] \geq 0$$

Using similar arguments, it is conclusive that $\mathcal{L}_{r,\omega_T}$ is also nonnegative and irreducible.

Theorem 3.2 Let $\mathcal{L}_{r,\omega} = (I - rE_B)^{-1}[(1-\omega)I + (\omega - r)E_B + \omega F_B]$ and $\mathcal{L}_{r,\omega_T} = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$ be the AOR and preconditioned AOR iterative matrices corresponding to systems (3.2) and (3.3) respectively. Suppose A is an irreducibly diagonally dominant $L-$matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n-1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then

(i) $\rho(\mathcal{L}_{r,\omega_T}) < \rho(\mathcal{L}_{r,\omega})$, if $\rho(\mathcal{L}_{r,\omega}) < 1$;

(ii) $\rho(\mathcal{L}_{r,\omega_T}) = \rho(\mathcal{L}_{r,\omega})$, if $\rho(\mathcal{L}_{r,\omega}) = 1$;

(iii) $\rho(\mathcal{L}_{r,\omega_T}) > \rho(\mathcal{L}_{r,\omega})$, if $\rho(\mathcal{L}_{r,\omega}) > 1$.

PROOF: It is established, from Theorem 3.1, that $\mathcal{L}_{r,\omega}$ and $\mathcal{L}_{r,\omega_T}$ are nonnegative and irreducible matrices. Therefore, suppose $\eta = \rho(\mathcal{L}_{r,\omega})$, then by Lemma 3.1 there exists a positive vector y, such that

$$\mathcal{L}_{r,\omega}y = \eta y$$

Equivalently,

$$(I - rE_B)^{-1}[(1-\omega)I + (\omega - r)E_B + \omega F_B]y = \eta y$$

$$[(1-\omega)I + (\omega - r)E_B + \omega F_B] = \eta(I - rE_B)$$

$$\omega F_B = (\eta + \omega - 1)I + (r - \omega - \eta r)E_B \tag{3.7}$$

Therefore, for this $y > 0$,

$$\mathcal{L}_{r,\omega_T}y - \eta y = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]y - \eta y$$

$$= (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]y - \eta(D_T - rE_T)^{-1}(D_T - rE_T)y$$

$$= (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T - \eta(D_T - rE_T)]y$$

From the identity,

$$\eta(D_T - rE_T) = \eta(1-r)D_T + \eta r(D_T - E_T),$$

it implies

$$\mathcal{L}_{r,\omega_T}y - \eta y = (D_T - rE_T)^{-1}[(1-\omega)D_T + (\omega - r)E_T + \omega F_T - \eta(1-r)D_T - \eta r(D_T - E_T)]y$$

$$= (D_T - rE_T)^{-1}[D_T - \omega D_T + \omega E_T - rE_T + \omega F_T - \eta D_T + \eta r D_T - \eta r D_T + \eta r E_T]y$$

$$= (D_T - rE_T)^{-1}[D_T - rD_T - \eta D_T + \eta r D_T - \omega D_T + rD_T - \eta r D_T + \omega E_T - rE_T + \eta r E_T + \omega F_T]y$$

$$= (D_T - rE_T)^{-1}[(1-\eta)(1-r)D_T - (\omega - r + \eta r)(D_T - E_T) + \omega F_T]y$$

$$= (D_T - rE_T)^{-1}[(1-\eta)(1-r)(I + D_*) - (\omega - r + \eta r)(I + D_*) + (\omega - r + \eta r)(E_B + E_S + E_*) + \omega(F_B + F_S + F_*)]y$$

$$= (D_T - rE_T)^{-1}[(1 - \omega - \eta)(I + D_*) + (\omega - r + \eta r)(E_B + E_S + E_*) + \omega(F_B + F_S + F_*)]y$$

$$= (D_T - rE_T)^{-1}[(1 - \omega - \eta)I + \omega F_B - (r - \omega - \eta r)E_B + (1 - \omega - \eta)D_* + (\omega - r + \eta r)(E_S + E_*) + \omega(F_F + F_*)]y$$

From (3.7),

$$= (D_T - rE_T)^{-1}[(1-\eta)D_* - \omega(D_* - L_* - U_*) + \omega(E_S + F_S) - r(1-\eta)(E_S + E_*)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_*) + (\eta - 1)(rE_S + rL_*) - \omega(-SE_B - SF_B) + \omega(-S)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + \omega S E_B + \omega S F_B - \omega S]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + (1-\omega)S + \omega S F_B - S(I - \omega E_B)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + (1-\omega)S + \omega S E_B - rSE_B + \omega S F_B - SI + rSE_B)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + (1-\omega)S + (\omega - r)SE_B + \omega S F_B - S(I - rE_B)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + S\{(1-\omega) + (\omega - r)E_B + \omega F_B\} - S(I - rE_B)]y$$

And from (3.7),

$$(1-\omega)I + (\omega - r)E_B + \omega F_B = \eta(I - r)E_B \tag{3.8}$$

$$\mathcal{L}_{r,\omega_T} y - \eta y = (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + \eta S(I - rE_B) - S(I - rE_B)]y$$

$$= (D_T - rE_T)^{-1}[(\eta - 1)(-D_* + rE_S + rE_*) + (\eta - 1)S(I - rE_B)]y$$

By employing (3.8),

$$= (\eta - 1)(D_T - rE_T)^{-1}[-D_* + rE_S + rE_* + [(1 - \omega)S + (\omega - r)SE_B + \omega SF_B]/\eta]y$$

$$= [(\eta - 1)/\eta](D_T - rE_T)^{-1}[-\eta D_* + r\eta E_S + r\eta E_* + (1 - \omega)S + (\omega - r)SE_B + \omega SF_B]y$$

It is obvious that $-\eta D_* + r\eta E_S + r\eta L_* \geq 0$, provided $a_{q,q+1}a_{q+1,1} + a_{q1} \geq 0$ $(q = 2, \cdots, n-1)$ and $\eta r a_{n1} - (1 - \omega)a_{n1} \geq 0$, $(1 - \omega)S \geq 0$, $(\omega - r)SE_B \geq 0$ and $\omega SF_B \geq 0$. Suppose $D_T - rE_T$ is a splitting of some matrix X. From observation, D_T is an $M-$matrix and $rE_T \geq 0$. Consequently, $D_T - rE_T$ is an $M-$splitting of X. Also, $r{D_T}^{-1}E_T$, being a strictly lower triangular matrix, has its eigenvalues lying on its main diagonal, and they are all zeros. Therefore, $\rho(r{D_T}^{-1}E_T) = 0 < 1$. And by Lemma 3, X is a nonsingular $M-$matrix. consequently, $X^{-1} = (D_T - rE_T)^{-1} \geq 0$. We are now ready to deduce $(i) - (iii)$, by employing Lemma 2 thus.

(1) If $\eta < 1$, then $\mathcal{L}_{r,\omega_T}y - \eta y \leq 0$ but not equal to 0. Therefore, $\mathcal{L}_{r,\omega_T}y \leq \eta y$. By Lemma 2, we obtain $\rho(\mathcal{L}_{r,\omega_T}) < \eta = \rho(\mathcal{L}_{r,\omega})$.

(2) If $\eta = 1$, then $\mathcal{L}_{r,\omega_T}y - \eta y = 0$. Therefore, $\mathcal{L}_{r,\omega_T}y = \eta y$. By Lemma 2, we obtain $\rho(\mathcal{L}_{r,\omega_T}) = \eta = \rho(\mathcal{L}_{r,\omega})$.

If $\eta > 1$, then $\mathcal{L}_{r,\omega_T}y - \eta y \geq 0$ but not equal to 0. Therefore, $\mathcal{L}_{r,\omega_T}y \geq \eta y$. By Lemma 2, we obtain $\rho(\mathcal{L}_{r,\omega_T}) > \eta = \rho(\mathcal{L}_{r,\omega})$.

Theorem 3.3 Let $0 < r_1 < r_2 \leq \omega \leq 1$ and $A^{-1} \geq 0$. Under the hypothesis of Theorem 3.2, then $1 > \rho(\mathcal{L}_{r_1,\omega_T}) > \rho(\mathcal{L}_{r_2,\omega_T}) > 0$, if $0 < \eta < 1$.

PROOF: Let

$$T = M_{r,\omega} - N_{r,\omega}$$

where $M_{r,\omega} = (1/\omega)(D_T - rE_T)$ and $N_{r,\omega} = (1/\omega)[(1-\omega)D_T + (\omega - r)E_T + \omega F_T]$. Suppose also that $T = M_{r_1,\omega} - N_{r_1,\omega}$ and $T = M_{r_2,\omega} - N_{r_2,\omega}$ are two regular splittings of T, where $M_{r_1,\omega} = (1/\omega)(D_T - r_1 E_T)$, $N_{r_1,\omega} = (1/\omega)[(1-\omega)D_T + (\omega - r_1)E_T + \omega F_T]$, $M_{r_2,\omega} = (1/\omega)(D_T - r_2 E_T)$ and $N_{r_2,\omega} = (1/\omega)[(1-\omega)D_T + (\omega - r_2)E_T + \omega F_T]$. Since $0 < r_1 < r_2 \leq \omega \leq 1$, then $N_{r_1,\omega} \geq N_{r_2,\omega} \geq 0$, equality excluded, then in the light of Lemma 4, we have that

$$1 > \rho\left(\mathcal{L}_{r_1,\omega_T}\right) > \rho\left(\mathcal{L}_{r_2,\omega_T}\right) > 0$$

Corollary 3.1 Let $\mathcal{L}_\omega = (I - \omega L)^{-1}[(1-\omega)I + \omega U]$ and $\mathcal{L}_{\omega_T} = (D_T - \omega E_T)^{-1}[(1-\omega)D_T + \omega F_T]$ be the SOR and preconditioned SOR iterative matrices respectively. Suppose A is an irreducibly diagonally dominant $L-$ matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n-1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then

(i) $\rho(\mathcal{L}_{\omega_T}) < \rho(\mathcal{L}_\omega)$, if $\rho(\mathcal{L}_\omega) < 1$;

(ii) $\rho(\mathcal{L}_{\omega_T}) = \rho(\mathcal{L}_\omega)$, if $\rho(\mathcal{L}_\omega) = 1$;

(iii) $\rho(\mathcal{L}_{\omega_T}) > \rho(\mathcal{L}_\omega)$, if $\rho(\mathcal{L}_\omega) > 1$.

Corollary 3.2 Let $0 < \omega_1 < \omega_2 \leq 1$ and $A^{-1} \geq 0$. Under the hypothesis of Corollary 3.1, then $1 > \rho\left(\mathcal{L}_{\omega_{1T}}\right) > \rho\left(\mathcal{L}_{\omega_{2T}}\right) > 0$, if $0 < \eta < 1$.

3.3 Formulation of Second Refinement of Preconditioned AOR (SRPAOR) Method

The matrix T of (3.3) has the splitting $T = D_T - E_T - F_T$. This is further transformed into

$$D_T^{-1}(D_T - E_T - F_T)x = D_T^{-1}k$$

That is,

$$Hx = c \tag{3.9}$$

which has the splitting

$$H = I - E - F$$

where I is the identity matrix of order n, $-E$ and $-F$ being the strictly lower and strictly upper triangular parts of H respectively.

