ORIGINAL ARTICLE

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

Hailu Muleta¹ and Genanew Gofe²

Abstract

In this paper, refinement of generalized accelerated over relaxation (RGAIOR) 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 the 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 ($RSON2GNM1$, $RSON2GNM1$) 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 is 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 errors as compared to the direct methods.


Preliminaries

Let us consider the linear system \( Ax - b = 0, (\det A \neq 0) \), or
\[
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - b_i = 0, \quad i = 1, 2, \ldots, n
\] (1)

Suppose that the matrix \( A \) is strictly diagonally dominant (SDD), i.e.,
\[
|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, 3, \ldots, n.
\]

Using the Nekrassov–Mehmke iteration scheme (Mehmke, R. and Nekrassov, P., 1892) the sequence of consecutive approximations \( x_i^{(k)} \), is computed as follows:
\[
x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \quad i = 1, 2, \ldots, n
\] (2)

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 \( x_i, \quad i = 1, \ldots, n \)

In spite of this fact the following modification of the Nekrassov–Mehmke method is known (Faddeev, D. and Faddeeva, V., 1963):
\[
x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} + \frac{b_i}{a_{ii}}, \quad i = n, n-1, \ldots, 1
\] (3)

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
\[
t_{ij} = \begin{cases} a_{ij} & \text{if } |i - j| \leq m \\ 0 & \text{otherwise} \end{cases}
\]
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

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

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

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

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

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

Let $\omega$ 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):

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

$$k = 0, 1, 2$$

Let $G^{(m)}_{G0R}(\omega)$ be the iteration matrix of the method (5), i.e.

$$G^{(m)}_{G0R}(\omega) = (T_m - \omega E_m)^{-1}(\omega F_m + (1 - \omega)T_m).$$
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)


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

$k = 0,1,2, ..., $ 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.

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

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

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

Definition 2: Let $A \in \mathbb{R}^{n \times n}$. 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 $B = (b_{ij})$ 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). And $A$ is an $n$ 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 $\rho (T_m^{-1}E_m) < 1$ (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., $\rho (G_{AOR}(\gamma, \omega)) < 1$ (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., $\rho (G^{(m)}_{GAOR}(\gamma, \omega)) < 1$.

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 $\rho(T_m^{-1}E_m) < 1$.

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

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

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

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

Refinement of Generalized Accelerated Over Relaxation Method based on the Nekrassov-Mehmke1-method (GNM1)-(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 Genanev, G. G., 2011).

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

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

After $k^{th}$ iteration, we obtain

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

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

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

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:

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

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

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

Hailu M. & Genanew G.

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

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

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

That is,

\[\hat{x}^{(k+1)} = x^{(k+1)} + (T_m - \omega E_m)^{-1}(b\omega - \omega Ax^{(k+1)})\]

From Eq. (5), we have

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

Therefore, the \[G^{(m)}_{R1GADR}\] becomes

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

\[k = 0, 1, 2, ...\]

(7)

We shall call the matrix \[G^{(m)}_{R1GADR} = \left[(T_m - \omega E_m)^{-1}(1 - \omega)T_m + \omega F_m\right]^2\] as refinement of generalized accelerated over relaxation iteration matrix and \[(T_m - \omega E_m)^{-1}[I + (T_m - \omega E_m)^{-1}(1 - \omega)T_m + \omega F_m]\omega b\] 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

\[\|\hat{x}^{(k+1)} - x\|_\infty \leq \|x^{(k+1)} - x\|_\infty + \omega \|(T_m - \omega E_m)^{-1}\|_\infty \|b - Ax^{(k+1)}\|_\infty\]

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

As a result, \[\|\hat{x}^{(k+1)} - x\|_\infty \to 0\] 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 $G^{(m)}_{R_{1GAOR}}$. 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, $N_m = 0$. 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 $\rho \left( (T_m - \omega E_m)^{-1}(1 - \omega)T_m + \omega F_m \right) < 1$.

