ORIGINAL ARTICLE

Refinement of Generalized Accelerated Over Relaxation Method for Solving System of Linear Equations Based on the Nekrassov-Mehmke1-Method

Hailu Muleta[1] and Genanew Gofe[2]

Abstract

In this paper, refinement of generalized accelerated over relaxation (RGAOR) iterative method is presented based on the Nekrassov-Mehmke 1- method (NM1) procedure for solving system of linear equations of the form $Ax = b$, where A is a nonsingular real matrix of order n, b is a given n −dimensional real vector. The coefficient matrix $A$is split as in $A = T_m - E_m - F_m$, where T_m is a banded matrix of band width $2m + 1$ and $-E_m$ and $-F_m$ are strictly lower and strictly upper triangular parts of the matrix $A - T_m$ respectively. The finding shows that the iterative matrix of the new method is the square of generalized accelerated successive over relaxation iterative matrix. The convergence of the new method is studied and few numerical examples are considered to show the efficiency of the proposed methods. As compared to generalized accelerated successive over relaxation (SOR2GNM1, SOR1GNM1), the results reveal that the present method (RSOR1GNM1, RSOR2GNM1) converges faster and its error at any predefined error of tolerance is less than the other methods used for comparison.

Keywords: *Convergence, M-matrix, Nekrassov-Mehmke 1- method, Refinement of Generalized accelerated over relaxation*

Department of Mathematics, Jimma University, Jimma, Ethiopia

INTRODUCTION

A collection of linear equations is called linear systems of equations. They involve same set of variables. Various methods have been introduced to solve systems of linear equations (Noreen, J., 2012 and Saeed, N.A., Bhatti, A., 2008). 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 volume of computations involved is huge. Another issue in the accuracy problem for the solutions rounding off errors involved in executing these computations.

Systems of linear equations arise in a large number of areas both directly in modeling physical situations and indirectly in the numerical solutions of the other mathematical models. These applications occur in all areas of the physical, biological, social science and engineering etc. The linear system problem is, "Given an n×n nonsingular matrix A and an n-vector b, the problem is to find an n-vector x such that Ax=b". The most common source of the above problem is the numerical solution of differential equations. A system of differential equations is normally solved by discretizing the system by means of finite difference methods. The efficiency of any method can be judged by two criteria namely, how fast it is i.e. how many operations are involved? And how accurate is the computer solution? (Anamul, H., L. and Samira, B., 2014).

Direct methods are not appropriate for solving large number of equations in a system, particularly when the coefficient matrix is sparse, i.e. when most of the elements in a matrix are zero (Noreen, J., 2012 and Anita, H., M., 2002). In contrast, Iterative methods are suitable for solving linear equations when the number of equations in a system is very large.

Iterative methods are very effective concerning computer storage and time requirements. One of the advantages of using iterative methods is that they require fewer multiplications for large systems. In general, it can be easily realize that direct methods are not appropriate for solving large number of equations in a system when the coefficient matrix is sparse i.e. when most of the elements in a matrix are zero. On the other hand iterative methods are suitable for solving linear equations when the number of equations in a system is very large. Iterative methods are very effective concerning computer storage and time requirements. One of the advantages of using iterative methods is that they require fewer multiplications for large systems. Iterative methods are fast and simple to use when the coefficient matrix is sparse. Also these methods have fewer rounds off errors as compared to the direct methods.

Preliminaries

Let us consider the linear system $Ax - b = 0, (detA \neq 0)$, or

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - b_i = 0, i = 1,2, \dots, n \qquad (1)$$

Suppose that the matrix A is strictly diagonally dominant (SDD), i.e.,

$$|a_{ii}| > \sum_{j \neq i}^{n} |a_{ij}|, i = 1,2,3, \dots, n.$$

Using the Nekrassov−Mehmke iteration scheme (Mehmke, R. and Nekrassov, P., 1892) the sequence of consecutive approximations $x_i^{(k)}$, is computed as follows:

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \quad i = 1,2, \dots, n \qquad (2)$$

$$k = 0,1,2, \dots$$

The scheme in Eq. (2) is called the *Nekrassov−Mehmke 1−method (NM1).* In a number of cases the success of the procedures of type (2) depends on the proper ordering of the equations and x_i, $i = 1, \dots, n$

In spite of this fact the following modification of the *Nekrassov−Mehmke method* is known (Faddeev, D. and Faddeeva, V., 1963):

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} + \frac{b_i}{a_{ii}}, \quad i = n, n-1, \dots, 1 \qquad (3)$$

$$k = 0,1,2, \dots$$

The Scheme in Eq. (3) is called the *Nekrassov−Mehmke 2−method (NM2).*

The *(NM2)* –method does not possess better convergence in comparison with method (NM1). But under circumstances, if A is positive definite then the Eigen−values of matrix G in the matrix equations $x = Gx + C$are real and this allows to apply different methods for improving rate of convergence (Faddeev, D. and Faddeeva, V.,1963).

Let $A = (a_{ij})$ be an $n \times n$ nonsingular matrix and $T_m = (t_{ij})$ be a banded matrix of band width $2m + 1$ is defined as

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

We consider the decomposition

$$A = T_m - E_m - F_m$$

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

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

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

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

Applying the Nekrassov$-$*Mehmke 1-method (NM1)* to the system in Eq. (1) with the decomposition

$A = T_m - E_m - F_m$, we have

$$x^{(k+1)} = (T_m - E_m)^{-1} F_m x^{(k)} + (T_m - E_m)^{-1}, k = 0,1\ 2, \ldots \quad (4)$$

Let ω be a parameter such that the matrix $T_m - \omega E_m$ be nonsingular.

Salkuyeh, D. (2007) considers the following Successive Over Relaxation Generalized Nekrassov-Mehmke method (GNM1)-(SORGNM1):

$$x^{(k+1)} = (T_m - \omega E_m)^{-1}(\omega F_m + (1-\omega)T_m)x^{(k)} + (T_m - \omega E_m)^{-1}\omega b \quad (5)$$

$$k = 0, 1, 2$$

Let $G^{(m)}{}_{GAOR}(\omega)$be the iteration matrix of the method (5), i.e.

$$G^{(m)}{}_{GAOR}(\omega) = (T_m - \omega E_m)^{-1}(\omega F_m + (1-\omega)T_m).$$

Theorem 1: Let A and T_m be strictly diagonally dominant (SDD). Then for every $0 < \omega < 2$, the method (SORGNM1) converges.

Proof: see (Zaharieva, D. and Malinova, A., 2011)

Salkuyeh, D. (2011) proposed Generalized Accelerated Over relation method-(GAOR), based on the Nekrassov-Mehmke mehod (GNM1) :

$$x^{(k+1)} = (T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega - \gamma)E_m + \omega F_m\big)x^{(k)} + (T_m - \gamma E_m)^{-1}\omega b \quad (6)$$

$k = 0,1,2, \dots$, based on method (5), where$0 \leq \gamma < \omega \leq 1$.

Let $G^{(m)}{}_{GAOR}(\gamma,\omega)$be the iteration matrix of the method (6), i.e.

$$G^{(m)}{}_{GAOR}(\gamma,\omega) = (T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega - \gamma)E_m + \omega F_m\big)$$

Procedure (6) is valid in the case where A is an$M-$ matrix.

Definition 1: $\boldsymbol{A}$is an $M-$matrix if$a_{ii} > 0$for $i = 1, 2, \dots, n$, $a_{ij} \leq 0$ for $i \neq j$, A is nonsingular and $A^{-1} \geq 0$**.**

Definition 2: Let $A \in \Re^{n\times n}$. The splitting $A = M - N$ is called:

a. Weak regular if $M^{-1} \geq 0$ and $M^{-1}N \geq 0$;
b. Regular if $M^{-1} \geq 0$ and $N \geq 0$

Theorem 2: Let $A = (a_{ij})$ and $B = \big(b_{ij}\big)$ be two matrices such that $A \leq B$ and $b_{ij} \leq 0$ for all $i \neq j$. Then, if A is an M-matrix, so is the matrix B (Saad, Y., 1995).$A$and B are n dimensional square matrices.