Following Assefa and Teklehaymanot (2021), a reformulation of second refinement of preconditioned AOR method is derived from (3.9) thus,

$$(I - E - F)x = c$$

$$Ix - rEx + \omega(I - E - F)x = Ix - rEx + \omega c$$

$$(I - rE)x = (I - rE)x - \omega(I - E - F)x + \omega c$$

$$(I - rE)x = (I - rE)x + \omega(c - Hx)$$

$$x = x + \omega(I - rE)^{-1}(c - Hx)$$

Consequently, the second refinement of AOR is defined as

$$x^{(n+1)} = x^{(n+1)} + \omega(I - rE)^{-1}\left(c - Hx^{(n+1)}\right) \quad (3.10)$$

where $x^{(n+1)}$ that appeared on the right-hand side is the $(\mathrm{n}+1)th$ approximation of refinement of AOR of Vatti *et al.* (2018) defined by

$$x^{(n+1)} = {\mathcal{L}_{r,\omega}}^2 x^{(n)} + d \quad (3.11)$$

where ${\mathcal{L}_{r,\omega}}^2 = [(I - rE)^{-1}\{(1-\omega)I + (\omega - r)E + \omega F\}]^2$, $d = \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c$.

Substituting (3.11) in (3.10),

$$x^{(n+1)} = {\mathcal{L}_{r,\omega}}^2 x^{(n)} + \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c + \omega(I - rE)^{-1}(c - (I - E - F)\{{\mathcal{L}_{r,\omega}}^2 x^{(n)} + \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c\})$$

$$x^{(n+1)} = {\mathcal{L}_{r,\omega}}^2 x^{(n)} + \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c + \omega(I - rE)^{-1}c - \omega(I - rE)^{-1}(I - E - F)\{{\mathcal{L}_{r,\omega}}^2 x^{(n)} + \omega\left[I + \mathcal{L}_{r,\omega}\right](I - rE)^{-1}c\}$$

$$x^{(n+1)} = {\mathcal{L}_{r,\omega}}^2 x^{(n)} - (I - rE)^{-1}(\omega I - \omega E - \omega F){\mathcal{L}_{r,\omega}}^2 x^{(n)} + \omega(I + \mathcal{L}_{r,\omega})(I - rE)^{-1}c + \omega(I - rE)^{-1}c - (I - rE)^{-1}(\omega I - \omega E - \omega F)\omega(I + \mathcal{L}_{r,\omega})(I - rE)^{-1}c$$

$$x^{(n+1)} = \mathcal{L}_{r,\omega}{}^2[I - (I - rE)^{-1}(\omega I - \omega E - \omega F)]x^{(n)} + \omega[I + \mathcal{L}_{r,\omega} + I - (I - rE)^{-1}(\omega I - \omega E - \omega F)(I + \mathcal{L}_{r,\omega})](I - rE)^{-1}c$$

$$x^{(n+1)} = \mathcal{L}_{r,\omega}{}^2[I - (I - rE)^{-1}(\omega I - \omega E - \omega F)]x^{(n)} + \omega[I + (I + \mathcal{L}_{r,\omega})(I - (I - rE)^{-1}(\omega I - \omega E - \omega F)(I + \mathcal{L}_{r,\omega}))](I - rE)^{-1}c$$

Note that,

$$I - (I - rE)^{-1}(\omega I - \omega E - \omega F)$$
$$= (I - rE)(I - rE)^{-1} - (I - rE)^{-1}\omega I + (I - rE)^{-1}\omega E + (I - rE)^{-1}\omega F$$
$$= (I - rE)^{-1}\{(1 - \omega)I + (\omega - r)E + \omega F\} = \mathcal{L}_{r,\omega}$$

$$x^{(n+1)} = \mathcal{L}_{r,\omega}{}^2\left[\mathcal{L}_{r,\omega}\right]x^{(n)} + \omega[I + (I + \mathcal{L}_{r,\omega})\mathcal{L}_{r,\omega}](I - rE)^{-1}c$$

$$x^{(n+1)} = \mathcal{L}_{r,\omega}{}^3 x^{(n)} + \omega\left[I + \mathcal{L}_{r,\omega} + \mathcal{L}_{r,\omega}{}^2\right](I - rE)^{-1}c \qquad (3.12)$$

Or more compactly,

$$x^{(n+1)} = \mathcal{L}_{r,\omega}{}^3 x^{(n)} + d \qquad (3.13)$$

Where,

$$\mathcal{L}_{r,\omega}{}^3 = [(I - rE)^{-1}\{(1 - \omega)I + (\omega - r)E + \omega F\}]^3$$

and

$$d = \omega\left[I + \mathcal{L}_{r,\omega} + \mathcal{L}_{r,\omega}{}^2\right](I - rE)^{-1}c.$$

3.4 Convergence of SRPAOR Method

Theorem 3.4 Let A be an irreducibly diagonally dominant $L-$matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n - 1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then SRPAOR method converges for any arbitrary choice of the initial approximation.

PROOF: Denote the exact solution of system (3.1) by x_E. Let $\bar{x}^{(n+1)}$ be the $(\mathrm{n}+1)th$ approximate solution of (3.1) by the SRPAOR method (3.10). Then,

$$\left\|\bar{x}^{(n+1)} - x_E\right\| = \left\|x^{(n+1)} + \omega(I - rE)^{-1}\left(c - Hx^{(n+1)}\right) - x_E\right\|$$

$$\leq \left\|x^{(n+1)} - x_E\right\|\left\|\left(c - Hx^{(n+1)}\right)\right\|\|\omega(I - rE)^{-1}\|$$

By Vatti *et al.* (2018), $x^{(n+1)}$, which is the Refinement of AOR method, is convergent. And by implication,

$$\left\|x^{(n+1)} - x_E\right\| \to 0 \text{ and } \left\|\left(c - Hx^{(n+1)}\right)\right\| \to 0.$$

Thus we have,

$$\left\|\bar{x}^{(n+1)} - x_E\right\| \to 0.$$

Hence, the Second Refinement of Preconditioned AOR (SRPAOR) converges to the solution of the linear system (3.1).

Theorem 3.5 Let A be an irreducibly diagonally dominant $L-$matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n - 1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then $\left\|{\mathcal{L}_{r,\omega}}^3\right\|_\infty = \left\|\mathcal{L}_{r,\omega}\right\|_\infty^3 < 1$.

PROOF: Consider $\left\|{\mathcal{L}_{r,\omega}}^3\right\|_\infty$; then we have,

$$\left\|{\mathcal{L}_{r,\omega}}^3\right\|_\infty = \|[(I - rE)^{-1}\{(1 - \omega)I + (\omega - r)E + \omega F\}]^3\|_\infty$$

$$= \|[(I - rE)^{-1}\{(1 - \omega)I + (\omega - r)E + \omega F\}]\|_\infty^3$$

$$= \left\|\mathcal{L}_{r,\omega}\right\|_\infty^3 < 1 \quad \text{by Theorem 3.4}$$

Theorem 3.6 Let A be an irreducibly diagonally dominant $L-$matrix with $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n - 1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$. Then $\left\|{\mathcal{L}_{r,\omega}}^3\right\|_\infty < \left\|\mathcal{L}_{r,\omega}\right\|_\infty$.

PROOF: By Theorem 3.5, we have,

$$\left\|\mathcal{L}_{r,\omega}{}^3\right\|_\infty = \left\|\mathcal{L}_{r,\omega}\right\|_\infty^3 < \left\|\mathcal{L}_{r,\omega}\right\|_\infty.$$

Theorem 3.7 The Second Refinement of Preconditioned AOR method converges faster than the Refinement of AOR method if Refinement of AOR method is convergent.

PROOF: Let $\bar{\bar{x}}$ be the solution of linear system (3.1) obtained by the Second Refinement of Preconditioned AOR method (3.13) and $\hat{x}$ be the solution of (3.1) obtained by the Preconditioned AOR method (3.5).

From (3.13), we get

$$\bar{\bar{x}} = \mathcal{L}_{r,\omega}{}^3\hat{x} + d$$

$$\bar{\bar{x}}^{(n+1)} - \bar{\bar{x}} = \mathcal{L}_{r,\omega}{}^3 x^{(n)} + d - \bar{\bar{x}}$$

$$= \mathcal{L}_{r,\omega}{}^3\left(x^{(n)} - \hat{x}\right) + d - \bar{\bar{x}} + \mathcal{L}_{r,\omega}{}^3\hat{x}$$

$$= \mathcal{L}_{r,\omega}{}^3\left(x^{(n)} - \hat{x}\right) - \bar{\bar{x}} + (\mathcal{L}_{r,\omega}{}^3\hat{x} + d)$$

$$= \mathcal{L}_{r,\omega}{}^3\left(x^{(n)} - \hat{x}\right) - \bar{\bar{x}} + \bar{\bar{x}}$$

$$= \mathcal{L}_{r,\omega}{}^3\left(x^{(n)} - \hat{x}\right)$$

Now,

$$\left\|\bar{\bar{x}}^{(n+1)} - \bar{\bar{x}}\right\|_\infty = \left\|\mathcal{L}_{r,\omega}{}^3\left(x^{(n)} - \hat{x}\right)\right\|_\infty$$

$$\leq \left\|\mathcal{L}_{r,\omega}{}^3\right\|_\infty \left\|\left(x^{(n)} - \hat{x}\right)\right\|_\infty$$

$$\leq \left\|\mathcal{L}_{r,\omega}\right\|_\infty^3 \left\|\left(x^{(n)} - \hat{x}\right)\right\|_\infty$$

Therefore, by Theorems 3.4 and 3.5 the Second Refinement of Preconditioned AOR method converges faster than the Preconditioned AOR method.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Numerical Experiments

In order to validate the results of the Theorems advanced in the preceding chapter, the following problems are considered. In all the problems considered we seek convergence for the linear system $Ax = b$ of (3.1).

4.1.1 Problem 1

Let the coefficient matrix of the linear system (3.1) be given by

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

and

$$b = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 3 \\ 1 \\ 2 \end{pmatrix}$$

considered by Youssef and Taha (2013).

4.1.2 Problem 2

Consider the case which gives the following coefficient matrix and constant vector for the linear system (3.1).

$$A = \begin{pmatrix} 35 & -2 & -3 & -1 & 0 & -2 & -3 & -1 \\ -5 & 27 & -3 & -4 & -4 & -1 & -2 & 0 \\ -7 & -4 & 71 & -9 & -2 & -6 & 0 & -3 \\ -1 & -1 & -2 & 20 & -4 & -3 & -2 & -4 \\ -2 & -2 & -3 & 0 & 71 & -2 & -1 & -1 \\ 0 & -2 & -5 & -4 & -3 & 53 & -5 & -4 \\ -3 & -2 & -1 & -3 & -5 & -4 & 32 & -3 \\ -4 & -3 & -2 & -5 & -1 & -2 & -7 & 31 \end{pmatrix}$$

$$b = \begin{pmatrix} 23 \\ 8 \\ 40 \\ 3 \\ 60 \\ 30 \\ 11 \\ 7 \end{pmatrix}$$

This problem can be found in Vatti *et al.* (2020).

4.1.3 Problem 3

Consider the 4×4 matrix

$$A = \begin{pmatrix} 43/9 & -4/3 & -10/9 & 0 \\ -5/3 & 49/9 & 0 & -10/9 \\ -13/9 & 0 & 49/9 & -4/3 \\ 0 & -13/9 & -5/3 & 55/9 \end{pmatrix}$$

where,

$$b = \begin{pmatrix} 5/9 \\ 8/27 \\ 22/9 \\ 62/27 \end{pmatrix}$$

Source: (Ndanusa and Adeboye, 2012).

