REFINEMENT AND SECOND DEGREE ITERATION OF GENERALIZED SUCCESSIVE OVERRELAXATION METHODS FOR SOLVING LARGE SYSTEM OF LINEAR EQUATIONS

A THESIS SUBMITED TO THE DEPARTMENT OF MATHEMATICS, JIMMA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE REQUIREMENTS OF THE DEGREE OF MASTERS OF SCIENCE IN MATHEMATICS (NUMERICAL ANALYSIS)

BY:

FIREW HAILU

UNDER THE SUPERVISION OF

GENANEW GOFE (PHD)

HAILU MULETA (MSC)

JUNE, 2016

JIMMA, ETHIOPIA

Declaration

I undersigned declare that this thesis entitled **“Refinement of Generalized Successive Over Relaxation and its Second Degree Iteration for Solving Large System of Linear Equations”** is my own original work and it has not been submitted for the award of any academic degree or the like in any other institution or university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

Name: FIREW HAILU

Signature ______________

Date _______________________

The work has been done under the supervision of the advisor:

Name: Genanew Gofe (PhD)

Signature: ___________________

Date _________________________

Acknowledgment

Above all, I would like to praise the almighty God for helping me in all direction to successful in this thesis development. Next, I would like to express my deepest gratitude to my advisor, Genanew Gofe (PhD). I am grateful for his encouragement, persistent help, constructive suggestions and guidance throughout the period of my thesis work.

My heartfelt gratitude also goes to my co-advisor, Hailu Muleta (MSC), for his effective cooperation in coding the software and unlimited support and immediate responses in the work of my thesis.

I am also indebted to extend my thanks to my wife W/ro Tigist Fetene, my daughter Henon Firew and my family for their moral, financial support and valuable advice.

I am also glad to express my heartfelt thanks to friends, collogues who gave consistent guidance and assistance all the way throughout.

Firew Hailu

Table of Content

List of Tables

Acronyms

SDGGS- Second Degree Generalized Gauss- Seidel Method

RSOR- Refinement of Successive Over Relaxation Method

GSOR- Generalized Successive Over Relaxation Method

RGSOR- Refinement of Generalized Successive Over Relaxation Method

SDGSOR- Second Degree Generalized Successive Over Relaxation Method

Abstract

In this thesis, we present a refinement and second degree iteration of generalized successive over relaxation methods for solving large system of linear equations and their convergence properties are discussed.

Some numerical examples are considered to show the efficiency of the proposed methods. The present methods are also compared against the other methods based on the number of iterations, computational running time and accuracy of each method. The results presented in tables show that Refinement and Second Degree Iteration of Generalized Successive Over Relaxation are more efficient than the other methods considered in this thesis.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The limitations of analytical methods in practical applications are led mathematicians and other scientist to evolve numerical methods. It is clear that exact methods often fail in drawing reasonable inference from a given set of tabulated data or in finding solutions for different equations. There are many more situations where analytical methods are unable to produce desirable results. Even if analytical solutions are available, these are not amenable to direct numerical interpretations (Goyal, 2007).

The ultimate aim of numerical analysis is therefore, to provide efficient methods for obtaining useful solutions to such problems and extracting useful information from available solutions.

Numerical analysis is the branch of mathematics concerned with the theoretical foundation of numerical algorithms for the solution of problems arising in scientific applications. The subject addresses a variety of questions ranging from the approximation of functions and integrals to the approximate solution of algebraic, transcendental, differential and integral equations with particular emphasis on the convergence, accuracy, efficiency and reliability of numerical algorithms (Lay, 1994).

A system of linear equations is one of the important topics that studied in numerical analysis. It is one of the methods in the field of computational mathematics which plays a vital role in the numerical solution of mathematical problems.

Many practical problems can be reduced to system of linear equations $Ax = b,$ where A is known non singular matrix, b is known vector and x is unknown vector. This type of equation plays a prominent role in finance, industry, economics, engineering, physics, chemistry, computer sciences (Iqbal, 2012).

The system of linear equations can be solved using both direct and iterative methods. The best known direct method is Gauss elimination method (Grear., 2011 and Strassen, 1969). Turing introduced LU decomposition of a matrix for solving system of linear equations. Choleski decomposed the matrix A into the product of lower triangular matrix and their transpose.

The Choleski method is more efficient than LU decomposition for solving symmetric and positive definite linear system (Burden and Faires, 2006).

Direct Methods produce new matrices at each step and therefore they are sensitive to rounding errors. And they are not efficient in terms of computer storage so these methods are prohibitively expensive for large systems. For these reasons, researchers have long since move to iterative methods for solving such system of equations. Iterative methods are very efficient when they are applied to large and sparse systems of equations that arise in practical problems (Iqbal, 2012).

The Iterative method is a technique that starts with an initial guess and attempts to solve a problem or a solution of a linear system of equation by finding successive approximations to the solution. Iterative methods are suitable for solving linear equations when the number of equations in a system is very large, and they are fast and simple to use when the coefficient matrix is sparse. That is iterative methods are very much effective regarding the time requirements.

The major factors to be considered in comparing different numerical methods are the accuracy of the numerical solutions and its computational time (Bedet et al., 1975). Further it is indicated that the comparison of numerical methods is not so simple because their performance may depend on the characteristic of the problem at hand (Bedet et al., 1975 and Salkuyeh, 2007). It should also be noted that there are other factors to be considered such as stability, proof against run –time error, and so on, which are being considered in most of the MATLAB built-in routines (Atkinson, 1978).

Different methods are being used for the solution of system of linear equations. There is no single method that is best for all situations. These methods should be determined according to sped and accuracy (Saeed, 2008).

Iterative refinement of system of linear equations is defined as a process by which a first computed solution can sometimes be improved to yield a more accurate solution that could be continued until the residuals stabilize at or very near to zero. In practice one step of iterative refinement usually suffices if iterative refinement fails to stabilize it is likely that meaningful solutions cannot be obtained using conventional computing method.

In this comparison of the indirect methods the criteria considered are, number of iterations, computational running time and accuracy of the solution.

1.2 Statement of the Problem

The numerical solution of systems of linear equations enter at some stage in almost all applications in many fields of science, engineering and Technology. The increasing of desire for the numerical solutions to mathematical problems, which are more difficult or impossible to solve explicitly, has become the present- day scientific research. The numerical method used to find approximate solution of systems of linear equations has an impressive importance due to its wide applications in scientific and engineering researchers. So, iterative method is one of the methods used to find approximate solution of system of linear equations.

Various methods have been introduced to solve systems of linear equations by many authors like (Salkuyeh, 2007,Kalambi, 2008, and Kumer and Genanew, 2011and Kumer 2015).There is no single method that is best for all situations. These methods should be determined according to speed and accuracy. Speed is an important factor in solving large systems of equations because the operation cost involved is very large. Another issue in the accuracy problem for the solution rounding off errors involved in executing these computations. Thus the intention of this study is to establish a numerical method that approximates the solution of linear system of Equation by providing the accuracy and efficiency of the numerical solution.

Therefore, this research is intended to answer the following basic research questions:

1. What are the procedures and techniques that can be followed to develop the methods RGSOR and SDGSOR?
2. To what extent the present methods converge?
3. To what extent the present methods approximate the exact solution?
4. What is the advantage of the present methods over the other?

1.3. Objective of the Study

1.3.1. General Objective

The general objective of this study is to present refinement of Generalized successive over relaxation and its second degree iteration methods for solving large system of linear equations.

1.3.2. Specific Objective

The specific objectives of the study are:

- To describe procedures and techniques followed to develop the RGSOR and SDGSOR methods.
- To establish the convergence of the present methods by means of error analysis.
- To compare the accuracy of the present methods with exact solutions of system of linear equations.
- To compare the advantage of the present methods over the other.

1.4. Significance of the Study

The outcomes of this work may have the following importance:

- It provides some background information for other researchers who want to work on similar topics.
- Further, this research would be useful for the graduate program of the department and enhances the research skill and scientific communication of the researcher too.

1.5. Delimitation of the Study

This study is delimited to the indirect methods for solving system of linear equations. In particular, it is delimited to “Refinement of Generalized Successive Over-relaxation method” and “Second Degree Generalized Successive Over-relaxation method” among many other indirect schemes for solving system of linear equations

1.6. Definitions of Basic Operational Terms

Definition1.6.1 A banded matrix is a square matrix with zeros after “m” elements above and below the main diagonal, where m is less than the size of the matrix. *i.e* if the matrix is $N \times N$ then $m < N$.

Definition 1.6.2 The term “iteration method” refers to a wide range of techniques that use successive approximations to obtain more accurate solution to a linear system at each step by beginning with initial approximation, these methods modify the components of the approximation, until convergence is achieved.

Definition 1.6.3 A matrix A is called sparse if many of its entries are zero. Otherwise, A is called dense or full.

Definition 1.6.4 A matrix **A** is said to be reducible, if there exists a permutation matrix P such that PAP^T is a block upper triangular matrix, otherwise it is an irreducible.