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 $\gamma$ 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,

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

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

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

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

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\[ \tilde{x}^{(k+1)} = (T_m - \gamma E_m)^{-1}((1 - \omega)T_m + (\omega - \gamma)E_m + \omega F_m)x^{(k)} + (T_m - \gamma E_m)^{-1}\omega b + (T_m - \gamma E_m)^{-1}\omega a \]

Therefore, the \( G^{(m)}_{R2GAOR} \) becomes\[ x^{(k+1)} = [(T_m - \gamma E_m)^{-1}((1 - \omega)T_m + (\omega - \gamma)E_m + \omega F_m)]x^{(k)} + (T_m - \gamma E_m)^{-1}[I + (T_m - \gamma E_m)^{-1}((1 - \omega)T_m + (\omega - \gamma)E_m + \omega F_m)]\omega b \]

where \( 0 \leq \gamma < \omega \leq 1 \).

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

We shall call the matrix \( G^{(m)}_{R2GAOR} = [(T_m - \gamma E_m)^{-1}((1 - \omega)T_m + (\omega - \gamma)E_m + \omega F_m)]^2 \) as refinement of generalized accelerated over relaxation iteration matrix and \( (T_m - \gamma E_m)^{-1}[(I + (T_m - \gamma E_m)^{-1}((1 - \omega)T_m + (\omega - \gamma)E_m + \omega F_m)]\omega b \) 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 \( \| x_i^{(k+1)} - x_i^{(k)} \| \leq 5 \times 10^{-7} \), where \( x_i^{(k+1)} \) and \( x_i^{(k)} \) are the computed solutions at the \((k + 1)\) and \(k\)th 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).

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

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>SOR2GNM1</th>
<th>SOR1GNM1</th>
<th>RSOR1GNM1</th>
<th>RSOR2GNM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral radius</td>
<td>0.4269416899692237</td>
<td>0.3286647942326976</td>
<td>0.1080205469680216</td>
<td>0.1822792066337767</td>
</tr>
</tbody>
</table>
Table 2: Numerical solution of example1 by SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 when m = 1

<table>
<thead>
<tr>
<th>n</th>
<th>(x_1^{(n)})</th>
<th>(x_2^{(n)})</th>
<th>(x_3^{(n)})</th>
<th>(x_4^{(n)})</th>
<th>(x_1^{(n)})</th>
<th>(x_2^{(n)})</th>
<th>(x_3^{(n)})</th>
<th>(x_4^{(n)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>22.500000</td>
<td>6.750000</td>
<td>27.000000</td>
<td>3.656250</td>
<td>31.647656</td>
<td>9.207422</td>
<td>33.539062</td>
<td>8.397509</td>
</tr>
<tr>
<td>3</td>
<td>35.100494</td>
<td>10.950096</td>
<td>36.842166</td>
<td>11.711346</td>
<td>37.318739</td>
<td>12.371311</td>
<td>37.382322</td>
<td>12.354827</td>
</tr>
<tr>
<td>5</td>
<td>37.073936</td>
<td>12.200801</td>
<td>37.493993</td>
<td>12.437939</td>
<td>37.499800</td>
<td>12.499857</td>
<td>37.499870</td>
<td>12.499839</td>
</tr>
<tr>
<td>6</td>
<td>37.318739</td>
<td>12.371311</td>
<td>37.499800</td>
<td>12.437939</td>
<td>37.499980</td>
<td>12.499857</td>
<td>37.499970</td>
<td>12.499970</td>
</tr>
<tr>
<td>7</td>
<td>37.422733</td>
<td>12.444881</td>
<td>37.499980</td>
<td>12.437939</td>
<td>37.499993</td>
<td>12.499995</td>
<td>37.499995</td>
<td>12.499995</td>
</tr>
<tr>
<td>8</td>
<td>37.467034</td>
<td>12.476434</td>
<td>37.499993</td>
<td>12.437939</td>
<td>37.499999</td>
<td>12.499995</td>
<td>37.499999</td>
<td>12.499999</td>
</tr>
<tr>
<td>13</td>
<td>37.499533</td>
<td>12.499665</td>
<td>37.499997</td>
<td>12.499624</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>37.499900</td>
<td>12.499857</td>
<td>37.499871</td>
<td>12.499839</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>37.499915</td>
<td>12.499939</td>
<td>37.499945</td>
<td>12.499931</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>37.499964</td>
<td>12.499974</td>
<td>37.499976</td>
<td>12.499971</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>37.499984</td>
<td>12.499988</td>
<td>37.499990</td>
<td>12.499987</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>37.499993</td>
<td>12.499995</td>
<td>37.499996</td>
<td>12.499995</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>37.499997</td>
<td>12.499998</td>
<td>37.499998</td>
<td>12.499998</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>37.499999</td>
<td>12.499999</td>
<td>37.499999</td>
<td>12.499999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>37.500000</td>
<td>12.500000</td>
<td>37.500000</td>
<td>12.500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>37.500000</td>
<td>12.500000</td>
<td>37.500000</td>
<td>12.500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CPU time (in seconds) = 0.050323