Maple19 software package was deployed to compute the results of several experiments on problems 1, 2 and 3. The results are presented in Table 4.1 to 4.30. The following notations are engaged in what follows.

AOR = Accelerated Overrelaxation method

PAOR = Preconditioned Accelerated Overrelaxation method

RAOR = Refinement of Accelerated Overrelaxation method

RPAOR = Refinement of Preconditioned Accelerated Overrelaxation method

SRPAOR = Second Refinement of Preconditioned Accelerated Overrelaxation method

$\rho(\text{AOR})$ = Spectral radius of AOR method

$\rho(\text{PAOR})$ = Spectral radius of PAOR method

$\rho(\mathrm{RAOR})$ = Spectral radius of RAOR method

$\rho(\mathrm{RPAOR})$ = Spectral radius of RPAOR method

$\rho(\mathit{S}\mathrm{RPAOR})$ = Spectral radius of SRPAOR method

R(AOR) = Rate of convergence of AOR

R(PAOR) = Rate of convergence of PAOR

R(RAOR) = Rate of convergence of RAOR

R(RPAOR) = Rate of convergence of RPAOR

R(SRPAOR) = Rate of convergence of SRPAOR

ω = Relaxation parameter

r = Acceleration parameter

4.2 Comparison of Spectral Radii of Various Iteration Matrices

Table 4.1 Comparison of spectral radii of AOR and SRPAOR iteration matrices for Problem 1

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.009893293	1.848680312
1.99	2.0	0.9998927604	1.793573334
1.82	1.83	0.8298817777	1.020442454
1.79	1.80	0.7998793515	0.9133893344
1.69	1.7	0.6998697756	0.6120230179
1.59	1.6	0.5998570051	0.3866851466
1.49	1.5	0.4998391242	0.2255727860
1.39	1.4	0.3998122984	0.1173133886
1.29	1.3	0.2997675715	0.05096283204
1.19	1.2	0.1996780414	0.01599836683
1.0	1.1	0.3007043648	0.001000000000
0.99	1.0	0.3704668386	0.004251211905
0.89	0.9	0.4777155236	0.02506663540
0.79	0.8	0.5645296200	0.06344736088
0.69	0.7	0.6391159035	0.1197686057
0.59	0.6	0.7051521109	0.1940640691
0.49	0.5	0.7647069984	0.2861536937
0.39	0.4	0.8190985010	0.3957219909
0.29	0.3	0.8692356623	0.5223661540
0.19	0.2	0.9157811211	0.6656271037
0.09	0.1	0.9592379015	0.8250102598
-0.01	0	1	1

Table 4.2 Comparison of spectral radii of PAOR and SRPAOR iteration matrices for Problem 1

r	ω	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.227309065	1.848680312
1.99	2.0	1.214990959	1.793573334
1.82	1.83	1.006768247	1.020442454
1.79	1.80	0.9702537133	0.9133893344
1.69	1.7	0.8490291154	0.6120230179
1.59	1.6	0.7285384808	0.3866851466
1.49	1.5	0.6087358825	0.2255727860
1.39	1.4	0.4895336224	0.1173133886
1.29	1.3	0.3707528664	0.05096283204
1.19	1.2	0.2519756369	0.01599836683
1.0	1.1	0.1000000000	0.001000000000
0.99	1.0	0.1619959862	0.004251211905
0.89	0.9	0.2926613339	0.02506663540
0.79	0.8	0.3988453386	0.06344736088
0.69	0.7	0.4929251742	0.1197686057
0.59	0.6	0.5789597599	0.1940640691
0.49	0.5	0.6589712262	0.2861536937
0.39	0.4	0.7341701598	0.3957219909
0.29	0.3	0.8053630090	0.5223661540
0.19	0.2	0.8731261580	0.6656271037
0.09	0.1	0.9378926198	0.8250102598
-0.01	0	1	1

Table 4.3 Comparison of spectral radii of RAOR and SRPAOR iteration matrices for Problem 1

r	ω	$\rho(\mathbf{RAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.019884463	1.848680312
1.99	2.0	0.9997855336	1.793573334
1.82	1.83	0.6887037657	1.020442454
1.79	1.80	0.6398069818	0.9133893344
1.69	1.7	0.4898177034	0.6120230179
1.59	1.6	0.3598284264	0.3866851466
1.49	1.5	0.2498391500	0.2255727860
1.39	1.4	0.1598498739	0.1173133886
1.29	1.3	0.08986059746	0.05096283204
1.19	1.2	0.03987132039	0.01599836683
1.0	1.1	0.09042311477	0.001000000000
0.99	1.0	0.1372456785	0.004251211905
0.89	0.9	0.2282121220	0.02506663540
0.79	0.8	0.3186936911	0.06344736088
0.69	0.7	0.4084691386	0.1197686057
0.59	0.6	0.4972394992	0.1940640691
0.49	0.5	0.5847767892	0.2861536937
0.39	0.4	0.6709223525	0.3957219909
0.29	0.3	0.7555706408	0.5223661540
0.19	0.2	0.8386550511	0.6656271037
0.09	0.1	0.9201373491	0.8250102598
-0.01	0	1	1

Table 4.4 Comparison of spectral radiiof RPAOR and SRPAOR iteration matrices for Problem 1

r	ω	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.506287517	1.848680312
1.99	2.0	1.476203029	1.793573334
1.82	1.83	1.013582288	1.020442454
1.79	1.80	0.9413922592	0.9133893344
1.69	1.7	0.7208504418	0.6120230179
1.59	1.6	0.5307683201	0.3866851466
1.49	1.5	0.3705593737	0.2255727860
1.39	1.4	0.2396431707	0.1173133886
1.29	1.3	0.1374576881	0.05096283204
1.19	1.2	0.06349172135	0.01599836683
1.0	1.1	0.01000000000	0.001000000000
0.99	1.0	0.02624269922	0.004251211905
0.89	0.9	0.08565065659	0.02506663540
0.79	0.8	0.1590776049	0.06344736088
0.69	0.7	0.2429752275	0.1197686057
0.59	0.6	0.3351944015	0.1940640691
0.49	0.5	0.4342430792	0.2861536937
0.39	0.4	0.5390058215	0.3957219909
0.29	0.3	0.6486095719	0.5223661540
0.19	0.2	0.7623492840	0.6656271037
0.09	0.1	0.8796425683	0.8250102598
-0.01	0	1	1

Table 4.5 Comparison of spectral radii of AOR and SRPAOR iteration matrices for Problem 2

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.086141240	1.302639086
1.99	2.0	1.076145681	1.264861418
1.82	1.83	0.9052579404	0.7295606329
1.79	1.80	0.8749168765	0.6545129583
1.69	1.7	0.7733956238	0.4416465038
1.59	1.6	0.6713025117	0.2805291441
1.49	1.5	0.5686635325	0.1638171188
1.39	1.4	0.4655027235	0.08425053731
1.29	1.3	0.3618356085	0.04156210740
1.19	1.2	0.2576567294	0.01476825437
1.0	1.1	0.2551097687	0.002855762565
0.99	1.0	0.3274088010	0.01069930790
0.89	0.9	0.4292694335	0.03831926478
0.79	0.8	0.5167206131	0.08268898026
0.69	0.7	0.5946811896	0.1437009750
0.59	0.6	0.6655917159	0.2210473853
0.49	0.5	0.7309134208	0.3142943979
0.39	0.4	0.7916211275	0.4229316603
0.29	0.3	0.8484127557	0.5464040117
0.19	0.2	0.9018133690	0.6841323274
0.09	0.1	0.9522327900	0.8355275037
-0.01	0	1	1.000000001

Table 4.6 Comparison of spectral radii of PAOR and SRPAOR iteration matrices for Problem 2

r	ω	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.092130916	1.302639086
1.99	2.0	1.081469598	1.264861418
1.82	1.83	0.9002306549	0.7295606329
1.79	1.80	0.8682392547	0.6545129583
1.69	1.7	0.7615380329	0.4416465038
1.59	1.6	0.6546251133	0.2805291441
1.49	1.5	0.5471668321	0.1638171188
1.39	1.4	0.4383868916	0.08425053731
1.29	1.3	0.3463904017	0.04156210740
1.19	1.2	0.2453445324	0.01476825437
1.0	1.1	0.1418754876	0.002855762565
0.99	1.0	0.2203527947	0.01069930790
0.89	0.9	0.3371364609	0.03831926478
0.79	0.8	0.4356615288	0.08268898026
0.69	0.7	0.5237852166	0.1437009750
0.59	0.6	0.6046375703	0.2210473853
0.49	0.5	0.6799007918	0.3142943979
0.39	0.4	0.7506256447	0.4229316603
0.29	0.3	0.8175317490	0.5464040117
0.19	0.2	0.8811436306	0.6841323274
0.09	0.1	0.9418612254	0.8355275037
-0.01	0	1	1.000000001

Table 4.7 Comparison of spectral radii of RAOR and SRPAOR iteration matrices for Problem 2

r	ω	$\rho(\mathbf{RAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.179702788	1.302639086
1.99	2.0	1.158089536	1.264861418
1.82	1.83	0.8194919381	0.7295606329
1.79	1.80	0.7654795405	0.6545129583
1.69	1.7	0.5981407923	0.4416465038
1.59	1.6	0.4506470575	0.2805291441
1.49	1.5	0.3233782157	0.1638171188
1.39	1.4	0.2166927859	0.08425053731
1.29	1.3	0.1309250064	0.04156210740
1.19	1.2	0.06638699001	0.01476825437
1.0	1.1	0.06508099396	0.002855762565
0.99	1.0	0.1071965225	0.01069930790
0.89	0.9	0.1842722465	0.03831926478
0.79	0.8	0.2670001934	0.08268898026
0.69	0.7	0.3536457184	0.1437009750
0.59	0.6	0.4430123315	0.2210473853
0.49	0.5	0.5342344253	0.3142943979
0.39	0.4	0.6266640101	0.4229316603
0.29	0.3	0.7198042025	0.5464040117
0.19	0.2	0.8132673489	0.6841323274
0.09	0.1	0.9067472857	0.8355275037
-0.01	0	1	1.000000001

Table 4.8 Comparison of spectral radii of RPAOR and SRPAOR iteration matrices for Problem 2

r	ω	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.192749937	1.302639086
1.99	2.0	1.169576493	1.264861418
1.82	1.83	0.8104152289	0.7295606329
1.79	1.80	0.7538394015	0.6545129583
1.69	1.7	0.5799401768	0.4416465038
1.59	1.6	0.4285340382	0.2805291441
1.49	1.5	0.2993915393	0.1638171188
1.39	1.4	0.1921830669	0.08425053731
1.29	1.3	0.1199863119	0.04156210740
1.19	1.2	0.06019394063	0.01476825437
1.0	1.1	0.02012865380	0.002855762565
0.99	1.0	0.04855535384	0.01069930790
0.89	0.9	0.1136609927	0.03831926478
0.79	0.8	0.1898009693	0.08268898026
0.69	0.7	0.2743509539	0.1437009750
0.59	0.6	0.3655865880	0.2210473853
0.49	0.5	0.4622650879	0.3142943979
0.39	0.4	0.5634388655	0.4229316603
0.29	0.3	0.6683581566	0.5464040117
0.19	0.2	0.7764140895	0.6841323274
0.09	0.1	0.8871025619	0.8355275037
-0.01	0	1	1.000000001

Table 4.9 Comparison of spectral radii of AOR and SRPAOR iteration matrices for Problem 3