Definition 1.6.5 A matrix A is said to be strictly diagonally dominant (SDD) if

$$|a_{ii}| > \sum_{j=1,\, j\neq i}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n.$$

and is said to be weakly diagonally dominant (WDD) if

$$|a_{ii}| \geq \sum_{j=1,\, j\neq i}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n.$$

Definition 1.6.6 A matrix A is said to be irreducibly weakly diagonally dominant (IWDD) if A is WDD and irreducible.

Definition 1.6.7.A real matrix A is said to be positive definite or positive real if $(Ax, x) > 0$,

$\forall x \in \Re^N, x \neq 0$

Definition1.6.8. Iterative refinement is a process by which a first computed solution can sometimes be improved to yield a more accurate solution.

CHAPTER TWO

LITERATURE REVIEW

The approximate methods for solving system of linear equations makes it possible to obtain the values of the roots system with the specified accuracy as the limit of the sequence of some vectors. This process of constructing such a sequence is known as iteration. Unlike the direct methods, which attempt to calculate an exact solution in a finite number of operations, indirect methods start with an initial approximation and generate successively improved approximations in an infinite sequence whose limit is the exact solution (Yarlett, 1980). In practical situation, this has more advantage because the direct solution will be subject to rounding off errors. A code is more efficient if it solves problems in less CPU times. However, this criterion is problem dependent, and hence it is necessary to test efficiency by considering problem (Hull.et al., 1972).

The efficiency of any method will be judged by two criteria:

i. How fast it is? That is how many operations are involved?
ii. How accurate is the computer solution?

Because of the large amount of computations required to linear equations for large system, the need to answer the first question is necessary. The need to answer the second, arise because small round off error may cause errors in the computer solution out of all proportions to their size. Furthermore because of the large number of operations involved in solving higher order system, the potential round off errors could cause substantial loss of accuracy (Kalambi, 2008).

(Bedet et al., 1975 and Salkuyeh, 2007) indicated that it is important to note that the evaluation/ comparison of numerical methods is not so simple because their performances may depend on the characteristic of the problem at hand. It should also be noted that there are other factors to be considered such as stability, proof against run-time, error and so on, which are being considered in most of the MATLAB built-in routines (Censor, 1981 and Amos, 2015).

Performance actually depends on several factors the computation time taken for one iteration of the algorithm, the time step for one iteration which represents the time discretization required to reach a given accuracy or numerical stability for a given method, the desired accuracy of the method, the numerical stability of the method which also limits the time step for a given method (Volino, and Thalmann, 2000).

The direct methods of solving linear equations are known to have their difficulties. For example the problem with Gauss elimination system of approach lies in control of the accumulation of rounding errors (Turner, 1989). To get rid of these problems many authors like (Kalambi, 2008 and Rajasekaran, 1992) were encouraged to investigate solutions of linear equations by indirect methods. Most researchers dealt with the iterative methods for solving linear systems of equations and inequalities for sparse Matrices.

Various methods have been developed to solve systems of linear equations by many authors. There is no single method that is best for all situations. These methods should be determined according to their speed and accuracy (Saeed, 2008).

In this thesis we present two indirect methods namely, Refinement of Generalized Successive Over Relaxation schemes and Second Degree Generalized Successive Over Relaxation methods for solving large system of linear equations and explain the efficiency of the present methods in terms of number of iteration, required time to converge and accuracy of the result.

2.1. Successive Over Relaxation Method

The SOR method seems to have appeared in the 1930's (Southwell, 1946 as cited in Hadjidimos, 2000). However, formally its theory was established almost simultaneously by Frankel and Young's (Frankel, 1950 and Young, 1950, as cited by Hadjidimos, 2000).

The Gauss- Seidel iteration was the starting point for the successive over relaxation method which dominated much of the literature on Iterative methods for a big part of the second half of the 19^{th} century (Saad, 2000). The successive over relaxation method, is devised 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. The idea is to choose a value for optimum relaxation factor that will accelerate the rate of convergence of iterates to the solution (Kalambi, 2008).

According to (Saad, 2000), the blossoming of Successive over relaxation techniques seems to have been initiated by the PhD work of David Young. Young introduced important notions such as consistent ordering and property A, which he used for formulation of an elegant theory for the convergence of these methods. Generalizations of Young's result to other relevant classes of matrices were due to Varga who published his book on matrix iterative analysis in 1962.

It covered important notions such as regular splitting a rather a complete theory of Stieljes and M-matrices and a treatment of semi-iterative methods including the Chebyshev Semi iterative method.

The accelerated Gauss Seidel method has motivated important developments in the theory of matrix linear algebra. In particular relevant properties for M-matrices introduced by Ostrowski were uncovered and convergence result for so called regular splitting, introduced by Varga were established. A corner stone in the convergence theory was the theorem of Stein-Rosbenberg (Stein-Rosbenberg , 1948 as cited by Saad, 2000) which proved relation between the asymptotic rate of convergence for the successive over relaxation methods including the Gauss Seidel and Gauss Jacobi method.

Sufficient conditions for convergence of the SOR methods were given by theorem Ostrowski and Reich. Lower bounds for the spectral radius of the SOR iteration matrix were derived by Kahan. This together provided the basis for a theory for iterative methods published in Varga book from which many methods emerged (Saad, 2000).

2.2. Iterative Refinement

The technique of iterative refinement for improving the computed solution to a linear system were probably first used in a computer program by Wilkinson in 1948, during the design and building of the ACE computer at the National Physical Laboratory (Wilkinson, 1948 as cited by Higham, 1997). Iterative refinement has achieved wide use ever since, and is exploited, for example, by most of the linear system expert drivers in LAPACK (Anderson *et al.* 1995 as cited by Higham, 1997).

The refinement process for a computed solution x to $Ax = b$, where A is $n \times n$ is nonsingular, is simple to describe: compute the residual $r = b - Ax$ solve the system $Ad = r$ for the correction d, and form the updated solution $y = x + d$. If there is not a sufficient improvement in passing from x to y the process can be repeated, with x replaced by y (Higham, 1997).

2.3. Refinement of SOR Method

It is a modification of SOR iterative method which is presented by (Kumar, 2015). It is an iterative method used to solve system of linear equations. It solves a matrix whose main diagonal elements are non zero and row strictly diagonally dominant. Proceeding with the SOR method and supposing that the equations are examined in a sequence and also the previously computed results are used as soon as they are available, we get the Refinement of SOR method. We start with an initial approximation and substitute the solution in the given equation. We shall use the most recent value in this method. The iteration process is to be continued until the relative error is less than the pre-specified tolerance. If A is a row strictly diagonally dominant matrix, then the SOR method converges for any arbitrary choice of the initial approximation. Accordingly, the refinement of SOR method converges faster than the SOR method when SOR method is convergent.

2.4. Generalized SOR Method

(Salkuyeh, 2007) introduced generalized SOR method which is more efficient than conventional SOR method. Like SOR method, it is also an iterative method used for the solution of linear system of equations. If the matrix is symmetric positive definite the method is much faster than conventional SOR iterative method and it is fast and simple to use when the coefficient matrix is sparse as well as accuracy is developed in every iteration that is continue the iteration process until the relative error is less than pre specified error of tolerance.

CHAPTER THREE

METHODOLOGY

3.1. Study Area and Period

The study has been conducted at Jimma University College of Natural sciences Department of Mathematics in 2015/2016 Academic year.

3.2. Source of Information

The data has been collected from the relevant source of information to achieve the objective of the study and experimental results obtained by using MATLAB software to validate the present methods.

3.3. Study Design

This study employed mixed-design (documentary review design and experimental design) for solving system of linear equations. Since the methods are coded and run using MATLAB software by properly inserting the problems so that numerical results are automatically generated. All algorithms have been made in the same condition, which use the same processor, having the same memory size, the same operating system, and using the same problems. The processor used is Intel(R) core (TM) i3-31110M CPU @240GHZ 2.40GHZ with 4GM memory (RAM),with 64 bits operating system (Window 7 home premium). The language program used is MATLAB version 7.60(R2008a)

Two major programs (code) have been written to solve system of linear equations using RGSOR and SDGSOR methods. The code contains equation definition line, input arguments, commands (equation body), and output arguments which are written in the script file of MATLAB. The equation definition line contains type of numerical method, equation, left hand equation and right hand column vector, initial value and number of steps. The input arguments are written in order to insert the values after the code are saved and debugged using MATLAB.

In the equation body the formula for column vector, the formula for methods, formula for iteration number and formula for run time have been coded. In the output argument approximate notation of the out puts such as the iteration number (k in our case), the corresponding numerical value of the determined column vector (in our case y), exact values (in our case x), error (in our case e) and the elapsed time (t) have been written.

3.4. Study Procedures

Important materials and data for the study have been collected using documentary analysis as an instrument. In order to achieve the intended objectives the study follows the following mathematical steps.

- Step 1, write the system of equations $Ax = b$ in the form of

$$A = T_m + E_m + F_m$$

where $A = (a_{ij})$ is an $n \times n$ non singular matrix , $T_m = (t_{ij})$ is a banded matrix with band length of *2m+1* where m is less than the size of the matrix. E_m and F_m are strictly lower and strictly upper triangular parts of $A - T_m$ respectively.