CPU time (in seconds) = 0.0238879
Refinement of Generalized Accelerated

<table>
<thead>
<tr>
<th>n</th>
<th>$x_1^{(n)}$</th>
<th>$x_2^{(n)}$</th>
<th>$x_3^{(n)}$</th>
<th>$x_4^{(n)}$</th>
<th>$x_1^{(n)}$</th>
<th>$x_2^{(n)}$</th>
<th>$x_3^{(n)}$</th>
<th>$x_4^{(n)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>22.500000</td>
<td>7.350000</td>
<td>29.400000</td>
<td>6.716250</td>
<td>32.876156</td>
<td>10.319469</td>
<td>35.233252</td>
<td>10.390640</td>
</tr>
<tr>
<td>2</td>
<td>32.876156</td>
<td>10.319469</td>
<td>35.233252</td>
<td>10.390641</td>
<td>37.035904</td>
<td>12.218478</td>
<td>37.275424</td>
<td>12.260334</td>
</tr>
<tr>
<td>5</td>
<td>37.349136</td>
<td>12.405276</td>
<td>37.426955</td>
<td>12.420776</td>
<td>37.499426</td>
<td>12.499630</td>
<td>37.499722</td>
<td>12.499694</td>
</tr>
<tr>
<td>6</td>
<td>37.450653</td>
<td>12.468553</td>
<td>37.476099</td>
<td>12.473899</td>
<td>37.499938</td>
<td>12.499960</td>
<td>37.499969</td>
<td>12.499967</td>
</tr>
<tr>
<td>7</td>
<td>37.483815</td>
<td>12.489620</td>
<td>37.492159</td>
<td>12.491413</td>
<td>37.499993</td>
<td>12.499995</td>
<td>37.499996</td>
<td>12.499999</td>
</tr>
<tr>
<td>9</td>
<td>37.498254</td>
<td>12.498876</td>
<td>37.499154</td>
<td>12.499072</td>
<td>37.500000</td>
<td>12.500000</td>
<td>37.500000</td>
<td>12.500000</td>
</tr>
<tr>
<td>10</td>
<td>37.499426</td>
<td>12.499630</td>
<td>37.499722</td>
<td>12.499695</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>37.499811</td>
<td>12.499879</td>
<td>37.499909</td>
<td>12.499000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>37.499938</td>
<td>12.499960</td>
<td>37.499970</td>
<td>12.499967</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>37.499980</td>
<td>12.499987</td>
<td>37.499990</td>
<td>12.499989</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>37.499993</td>
<td>12.499996</td>
<td>37.499997</td>
<td>12.499996</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>37.499998</td>
<td>12.499999</td>
<td>37.499999</td>
<td>12.499999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>37.499999</td>
<td>12.499999</td>
<td>37.499999</td>
<td>12.500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>37.500000</td>
<td>12.500000</td>
<td>37.500000</td>
<td>12.500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CPU time (in seconds) = 0.043437
CPU time (in seconds) = 0.019739
Table 3: Numerical solution of example 1 by SOR2GNM1, RSOR2GNM1, SOR1GNM1 and RSOR1GNM1 when m = 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Spectral Radius</th>
<th>Iteration Number</th>
<th>CPU time (in second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR2GNM1</td>
<td>0.2912208524</td>
<td>16</td>
<td>0.062166</td>
</tr>
<tr>
<td>SOR1GNM1</td>
<td>0.2142471721</td>
<td>13</td>
<td>0.031271</td>
</tr>
<tr>
<td>RSOR2GNM1</td>
<td>0.0840958490</td>
<td>9</td>
<td>0.025449</td>
</tr>
<tr>
<td>RSOR1GNM1</td>
<td>0.0459018507</td>
<td>7</td>
<td>0.016730</td>
</tr>
</tbody>
</table>