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.009983181	1.730048988
1.99	2.0	0.9999830975	1.679119523
1.82	1.83	0.8299813659	0.961804435
1.79	1.80	0.7999809844	0.8619876784
1.69	1.7	0.6999794757	0.5801306648
1.59	1.6	0.5999774624	0.3683177789
1.49	1.5	0.4999746450	0.2160610214
1.39	1.4	0.3999704175	0.1131444290
1.29	1.3	0.2999633744	0.04962725472
1.19	1.2	0.1999492841	0.01584281244
1.0	1.1	0.1865127564	0.001270386970
0.99	1.0	0.2675950738	0.0007036148046
0.89	0.9	0.3889501242	0.01109632675
0.79	0.8	0.4867116564	0.03682664372
0.69	0.7	0.5715207421	0.08055908117
0.59	0.6	0.6475230708	0.1441541876
0.49	0.5	0.7169138494	0.2289808309
0.39	0.4	0.7810467239	0.3360689052
0.29	0.3	0.8408376562	0.4661987189
0.19	0.2	0.8969466513	0.6199590652
0.09	0.1	0.9498713341	0.7977875647
-0.01	0	1	1

Table 4.10 Comparison of spectral radii of PAOR and SRPAOR iteration matrices for Problem 3

r	ω	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.200474117	1.730048988
1.99	2.0	1.188576678	1.679119523
1.82	1.83	0.9871025140	0.961804435
1.79	1.80	0.9517006211	0.8619876784
1.69	1.7	0.8340177116	0.5801306648
1.59	1.6	0.7168157878	0.3683177789
1.49	1.5	0.6000564957	0.2160610214
1.39	1.4	0.4836646991	0.1131444290
1.29	1.3	0.3674853941	0.04962725472
1.19	1.2	0.2511563106	0.01584281244
1.0	1.1	0.1083042112	0.001270386970
0.99	1.0	0.08894297642	0.0007036148046
0.89	0.9	0.2230453018	0.01109632675
0.79	0.8	0.3327009542	0.03682664372
0.69	0.7	0.4318883616	0.08055908117
0.59	0.6	0.5243352888	0.1441541876
0.49	0.5	0.6117862466	0.2289808309
0.39	0.4	0.6952528490	0.3360689052
0.29	0.3	0.7753962411	0.4661987189
0.19	0.2	0.8526831330	0.6199590652
0.09	0.1	0.9274612064	0.7977875647
-0.01	0	1	1

Table 4.11 Comparison of spectral radii of RAOR and SRPAOR iteration matrices for Problem 3

r	ω	$\rho(\mathrm{RAOR})$	$\rho(\mathrm{SRPAOR})$
2.0	2.01	1.020066025	1.730048988
1.99	2.0	0.9999661949	1.679119523
1.82	1.83	0.6888690663	0.961804435
1.79	1.80	0.6399695755	0.8619876784
1.69	1.7	0.4899712647	0.5801306648
1.59	1.6	0.3599729551	0.3683177789
1.49	1.5	0.2499746454	0.2160610214
1.39	1.4	0.1599763359	0.1131444290
1.29	1.3	0.08997802600	0.04962725472
1.19	1.2	0.03997971631	0.01584281244
1.0	1.1	0.03478700819	0.001270386970
0.99	1.0	0.07160712315	0.0007036148046
0.89	0.9	0.1512821990	0.01109632675
0.79	0.8	0.2368882390	0.03682664372
0.69	0.7	0.3266359600	0.08055908117
0.59	0.6	0.4192861275	0.1441541876
0.49	0.5	0.5139654660	0.2289808309
0.39	0.4	0.6100339841	0.3360689052
0.29	0.3	0.7070079673	0.4661987189
0.19	0.2	0.8045132960	0.6199590652
0.09	0.1	0.9022555517	0.7977875647
-0.01	0	1	1

Table 4.12 Comparison of spectral radii of RPAOR and SRPAOR iteration matrices for Problem 3

r	ω	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.506287517	1.730048988
1.99	2.0	1.476203029	1.679119523
1.82	1.83	1.013582288	0.961804435
1.79	1.80	0.9413922592	0.8619876784
1.69	1.7	0.7208504418	0.5801306648
1.59	1.6	0.5307683201	0.3683177789
1.49	1.5	0.3705593737	0.2160610214
1.39	1.4	0.2396431707	0.1131444290
1.29	1.3	0.1374576881	0.04962725472
1.19	1.2	0.06349172135	0.01584281244
1.0	1.1	0.01000000000	0.001270386970
0.99	1.0	0.02624269922	0.0007036148046
0.89	0.9	0.08565065659	0.01109632675
0.79	0.8	0.1590776049	0.03682664372
0.69	0.7	0.2429752275	0.08055908117
0.59	0.6	0.3351944015	0.1441541876
0.49	0.5	0.4342430792	0.2289808309
0.39	0.4	0.5390058215	0.3360689052
0.29	0.3	0.6486095719	0.4661987189
0.19	0.2	0.7623492840	0.6199590652
0.09	0.1	0.8796425683	0.7977875647
-0.01	0	1	1

4.3 Comparison of Rates of Convergence

Table 4.13 Convergence rates of AOR and SRPAOR for Problem 1

r	ω	$R(\text{AOR})$	$R(\text{SRPAOR})$	Ratio
2.0	2.01	-0.009844674	-0.6144720397	Divergence
1.99	2.0	0.0001072453	-0.5842099059	Divergence
1.82	1.83	0.1864720248	-0.0202363117	Divergence
1.79	1.80	0.2232943733	0.0905930551	0.4057113208
1.69	1.7	0.3568609961	0.4909853862	1.3758449132
1.59	1.6	0.5110639770	0.9501444918	1.8591498023
1.49	1.5	0.6934689839	1.4891123960	2.1473381371
1.39	1.4	0.9167600960	2.1429063900	2.3374778193
1.29	1.3	1.204747866	2.9766586950	2.4707731626
1.19	1.2	1.611049003	4.1352686350	2.5668174136
1.0	1.1	1.201627674	6.9077552790	5.7486652717
0.99	1.0	0.9929913427	5.4605511830	5.4990924374
0.89	0.9	0.7387398626	3.6862175840	4.9898723091
0.79	0.8	0.5717624256	2.7575446790	4.8228854425
0.69	0.7	0.4476694584	2.1221936840	4.7405371177
0.59	0.6	0.3493417393	1.6395669210	4.6933038242
0.49	0.5	0.2682625271	1.2512262220	4.6641856227
0.39	0.4	0.1995509325	0.9270433575	4.6456478348
0.29	0.3	0.1401410026	0.6493864926	4.6338079545
0.19	0.2	0.08797789363	0.4070256696	4.6264539057
0.09	0.1	0.04161616239	0.1923594566	4.6222295751
-0.01	0	-0.	-0	Divergence

Table 4.14 Convergence rates of PAOR and SRPAOR for Problem 1

r	ω	$R(\mathbf{PAOR})$	$R(\mathbf{SRPAOR})$	**Ratio**
2.0	2.01	-0.2048240207	-0.6144720397	Divergence
1.99	2.0	-0.1947366356	-0.5842099059	Divergence
1.82	1.83	-0.006745445244	-0.0202363117	Divergence
1.79	1.80	0.03019768158	0.0905930551	3.0000003434
1.69	1.7	0.1636617995	0.4909853862	2.9999999248
1.59	1.6	0.3167148328	0.9501444918	2.9999999792
1.49	1.5	0.4963707958	1.4891123960	3.0000000173
1.39	1.4	0.7143021321	2.1429063900	2.9999999912
1.29	1.3	0.9922195666	2.9766586950	2.9999999952
1.19	1.2	1.378422875	4.1352686350	3.0000000073
1.0	1.1	2.302585093	6.9077552790	3.0000000000
0.99	1.0	1.820183721	5.4605511830	3.0000000110
0.89	0.9	1.228739195	3.6862175840	2.9999999992
0.79	0.8	0.9191815598	2.7575446790	2.9999999996
0.69	0.7	0.7073978929	2.1221936840	3.0000000075
0.59	0.6	0.5465223031	1.6395669210	3.0000000214
0.49	0.5	0.4170754083	1.2512262220	2.9999999930
0.39	0.4	0.3090144519	0.9270433575	3.0000000058
0.29	0.3	0.2164621604	0.6493864926	3.0000000527
0.19	0.2	0.1356752227	0.4070256696	3.0000000111
0.09	0.1	0.06411981435	0.1923594566	3.0000002113
-0.01	0	-0.	-0	Divergence

Table 4.15 Convergence rates of RAOR and SRPAOR for Problem 1

r	ω	$R(\mathrm{RAOR})$	$R(\mathrm{SRPAOR})$	**Ratio**
2.0	2.01	-0.0196893493	-0.6144720397	Divergence
1.99	2.0	0.00021448940	-0.5842099059	Divergence
1.82	1.83	0.3729440486	-0.0202363117	Divergence
1.79	1.80	0.4465887391	0.0905930551	0.2028556638
1.69	1.7	0.7137219910	0.4909853862	0.6879224578
1.59	1.6	1.022127954	0.9501444918	0.9295749011
1.49	1.5	1.386937968	1.4891123960	1.0736690684
1.39	1.4	1.833520192	2.1429063900	1.1687389096
1.29	1.3	2.409495727	2.9766586950	1.2353865839
1.19	1.2	3.222098001	4.1352686350	1.2834087088
1.0	1.1	2.403255350	6.9077552790	2.8743326334
0.99	1.0	1.985982685	5.4605511830	2.7495462192
0.89	0.9	1.477479723	3.6862175840	2.4949361583
0.79	0.8	1.143524853	2.7575446790	2.4114427175
0.69	0.7	0.8953389156	2.1221936840	2.3702685620
0.59	0.6	0.6986834792	1.6395669210	2.3466519101
0.49	0.5	0.5365250615	1.2512262220	2.3320927796
0.39	0.4	0.3991018678	0.9270433575	2.3228239011
0.29	0.3	0.2802819996	0.6493864926	2.3169040235
0.19	0.2	0.1759558000	0.4070256696	2.3132267854
0.09	0.1	0.08323232758	0.1923594566	2.3111147098
-0.01	0	-0.	-0.	Divergence

Table 4.16 Convergence rates of RPAOR and SRPAOR for Problem 1

r	ω	$R(\text{RPAOR})$	$R(\text{SRPAOR})$	**Ratio**
2.0	2.01	-0.4096480255	-0.6144720397	1.5000000035
1.99	2.0	-0.3894732702	-0.5842099059	1.5000000015
1.82	1.83	-0.01349087552	-0.0202363117	1.4999998807
1.79	1.80	0.06039537270	0.0905930551	1.4999999348
1.69	1.7	0.3273235948	0.4909853862	1.4999999817
1.59	1.6	0.6334296616	0.9501444918	1.4999999991
1.49	1.5	0.9927415942	1.4891123960	1.5000000047
1.39	1.4	1.428604251	2.1429063900	1.5000000094
1.29	1.3	1.984439132	2.9766586950	1.4999999985
1.19	1.2	2.756845754	4.1352686350	1.5000000015
1.0	1.1	4.605170186	6.9077552790	1.5000000000
0.99	1.0	3.640367454	5.4605511830	1.5000000005
0.89	0.9	2.457478388	3.6862175840	1.5000000008
0.79	0.8	1.838363115	2.7575446790	1.5000000035
0.69	0.7	1.414795785	2.1221936840	1.5000000046
0.59	0.6	1.093044612	1.6395669210	1.5000000027
0.49	0.5	0.8341508114	1.2512262220	1.5000000059
0.39	0.4	0.6180289076	0.9270433575	1.4999999937
0.29	0.3	0.4329243274	0.6493864926	1.5000000035
0.19	0.2	0.2713504504	0.4070256696	1.4999999779
0.09	0.1	0.1282396264	0.1923594566	1.5000001326
-0.01	0	-0.	-0.	Divergence