- Step 2. Deriving Iterative refinement formula for GSOR and SDGSOR.
- Step 3. Proving the convergence of the proposed method.
- Step 4. Validating the proposed method using numerical examples
- Step 5. Writing MATLAB code to compare the numerical examples in step2 to determine the efficiency of the method.

3.5. Ethical Consideration

The researcher takes care of ethical considerations through official letter support from the department.

CHAPTER FOUR

DESCRIPTION OF METHODS, NUMERICAL RESULTS AND DISCUSSIONS

4.1 Description of the RGSOR Method

Consider a system of linear equations

$$Ax = b \tag{4.1}$$

where, A is an nxn nonsingular coefficient matrix, b is a column vector and x is solution vectors to be determined .

Based on splitting of the matrix A (Young, 1971) as

$$A = D + L + U \tag{4.2}$$

where, D is the diagonal matrix of order n $(a_{ij} = 0, \, i \neq j)$ and L and U are strictly lower and upper triangular matrix of order n with zero diagonal entries, respectively.

The Jacobi and Gauss Seidel methods for solving Eq. (4.1) are defined as

$$x^{(k+1)} = -\left(D\right)^{-1}\left(L+U\right)x^{(k)} + D^{-1}b$$

$$x^{(k+1)} = -\left(D+L\right)^{-1}Ux^{(k)} + (D+L)^{-1}b$$

To obtain successive over relaxation method multiply both sides of the equation Eq. (4.1) by ω, where ω is optimum relaxation parameter, as (Salkuyeh, 2007)

$$\omega Ax = \omega b.$$

Then the coefficient matrix ωA is decomposed in the form

$$\omega\left(D+L+U\right)x = \omega b$$

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

$$\left(D+\omega L\right)x = \left(\left(1-\omega\right)D - \omega U\right)x + \omega b$$

$$x = \left(D+\omega L\right)^{-1}\left(\left(1-\omega\right)D - \omega U\right)x + \omega\left(D+\omega L\right)^{-1}b$$

Then the iterative method of SOR method for solution of Eq. (4.1) is defined as (Young, 1971)

$$x^{k+1} = \left(D+\omega L\right)^{-1}\left(\left(1-\omega\right)D-\omega U\right)x^{k} + \omega\left(D+\omega L\right)^{-1} b$$

$$x^{k+1} = B_{SOR}x^{k} + C \tag{4.3}$$

where $B_{SOR} = \left(D+\omega L\right)^{-1}\left(\left(1-\omega\right)D-\omega U\right)$ is the iteration matrix for the SOR method and $C = \omega\left(D+\omega L\right)^{-1} b$ is the corresponding column vector.

To solve Eq. (4.1) we have given nonsingular matrix A and a known vector *b*, the problem is to find the unknown vector x, we start with an initial approximation $x^{(0)}$ to the exact solution x and produces a sequence of approximation $\left\{x^{k}\right\}_{k=0}^{\infty}$ that converges to x. Based on Eq. (4.2) iterative methods for solving Eq. (4.1) can be written in the form

$$x = Bx + C \tag{4.4}$$

for some $n\times n$ iteration matrix *B* that depends on *A*, and *C* is a column vector, where the iteration matrix *B* and a column vector represent different values in different methods .The sequence of approximate solution vector is generated by computing

$$x^{(k+1)} = Bx^{(k)} + C \quad \text{where } k = 0,1,2,\ldots \tag{4.5}$$

$x^{(K+1)} and\ x^{(K)}$ are approximate for x at $\left(k+1\right)^{th}$ *and* k^{th} iteration respectively in the limiting case when $k \to \infty$, $x^{(K)}$ convergences to the exact solution which is given by

$$x = A^{-1}b \tag{4.6}$$

Generalized Successive Over-Relaxation Method (GSOR)

Consider the linear system of Eq. (4.1) and splitting made by (Salkuyeh, 2007) as

$$A = T_m + E_m + F_m \tag{4.7}$$

where $A = (a_{ij})$ is an $n \; x \; n$ non singular matrix and $T_m = (t_{ij})$ is a banded matrix with band length *2m+1* is defined as follows.

$$t_{ij} = \begin{cases} a_{ij, |j-i| \le m} \\ 0, \; otherwise \end{cases}$$

where E_m and F_m are strictly lower and strictly upper triangular parts of $A - T_m$ respectively and they are defined as follows

$$T_m = \begin{bmatrix} a_{11} & \cdots & a_{1,m+1} & \\ \vdots & \ddots & \ddots & \\ a_{m+1,1} & \ddots & \ddots & a_{n-m,n} \\ \ddots & & \ddots & \vdots \\ & a_{n,n-m} & \cdots & a_{n,n} \end{bmatrix}$$

$$E_m = \begin{bmatrix} & & \\ & & \\ a_{m+2,1} & & \\ \vdots & \ddots & \\ a_{n,1} & \cdots & a_{n-m-1,n} \end{bmatrix} \qquad F_m = \begin{bmatrix} a_{1,m+2} & \cdots & a_{1,n} \\ & \ddots & \vdots \\ & & a_{n,n-m-1} \\ & & \end{bmatrix}$$

Then the generalized successive over relaxation method for solving Eq. (4.1) is given by (Salkuyeh, 2007) as

$$x^{(k+1)} = \left(T_m + \omega E_m\right)^{-1}\left(\left(1-\omega\right)\mathrm{T_m} - \omega\mathrm{F_m}\right)\mathrm{x}^{(k)} + (\mathrm{T_m} + \omega\mathrm{E_m})^{-1}\omega\mathrm{b} \tag{4.8}$$

$$x^{k+1} = B_{GSOR}x^k + C_s \quad \text{, where } k = 0,1,2,\ldots$$

$$B_{GSOR} = \left(T_m + \omega E_m\right)^{-1}\left(\left(1-\omega\right)\mathrm{T_m} - \omega\mathrm{F_m}\right) \tag{4.9}$$

Eq. (4.9) is the generalized successive over relaxation iteration matrix and

$$C_s = (\mathrm{T_m} + \omega\mathrm{E_m})^{-1}\omega\mathrm{b} \tag{4.10}$$

its iteration vector.

Refinement of Generalized Successive Over Relaxation in Matrix form

Putting Eq. (4.7) in to Eq. (4.1), we get

$$\left(T_m + E_m + F_m\right)x = b$$

Multiplying both sides by ω we obtain

$$\omega\left(T_m + E_m + F_m\right)x = \omega b$$

$$\left[\left(T_m + \omega E_m\right) - \left((1-\omega)T_m - \omega F_m\right)\right]x = \omega b$$

$$\left(T_m + \omega E_m\right)x = \left((1-\omega)T_m - \omega F_m\right)x + \omega b$$

$$\left(T_m + \omega E_m\right)x = \left((1-\omega)T_m + \omega\left(T_m + E_m - A\right)\right)x + \omega b$$

$$\left(T_m + \omega E_m\right)x = \left((1-\omega)T_m + \omega\left(T_m + E_m\right)\right)x + \omega\left(b - Ax\right)$$

$$x = \left(T_m + \omega E_m\right)^{-1}\left((1-\omega)T_m + \omega\left(T_m + E_m\right)\right)x + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax\right)$$

$$x = \left(T_m + \omega E_m\right)^{-1}\left(T_m - \omega T_m + \omega T_m + \omega E_m\right)x + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax\right)$$

$$x = \left(T_m + \omega E_m\right)^{-1}\left(T_m + \omega E_m\right)x + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax\right)$$

$$x = x + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax\right)$$

Now the Refinement of generalized successive over Relaxation is defined as

$$\overline{x}^{(k+1)} = x^{(k+1)} + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax^{(k+1)}\right) \tag{4.11}$$

where $x^{(k+1)}$ appeared in the right side is of Eq. (4.8). Substituting Eq. (4.8) in to Eq. (4.11) we obtain:

$$\overline{x}^{(k+1)} = \left(T_m + \omega E_m\right)^{-1}\left(\left(1-\omega\right)T_{\mathrm{m}} - \omega \mathrm{F_m}\right)\mathrm{x}^{(\mathrm{k})} + \left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} + \left(T_m + \omega E_m\right)^{-1}\omega\left(b - Ax^{(k+1)}\right)$$

$$\overline{x}^{(k+1)} = B_{GSOR}\mathrm{x}^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - \left(T_m + \omega E_m\right)^{-1}\omega A x^{(k+1)}$$

$$\overline{x}^{(k+1)} = B_{GSOR}\mathrm{x}^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - \left(T_m + \omega E_m\right)^{-1}\omega\left(T_m + E_m + F_m\right)x^{(k+1)}$$

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - \left(T_m + \omega E_m\right)^{-1}\left(\omega T_m + \omega E_m + \omega F_m + T_m - T_m\right)x^{(k+1)}$$

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - \left(T_m + \omega E_m\right)^{-1}(T_m + \omega E_m - ((1-\omega)T_{\mathrm{m}} - \omega F_m)x^{(k+1)}$$

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - \left(I - B_{GSOR}\right)x^{(k+1)} \tag{4.12}$$

Similarly substituting the right side $x^{(k+1)}$ of Eq. (4.12) by Eq. (4.8) we obtain:

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + 2\left(T_m + \omega\mathrm{E}_m\right)^{-1}\omega\mathrm{b} - \left(I - B_{GSOR}\right)\left(B_{GSOR}x^{(\mathrm{k})} + \left(T_m + \omega\mathrm{E}_m\right)^{-1}\omega\mathrm{b}\right)$$

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + 2\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - B_{GSOR}x^{(\mathrm{k})} + \left(B_{GSOR}\right)^2 x^{(\mathrm{k})} - \left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} + B_{GSOR}\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b}$$

$$\overline{x}^{(k+1)} = B_{GSOR}x^{(\mathrm{k})} + \left(\mathrm{T_m} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b} - B_{GSOR}x^{(\mathrm{k})} + \left(B_{GSOR}\right)^2 x^{(\mathrm{k})} - B_{GSOR}\left(\mathrm{T_m} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b}$$

$$\overline{x}^{(k+1)} = \left(B_{GSOR}\right)^2 x^{(\mathrm{k})} + \left(I + B_{GSOR}\right)\left(T_{\mathrm{m}} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b}$$

$$\overline{x}^{(k+1)} = B_{RGSOR}x^{(\mathrm{k})} + \overline{C}_s \tag{4.13}$$

where $B_{RGSOR} = (B_{GSOR})^2 = \left[\left(T_m + \omega E_m\right)^{-1}\left(\left(1-\omega\right)\mathrm{T_m} - \omega\mathrm{F_m}\right)\right]^2$ (4.14)

Eq. (4.14) is the iteration matrix of refinement Generalized Successive over relaxation method and

$$\overline{C_s} = \left(\left(I + B_{GSOR}\right)\left(\left(\mathrm{T_m} + \omega\mathrm{E_m}\right)^{-1}\omega\mathrm{b}\right)\right) \tag{4.15}$$

Eq. (4.15) is its corresponding column vector.

Error Analysis for Refinement of Generalized Successive Over Relaxation

Let consider Eq. (4.1) and the splitting of A in Eq. (4.7), such that

$x = Bx + C$ and the iteration $x^{(k)} = Bx^{(k-1)} + C$

The error at the k^{th} iteration is $e^k = x^k - x$ and the iteration matrix of RGSOR is given as

$B_{RGSOR} = \left[\left(T_m + \omega E_m \right)^{-1} \left((1-\omega) \mathrm{T_m} - \omega \mathrm{F_m} \right) \right]^2$, then the error e^k satisfies

$$e^{(k)} = x^{(k)} - x = \left[\left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \left((1-\omega) T_{\mathrm{m}} - \omega \mathrm{F_m} \right) \right]^2 x^{(k-1)} + \left(I + B_{GSOR} \right) \left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \omega \mathrm{b} - x$$

But we have $x = Bx + C$

$$x = \left[\left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \left((1-\omega) T_{\mathrm{m}} - \omega \mathrm{F_m} \right) \right]^2 x + \left(I + B_{GSOR} \right) \left(T_{\mathrm{m}} + \omega \mathrm{E_m} \right)^{-1} \omega \mathrm{b}$$

So

$$\begin{aligned} e^{(k)} = & \left[\left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \left((1-\omega) \mathrm{T_m} - \omega \mathrm{F_m} \right) \right]^2 x^{(k-1)} + \left(I + B_{GSOR} \right) \left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \omega \mathrm{b} \\ & - \left\{ \left[\left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \left((1-\omega) \mathrm{T_m} - \omega \mathrm{F_m} \right) \right]^2 + \left(I + B_{GSOR} \right) \left(\mathrm{T_m} + \omega \mathrm{E_m} \right)^{-1} \omega \mathrm{b} \right\} \end{aligned}$$

$$e^{(k)} = \left[\left(T_m + \omega E_m \right)^{-1} \left((1-\omega) \mathrm{T_m} - \omega \mathrm{F_m} \right) \right]^2 \left(x^{(k-1)} - x \right)$$

$e^{(k)} = \left(B_{GSOR} \right)^2 e^{(k-1)}.$ which implies that it is second order convergent method.

Since $\lim_{k \to \infty} \frac{e^{(k)}}{e^{(k-1)}} = \left(B_{GSOR} \right)^2$ which is quadratic convergence.

4.2 Conditions for Convergence of the Method

Theorem 4.1: The SOR iteration method converge for any initial approximation if ω lies inside side the interval *(0, 2)*.

Proof:- The proof is similar to Varga , 1999

Recall that the SOR iteration matrix B_{SOR} is given by

$$B_{SOR} = (D+\omega L)^{-1}\left((1-\omega)D - \omega U\right) \quad \text{where ,} \qquad A = \left[a_{ij}\right] = D + L + U$$

The matrix $(D+\omega L)^{-1}$ is lower triangular matrix with $\frac{1}{a_{ii}}, i = 1, 2, \ldots, n$ as diagonal entries and the matrix $\left((1-\omega)D - \omega U\right)$ is an upper triangular matrix with $(1-\omega)a_{ii}$, $i = 1, 2, \ldots, n$ as diagonal entries.

Therefore, $\det(B_{SOR}) = (1-\omega)^n$. Since the determinant of the matrix is equal to the product of its eigenvalues. We conclude that $\rho(B_{SOR}) \geq |1-\omega|$, where $\rho(B_{SOR})$ is the spectral radius of B_{SOR} .

Since the spectral radius of the iteration matrix should be less than1, for convergence, we conclude that $0 < \omega < 2$ is the required for convergence of the SOR method.

Theorem 4.2: Let A and T_{m} be symmetric positive definite matrices. Then for every $0 < \omega < 2$ the RGSOR method converges with any initial guess x_0 .

Proof:- The proof similar to the method SOR given by (Xiao-Qing and Yi-Min WEI, 2008).

The iteration matrix of RGSOR is given by $B_{RGSOR} = \left[(T_m + \omega E_m)^{-1}\left((1-\omega)T_{\text{m}} - \omega \text{F}_{\text{m}}\right)\right]^2$

Let λ be an eigenvalue of the iteration matrix of RGSOR and x be the corresponding eigenvectors. Then we have

$$B_{RGSOR}x = \lambda x$$

$$\Rightarrow \left[(T_m + \omega E_m)^{-1}\left((1-\omega)T_{\text{m}} - \omega \text{F}_{\text{m}}\right)\right]^2 x = \lambda x$$

$$\Rightarrow \left((T_m + \omega E_m)^{-1}\right)^2 \left((1-\omega)T_{\text{m}} - \omega \text{F}_{\text{m}}\right)^2 = \lambda x$$

$$\Rightarrow\left((1-\omega)\mathrm{T_m}-\omega\mathrm{F_m}\right)^2=\lambda\left(T_m+\omega E_m\right)^2 x$$

$$\Rightarrow\left((1-\omega)\mathrm{T_m}-\omega\mathrm{E}_m^T\right)^2=\lambda\left(T_m+\omega E_m\right)^2 x\text{, (}\ F_m=E_m^T\ \text{ as } A \text{ is symmetric)}$$

Let x^* be the conjugate transpose of x , then we have

$$\left(x^*\right)^2\left((1-\omega)\mathrm{T_m}-\omega\mathrm{E}_m^T\right)^2 x=\lambda\left(x^*\right)^2\left(T_m+\omega E_m\right)^2 x\ \text{, (multiplying both sides by}\left(x^*\right)^2\text{)}$$

$$\left[x^*\left((1-\omega)\mathrm{T_m}-\omega\mathrm{E}_m^T\right)x\right]^2=\lambda\left[\left(x^*(T_m+\omega E_m\right)x)\right]^2\ \text{ (multiplying both sides by } x\text{)}$$

Let $x^*\mathrm{T_m}x=\delta$ and $x^*\mathrm{E_m}x=\alpha+i\beta$ Then,

$$x^*E_m{}^T x=\left(E_m x\right)^*\mathrm{x}=\overline{x^*E_m x}=\alpha-i\beta\quad\text{(Since } A^TB^T=(BA)^T \text{ and } \left(A^*\right)^*=A\text{)}$$

we then have:

$$\left[(1-\omega)\delta-\omega(\alpha-i\beta)\right]^2=\lambda\left[\delta+\omega(\alpha+i\beta)\right]^2.$$

Taking the Modulus on both sides we get

$$\left[\left((1-\omega)\delta-\omega\alpha\right)^2+\omega^2\beta^2\right]^2=\lambda^2\left[\left(\delta+\omega\alpha\right)^2+\omega^2\beta^2\right]^2$$

and solving for λ we get

$$|\lambda|^2=\frac{\left[\left((1-\omega)\delta-\omega\alpha\right)^2+\omega^2\beta^2\right]^2}{\left[\left(\delta+\omega\alpha\right)^2+\omega^2\beta^2\right]^2}\tag{4.16}$$

On the other hand $0\le x^*Ax=x^*\left(T_m+E_m+F_m\right)x=\delta+\left(\alpha+i\beta\right)+\left(\alpha-i\beta\right)=\delta+2\alpha$