Theorem 3: Let A be an M-matrix and $A = M - N$ regular or weak regular splitting of A. Then, $\rho(M^{-1}N) < 1$(Wang, L. and Song, Y., 2009).

Lemma 1: Let A be an M-matrix and$A = T_m - E_m - F_m$ be the splitting of A. Then T_m is an M-matrix and $\rho\big({T_m}^{-1}E_m\big) < 1$ (Salkuyeh, D., 2011).

Theorem 4: If A is an M-matrix and $0 \leq \gamma \leq \omega \leq 1$, with $\omega \neq 0$, then the AOR iterative method is convergent, i.e., $\rho\big(G_{AOR}(\gamma,\omega)\big) < 1$ (Wu, M. et al., 2007**).**

Theorem 5: If A is an M- matrix and $0 \leq \gamma \leq \omega \leq 1$with $\omega \neq 0$, then the method (6) is convergent, i.e.,$\rho\left(G^{(m)}{}_{GAOR}(\gamma,\omega)\right) < 1$.

Proof: In the GAOR iterative method, we have $A_m = M_m - N_m$, where $M_m = T_m - \gamma E_m$, and $N_m = (1-\omega)T_m + (\omega - \gamma)E_m + \omega F_m$.

Evidently, we have $A \leq M_m$. Therefore, by Theorem 2, M_m is an M-matrix and ${M_m}^{-1} \geq 0$.

From Lemma 1, we have $\rho\left({T_m}^{-1}E_m\right) < 1$.

Since $0 \leq \gamma \leq 1$, we have $\rho\left(\gamma {T_m}^{-1}E_m\right) < 1$, and therefore,

$$M_m{}^{-1}N_m = (T_m - \gamma E_m)^{-1}\left((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\right)$$

$$=\left(I - \gamma {T_m}^{-1}E_m\right)^{-1}\left((1-\omega)I + (\omega-\gamma){T_m}^{-1}E_m + \omega {T_m}^{-1}F_m\right) \quad \geq 0.$$

Therefore, we conclude that $\omega A = M_m - N_m$ is a weak splitting of ωA. Now, from Theorem 3, we realize that $\rho\left(\gamma {M_m}^{-1}N_m\right) < 1$ and this completes the proof.

Refinement of Generalized Accelerated Over Relaxation Method based on the Nekrassov-Mehmke1-method (GNM1)-($\boldsymbol{RGAOR}$).

Since the rate of convergence of stationary iterative process depends on spectral radius of the iterative matrix, any reasonable modification of the iterative matrix that will reduce the spectral radius increases the rate of convergence of that method (Vatti, V. and Genanew, G. G.,2011).

Let $x^{(1)}$ be an initial approximation for the solution of the system in Eq.(1) and

$$b_i{}^{(1)} = \sum_{j=1}^{n} a_{ij}x_j{}^{(1)}, i = 1,2,\dots,n.$$

After k^{th} iteration, we obtain $b_i{}^{(k+1)} = \sum_{j=1}^{n} a_{ij}x_j{}^{(k+1)}, i = 1,2,\dots,n.$

This obtained solution is refined as $b_i{}^{(k+1)} \to b_i$.

Assume that $\tilde{x}^{(k+1)} = \left(\tilde{x}_1{}^{(k+1)},\dots,\tilde{x}_n{}^{(k+1)}\right)$ is good approximation for the solution of the system in Eq.(1), i.e., $\tilde{x}^{(k+1)} \to x$, where x is the exact solution of Eq.(1), and $b_i = \sum_{j=1}^{n} a_{ij}\tilde{x}_j{}^{(k+1)}, i = 1,2,\dots,n.$

Since all $\tilde{x}_j{}^{(k+1)}$ are unknown, we define it as follows, $\tilde{x}^{(k+1)} = x^{(k+1)} + b^{(k+1)} - b$.