Table 4.17 Convergence rates of AOR and SRPAOR for Problem 2

r	ω	$R(\mathrm{AOR})$	$R(\mathrm{SRPAOR})$	**Ratio**
2.0	2.01	-0.08263126830	-0.2643922728	Divergence
1.99	2.0	-0.07338584384	-0.2349625652	Divergence
1.82	1.83	0.09953535888	0.3153127988	3.1678471083
1.79	1.80	0.1336263954	0.4238638951	3.1720072507
1.69	1.7	0.2569645582	0.8172454820	3.1803821030
1.59	1.6	0.3985354065	1.271077659	3.1893719812
1.49	1.5	0.5644663510	1.809004603	3.2048050336
1.39	1.4	0.7646373316	2.473960332	3.2354689338
1.29	1.3	1.016565290	3.180566407	3.1287379554
1.19	1.2	1.356127086	4.215275377	3.1083188445
1.0	1.1	1.366061361	5.858416373	4.2885455517
0.99	1.0	1.116545733	4.537576222	4.0639412143
0.89	0.9	0.8456705071	3.261802512	3.8570607401
0.79	0.8	0.6602529507	2.492668935	3.7753241880
0.69	0.7	0.5197298328	1.940020701	3.7327483984
0.59	0.6	0.4070788356	1.509378187	3.7078277105
0.49	0.5	0.3134602656	1.157425159	3.6924142739
0.39	0.4	0.2336723760	0.8605446726	3.6826974901
0.29	0.3	0.1643880214	0.6043966287	3.6766464098
0.19	0.2	0.1033476883	0.3796039190	3.6730760527
0.09	0.1	0.04894574677	0.1796920126	3.6712487695
-0.01	0	-0.	-0.00000000009	Divergence

Table 4.18 Convergence rates of PAOR and SRPAOR for Problem 2

r	ω	$R(\text{PAOR})$	$R(SRPAOR)$	Ratio
2.0	2.01	-0.08813075658	-0.2643922728	Divergence
1.99	2.0	-0.07832085506	-0.2349625652	Divergence
1.82	1.83	0.1051042653	0.3153127988	3.0000000276
1.79	1.80	0.1412879632	0.4238638951	3.0000000389
1.69	1.7	0.2724151632	0.8172454820	2.9999999721
1.59	1.6	0.4236925533	1.271077659	2.9999999979
1.49	1.5	0.6030015284	1.809004603	3.0000000295
1.39	1.4	0.8246534443	2.473960332	2.9999999989
1.29	1.3	1.060188811	3.180566407	2.9999999755
1.19	1.2	1.405091802	4.215275377	2.9999999794
1.0	1.1	1.952805454	5.858416373	3.0000000056
0.99	1.0	1.512525405	4.537576222	3.0000000046
0.89	0.9	1.087267502	3.261802512	3.0000000055
0.79	0.8	0.8308896472	2.492668935	2.9999999921
0.69	0.7	0.6466735707	1.940020701	2.9999999828
0.59	0.6	0.5031260578	1.509378187	3.0000000270
0.49	0.5	0.3858083859	1.157425159	3.0000000034
0.39	0.4	0.2868482273	0.8605446726	2.9999999676
0.29	0.3	0.2014655403	0.6043966287	3.0000000387
0.19	0.2	0.1265346350	0.3796039190	3.0000001106
0.09	0.1	0.05989733437	0.1796920126	3.0000001584
-0.01	0	-0.	-0.00000000009	Divergence

Table 4.19 Convergence rates of RAOR and SRPAOR for Problem 2

r	ω	$R(\text{RAOR})$	$R(\text{SRPAOR})$	Ratio
2.0	2.01	-0.1652625322	-0.2643922728	Divergence
1.99	2.0	-0.1467716957	-0.2349625652	Divergence
1.82	1.83	0.1990707184	0.3153127988	1.5839235490
1.79	1.80	0.2672527912	0.4238638951	1.5860036230
1.69	1.7	0.5139291141	0.8172454820	1.5901910586
1.59	1.6	0.7970708235	1.271077659	1.5946859696
1.49	1.5	1.128932694	1.809004603	1.6024025282
1.39	1.4	1.529274662	2.473960332	1.6177344682
1.29	1.3	2.033130590	3.180566407	1.5643689700
1.19	1.2	2.712254175	4.215275377	1.5541594205
1.0	1.1	2.732122724	5.858416373	2.1442727743
0.99	1.0	2.233091470	4.537576222	2.0319706035
0.89	0.9	1.691341014	3.261802512	1.9285303703
0.79	0.8	1.320505896	2.492668935	1.8876621017
0.69	0.7	1.039459662	1.940020701	1.8663742057
0.59	0.6	0.8141576730	1.509378187	1.8539138512
0.49	0.5	0.6269205377	1.157425159	1.8462071178
0.39	0.4	0.4673447510	0.8605446726	1.8413487490
0.29	0.3	0.3287760449	0.6043966287	1.8383231932
0.19	0.2	0.2066953811	0.3796039190	1.8365379864
0.09	0.1	0.09789149426	0.1796920126	1.8356243712
-0.01	0	-0.	-0.00000000009	Divergence

Table 4.20 Convergence rates of RPAOR and SRPAOR for Problem 2

r	ω	$R(\mathbf{RPAOR})$	$R(\mathbf{SRPAOR})$	**Ratio**
2.0	2.01	-0.1762615126	-0.2643922728	Divergence
1.99	2.0	-0.1566417115	-0.2349625652	Divergence
1.82	1.83	0.2102085344	0.3153127988	1.4999999867
1.79	1.80	0.2825759290	0.4238638951	1.5000000057
1.69	1.7	0.5448303242	0.8172454820	1.4999999921
1.59	1.6	0.8473851084	1.271077659	1.4999999958
1.49	1.5	1.206003066	1.809004603	1.5000000033
1.39	1.4	1.649306888	2.473960332	1.5000000000
1.29	1.3	2.120377610	3.180566407	1.4999999962
1.19	1.2	2.810183586	4.215275377	1.4999999993
1.0	1.1	3.905610917	5.858416373	1.4999999994
0.99	1.0	3.025050816	4.537576222	1.4999999993
0.89	0.9	2.174535009	3.261802512	1.4999999993
0.79	0.8	1.661779286	2.492668935	1.5000000036
0.69	0.7	1.293347139	1.940020701	1.4999999942
0.59	0.6	1.006252125	1.509378187	1.4999999995
0.49	0.5	0.7716167691	1.157425159	1.5000000069
0.39	0.4	0.5736964421	0.8605446726	1.5000000165
0.29	0.3	0.4029310866	0.6043966287	1.4999999970
0.19	0.2	0.2530692806	0.3796039190	1.4999999925
0.09	0.1	0.1197946755	0.1796920126	1.4999999946
-0.01	0	-0.	-0.00000000009	Divergence

Table 4.21 Convergence rates of AOR and SRPAOR for Problem 3

r	ω	$R(\mathbf{AOR})$	$R(\mathbf{SRPAOR})$	**Ratio**
2.0	2.01	-0.0099336782	-0.5481497249	Divergence
1.99	2.0	0.0000169026	-0.5182695626	Divergence
1.82	1.83	0.1863520292	0.03894413900	0.2089815666
1.79	1.80	0.2231673211	0.1485143026	0.6654840945
1.69	1.7	0.3567042648	0.5445019167	1.5264799736
1.59	1.6	0.5108631871	0.9988091838	1.9551402587
1.49	1.5	0.6931978918	1.532194405	2.2103275603
1.39	1.4	0.9163646909	2.179090144	2.3779726190
1.29	1.3	1.204094897	3.003215106	2.4941681204
1.19	1.2	1.609691524	4.145039355	2.5750519856
1.0	1.1	1.679255643	6.668433724	3.9710652466
0.99	1.0	1.318280360	7.259279504	5.5066279710
0.89	0.9	0.9443041590	4.501141149	4.7666221800
0.79	0.8	0.7200834126	3.301533682	4.5849322790
0.69	0.7	0.5594545023	2.518764436	4.5021792222
0.59	0.6	0.4346008555	1.936871805	4.4566681830
0.49	0.5	0.3327995999	1.474116987	4.4294433871
0.39	0.4	0.2471203052	1.090439065	4.4125838389
0.29	0.3	0.1733566743	0.7631433003	4.4021570175
0.19	0.2	0.1087588933	0.4781018270	4.3959791470
0.09	0.1	0.05142874135	0.2259129266	4.3927368368
-0.01	0	-0.	-0.	Divergence

Table 4.22 Convergence rates of PAOR and SRPAOR for Problem 3

r	ω	$R(\mathrm{PAOR})$	$R(\mathrm{SRPAOR})$	Ratio
2.0	2.01	-0.1827165763	-0.5481497249	Divergence
1.99	2.0	-0.1727565224	-0.5182695626	Divergence
1.82	1.83	0.01298138071	0.03894413900	2.9999997589
1.79	1.80	0.04950476728	0.1485143026	3.0000000154
1.69	1.7	0.1815006399	0.5445019167	2.9999999835
1.59	1.6	0.3329363922	0.9988091838	3.0000000216
1.49	1.5	0.5107314687	1.532194405	2.9999999978
1.39	1.4	0.7263633828	2.179090144	2.9999999939
1.29	1.3	1.001071705	3.003215106	2.9999999910
1.19	1.2	1.381679782	4.145039355	3.0000000065
1.0	1.1	2.222811241	6.668433724	3.0000000004
0.99	1.0	2.419759829	7.259279504	3.0000000070
0.89	0.9	1.500380381	4.501141149	3.0000000040
0.79	0.8	1.100511228	3.301533682	2.9999999982
0.69	0.7	0.8395881464	2.518764436	2.9999999962
0.59	0.6	0.6456239352	1.936871805	2.9999999991
0.49	0.5	0.4913723277	1.474116987	3.0000000079
0.39	0.4	0.3634796881	1.090439065	3.0000000019
0.29	0.3	0.2543811014	0.7631433003	2.9999999847
0.19	0.2	0.1593672741	0.4781018270	3.0000000295
0.09	0.1	0.07530431136	0.2259129266	2.9999999007
-0.01	0	-0.	-0.	Divergence

Table 4.23 Convergence rates of RAOR and SRPAOR for Problem 3

r	ω	$R(\mathbf{RAOR})$	$R(\mathbf{SRPAOR})$	**Ratio**
2.0	2.01	-0.01986735559	-0.5481497249	Divergence
1.99	2.0	0.000033805671	-0.5182695626	Divergence
1.82	1.83	0.3727040604	0.03894413900	0.1044907827
1.79	1.80	0.4463346420	0.1485143026	0.3327420474
1.69	1.7	0.7134085331	0.5445019167	0.7632399830
1.59	1.6	1.021726375	0.9988091838	0.9775701286
1.49	1.5	1.386395785	1.532194405	1.1051637790
1.39	1.4	1.832729375	2.179090144	1.1889863139
1.29	1.3	2.408189794	3.003215106	1.2470840602
1.19	1.2	3.219383046	4.145039355	1.2875259936
1.0	1.1	3.358511290	6.668433724	1.9855326209
0.99	1.0	2.636560725	7.259279504	2.7533139803
0.89	0.9	1.888608319	4.501141149	2.3833110888
0.79	0.8	1.440166814	3.301533682	2.2924661573
0.69	0.7	1.118909000	2.518764436	2.2510896203
0.59	0.6	0.8692017103	1.936871805	2.2283340933
0.49	0.5	0.6655992026	1.474116987	2.2147216842
0.39	0.4	0.4942406117	1.090439065	2.2062919137
0.29	0.3	0.3467133440	0.7631433003	2.2010785380
0.19	0.2	0.2175177857	0.4781018270	2.1979895826
0.09	0.1	0.1028574823	0.2259129266	2.1963684270
-0.01	0	-0.	-0.	Divergence