Example 2. 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 \( x = (x_1, \ldots, x_6)^T \) and \( b = (1, 0, 0, 0, 0)^T \) 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 = 1 of example 2

<table>
<thead>
<tr>
<th>Method</th>
<th>SOR2GNM1</th>
<th>SOR1GNM1</th>
<th>RSOR1GNM1</th>
<th>RSOR2GNM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral radius</td>
<td>0.382053999242000</td>
<td>0.286203173633144</td>
<td>0.081912256597683</td>
<td>0.145965258336806</td>
</tr>
</tbody>
</table>

Table 5: Numerical solution of example 2 by SORG2NM1, RSORG2NM1, SOR1GNM1 and RSOR1GNM1 when m = 1

<table>
<thead>
<tr>
<th>n</th>
<th>SOR2GNM1</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x_1^{(n)}</td>
<td>x_2^{(n)}</td>
</tr>
<tr>
<td>1</td>
<td>0.2410714285</td>
<td>0.0642857142</td>
</tr>
<tr>
<td>2</td>
<td>0.2748368030</td>
<td>0.0780940233</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>0.2948237169</td>
<td>0.0931672842</td>
</tr>
<tr>
<td>13</td>
<td>0.2948239025</td>
<td>0.0931675422</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>SOR1GNM1</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x_1^{(n)}</td>
<td>x_2^{(n)}</td>
</tr>
<tr>
<td>1</td>
<td>0.2410714285</td>
<td>0.0642857142</td>
</tr>
<tr>
<td>2</td>
<td>0.2825633883</td>
<td>0.0839978134</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>0.2948236728</td>
<td>0.0931672315</td>
</tr>
<tr>
<td>11</td>
<td>0.2948239191</td>
<td>0.0931675671</td>
</tr>
<tr>
<td>n</td>
<td>$x_1^{(n)}$</td>
<td>$x_2^{(n)}$</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>0.2748368030</td>
<td>0.0780940233</td>
</tr>
<tr>
<td>2</td>
<td>0.2925673346</td>
<td>0.0908550744</td>
</tr>
<tr>
<td>6</td>
<td>0.2948232279</td>
<td>0.0931666091</td>
</tr>
<tr>
<td>7</td>
<td>0.2948239025</td>
<td>0.0931675422</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>$x_1^{(n)}$</th>
<th>$x_2^{(n)}$</th>
<th>$x_3^{(n)}$</th>
<th>$x_4^{(n)}$</th>
<th>$x_5^{(n)}$</th>
<th>$x_6^{(n)}$</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2825633883</td>
<td>0.0839978134</td>
<td>0.0236727633</td>
<td>0.0801489119</td>
<td>0.0445921699</td>
<td>0.0168360213</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.2940394571</td>
<td>0.0923461701</td>
<td>0.0276790386</td>
<td>0.0857254153</td>
<td>0.0492453211</td>
<td>0.0191956668</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.2948236728</td>
<td>0.0931672315</td>
<td>0.0281570262</td>
<td>0.0861281814</td>
<td>0.0496891892</td>
<td>0.0194615237</td>
<td>0.012419</td>
</tr>
<tr>
<td>6</td>
<td>0.2948239888</td>
<td>0.0931676632</td>
<td>0.0281573229</td>
<td>0.0861283496</td>
<td>0.0496894203</td>
<td>0.0194616832</td>
<td></td>
</tr>
</tbody>
</table>
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 Tables 1, 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 SOR2GNM1 in 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