By the decomposition:

$$\omega A = (T_m - \omega E_m) - [(1-\omega)T_m + \omega F_m]$$

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

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

$$(T_m - \omega E_m)x = [T_m - \omega A - \omega E_m]x + \omega b$$

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

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

That is, $$\tilde{x}^{(k+1)} = x^{(k+1)} + (T_m - \omega E_m)^{-1}(b\omega - \omega A x^{(k+1)})$$

From Eq. (5), we have

$$\tilde{x}^{(k+1)} = (T_m - \omega E_m)^{-1}(\omega E_m + (1-\omega)T_m)x^{(k)} + (T_m - \omega E_m)^{-1}\omega b +$$
$$(T_m - \omega E_m)^{-1}\left[\omega b - \omega A\left[(T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)x^{(k)} + (T_m - \omega E_m)^{-1}\omega b\right]\right]$$

Therefore, the $G^{(m)}{}_{R1GAOR}$ becomes

$$x^{(k+1)} = \left[(T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)\right]^2 x^{(k)} +$$
$$(T_m - \omega E_m)^{-1}\left[I + (T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)\right]\omega b$$
$$k\ 0,1,2,\ldots \qquad (7)$$

We shall call the matrix $G^{(m)}{}_{R1GAOR} = \left[(T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)\right]^2$ as refinement of generalized accelerated over relaxation iteration matrix and $(T_m - \omega E_m)^{-1}\left[I + (T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)\right]\omega b$ the corresponding refinement of generalized accelerated over relaxation vector.

Theorem 6: Let A be strictly diagonally dominant (SDD) matrix of order n. Then for any natural number $m \leq n$ the (RSOR1GNM1) method is convergent for any initial guess $x^{(0)}$.

Proof: Assume x is the exact solution of Eq. (1), as A is SDD matrix, by Theorem 1, a (SOR1GNM1) is convergent.

Let $x^{(k+1)} \to x$. Then

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

Evidently, $\left\|x^{(k+1)} - x\right\|_\infty \to 0$, we have $\left\|b - Ax^{(k+1)})\right\|_\infty \to 0$.

As a result, $\left\|\tilde{x}^{(k+1)} - x\right\|_\infty \to 0$ and a (RSOR1GNM1) method is convergent.

Theorem 7: Let A be an M-matrix of order n. Then for any natural number $m \leq n$ then the (RSOR1GN1) method is convergent for any initial guess $x^{(0)}$.

***Proof*:** Let $M_m = T_m - \omega E_m$ and $N_m = (1-\omega)T_m + \omega F_m$ in $G^{(m)}{}_{R1GAOR}$. Evidently, $A \leq T_m - \omega E_m$.

Hence by Theorem 2, we conclude that the matrix M_m is an M-matrix. On the other hand, $N_m \geq 0$. Thus, $A = M_m - N_m$ is a regular splitting of the matrix A. Bearing in mind that $A^{-1} \geq 0$ and making use of Theorem 3, we conclude that $\rho\left((T_m - \omega E_m)^{-1}\big((1-\omega)T_m + \omega F_m\big)\right) < 1$.

We realize that the iteration matrix of refinement of generalized accelerated over relaxation method is the square of the iteration matrix of generalized accelerated over relaxation iteration matrix, i.e. $G^{(m)}{}_{RGAOR}(\omega) = \left[G^{(m)}{}_{GAOR}(\omega)\right]^2$.

Evidently, $\rho\left(G^{(m)}{}_{RGAOR}(\omega)\right) = \left[\rho\left(G^{(m)}{}_{GAOR}(\omega)\right)\right]^2$, where $\rho\left(G^{(m)}{}_{GAOR}(\omega)\right)$ is the spectral radius of GAOR iteration matrix, whereas $\left[\rho\left(G^{(m)}{}_{GAOR}(\omega)\right)\right]^2$ is the spectral radius of RGAOR iteration matrix. Since GAOR converges, $\rho\left(G^{(m)}{}_{GAOR}(\omega)\right) < 1$, then $\rho\left(G^{(m)}{}_{RGAOR}(\omega)\right) < \rho\left(G^{(m)}{}_{GAOR}(\omega)\right) < 1$.