Table 4.24 Convergence rates of RPAOR and SRPAOR for Problem 3

r	ω	$R(\text{RPAOR})$	$R(\text{SRPAOR})$	**Ratio**
2.0	2.01	-0.4096480255	-0.5481497249	Divergence
1.99	2.0	-0.3894732702	-0.5182695626	Divergence
1.82	1.83	-0.0134908755	0.03894413900	Divergence
1.79	1.80	0.06039537270	0.1485143026	2.4590344584
1.69	1.7	0.3273235948	0.5445019167	1.6634973016
1.59	1.6	0.6334296616	0.9988091838	1.5768273012
1.49	1.5	0.9927415942	1.532194405	1.5433970068
1.39	1.4	1.428604251	2.179090144	1.5253280553
1.29	1.3	1.984439132	3.003215106	1.5133823243
1.19	1.2	2.756845754	4.145039355	1.5035441678
1.0	1.1	4.605170186	6.668433724	1.4480319846
0.99	1.0	3.640367454	7.259279504	1.9941062532
0.89	0.9	2.457478388	4.501141149	1.8316096577
0.79	0.8	1.838363115	3.301533682	1.7959094452
0.69	0.7	1.414795785	2.518764436	1.7803024738
0.59	0.6	1.093044612	1.936871805	1.7719970290
0.49	0.5	0.8341508114	1.474116987	1.7672068010
0.39	0.4	0.6180289076	1.090439065	1.7643819756
0.29	0.3	0.4329243274	0.7631433003	1.7627637257
0.19	0.2	0.2713504504	0.4781018270	1.7619348938
0.09	0.1	0.1282396264	0.2259129266	1.7616467931
-0.01	0	-0.	-0.	Divergence

Table 4.25 Comparison of spectral radii of various methods for Problem 1

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{RAOR})$	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.009893293	1.227309065	1.019884463	1.506287517	1.848680312
1.99	2.0	0.9998927604	1.214990959	0.9997855336	1.476203029	1.793573334
1.82	1.83	0.8298817777	1.006768247	0.6887037657	1.013582288	1.020442454
1.79	1.80	0.7998793515	0.9702537133	0.6398069818	0.9413922592	0.9133893344
1.69	1.7	0.6998697756	0.8490291154	0.4898177034	0.7208504418	0.6120230179
1.59	1.6	0.5998570051	0.7285384808	0.3598284264	0.5307683201	0.3866851466
1.49	1.5	0.4998391242	0.6087358825	0.2498391500	0.3705593737	0.2255727860
1.39	1.4	0.3998122984	0.4895336224	0.1598498739	0.2396431707	0.1173133886
1.29	1.3	0.2997675715	0.3707528664	0.08986059746	0.1374576881	0.05096283204
1.19	1.2	0.1996780414	0.2519756369	0.03987132039	0.06349172135	0.01599836683
1.0	1.1	0.3007043648	0.1000000000	0.09042311477	0.01000000000	0.001000000000
0.99	1.0	0.3704668386	0.1619959862	0.1372456785	0.02624269922	0.004251211905
0.89	0.9	0.4777155236	0.2926613339	0.2282121220	0.08565065659	0.02506663540
0.79	0.8	0.5645296200	0.3988453386	0.3186936911	0.1590776049	0.06344736088
0.69	0.7	0.6391159035	0.4929251742	0.4084691386	0.2429752275	0.1197686057
0.59	0.6	0.7051521109	0.5789597599	0.4972394992	0.3351944015	0.1940640691
0.49	0.5	0.7647069984	0.6589712262	0.5847767892	0.4342430792	0.2861536937
0.39	0.4	0.8190985010	0.7341701598	0.6709223525	0.5390058215	0.3957219909
0.29	0.3	0.8692356623	0.8053630090	0.7555706408	0.6486095719	0.5223661540
0.19	0.2	0.9157811211	0.8731261580	0.8386550511	0.7623492840	0.6656271037
0.09	0.1	0.9592379015	0.9378926198	0.9201373491	0.8796425683	0.8250102598
-0.01	0	1	1	1	1	1

Table 4.26 Comparison of spectral radii of various methods for Problem 2

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{RAOR})$	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.086141240	1.092130916	1.179702788	1.192749937	1.302639086
1.99	2.0	1.076145681	1.081469598	1.158089536	1.169576493	1.264861418
1.82	1.83	0.9052579404	0.9002306549	0.8194919381	0.8104152289	0.7295606329
1.79	1.80	0.8749168765	0.8682392547	0.7654795405	0.7538394015	0.6545129583
1.69	1.7	0.7733956238	0.7615380329	0.5981407923	0.5799401768	0.4416465038
1.59	1.6	0.6713025117	0.6546251133	0.4506470575	0.4285340382	0.2805291441
1.49	1.5	0.5686635325	0.5471668321	0.3233782157	0.2993915393	0.1638171188
1.39	1.4	0.4655027235	0.4383868916	0.2166927859	0.1921830669	0.08425053731
1.29	1.3	0.3618356085	0.3463904017	0.1309250064	0.1199863119	0.04156210740
1.19	1.2	0.2576567294	0.2453445324	0.06638699001	0.06019394063	0.01476825437
1.0	1.1	0.2551097687	0.1418754876	0.06508099396	0.02012865380	0.002855762565
0.99	1.0	0.3274088010	0.2203527947	0.1071965225	0.04855535384	0.01069930790
0.89	0.9	0.4292694335	0.3371364609	0.1842722465	0.1136609927	0.03831926478
0.79	0.8	0.5167206131	0.4356615288	0.2670001934	0.1898009693	0.08268898026
0.69	0.7	0.5946811896	0.5237852166	0.3536457184	0.2743509539	0.1437009750
0.59	0.6	0.6655917159	0.6046375703	0.4430123315	0.3655865880	0.2210473853
0.49	0.5	0.7309134208	0.6799007918	0.5342344253	0.4622650879	0.3142943979
0.39	0.4	0.7916211275	0.7506256447	0.6266640101	0.5634388655	0.4229316603
0.29	0.3	0.8484127557	0.8175317490	0.7198042025	0.6683581566	0.5464040117
0.19	0.2	0.9018133690	0.8811436306	0.8132673489	0.7764140895	0.6841323274
0.09	0.1	0.9522327900	0.9418612254	0.9067472857	0.8871025619	0.8355275037
-0.01	0	1	1	1	1	1.000000001

Table 4.27 Comparison of spectral radii of various methods for Problem 3

r	ω	$\rho(\mathbf{AOR})$	$\rho(\mathbf{PAOR})$	$\rho(\mathbf{RAOR})$	$\rho(\mathbf{RPAOR})$	$\rho(\mathbf{SRPAOR})$
2.0	2.01	1.009983181	1.200474117	1.020066025	1.506287517	1.730048988
1.99	2.0	0.9999830975	1.188576678	0.9999661949	1.476203029	1.679119523
1.82	1.83	0.8299813659	0.9871025140	0.6888690663	1.013582288	0.961804435
1.79	1.80	0.7999809844	0.9517006211	0.6399695755	0.9413922592	0.8619876784
1.69	1.7	0.6999794757	0.8340177116	0.4899712647	0.7208504418	0.5801306648
1.59	1.6	0.5999774624	0.7168157878	0.3599729551	0.5307683201	0.3683177789
1.49	1.5	0.4999746450	0.6000564957	0.2499746454	0.3705593737	0.2160610214
1.39	1.4	0.3999704175	0.4836646991	0.1599763359	0.2396431707	0.1131444290
1.29	1.3	0.2999633744	0.3674853941	0.08997802600	0.1374576881	0.04962725472
1.19	1.2	0.1999492841	0.2511563106	0.03997971631	0.06349172135	0.01584281244
1.0	1.1	0.1865127564	0.1083042112	0.03478700819	0.01000000000	0.00127038697
0.99	1.0	0.2675950738	0.08894297642	0.07160712315	0.02624269922	0.00070361480
0.89	0.9	0.3889501242	0.2230453018	0.1512821990	0.08565065659	0.01109632675
0.79	0.8	0.4867116564	0.3327009542	0.2368882390	0.1590776049	0.03682664372
0.69	0.7	0.5715207421	0.4318883616	0.3266359600	0.2429752275	0.08055908117
0.59	0.6	0.6475230708	0.5243352888	0.4192861275	0.3351944015	0.1441541876
0.49	0.5	0.7169138494	0.6117862466	0.5139654660	0.4342430792	0.2289808309
0.39	0.4	0.7810467239	0.6952528490	0.6100339841	0.5390058215	0.3360689052
0.29	0.3	0.8408376562	0.7753962411	0.7070079673	0.6486095719	0.4661987189
0.19	0.2	0.8969466513	0.8526831330	0.8045132960	0.7623492840	0.6199590652
0.09	0.1	0.9498713341	0.9274612064	0.9022555517	0.8796425683	0.7977875647
-0.01	0	1	1	1	1	1

Table 4.28 Convergence rates of various methods for Problem 1

r	ω	$R(\mathbf{AOR})$	$R(\mathbf{PAOR})$	$R(\mathbf{RAOR})$	$R(\mathbf{RPAOR})$	$R(\mathbf{SRPAOR})$
2.0	2.01	-0.009844674	-0.2048240207	-0.0196893493	-0.4096480255	-0.6144720397
1.99	2.0	0.0001072453	-0.1947366356	0.00021448940	-0.3894732702	-0.5842099059
1.82	1.83	0.1864720248	-0.0067454452	0.3729440486	-0.0134908755	-0.0202363117
1.79	1.80	0.2232943733	0.03019768158	0.4465887391	0.06039537270	0.0905930551
1.69	1.7	0.3568609961	0.1636617995	0.7137219910	0.3273235948	0.4909853862
1.59	1.6	0.5110639770	0.3167148328	1.022127954	0.6334296616	0.9501444918
1.49	1.5	0.6934689839	0.4963707958	1.386937968	0.9927415942	1.4891123960
1.39	1.4	0.9167600960	0.7143021321	1.833520192	1.428604251	2.1429063900
1.29	1.3	1.204747866	0.9922195666	2.409495727	1.984439132	2.9766586950
1.19	1.2	1.611049003	1.378422875	3.222098001	2.756845754	4.1352686350
1.0	1.1	1.201627674	2.302585093	2.403255350	4.605170186	6.9077552790
0.99	1.0	0.9929913427	1.820183721	1.985982685	3.640367454	5.4605511830
0.89	0.9	0.7387398626	1.228739195	1.477479723	2.457478388	3.6862175840
0.79	0.8	0.5717624256	0.9191815598	1.143524853	1.838363115	2.7575446790
0.69	0.7	0.4476694584	0.7073978929	0.8953389156	1.414795785	2.1221936840
0.59	0.6	0.3493417393	0.5465223031	0.6986834792	1.093044612	1.6395669210
0.49	0.5	0.2682625271	0.4170754083	0.5365250615	0.8341508114	1.2512262220
0.39	0.4	0.1995509325	0.3090144519	0.3991018678	0.6180289076	0.9270433575
0.29	0.3	0.1401410026	0.2164621604	0.2802819996	0.4329243274	0.6493864926
0.19	0.2	0.08797789363	0.1356752227	0.1759558000	0.2713504504	0.4070256696
0.09	0.1	0.04161616239	0.06411981435	0.08323232758	0.1282396264	0.1923594566
-0.01	0	-0.	-0.	-0.	-0.	-0