Note that

$$\left((1-\omega)\delta-\omega\alpha\right)^2+\omega^2\beta^2-\left[\left(\delta+\omega\alpha\right)^2+\omega^2\beta^2\right]=\left((1-\omega)\delta-\omega\alpha\right)^2-\left(\delta+\omega\alpha\right)^2$$

$$=(1-\omega)\delta-\omega\alpha+(\delta+\omega\alpha)\big((1-\omega)\delta-\omega\alpha\big)-(\delta+\omega\alpha)$$, (the difference of two squares)

$$= \quad (2\delta-\omega\delta)(-\omega\delta-2\omega\alpha)$$

$$= \quad \omega\delta(\omega-2)(\delta+2\alpha)$$

Since A is symmetric positive definite, we have, $\delta>0$ and $\delta+2\alpha>0$

If $0<\omega<2$ we have $\big((1-\omega)\delta-\omega\alpha\big)^2-\omega^2\beta^2<(\delta+\omega\alpha)^2-\omega^2\beta^2$

Using the relation $a<b \Rightarrow a^2<b^2$ provided that a and b are positive, we obtain:

$$\left[\big((1-\omega)\delta-\omega\alpha\big)^2-\omega^2\beta^2\right]^2<\left[(\delta+\omega\alpha)^2-\omega^2\beta^2\right]^2 \tag{4.17}$$

Thus from Eqs. (4.16) and (4.17) for $0<\omega<2$ we obtain

$$|\lambda|^2=\frac{\left[\big((1-\omega)\delta-\omega\alpha\big)^2-\omega^2\beta^2\right]^2}{\left[(\delta+\omega\alpha)^2-\omega^2\beta^2\right]^2}<1$$

Therefore the RGSOR method converges.

Theorem 4.3: Let $0<\omega\leq 1$. Let A be an IWDD matrix and T_m be irreducible. Then the GSOR method is convergent for every initial guess x_o.

Proof:- See, (Davod Salkuyeh, 2007).

Theorem 4.4: Let A be an IWDD matrix and T_m be irreducible. Let also $0<\omega\leq 1$. Then the associated RGSOR method converges for every initial guess x_o.

Proof:- Let x be the exact solution of and $\overline{x}^{(k+1)}$ be the $(k+1)^{th}$ approximation to the solution of Eq. (4.1) by method of Eq. (4.13). Then the GSOR method is convergent as proved by Salkuyeh 2007.

If $\left(x^{(k+1)}-x\right)$, then

$$= \left\|(\overline{x}^{(k+1)}-x)\right\| = \left\|x^{(k+1)}+\left(T_m+\omega E_m\right)^{-1}\omega\left(b-Ax^{(k+1)}\right)-x\right\|$$

$$= \left\|\left(x^{(k+1)}-x\right)+\left(T_m+\omega E_m\right)^{-1}\omega\left(b-Ax^{(k+1)}\right)\right\|$$

$$\le \left\|\left(x^{(k+1)}-x\right)\right\|+\left\|\left(T_m+\omega E_m\right)^{-1}\omega\left(b-Ax^{(k+1)}\right)\right\|$$

From the fact that $\left\|(x^{(k+1)}-x)\right\| \to 0$ we have $\left\|\left(b-Ax^{(k+1)}\right)\right\| \to 0$

Therefore $\left\|\overline{x}^{(k+1)}-x\right\| \to 0$

Hence the Refinement of Generalized Successive Over Relaxation method is convergent.

4.3. Description of Second Degree Generalized Successive Over Relaxation Method

Let consider the linear stationary first degree iteration method defined by (Young, 1971) as

$$x^{(k+1)} = G_1 x^{(k)} + C \qquad (4.18)$$

Where, G_1 is an iteration matrix of the iterative method and C is the corresponding column vector. Moreover (Young, 1971) defined the linear stationary second degree method as

$$x^{(k+1)} = x^{(k)} + d\left(x^{(k)}-x^{(k-1)}\right)+e\left(x^{(k+1)}-x^{(k)}\right) \qquad (4.19)$$

where $x^{(k+1)}$ appearing in the right hand side of Eq. (4.19) is substituted by $x^{(k+1)}$ in Eq.(4.18) which is completely consistent for any constant d and e such that $e \neq 0$.

$$x^{(k+1)} = x^{(k)} + d\left(x^{(k)}-x^{(k-1)}\right)+e\left(G_1 x^{(k)}+C-x^{(k)}\right)$$

$$x^{(k+1)} = x^{(k)} + dx^{(k)} - ex^{(k-1)} + eG_1 x^{(k)} + eC - ex^{(k)}$$

$$x^{(k+1)} = \left[\left(1+d-e\right)I+eG_1\right]x^{(k)} - dx^{(k-1)} + eC$$

$$x^{(k+1)} = Gx^{(k)} + Hx^{(k-1)} + l$$

where $G = (1+d-e)I + eG_1$

$$H = -dI$$

$$l = eC$$

The second degree generalized SOR iteration method is defined by (Young, 1971) as

$$x^{(k+1)} = Gx^{(k)} + Hx^{(k-1)} + l \tag{4.20}$$

where $G = (1+d-e)I + eG_s$ (4.21)

$$H = -dI \tag{4.22}$$

$$l = eC_s \tag{4.23}$$

G_s is the iteration matrix of generalized successive over relaxation iterative method and

C_s is its corresponding column vector.

(*i.e* $G_s = B_{GSOR} = (T_m + \omega E_m)^{-1}((1-\omega)\mathrm{T_m} - \omega\mathrm{F_m})$ and $C_s = (\mathrm{T_m} + \omega\mathrm{E_m})^{-1}\omega\mathrm{b}$)

Using the idea of (Golub and Varga, 1961) Eq. (4.20) can be written in the form

$$\begin{pmatrix} x^{(k)} \\ x^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & I \\ H & G \end{pmatrix} \begin{pmatrix} x^{(k-1)} \\ x^{(k)} \end{pmatrix} + \begin{pmatrix} 0 \\ l \end{pmatrix}$$

The necessary and sufficient conditions for convergence of the method is that the spectral radius of $\widehat{G}$ must be less than unity in magnitude for any $x^{(0)}$ and $x^{(1)}$.

where $\widehat{G} = \begin{pmatrix} 0 & I \\ H & G \end{pmatrix}$ is the second degree iteration matrix.

$\rho(\hat{G}) < 1$ if and only if $\det\left(\lambda^2 I - \lambda G - H\right) = 0$ are less than unity in modulus

$$\det\left(\lambda^2 I - \lambda G - H\right) = 0 \tag{4.24}$$

Substituting Eqs. (4.21), (4.22) and (4.23) in to Eq. (4.24) we have

$$\det\left(\lambda^2 I - \lambda\left[(1+d-e)I + eG_s\right] + dI\right) = 0$$

$$\det\left(-e\lambda\left[G_s + \left(\frac{1+d-e}{e}\right)\mathrm{I} - \frac{(\lambda^2+d)I}{e\lambda}\right]\right) = 0$$

$$\det(-e\lambda) \times det\left[G_s + \left(\frac{1+d-e}{e}\right)\mathrm{I} - \frac{\left(\lambda^2+d\right)I}{e\lambda}\right] = 0$$, since, ($\det(\mathrm{AB}) = \det\mathrm{A} \times \det\mathrm{B}$)

$$\det\left(G_s + \left(\frac{1+d-e}{e}\right)I - \left(\frac{\lambda^2+d}{e\lambda}\right)I\right) = 0 \tag{4.25}$$

($\det(-e\lambda) \neq 0$, since it is a non zero 1by 1 matrix)

If μ is the eigenvalue of G_S we have the relation (Manteufeel, 1981, Tesfaye, 2014 and Young, 1971)

$$\mu + \left(\frac{1+d-e}{e}\right) = \left(\frac{\lambda^2+d}{e\lambda}\right) \tag{4.26}$$

i.e for each eigenvalues $\mu_{1,}\mu_{2,}\ldots,\mu_{N,}$ of G_s λ_i are the roots of Eq. (4.26) with $\mu = \mu_i$

As discussed in (Manteufeel, 1981 Tesfaye, 2014 and Young, 1971) if we let

$$\lambda = re^{i\theta} \tag{4.27}$$

Substituting Eq. (4.27) into Eq. (4.26)

$$\mu + \frac{1+d-e}{e} = \frac{\left(re^{i\theta}\right)^2 + d}{ere^{i\theta}} = \frac{\left(rcos\theta + irsin\theta\right)^2 + d}{er\left(cos\theta + isin\theta\right)}$$

After simplifying and collecting like terms we get:

$$\mu = \frac{e-1-d}{e} + \left(\frac{r^2+d}{er}\right)\cos\theta + i\left(\frac{r^2-d}{er}\right)\sin\theta \qquad (4.28)$$

From (4.28) we have:

$$\operatorname{Re}(\mu) = \frac{e-1-d}{e} + \left(\frac{r^2+d}{er}\right)\cos\theta \quad \text{and} \quad Im(\mu) = \left(\frac{r^2-d}{er}\right)\sin\theta$$

From $\operatorname{Re}(\mu)$ we get:

$$\operatorname{Re}(\mu) + \frac{1+d-e}{e} = \left(\frac{r^2+d}{er}\right)\cos\theta$$