Hence, RSOR1GNM1 method is convergent.

Thus, if GAOR and RGAOR converge, then the RGAOR converges faster than the GAOR method.

Let γ be a fixed parameter so that $T_m - \omega E_m$ be nonsingular.

By the decomposition:

$$\omega A = (T_m - \gamma E_m) - [(1-\omega)T_m + (\omega - \gamma)E_m + \omega F_m]$$

We have,

$$\big[(T_m - \gamma E_m) - [(1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m]\big]x = \omega b.$$

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

$$(T_m - \gamma E_m)x = [T_m - \gamma E_m - \omega A]x + \omega b$$

$$(T_m - \gamma E_m)x = (T_m - \gamma E_m)x + \omega(b - Ax)$$

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

That is, $$\tilde{x}^{(k+1)} = x^{(k+1)} + \omega(T_m - \gamma E_m)^{-1}(b - Ax^{(k+1)})$$

From method (6), we have

$$\tilde{x}^{(k+1)} = (T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\big)x^{(k)} + (T_m - \gamma E_m)^{-1}\omega b +$$

$$(T_m - \gamma E_m)^{-1}\Big[\omega b - \omega A\big[(T_m - \gamma E_m)^{-1}\big((1-\omega)T_m\big) + (\omega-\gamma)E_m + \omega F_m)x^{(k)} + (T_m - \gamma E_m)^{-1}\omega b\big]\Big]$$

Therefore, the $G^{(m)}{}_{R2GAOR}$ becomes

$$x^{(k+1)} = \big[(T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\big)\big]^2 x^{(k)} +$$

$$(T_m - \gamma E_m)^{-1}\big[I + (T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\big)\big]\omega b$$

$k = 0,1,2,\ldots$ (8),

where $0 \leq \gamma < \omega \leq 1$.

We shall call the method (8) the Refinement of (SOR2GNM1) method –(RSOR2GNM1)

We shall call the matrix $G^{(m)}{}_{R2GAOR} = \big[(T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\big)\big]^2$ as refinement of generalized accelerated over relaxation iteration matrix and $(T_m - \gamma E_m)^{-1}\big[I + (T_m - \gamma E_m)^{-1}\big((1-\omega)T_m + (\omega-\gamma)E_m + \omega F_m\big)\big]\omega b$ the corresponding refinement of generalized accelerated over relaxation vector.

Theorem 8 Let A be an M-matrix. Then for any natural number $m \leq n$ the (RSOR2GNM1) method is convergent for any initial guess $x^{(0)}$.

Proof: *The proof follows from Theorem 5 and 7, and will be omitted.*

Numerical Experiments

The numerical examples presented in this section are computed with some MATLAB codes on a personal computer Intel® Core™ i3-3420CPU@3.40GHZ having 2GB memory(RAM) with 32 bits operating system(window 7 home premium). The stopping criteria used is $\left\|{x_i}^{(k+1)} - {x_i}^{(k)}\right\| \leq 5 \times 10^{-7}$, where ${x_i}^{(k+1)}$ and ${x_i}^{(k)}$ are the computed solutions at the $(\boldsymbol{k+1})$ and $\boldsymbol{kth}$ step of each method, respectively.

Here we consider two examples to illustrate the theory developed in this paper. The efficiency of the proposed method (RSOR1GNM1 and RSOR2GNM1) is compared with SOR1GNM1 and SOR2GNM1.