Table 4.29 Convergence rates of various methods for Problem 2

r	ω	$R(\mathbf{AOR})$	$R(\mathbf{PAOR})$	$R(\mathbf{RAOR})$	$R(\mathbf{RPAOR})$	$R(\mathbf{SRPAOR})$
2.0	2.01	-0.08263126830	-0.0881307565	-0.1652625322	-0.1762615126	-0.2643922728
1.99	2.0	-0.07338584384	-0.0783208550	-0.1467716957	-0.1566417115	-0.2349625652
1.82	1.83	0.09953535888	0.1051042653	0.1990707184	0.2102085344	0.3153127988
1.79	1.80	0.1336263954	0.1412879632	0.2672527912	0.2825759290	0.4238638951
1.69	1.7	0.2569645582	0.2724151632	0.5139291141	0.5448303242	0.8172454820
1.59	1.6	0.3985354065	0.4236925533	0.7970708235	0.8473851084	1.271077659
1.49	1.5	0.5644663510	0.6030015284	1.128932694	1.206003066	1.809004603
1.39	1.4	0.7646373316	0.8246534443	1.529274662	1.649306888	2.473960332
1.29	1.3	1.016565290	1.060188811	2.033130590	2.120377610	3.180566407
1.19	1.2	1.356127086	1.405091802	2.712254175	2.810183586	4.215275377
1.0	1.1	1.366061361	1.952805454	2.732122724	3.905610917	5.858416373
0.99	1.0	1.116545733	1.512525405	2.233091470	3.025050816	4.537576222
0.89	0.9	0.8456705071	1.087267502	1.691341014	2.174535009	3.261802512
0.79	0.8	0.6602529507	0.8308896472	1.320505896	1.661779286	2.492668935
0.69	0.7	0.5197298328	0.6466735707	1.039459662	1.293347139	1.940020701
0.59	0.6	0.4070788356	0.5031260578	0.8141576730	1.006252125	1.509378187
0.49	0.5	0.3134602656	0.3858083859	0.6269205377	0.7716167691	1.157425159
0.39	0.4	0.2336723760	0.2868482273	0.4673447510	0.5736964421	0.8605446726
0.29	0.3	0.1643880214	0.2014655403	0.3287760449	0.4029310866	0.6043966287
0.19	0.2	0.1033476883	0.1265346350	0.2066953811	0.2530692806	0.3796039190
0.09	0.1	0.04894574677	0.05989733437	0.09789149426	0.1197946755	0.1796920126
-0.01	0	-0.	-0.	-0.	-0.	-0.00000000009

Table 4.30 Convergence rates of various methods for Problem 3

r	ω	$R(\mathbf{AOR})$	$R(\mathbf{PAOR})$	$R(\mathbf{RAOR})$	$R(\mathbf{RPAOR})$	$R(\mathbf{SRPAOR})$
2.0	2.01	-0.0099336782	-0.1827165763	-0.0198673555	-0.4096480255	-0.5481497249
1.99	2.0	0.0000169026	-0.1727565224	0.00003380567	-0.3894732702	-0.5182695626
1.82	1.83	0.1863520292	0.01298138071	0.3727040604	-0.0134908755	0.03894413900
1.79	1.80	0.2231673211	0.04950476728	0.4463346420	0.06039537270	0.1485143026
1.69	1.7	0.3567042648	0.1815006399	0.7134085331	0.3273235948	0.5445019167
1.59	1.6	0.5108631871	0.3329363922	1.021726375	0.6334296616	0.9988091838
1.49	1.5	0.6931978918	0.5107314687	1.386395785	0.9927415942	1.532194405
1.39	1.4	0.9163646909	0.7263633828	1.832729375	1.428604251	2.179090144
1.29	1.3	1.204094897	1.001071705	2.408189794	1.984439132	3.003215106
1.19	1.2	1.609691524	1.381679782	3.219383046	2.756845754	4.145039355
1.0	1.1	1.679255643	2.222811241	3.358511290	4.605170186	6.668433724
0.99	1.0	1.318280360	2.419759829	2.636560725	3.640367454	7.259279504
0.89	0.9	0.9443041590	1.500380381	1.888608319	2.457478388	4.501141149
0.79	0.8	0.7200834126	1.100511228	1.440166814	1.838363115	3.301533682
0.69	0.7	0.5594545023	0.8395881464	1.118909000	1.414795785	2.518764436
0.59	0.6	0.4346008555	0.6456239352	0.8692017103	1.093044612	1.936871805
0.49	0.5	0.3327995999	0.4913723277	0.6655992026	0.8341508114	1.474116987
0.39	0.4	0.2471203052	0.3634796881	0.4942406117	0.6180289076	1.090439065
0.29	0.3	0.1733566743	0.2543811014	0.3467133440	0.4329243274	0.7631433003
0.19	0.2	0.1087588933	0.1593672741	0.2175177857	0.2713504504	0.4781018270
0.09	0.1	0.05142874135	0.07530431136	0.1028574823	0.1282396264	0.2259129266
-0.01	0	-0.	-0.	-0.	-0.	-0.

4.4 Discussion of Results

A basic requirement for convergence of any iterative method demands that the spectral radius of its iteration matrix be less than 1. More so, it is not just sufficient for a method to converge, it has to converge effectively. The computational effectiveness of a convergent iterative method is directly related to the magnitude of the spectral radius of the iterative method. The rate of convergence is best when the spectral radius is near zero and poorest when it is near 1.

Table 4.1 through Table 4.12 are the results of comparing the spectral radii of SRPAOR against its preceding methods, AOR, PAOR, RAOR and RPAOR, which it seeks to improve.

Table 4.1 displays the spectral radii of AOR against those of SRPAOR for various values of relaxation and acceleration parameters ω and r respectively applied to Problem 1. These values are carefully chosen between 2.01 and -0.01. The SRPAOR is shown to exhibit faster convergence than the AOR whenever it converges. Its range of convergence extends between $0 < r < \omega < 1.83$ wherein it attains optimum convergence when r and ω are 1.0 and 1.1 respectively. For the AOR, the range of convergence is $0 < r < \omega \leq 2.0$ with optimum values at $r = 1.19$ and $\omega = 1.2$. In Table 4.2 the results of the spectral radii of PAOR alongside those of SRPAOR for various values of the parameters (relaxation ω and acceleration r) applied to Problem 1 are exhibited. Again, the SRPAOR converges more rapidly than the AOR. The range of convergence extends between $0 < r < \omega \leq 1.80$ and the optimum convergence is attained at when $r = 1.0$ and $\omega = 1.1$ for the two methods. Table 4.3 is the results of comparing the spectral radii of RAOR with those of SRPAOR using various values of relaxation ω and acceleration r between 2.01 and -0.01 for Problem 1. As expected, values of $\rho(\text{SRPAOR})$ are in smaller magnitudes than those of $\rho(\text{RAOR})$, implying faster convergence provided the parameters fall within the range $0 < r < \omega < 1.83$, whose optimum values are obtained at $r = 1.0$ and $\omega = 1.1$. The AOR however, converges provided the parameter fall within the range $0 < r < \omega \leq 2.0$ with optimum values at $r = 1.19$ and $\omega = 1.2$. In Table 4.4 the spectral radii of RPAOR are compared with those of SRPAOR in solving Problem 1. Faster convergence is shown to be attained by the SRPAOR method due to its smaller spectral radii across values of the parameters. This convergence is attained for the range of values $0 < r < \omega < 1.83$ in both methods. An optimum convergence is reached when r and ω are 1.0 and 1.1 respectively.

Table 4.5 shows the spectral radii of AOR against those of SRPAOR for various values of relaxation and acceleration parameters ω and r respectively applied to Problem 2. The SRPAOR is shown to exhibit faster convergence than the AOR. Both methods have convergence range extending between $0 < r < \omega < 2$ wherein they attains optimum convergence when r and ω are 1.0 and 1.1 respectively. In Table 4.6 the results of the spectral radii of PAOR alongside those of SRPAOR for various values of the parameters (relaxation ω and acceleration r) applied to Problem 2 are exhibited. Again, the SRPAOR converges more rapidly than the AOR. The range of convergence extends between $0 < r < \omega \leq 2$ and the optimum convergence is attained at when $r = 1.0$ and $\omega = 1.1$ for the two methods.

Table 4.7 is the results of comparing the spectral radii of RAOR with those of SRPAOR using various values of relaxation ω and acceleration r between 2.01 and -0.01 for Problem 2. As expected, values of $\rho(\text{SRPAOR})$ are in smaller magnitudes than those of $\rho(\text{RAOR})$, implying faster convergence provided the parameters fall within the range $0 < r < \omega < 2$, whose optimum values are obtained at $r = 1.0$ and $\omega = 1.1$. In Table 4.8 the spectral radii of RPAOR are compared with those of SRPAOR in solving Problem 2. Faster convergence is shown to be attained by the SRPAOR method due to its smaller spectral radii across values of the parameters. This convergence is attained for the range of values $0 < r < \omega < 2$ in both methods. An optimum convergence is reached when r and ω are 1.0 and 1.1 respectively.

Table 4.9 displays the spectral radii of AOR against those of SRPAOR for various values of relaxation and acceleration parameters ω and r respectively applied to Problem 3. The SRPAOR is shown to exhibit faster convergence than the AOR. Its range of convergence extends between $0 < r < \omega < 2$ wherein it attains optimum convergence when r and ω are 0.99 and 1 respectively. For the AOR, the range of convergence is $0 < r < \omega \leq 2$ with optimum values at $r = 1$ and $\omega = 1.1$.

In Table 4.10 the results of the spectral radii of PAOR alongside those of SRPAOR for various values of the parameters (relaxation ω and acceleration r) applied to Problem 3 are exhibited. Again, the SRPAOR converges more rapidly than the AOR. The range of convergence extends between $0 < r < \omega < 2$ and the optimum convergence is attained at when $r = 0.99$ and $\omega = 1$ for the two methods. Table 4.11 is the results of comparing the spectral radii of RAOR with those of SRPAOR for Problem 3. Values of $\rho(SRPAOR)$ are in smaller magnitudes than those of $\rho(\text{RAOR})$, implying faster convergence provided the parameters fall within the range $0 < r < \omega < 2$, whose optimum values are obtained at $r = 0.99$ and $\omega = 1$. The AOR however, converges provided the parameter fall within the range $0 < r < \omega \leq 2$ with optimum values at $r = 1$ and $\omega = 1.1$. In Table 4.12 the spectral radii of RPAOR are compared with those of SRPAOR in solving Problem 3. Faster convergence is shown to be attained by the SRPAOR method due to its smaller spectral radii across values of the parameters. This convergence is attained for the range of values $0 < r < \omega < 2$ with optimum convergence attained when r and ω are 0.99 and 1 respectively in the case of SRPAOR. For AOR however, convergence is attained in the range of values $0 < r < \omega < 1.83$ with optimum convergence attained when r and ω are 1 and 1.1 respectively.