Solving for $\cos\theta$ we have: $\cos\theta = \dfrac{\operatorname{Re}(\mu) + \frac{1+d-e}{e}}{\left(\frac{r^2+d}{er}\right)}$

$$(\cos\theta)^2 = \left(\frac{\operatorname{Re}(\mu) + \frac{1+d-e}{e}}{\left(\frac{r^2+d}{er}\right)}\right)^2 \qquad (4.29)$$

Similarly, from $Im(\mu) = \left(\frac{r^2-d}{er}\right)\sin\theta$ we have: $\sin\theta = \dfrac{Im(\mu)}{\left(\frac{r^2-d}{er}\right)}$ and

$$(\sin\theta)^2 = \left(\frac{Im(\mu)}{\frac{r^2-d}{er}}\right)^2 \qquad (4.30)$$

Adding Eqs. (4.29) and (4.30), we obtain

$$\left(\frac{\operatorname{Re}(\mu)+\frac{1+d-e}{e}}{\frac{r^2+d}{er}}\right)^2+\left(\frac{Im(\mu)}{\frac{r^2-d}{er}}\right)^2=1 \qquad (\text{Since } \sin^2\theta+\cos^2\theta=1) \qquad (4.31)$$

From the analysis of (Frankel, 1950 as cited by Young1972), that if μ is real and varies over the range, $\alpha \le \mu \le \beta < 1$then the choice of d and e which minimizes the spectral radius of $\widehat{G}$ is given by

$$d=\hat{\omega}_b-1 \qquad e=\frac{2\hat{\omega}_b}{2-(\beta+\alpha)} \qquad (4.32)$$

Where $$\hat{\omega}_b=\frac{2}{\left(1+\sqrt{(1-\sigma^2)}\right)} \qquad \sigma=\frac{(\beta-\alpha)}{\left(2-(\beta+\alpha)\right)} \qquad (4.33)$$

The corresponding values of the spectral radius of $\widehat{G}$ is

$$\rho(\hat{G})=d^{1/2}=\sqrt{\hat{\omega}_b-1} \qquad (4.34)$$

Thus with this choice of d and e we have

$$G=(1+d-e)I+eG_s=\hat{\omega}_b\left(\frac{2G_s}{2-(\beta+\alpha)}-\frac{(\beta+\alpha)I}{2-(\beta+\alpha)}\right)$$

$$H=-dI=(1-\hat{\omega}_b)I$$

$$l=eC_s=\frac{2\hat{\omega}_b C_s}{2-(\beta+\alpha)}$$

Hence Eq. (4.20) becomes

$$x^{(k+1)}=\hat{\omega}_b\left(\frac{2G_s}{2-(\beta+\alpha)}-\frac{(\beta+\alpha)I}{2-(\beta+\alpha)}\right)x^{(k)}+(1-\hat{\omega}_b)x^{(k-1)}+\frac{2\hat{\omega}_b C_s}{2-(\beta+\alpha)} \qquad (4.35)$$

$G_s = B_{GSOR}$is the iteration matrix of GSOR and C_s is its corresponding vector.

$$G_s=B_{GSOR}=(T_m+\omega E_m)^{-1}\left((1-\omega)\mathrm{T_m}-\omega\mathrm{F_m}\right)$$

$$C_s = \omega\left(\mathrm{T_m} + \omega\mathrm{E_m}\right)^{-1}\mathrm{b}$$

According to (David R. Kincaid, 1994 and Young's, 1971) if A is symmetric positive definite matrix then A has real non negative eigenvalues and we can apply the second degree iterative method using $\alpha = 0$ and $\beta = \mu$.

$$\sigma = \frac{\beta - \alpha}{2 - (\beta + \alpha)} = \frac{\mu}{2 - \mu}$$

where μ is the spectral radius of the Generalized SOR iteration matrix.

$$\hat{\omega}_b = \frac{2(2-\mu)}{2-\mu+2\sqrt{1-\mu}} = \frac{2(2-\mu)}{\left(1+\sqrt{1-\mu}\right)^2} \tag{4.36}$$

$$d = \hat{\omega}_b - 1 = \frac{2(2-\mu)}{\left(1+\sqrt{1-\mu}\right)^2} - 1 = \frac{2(2-\mu) - \left(1+\sqrt{1-\mu}\right)^2}{\left(1+\sqrt{1-\mu}\right)^2}$$

$$= \left(\frac{2-\mu-2\sqrt{1-\mu}}{\left(1+\sqrt{1-\mu}\right)^2}\right)\left(\frac{2-\mu+2\sqrt{1-\mu}}{2-\mu+2\sqrt{1-\mu}}\right) \quad \text{, (Rationalizing the numerator)}$$

$$\frac{(2-\mu)^2 - \left(2\sqrt{1-\mu}\right)^2}{\left(1+\sqrt{1-\mu}\right)^4} = \frac{\mu^2}{\left(1+\sqrt{1-\mu}\right)^4} \quad \text{, (by the difference of two squares)}$$

Thus,
$$d = \frac{\mu^2}{\left(1+\sqrt{1-\mu}\right)^4} \tag{4.37}$$

From (4.32), we have

$$e = \frac{2\sigma\hat{\omega}_b}{(\beta-\alpha)} = \frac{2\hat{\omega}_b}{2-(\beta+\alpha)} = \frac{2}{2-\mu}\left(\frac{2(2-\mu)}{\left(1+\sqrt{1-\mu}\right)^2}\right) = \frac{4}{\left(1+\sqrt{1-\mu}\right)^2} \tag{4.38}$$

$$\rho(\widehat{G}) = d^{1/2} = \sqrt{\hat{\omega}_b - 1} = \sqrt{\frac{\mu^2}{\left(1+\sqrt{1-\mu}\right)^4}} = \frac{\mu}{\left(1+\sqrt{1-\mu}\right)^2} \tag{4.39}$$

Therefore the second degree generalized SOR is given by

$$x^{(1)} = G_s x^{(0)} + C_s$$

$$x^{(k+1)} = \hat{\omega}_b \left(\frac{2B_{GSOR}}{2-(\beta+\alpha)} - \frac{(\beta+\alpha)I}{2-(\beta+\alpha)} \right) x^{(k)} + \left(1-\hat{\omega}_b\right) x^{(k-1)} + \frac{2\hat{\omega}_b C_s}{2-(\beta+\alpha)} \tag{4.40}$$

Substituting for β *and* α we get $(\beta = \mu \;\; \& \;\; \alpha = 0)$

$$x^{(k+1)} = \hat{\omega}_b \left(\frac{2B_{GSOR}}{2-\mu} - \frac{\mu I}{2-\mu} \right) x^{(k)} + \left(1-\hat{\omega}_b\right) x^{(k-1)} + \frac{2\hat{\omega}_b C_s}{2-\mu}$$

$$x^{(k+1)} = \frac{\hat{\omega}_b}{2-\mu}\left(2B_{GSOR}x^{(k)} + 2C_s\right) - \left(\frac{\hat{\omega}_b \mu}{2-\mu}\right) x^{(k)} + \left(1-\hat{\omega}_b\right) x^{(k-1)} + \left(x^{(k)} - x^{(k)}\right)$$

$$x^{(k+1)} = \frac{\hat{\omega}_b}{2-\mu}\left(2B_{GSOR}x^{(k)} + 2C_s\right) + \left(1 - \frac{\hat{\omega}_b \mu}{2-\mu}\right) x^{(k)} + \left(\hat{\omega}_b - 1\right)\left(x^{(k)} - x^{(k-1)}\right)$$

where $$\hat{\omega}_b = \frac{4-2\mu}{\left(1+\sqrt{1-\mu}\right)^2}$$

$$B_{GSOR} = \left(T_m + \omega E_m\right)^{-1}\left(\left(1-\omega\right)\mathrm{T_m} - \omega \mathrm{F_m}\right)$$

$$\mathrm{C_s} = \left(\mathrm{T_m} + \omega \mathrm{E_m}\right)^{-1} \omega \mathrm{b}$$

4.4. Numerical Examples

To illustrate the feasibility and efficiency of the present methods when employed to solve system of linear equations, we used two systems of linear equations one dense and the other sparse. We also compared the performance of RGSOR and SDGSOR with other methods based on the number of iteration, accuracy and the computational running times. These examples are chosen because they have been widely discussed in the literature and their exact solutions are available for comparison.

Example1. Consider the system of linear equations considered by (Noreen, 2012)

$$\begin{aligned}
4x_1 - x_2 - x_3 &= 0.5\\
-x_1 + 4x_2 - x_4 &= 1.3\\
-x_1 + 4x_3 - x_4 &= 1\\
-x_2 - x_3 + 4x_4 &= 1.8
\end{aligned}$$

Example2. Consider the system of linear equations considered by (Noreen, 2012)

$$\begin{aligned}
4x_1 - x_2 - x_4 &= 0.707\\
-x_1 + 4x_2 - x_3 - x_5 &= 1\\
-x_2 + 4x_3 - x_6 &= 0.707\\
-x_1 + 4x_4 - x_5 - x_7 &= 0\\
-x_2 - x_4 + 4x_5 - x_6 - x_8 &= 0\\
-x_3 - x_5 + 4x_6 - x_9 &= 0\\
-x_4 + 4x_7 - x_8 &= 0\\
-x_5 - x_7 + 4x_8 - x_9 &= 0\\
-x_6 - x_8 + 4x_9 &= 0
\end{aligned}$$

4.5. Numerical Experiments

4.5.1 Experimental Results for the RGSOR

The efficiency, computational running time, and accuracy of the RGSOR have been compared with SOR, GSOR, and RSOR, and results are given in Tables 1 through 5.