Example1. Consider the system of equations considered by (YOUNG, D. M., 1971; Vatti, V. and Genanew, G. G., 2011).

$$\begin{pmatrix} 4 & 0 & -1 & -1 \\ 0 & 4 & - & -1 \\ -1 & -1 & 4 & 0 \\ -1 & -1 & 0 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 100 \\ 0 \\ 100 \\ 0 \end{pmatrix}$$

This matrix is strictly diagonally dominant with positive diagonal and non-positive off-diagonal entries, and $A^{-1} \geq 0$. Hence, the coefficient matrix A is an M-matrix.

The solution of the above system is solved and tabulated by using the methods SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 taking the initial approximations for $x's$ as all zero vector and letting $\omega = 0.9$ and $\gamma = 0.5$.

Table 1: Spectral radii of SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 when m = 1 of example 1.

Method	**SOR2GNM1**	**SOR1GNM1**	**RSOR1GNM1**	**RSOR2GNM1**
Spectral radius	0.4269416899692237	0.3286647942326976	0.1080205469680216	0.1822792066337767

Table 2: Numerical solution of **example1** by SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 when m = 1

	SOR2GNM1				RSOR2GNM1			
n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	22.500000	6.750000	27.000000	3.656250	31.647656	9.207422	33.539062	8.397509
2	31.647656	9.207422	33.539063	8.397510	36.493959	11.811068	36.842166	11.711346
3	35.100494	10.950096	35.913573	10.681583	37.318739	12.371311	37.382322	12.354827
4	36.493959	11.811068	36.842166	11.711346	37.467034	12.476434	37.478627	12.473489
5	37.073936	12.200800	37.222778	12.160979	37.493993	12.495700	37.496106	12.495165
6	37.318739	12.371311	37.382322	12.354828	37.498905	12.499216	37.499290	12.499118
7	37.422733	12.444881	37.449885	12.437939	37.499800	12.499857	37.499870	12.499839
8	37.467034	12.476434	37.478628	12.473489	37.499963	12.499973	37.499976	12.499970
9	37.485930	12.489933	37.490880	12.488670	37.499993	12.499995	37.499995	12.499994
10	37.493994	12.495701	37.496107	12.495166	37.499998	12.499999	37.499999	12.499999
11	37.497436	12.498164	37.498338	12.497936	37.499999	12.500000	37.500000	12.500000
12	37.498905	12.499216	37.499290	12.499119	37.500000	12.500000	37.500000	12.500000
13	37.499533	12.499665	37.499697	12.499624				
14	37.499800	12.499857	37.499871	12.499839				
15	37.499915	12.499939	37.499945	12.499931				
16	37.499964	12.499974	37.499976	12.499971				
17	37.499984	12.499988	37.499990	12.499987				
18	37.499993	12.499995	37.499996	12.499995				
19	37.499997	12.499998	37.499998	12.499998				
20	37.499999	12.499999	37.499999	12.499999				
21	37.499999	12.500000	37.500000	12.500000				
22	37.500000	12.500000	37.500000	12.500000				
CPU time (in seconds) = 0.050323					**CPU time (in seconds) = 0.0238879**			

	SOR1GNM1				RSOR1GNM1			
n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	22.500000	7.350000	29.400000	6.716250	32.876156	10.319469	35.233252	10.390640
2	32.876156	10.319469	35.233252	10.390641	37.035904	12.218478	37.275424	12.260334
3	36.052992	11.688880	36.799482	11.780985	37.450653	12.468553	37.476098	12.473899
4	37.035904	12.218479	37.275424	12.260335	37.494685	12.496582	37.497425	12.497176
5	37.349136	12.405276	37.426955	12.420776	37.499426	12.499630	37.499722	12.499694
6	37.450653	12.468553	37.476099	12.473899	37.499938	12.499960	37.499969	12.499967
7	37.483815	12.489620	37.492159	12.491413	37.499993	12.499995	37.499996	12.499996
8	37.494685	12.496582	37.497425	12.497176	37.499999	12.499999	37.499999	12.499999
9	37.498254	12.498876	37.499154	12.499072	37.500000	12.500000	37.500000	12.500000
10	37.499426	12.499630	37.499722	12.499695				
11	37.499811	12.499879	37.499909	12.499000				
12	37.499938	12.499960	37.499970	12.499967				
13	37.499980	12.499987	37.499990	12.499989				
14	37.499993	12.499996	37.499997	12.499996				
15	37.499998	12.499999	37.499999	12.499999				
16	37.499999	12.499999	37.499999	12.500000				
17	37.500000	12.500000	37.500000	12.500000				
CPU time (in seconds) = 0.043437					CPU time (in seconds) = 0.019739			