The rate of convergence of an iterative method is the number $R(G) = -\log \rho(G)$, and is a measure of how rapidly a method is convergent, depending on the choice of the parameters r and ω. Using Maple software, the rates of convergence of the AOR, PAOR, RAOR RPAOR and SRPAOR are computed and compared. The results are presented in Table 4.13 through Table 4.24. Table 4.13 compares the rate of convergence of SRPAOR with AOR for various values of the parameters when applied to Problem 1. It reveals that the SRPAOR converges five and a half times faster than the AOR. Table 4.14 compares the rate of convergence of SRPAOR with PAOR for Problem 1. Results show that SRPAOR is faster than PAOR by a factor of three. In Table 4.15 a comparison of convergence rates of SRPAOR and RAOR for Problem 1 is undertaken. It shows that SRPAOR converges almost three times more rapidly than RAOR. Table 4.16 is the result of comparing the convergence rates of SRPAOR and RPAOR for Problem 1, where it is seen than SRPAOR is one a half times as fast as RPAOR.

Table 4.17 compares the rate of convergence of SRPAOR with AOR for various values of the parameters when applied to Problem 2. It reveals that the SRPAOR converges four times faster than the AOR. Table 4.18 compares the rate of convergence of SRPAOR with PAOR for Problem 2. Results show that SRPAOR is faster than PAOR by a factor of three. In Table 4.19 a comparison of convergence rates of SRPAOR and RAOR for Problem 2 is undertaken. It shows that SRPAOR converges two times more rapidly than RAOR. Table 4.20 is the result of comparing the convergence rates of SRPAOR and RPAOR for Problem 2, where it is seen than SRPAOR is one a half time as fast as RPAOR.

Table 4.21 compares the rate of convergence of SRPAOR with AOR for various values of the parameters when applied to Problem 3. It reveals that the SRPAOR converges five and a half times faster than the AOR. Table 4.22 compares the rate of convergence of SRPAOR with PAOR for Problem 3. Results show that SRPAOR is faster than PAOR by a factor of three. In Table 4.23 a comparison of convergence rates of SRPAOR and RAOR for Problem 3 is undertaken. It shows that SRPAOR converges two and a half times more rapidly than RAOR. Table 4.24 is the result of comparing the convergence rates of SRPAOR and RPAOR for Problem 3, where it is seen than SRPAOR is two times as fast as RPAOR.

Table 4.25, Table 4.26 and Table 4.27 compare the spectral radii of all the five methods in solving Problem 1, Problem 2 and Problem 3 respectively. A common trend in the results reveal that $\rho(\mathrm{SRPAOR}) < \rho(\mathrm{RPAOR}) < \rho(\mathrm{RAOR}) < \rho(\mathrm{PAOR}) < \rho(\mathrm{AOR})$, indicating that the proposed SRPAOR iterative converges faster than all the methods.

Table 4.28, Table 4.29 and Table 4.30 compare the rates of convergence of all the five methods in solving Problem 1, Problem 2 and Problem 3 respectively. At optimal solution of Problem 1, SRPAOR method is shown to converge faster than AOR, PAOR, RAOR and RPAOR by a factor of 5.75, 3, 2.87 and 1.5 respectively. While for Problem 2, SRPAOR method converges more rapidly than AOR, PAOR, RAOR and RPAOR by a factor of 4.3, 3, 2.14 and 1.5 respectively. And in Problem 3, SRPAOR method converges faster than AOR, PAOR, RAOR and RPAOR by a factor of 5.5, 3, 2.75 and 2 respectively.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

A second refinement of preconditioned Accelerated Overrelaxation (AOR) iterative method has been proposed. This method christened Second Refinement of Preconditioned Accelerated Overrelaxation (SRPAOR) method involves the successive application of preconditioning and second refinement techniques in improving the convergence rates of iterative methods towards the solution of linear algebraic systems.

A new preconditioner was derived and applied to the AOR method resulting in a preconditioned linear system. Convergence theorems were advanced and the resulting preconditioned AOR method was found to be convergent. A second refinement formulation of the preconditioned AOR method was further introduced and proven to be convergent as well.

In order to validate the results of theoretical convergence several numerical experiments were conducted and results compared with other methods. It was established that the SRPAOR converges almost 6 times faster than the classical AOR.

5.2 Recommendations

The following recommendations for further research are proposed:

1. A formula for obtaining the optimum relaxation and acceleration parameters for SRPAOR could be derived.
2. More general class of matrices, and not just irreducibly diagonally dominant $L-$ matrices could be accommodated.

5.3 Contribution to Knowledge

The following contributions have been made:

1. The techniques of preconditioning and second refinement have been exploited to introduce a new approach towards improving the rate of convergence of the AOR iterative method in solving linear system of equations.
2. The new method can be applied to solve irreducibly diagonally dominant $L-$matrix linear systems provided $0 < a_{12}a_{21} < 1$, $0 < a_{12}a_{21} + a_{23}a_{32} < 1$, $0 < a_{1i}a_{i1} + a_{i-1,i}a_{i,i-1} + a_{i,i+1}a_{i+1,i} < 1$ $(i = 3(1)n-1)$ and $0 < a_{1n}a_{n1} + a_{n-1,n}a_{n,n-1} < 1$.
3. The SRPAOR method converges faster than AOR, PAOR, RAOR and RPAOR by a factor of 5.75, 3, 2.87 and 1.5 respectively.
4. Optimum convergence is attained when $r = 1.0, \omega = 1.1$ and when $r = 0.99, \omega = 1.0$

REFERENCES

Abdullahi, 1., & Ndanusa, A. (2020). A new modified preconditioned accelerated overrelaxation (AOR) iterative method for L-matrix linear algebraic systems. *Science World Journal*, 15(2),

Adil, D., Kyng, R., Peng, R., & Sachdeva, S. (2019). Iterative refinement for ep-norm regression. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (pp. 1405-1424). *Society for Industrial and Applied Mathematics, 5*(3), 462 – 479.

Adsuara,J. E Cordero-Carri6n, L, Cerdå-Durån, P., & Aloy, M. A. (2016). Scheduled relaxation Jacobi method: improvements and applications. *Journal of Computational Physics*, 321, 369-413.

Ames, W. F. (1977). *Numerical methods for partial differential equations*. (2nd ed.). Great Britain: Thomas Nelson & Sons Ltd. pp 365

Assefa, W. L., & Teklehaymanot, A. W. (2021). Second refinement of accelerated over relaxation method for the Solution of Linear System. *Pure and Applied Mathematics Journal*, 10(1), 32 — 37.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., & Van der Vorst, H. (1994). Templates for the solution of linear systems: building blocks for iterative methods. Philadelphia, PA: SIAM. Pp. 124.

Byrne, C. L. (2008). *Applied iterative methods*. Wellesley: AK Peters.

Dehghan, M. & Hajarian, M. (2009). Improving preconditioned SOR-type iterative methods for L —matrices. *International Journal for Numerical Methods in Biomedical Engineering*, 27, 774-784.

Eneyew, T. K., Awgichew, G., Haile, E. & Abie, G. D. (2019). Second refinement of Jacobi iterative method for solving linear system of equations. *International Journal of Computing Science and Applied Mathematics*, 5(2), 41-47.

Eneyew, T. K., Awgichew, G., Haile, E. & Abie, G. D. (2020a). Second refinement of Gauss- Seidel iterative method for solving linear system of equations. *Ethiopian Journal of Science and Technology*, 13(1), 1-15.

Eneyew, T. K., Awgichew, G., Haile, E. & Abie, G. D. (2020b). Second refinement of generalized Jacobi iterative method for solving linear system of equations. *Journal of the Nigerian Mathematical Society*, 39(1), 117-133.

Faruk, A. I., & Ndanusa, A. (2019). Improvements of successive overrelaxation (SOR) iterative method for L-Matrices. *Savanna Journal of Basic and Applied Sciences*, 1(8): 218-223.

Gunawardena, A. D,, Jain, S. K. & Snyder, L. (1991). Modified iterative methods for consistent linear systems. *Linear Algebra and its Applications, 154*, 123-143.

Hadjidimos, A. (1978). Accelerated Overrelaxation Method. *Mathematics of Computation*, 32(141), 149— 157.

Kebede, T. (2017). Second degree refinement Jacobi iteration method for solving system of linear equation. *International Journal of Computing Science and Applied Mathematics*, 3(1), 5-10.

Kohno, T., Kotakemori, H., Niki, H. & Usui, M. (1997). Improving modified Gauss—Seidel method for Z-matrices. *Linear Algebra Appt.*, 267, 1 13—123.

Kotakemori, H., Harada, K., Morimoto, M. & Niki, H. (2002). A comparison theorem for the iterative method with the preconditioner (I + Smax). *Journal of Computational and Applied, Mathematics*, 145, 373-378.

Kotakemori, H., Niki, H. & Okamoto, N. (1996). Accelerated iteration method for Zmatrices. Journal of Computational and Applied Mathematics, 75, 87—97.

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

Milaszewicz, J. P. (1987). Improving jacobi and gauss-seidel iterations. *Linear Algebra and Its Applications*, *93*, 161-170.

Morimoto, M., Harada, K., Sakakihara, M. & Sawami, H. (2003). The Gauss—Seidel iterative method with the preconditioning matrix (I + Smax + Sm). *Japan J. Indust. Appl. Math*., 21, 25-34.

Naumov, M. (2011). Parallel solution of sparse triangular linear systems in the preconditioned iterative methods on the GPU. NVIDIA Corp., Westford, MA, USA, Tech. Rep. NVR-2011, 1.

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

Ndanusa, A. (2020). Convergence of Preconditioned Gauss-Seidel Iterative Method for L — Matrices. *Communication in Physical Sciences*, 6(1), 803 — 808.

Ndanusa, A., David, B. E., Ayantola, B. B. & Abubakar, A. W. (2020). Improving preconditioned Gauss-Seidel iterative method for L—matrices. FUD&L4 Journal ofSciences, 4(1), 453 — 459.

Niki, H., Harada, K., Morimoto, M. & Sakakihara, M. (2004). The survey of preconditioners used for accelerating the rate of convergemce in the Gauss-Seidel method. *Journal of Computational and Applied Mathematics*, 164, 587-600.

Noor, M. A., Noor, K. L, & Aftab, K, (2012). Some new iterative methods for solving nonlinear equations. *World Applied Sciences Journal*, 20(6), 870-874.

Saad, Y. (2000). Iterative methods for sparse linear systems, (2nd ed). Philadelphia: SIAM. pp 447.

Varga, R. S. (1962). *Matrix Iterative Analysis.* Englewood Cliffs, New York: Prentice Hall.

Vatti, V. B. K. Sri, R. & Mylapalli, M. S. K. (2018). A refinement of accelerated over relaxation method for the solution of linear systems. *International Journal of Pure and Applied Mathematics,* 8(18), 1571-1577.

Vatti, V. B. K., Rao, G. C. & Pai, S. S. (2020). Reaccelerated over relaxation (ROR) method. *Bulletin of the International Mathematical Virtual Institute*, 10(2), 315324.

Young, D. M. (1954). Iterative methods for solving partial difference equations ofelliptic type. *Transactions of the American Mathematical Society*, 76(1), 92—111

Youssef, I. K. & Taha, A. A. (2013). On the modified successive overrelaxation method. *Applied Mathematics and Computation,* 219, 4601—4613.