Table 1: Comparison of Numerical Test for System of Linear Equations of Example 1 (using SOR, GSOR, RSOR & RGSOR)

Iterative Method	Number Of Iteration	Exact solution (x)	Numerical Solution (y)	Errors (e=x-y)	CPU time in second
SOR ω=1.09	8	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4124999733216189 0.6124999908348020 0.5374999911576524 0.7374999955084681	2.667838105985254e-08 9.165197822902371e-09 8.842347520854332e-09 4.491531702122131e-09	0.002923
GSOR ω= 1.035	6	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4125000132749550 0.6125000144505559 0.5375000023448290 0.7375000021865872	-1.327495502412290e-08 -1.445055608506607e-08 -2.344829108658075e-09 -2.186587355623715e-09	0.003171
RSOR ω=1.09	5	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4124999996279540 0.6124999999433065 0.5374999999459216 0.7374999999872161	3.720459496037165e-10 5.669331670787869e-11 5.407829739567660e-11 1.278377403934883e-11	0.002642
RGSOR ω=1.03	4	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4124999999871906 0.6124999999866999 0.5374999999979344 0.7374999999979814	1.280942019121767e-11 1.329991672349706e-11 2.065458915012641e-12 2.018385458768535e-12	0.001512

Table 2 Number of Iteration for each Method for an Accuracy Varying form 0.5 x10^{-6} to 0.5x10^{-15} for System of Linear Equations of Example 1

Iterative Method	Number of iteration taken for getting the solution for an error less than										
	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}
SOR	7	8	9	10	11	12	12	14	15	16	17
RSOR	4	5	5	6	6	7	7	8	8	9	9
GSOR	5	6	7	8	8	9	9	11	11	12	13
RGSOR	4	4	4	5	5	5	6	6	6	7	7

Table 3 Comparison of Numerical Test for System of Linear Equations of Example 2 (using SOR & GSOR)

Iterative Method	Number Of Iteration	Exact solution (x)	Numerical Solution (y)	Errors (e=x-y)	CPU time in second
SOR ω=1.19	10	0.3317767857142857	0.3317768359363765	-5.022209081939622e-08	0.004272
		0.4692321428571428	0.4692321726778542	-2.982071134471909e-08	
		0.3317767857142857	0.3317767943344221	-8.620136437986758e-09	
		0.1508750000000000	0.1508750428272987	-4.282729873517077e-08	
		0.2133750000000000	0.2133750152486913	-1.524869128188833e-08	
		0.1508750000000000	0.1508750032604461	-3.260446107544723e-09	
		0.0583482142857143	0.05834821007760948	4.208104815994762e-09	
		0.08251785714285716	0.08251785702790632	1.149508405129396e-10	
		0.05834821428571428	0.05834821422645566	5.925861884525929e-11	
GSOR ω=1.09	8	0.3317767857142857	0.3317767793618265	6.352459169800540e-09	0.004580
		0.4692321428571428	0.4692321338733815	8.983761345948693e-09	
		0.3317767857142857	0.3317767793618265	6.352459225311691e-09	
		0.1508750000000000	0.1508749976810884	2.318911562770865e-09	
		0.2133750000000000	0.2133749967205469	3.279453070215155e-09	
		0.1508750000000000	0.1508749976810884	2.318911590526440e-09	
		0.0583482142857143	0.05834821377399961	5.117146845146081e-10	
		0.08251785714285716	0.08251785641917157	7.236855836811884e-10	
		0.05834821428571428	0.05834821377399960	5.117146845146081e-10	

Table 4 Comparison of Numerical Test for System of Linear Equations of Example 2 (using RSOR& RGSOR)

Iterative Method	Number Of Iteration	Exact solution (x)	Numerical Solution (y)	Errors (e=x-y)	CPU time in seconds
RSOR ω=1.19	6	0.3317767857142857	0.3317767845407915	1.173494190709334e-09	0.003237
		0.4692321428571428	0.4692321415488724	1.308270380917520e-09	
		0.3317767857142857	0.3317767849770751	7.372105703673526e-10	
		0.1508750000000000	0.1508749997562597	2.437403334720756e-10	
		0.2133750000000000	0.2133749994686667	5.313332829270934e-10	
		0.1508750000000000	0.1508749996795315	3.204685128377349e-10	
		0.0583482142857143	0.05834821410038821	1.853260878093010e-10	
		0.08251785714285716	0.08251785688944602	2.534111392948901e-10	
		0.05834821428571428	0.0583482141564517	1.292625798066105e-10	
RGSOR ω=1.09	5	0.3317767857142857	0.3317767856757489	3.853678487431012e-11	0.002483
		0.4692321428571428	0.4692321428026435	5.449929396661446e-11	
		0.3317767857142857	0.3317767856757490	3.853667385200765e-11	
		0.1508750000000000	0.1508749999875549	1.244512826126254e-11	
		0.2133750000000000	0.2133749999824001	1.759983825344591e-11	
		0.1508750000000000	0.1508749999875549	1.244512826126254e-11	
		0.0583482142857143	0.05834821428307849	2.635801299444296e-12	
		0.08251785714285716	0.08251785713912956	3.727601560754579e-12	
		0.05834821428571428	0.08251785713912956	2.635794360550392e-12	

Table 5 Number of Iteration for each Method for an Accuracy Varying form 0.5 x10^{-6} to 0.5x10^{-15} for System of Linear Equations of Example 2

Iterative Method	Number of iteration taken for getting the solution for an error less than										
	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}
SOR	9	10	12	13	14	16	17	18	20	22	23
RSOR	5	6	7	7	8	9	9	10	11	12	12
GSOR	7	8	9	10	11	12	12	13	14	14	16
RGSOR	4	5	5	6	6	7	7	7	8	8	9

4.5.2 Experimental Results for the SDGSOR

The efficiency, computational running times, and accuracy of the SDGSOR have been compared against SOR, SDGGS GSOR and results are given in Table 6 and 7.

Table 6 Comparison of Numerical Test for System of Linear Equations of Example 1 (using SOR, SDGGS, GSOR & SDGSOR)

Iterative Method	Number Of Iteration	Exact solution (x)	Numerical Solution (y)	Errors (e=x-y)	CPU time in seconds
SOR ω=1.09	8	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4124999733216189 0.6124999908348020 0.5374999911576524 0.7374999955084681	2.667838105985254e-08 9.165197822902371e-09 8.842347520854332e-09 4.491531702122131e-09	0.002923
SDGGS	6	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4124999990375529 0.6124999990266025 0.5374999996784543 0.7374999996762642e	9.624470664881812e-10 9.733973627135129e-10 3.215455679494994e-10 3.237355938878750e-10	0.003139
GSOR ω=1.035	6	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4125000132749550 0.6125000144505559 0.5375000023448290 0.7375000021865872	-1.327495502412290e-08 -1.445055608506607e-08 -2.344829108658075e-09 -2.186587355623715e-09	0.003171
SDGSOR ω=1.03	5	0.4125000000000000 0.6124999999999998 0.5374999999999999 0.7374999999999998	0.4125000012955918 0.6124999995333953 0.5375000003615139 0.7374999998011715	-1.295591856020906e-09 4.666045327894608e-10 -3.615140409252149e-10 1.988282871678848e-10	0.002836

Table 7 Comparison of Numerical Test for System of Linear Equations of Example 2 (using, SOR, GSOR SDGGS & SDGSOR)