Table 3: Numerical solution of **example1** by SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 when m = 2

Method	**Spectral Radius**	**Iteration Number**	**CPU time (in second)**
SOR2GNM1	0.2912208524	16	0.062166
SOR1GNM1	0.2142471721	13	0.031271
RSOR2GNM1	0.0840958490	9	0.025449
RSOR1GNM1	0.0459018507	7	0.016730

Example2. Consider 2-cyclic matrix, which arises from discretization of the Poisson's equation $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y)$ on the unit square as considered by (Dafchahi, F. N., 2008; Vatti, V. and Genanew, G.G., 2011).

Now consider $Ax = b$, where $x = (x_1, \dots, x_6)^T$ and $b = (1, 0,0, 0, 0, 0)^T$ or

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This matrix is strictly diagonally dominant with positive diagonal and non-positive off-diagonal entries $A^{-1} \geq 0$.

Hence, the coefficient matrix A is an M-matrix.

The solution of the above system is solved and tabulated below by using the iterative methods SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 taking the initial approximations for $x's$ as all zero vector letting $\omega = 0.9$ and $\gamma = 0.5$.

Table 4: Spectral radii of SORG2NM1, RSORG2NM1, SOR1GNM1 and RSOR1GNM1 when m = 1of example 2

Method	SOR2GNM1	SOR1GNM1	RSOR1GNM1	RSOR2GNM1
Spectral radius	0.382053999242000	0.286203173633144	0.081912256597683	0.145965258336806

Table 5: Numerical solution of **example2** by SORG2NM1, RSORG2NM1, SOR1GNM1 and RSOR1GNM1 when m = 1

	SOR2GNM1						**CPU**
n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$x_6^{(n)}$	**time(sec)**
1	0.2410714285	0.0642857142	0.0160714285	0.0347257653	0.0183673469	0.0066007653	
2	0.2748368030	0.0780940233	0.0210086780	0.0710383389	0.0382528113	0.0139973232	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	0.016151
12	0.2948237169	0.0931672842	0.0281570584	0.0861281217	0.0496891025	0.0194614614	
13	0.2948239025	0.0931675422	0.0281572381	0.0861282720	0.0496893116	0.0194616071	
	SOR1GNM1						**CPU**
n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$x_6^{(n)}$	**time(sec)**
1	0.2410714285	0.0642857142	0.0160714285	0.0625063775	0.0330612244	0.0118813775	
2	0.2825633883	0.0839978134	0.0236727633	0.0801489119	0.0445921699	0.0168360213	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	0.015458
10	0.2948236728	0.0931672315	0.0281570262	0.0861281814	0.0496891892	0.0194615237	
11	0.2948239191	0.0931675671	0.0281572564	0.0861283124	0.0496893689	0.0194616475	

n	**RSOR2GNM1**						**CPU time(sec)**
	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$x_6^{(n)}$	
1	0.2748368030	0.0780940233	0.0210086780	0.0710383389	0.0382528113	0.0139973232	
2	0.2925673346	0.0908550744	0.0268480955	0.0843303828	0.0478213664	0.0183948694	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	0.013495
6	0.2948232279	0.0931666091	0.0281565911	0.0861277259	0.0496885554	0.0194610823	
7	0.2948239025	0.0931675422	0.0281572381	0.0861282720	0.0496893116	0.0194616071	
	RSOR1GNM1						**CPU time(sec)**
n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$x_6^{(n)}$	
1	0.2825633883	0.0839978134	0.0236727633	0.0801489119	0.0445921699	0.0168360213	
2	0.2940394571	0.0923461701	0.0276790386	0.0857254153	0.0492453211	0.0191956668	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	0.012419
5	0.2948236728	0.0931672315	0.0281570262	0.0861281814	0.0496891892	0.0194615237	
6	0.2948239888	0.0931676632	0.0281573229	0.0861283496	0.0496894203	0.0194616832	