Iterative Method	Number Of Iteration	Exact solution (x)	Numerical Solution (y)	Errors (e=x-y)	CPU time in seconds
SOR ω=1.19	10	0.3317767857142857	0.3317768359363765	-5.022209081939622e-08	0.004272
		0.4692321428571428	0.4692321726778542	-2.982071134471909e-08	
		0.3317767857142857	0.3317767943344221	-8.620136437986758e-09	
		0.1508750000000000	0.1508750428272987	-4.282729873517077e-08	
		0.2133750000000000	0.2133750152486913	-1.524869128188833e-08	
		0.1508750000000000	0.1508750032604461	-3.260446107544723e-09	
		0.0583482142857143	0.05834821007760948	4.208104815994762e-09	
		0.08251785714285716	0.08251785702790632	1.149508405129396e-10	
		0.05834821428571428	0.05834821422645566	5.925861884525929e-11	
GSOR ω=1.035	8	0.3317767857142857	0.3317767793618265	6.352459169800540e-09	0.004580
		0.4692321428571428	0.4692321338733815	8.983761345948693e-09	
		0.3317767857142857	0.3317767793618265	6.352459225311691e-09	
		0.1508750000000000	0.1508749976810884	2.318911562770865e-09	
		0.2133750000000000	0.2133749967205469	3.279453070215155e-09	
		0.1508750000000000	0.1508749976810884	2.318911590526440e-09	
		0.0583482142857143	0.05834821377399961	5.117146845146081e-10	
		0.08251785714285716	0.08251785641917157	7.236855836811884e-10	
		0.05834821428571428	0.0583482137739996	5.117146845146081e-10	
SDGGS	7	0.3317767857142857	0.3317767745107025	1.120358322781456e-08	0.005013
		0.4692321428571428	0.4692321270133500	1.584379283325887e-08	
		0.3317767857142857	0.3317767745107025	1.120358322781456e-08	
		0.1508750000000000	0.1508749885183459	1.148165404418755e-08	
		0.2133750000000000	0.2133749837625242	1.623747580836721e-08	
		0.1508750000000000	0.1508749885183459	1.148165404418755e-08	
		0.0583482142857143	0.05834820984542199	4.440292307450466e-09	
		0.08251785714285716	0.08251785086334205	6.279515102347588e-09	
		0.05834821428571428	0.05834820984542199	4.440292293572679e-09	
SDGSOR ω=1.03	6	0.3317767857142857	0.3317767842293651	1.484920575389737e-09	0.004437
		0.4692321428571428	0.4692321407681666	2.088976214231764e-09	
		0.3317767857142857	0.3317767842293651	1.484920575389737e-09	
		0.1508750000000000	0.1508749943453905	5.654609475103101e-09	
		0.2133750000000000	0.2133749920029026	7.997097373202422e-09	
		0.1508750000000000	0.1508749943453905	5.654609475103101e-09	
		0.0583482142857143	0.05834821300094367	1.284770623222187e-09	
		0.08251785714285716	0.08251785532592118	1.816935976672873e-09	
		0.05834821428571428	0.05834821300094367	1.284770609344399e-09	

4.6. Discussion

In this thesis, Refinement and Second Degree Iteration of Generalized Successive Over Relaxation methods for solving large system of Linear Equations have been presented. Two practical examples, a $4X\,4$, and $9X\,9$ system of linear equations, were considered. The initial approximation for both systems is taken as all zero vectors. The stopping criterion $\left\|x^{(k+1)}-x^{(k)}\right\|<10^{-6}$ was used. We let m=1 and in this case T_m is a tri-diagonal matrix. A simple experimental determination of ω is used to find the optimum relaxation factor. We tried different values of ω and compared the rates of convergence and continued the experiment with the value of ω which gives better approximation. The results obtained by the present methods have been compared with numerical results obtained by other methods used for comparison in Tables 1 through 7.

As it can be observed from Tables 1 through 5, the refinement of generalized successive over relaxation requires less computational running time , less number of iterations and approximates the exact solution better than the other methods used for comparison. Further, the results presented in Tables 2 and 5 revealed that, as the error of tolerance decreases the number of iterations taken for convergence by the refinement of generalized successive over relaxation is the smallest of all other methods presented in this thesis. Thus the refinement of generalized successive over relaxation could be considered as more efficient method than others.

With the same stopping criterion, $\left\|x^{(k+1)}-x^{(k)}\right\|<10^{-6}$ and the initial guess taken to be zero vectors, the Second Degree Generalized Successive Over Relaxation is more efficient than the other methods used for comparison in terms of accuracy, number of iteration and computational running times.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1. Conclusion

In this thesis, we have presented Refinement of Generalized Successive Over Relaxation (RGSOR) and Second Degree Generalized Successive Over Relaxation (SDGSOR) and studied their convergence properties for symmetric positive definite matrices. Two systems of linear equations were studied and the results are presented in tables. We have compared the present methods with successive over relaxation, refinement successive over relaxation, Generalized successive over relaxation and second degree generalized Gauss Seidel for solving system of linear equations by considering number of iterations, accuracy of the numerical results and computational running times. The numerical results obtained show that both the Refinement of Generalized Successive Over Relaxation and Second Degree Generalized Successive Over Relaxation are efficient than the other methods used for comparison in this thesis.

Thus the Refinement of Generalized Successive Over Relaxation and Second Degree Generalized Successive Over Relaxation could be considered as more efficient than the Generalized Successive Over Relaxation for solving system of linear equations.

5.2. Future Work

To make this work more effective and realistic, it would be interesting to investigate optimum parameter.

REFERENCES

Atkinson, K.E. 1978. An Introduction to Numerical Analysis, 2nd Edition John Wiley &Sons, Inc,PP.325-355..

Bedet, R.A., Enright ,W.H., and Hall , T .E 1975. STIFF DETEST : A program for comparing numerical methods for stiff Ordinary differential equations. Computing Surveys 17(1) 25-48.

Burden, R.L. and Faires, J.D. 2006. Numerical analysis (7th edition), The PWS publishing company Boston

Censor , Y. 1981. Row-action methods for huge and sparse systems and their applications. p 444–466.

Davod , K. Salkuyeh, 2007. Generalized Jacobi and Gauss-Seidel Methods for Solving Linear Systems of Equations. *Numer. Math. J. Chinese Uni (Englisher) issue* 2, Vol.16 .

Davod, K. Salkuyeh, 2007. A Generalization of the SOR method for solving Linear System of equations. *J. Applied Mathematics, Islamic Azad university of Lahijan.* .

Golub, G.H and Varga, R.S. 1961. Chebyshev Semi-iterative methods, Successive Over-Relaxation iterative methods and Second Order Richardson iterative methods. Numerical mathematics. 3 P 147-168.

Goyal, M. 2007. Computer Based Numerical and Statistical Techniques, INFINITY SCIENCE PRESS LLC, New Delhi,

Grear, .J.F. 2011. How ordinary elimination became Gaussian elimination,

Hadjidimos, 2000. Successive over relaxation (SOR) and related method. *Journal of computational and applied mathematics* p177-199.

Hull, T.E, et al Enright , W.H, B.M. and Sedgwick, A.E. 1972. Comparing numerical methods for ordinary differential equations. *SIAM, J. Nmer.Anal.* VOL. 9:603-637.

Javad Iqbal, 2012. Iterative method for solving systems of linear equations, PhD Thesis, COMSATS Institute of information technology, Islamabad.

Kalambi, I. 2008. A Comparison of three Iterative Methods for the Solution of Linear Equations. *J. Appl. Sci. Environ. Manage*. Vol. 12(4), 53-55

Kalambi, I. 1998. Solution of simultaneous equations by iterative methods. Vol3

David, R. Kincaid, 1994. Stationary second degree iterative methods. *Applied Numerical Mathematics, 16, p 227-237.*

Kumer Vatti, V.B. A. and Genanew Gofe Gonfa, 2011. Refinement of Generalized Jacobi (RGJ) Method for solving of Linear system of equations. Int. *J. Contemp. and Math. sciences* Vol. (6), P 109-116.

Kumer Vatti, V.B. A. 2015. Refinement of Successive Over-relaxation(RSOR) method for solving of Linear system of equations. *J. Advanced information science and Technology,* Vol. (40),

Lay D. C., 1994. Linear Algebra and its Applications, New York

Manteufel, T.A. 1982. Optimal parameters for linear second degree stationary iterative methods. *SIAM. J. NUMER. ANAL.* ,Vol. 19, p 833-839

Nicholas J.Higham., *1997.* Iterative refinement for linear system and LAPACK, IMA.J. Numerical Analysis

Noren Jamil, A. 2012.Comparison of Direct and indirect solvers for linear systems of equations. *Int. J. Emerge. Sci.* Vol. (2)2 p 310-321 New York.

Richard, S. Varga, 1999. Matrix Iterative Analysis 2nd Edition, New York.

Rajasekaran, S. 1992. Numerical methods in Science and Engineering .Wheeler and Co. Ltd Allahabad, P 11

Saad, Y. 2000. Iterative solution of linear systems in the 20th century. *Journal of computational and applied mathematics,* p 1-33.

Saeed, N.A. and Bhatti, A., 2008. Numerical Analysis. Shahryar, Allahad.

Strassen, V.1969. Gaussian is not optimal, Numerical Math 13(4) 354-356

Tesfaye Kebede. 2014. Second degree generalized Gauss Seidel iteration method for solving linear system of equations. *Ethiop. J. Sci & Techno.* 7(2), p 115-124

Turner, P. 1989. Guide to numerical analysis. Macmillan Education Ltd., Hong Kong.

Volino P. and Magnenat-Thalmann, N. 2000. Comparing efficiency of integrations methods for cloth simulstion. VSMM,*5* p 109-118.

Wiley Amos Gilat, 2015. An introduction with application of MATLAB fifth edition, Ohio state university p 175-221.

Xiao-Qing, JIL. and Yi-Min, WEI. 2008. Numerical Linear Algebra And Its Application.

Yarlett, B.N. 1980. A new look at the lanczos algorithm for solving symmetric system of linear equations. Lin. Alg. Appl., P323-346

Yousef Saad and Vander Vorst, 2000. Iterative solution of linear system in the 20^{th} century. *J. computational applied mathematics*, p 1-33

Young, D.M .1971 Iterative solution of large linear systems. Academic press, New York,.

Young, D.M .1972. Second degree Iterative methods for the solution of Large linear systems. *Journal of approximation theory* Vol. (5) p 137-148.