CONCLUSION

In this paper, the refinement of generalized accelerated over relaxation method, based on the Nekrassov-Mehmke 1- method (GNM1), for solving system of linear equations is proposed and its convergence properties for SDD and M-matrices is studied. Two numerical examples (a 4X4 and 6X6 system of linear equations) are presented and investigated by using MATLAB version 7.60(R2008a) software package to show the effectiveness of the proposed method. The results obtained by RSOR1GNM1 and RSOR2GNM1 are compared with that of SOR1GNM1 and SOR2GNM1 as depicted in Tables1, 2, 3, 4 and 5. The analysis of the results in tables shows that the proposed method converges to the exact solution faster than the SOR1GNM1 and SOR2GNM1in terms of iteration numbers and computational running times. As a result, RSOR1GNM1 and RSOR2GNM1 require less memory than SOR1GNM1 and SOR2GNM1.

ACKNOWLEDGEMENT

The authors are grateful to Jimma University for the reason that we have used all the necessary resources of Jimma University while conducting this research.

REFERENCES

Anamul, H. L. and Samira, B., (2014). Refinement of Iterative Methods for the Solution of System of Linear Equations Ax=b. IOSR Journal of Mathematics (IOSR-JM), **10(3),** 70-73.

Dafchahi, F.N. (2008). A new refinement of Jacobi method for solution of linear system of equations. Int. J. Contemp. Math. Sci. **3**(17), 819-827.

Faddeev, D. and Faddeeva, V. (1963). On one approach for improving iteration processes. Comp., Rend. Acad. SSSR, 74, 1051-1052.

Faddeev, D. and Faddeeva, V.(1963). Computational methods of linear algebra, 2nd ed., Mishawaka, U.S.A: undergraduate mathematics books.

Mehmke, R. and Nekrassov, P. (1892). Solution of linear system equations by means of successive approximations. Math. Sb., **16**, 437-459.

Nickolay, K. and Anton, I.(2013). A refinement of some over relaxation algorithms for solving a system of linear equations. Serdica J. Computing, **7**(3), 245-256.

Noreen, J. (2012). A Comparison of Direct and Indirect Solvers for Linear Systems of Equations. Int. J. Emerg. Sci., **2(2)**, 310-321.

Saad, Y.(1995). Iterative Methods for Sparse Linear Systems, PWS press, New York.

Saeed, N. A. and Bhatti, A. (2008). Numerical Analysis. Shahryar.

Salkuyeh, D.(2007).A generalization of the SOR method for linear system of equations, J. of Appl. Math. (Islamic Azad Univ. of Lahijan), **4**(15), 31-38.

Salkuyeh, D.(2011). Generalized AOR method for solving system of linear equations, Australian J. of Basic and Appl. Sci. ,**5**(3), 351-358.

Vatti, V. and Genanew, G.G.(2011). Refinement of generalized Jacobi (RGJ) method for solving system of linear equations. Int. J. Contemp. Math. Sci., **6**(3), 109-116.

WANG, L. and SONG,Y. (2009). Preconditioned AOR iterative methods for M-matrices, Journal of Computational and Applied Mathematics.**226**, 114-124.

WU,M. et al.(2007). Preconditioned AOR iterative method for linear systems. Applied Numerical Mathematics, 57,672-685.

Young, D. M. (1971).Iterative solution of large linear systems. Acad. Press, New York and London.

Zaharieva, D. and Malinova, A. (2011).On some (AOR) iterative algorithms for solving system of linear equations. PLOVDIV univ. “Paissiihilendarski”, Bulgaria, Sci. works, **38**(3), 133-144.