Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(3): 6504–6519.
DOI: 10.3934/math.2023328
Received: 29 August 2022
Revised: 14 December 2022
Accepted: 20 December 2022
Published: 04 January 2023

Research article

A generalized iterative scheme with computational results concerning the systems of linear equations

Kamsing Nonlaopon[1,*], **Farooq Ahmed Shah**[2], **Khaleel Ahmed**[2] **and Ghulam Farid**[2]

[1] Department of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

[2] Department of Mathematics, COMSATS University Islamabad, Attock Campus, Pakistan

* **Correspondence:** Email: nkamsi@kku.ac.th; Tel: +66866421582; Fax: +66043202376.

Abstract: In this article, a new generalized iterative technique is presented for finding the approximate solution of a system of linear equations $Ax = b$. The efficiency of iterative technique is analyzed by implementing it on some examples, and then comparing with existing methods. A parameter introduced in the method plays very vital role for a better and rapid solution. Convergence analysis is also examined. Findings of this paper may stimulate further research in this area.

Keywords: iterative method; convergence analysis; Gauss-Seidel method; sparse matrix; AOR method
Mathematics Subject Classification: 65F10, 90C30

1. Introduction

Consider the general frame work of system of linear equations

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is the coefficients matrix, $b \in \mathbb{R}^n$ is a constant vector and $x \in R^n$ is an unknown vector. Various problems arising in different fields such as computer science, electrical engineering, mechanical engineering and economics are modeled in this general frame work of system of linear equations (1.1).

Importance of the methods for systems of linear equations can not be denied due to the requirement of solutions of systems occurring in almost all fields. Babylonians first introduced the system of linear equations with two unknowns about 4000 years ago. Later on Cramer [1] gave the idea for solving the systems of linear equations by using determinants. In the nineteenth century, Gauss introduced a method to solve the linear system (1.1) by elimination of variables one by one and later on using backward substitutions. There also exist many other methods in the literature to solve (1.1). Usually, these methods are classified into two categories, called direct and iterative methods.

The objective of a direct method is to get an exact solution in minimal number of operations. While, an iterative method starts with an initial guess and produces an infinite sequence of approximations in the direction of exact solution. This sequence can be limited by using a suitable stopping criteria. Direct methods involve the Gauss elimination method, Gauss-Jordan elimination method, Cholasky method, LU decomposition method [2]. Large and sparsely populated systems often arise in solving partial differential equations numerically or dealing with optimization problems. For such cases the conjugate gradient method is implemented and also suggested for sparse systems [3]. Direct methods are ineffective for a system consisting on a large number of equations, mostly when the coefficient matrix is sparse.

Iterative methods consist on successive approximations that are used to gain approximate solution for system (1.1) at each step, starting with a given initial approximation. Moreover, iterative methods can be further categorized into stationary and non-stationary methods. Stationary methods are older and more straightforward methods involving an iteration matrix that remains constant throughout the whole iterations during calculation. Examples of stationary iterative methods are the Jacobi method, Gauss-Seidel method, Successive Over Relaxation method [2]. The computations in non-stationary methods involve information that changes at each iteration. These iterative methods are used to derive the inner products of residuals [2].

We observe that in any iterative method the system may be represented in the form of $x = Px + c$, and the iterative scheme $x^{(k+1)} = Px^{(k)} + c$ is suggested by using an initial approximation $x^{(0)}$ to obtain the best approximate solution. The iterative method is convergent if and only if $\rho(P) < 1$, where $\rho(P)$ is spectral radius of P. In order to obtain the iterative scheme we partition $A = (a_{ij})$ as $A = D - L - U$, where $D = diag(a_{ii})$, L and U are strictly, lower and upper triangular matrices respectively.

Jacobi method and Gauss-Seidel method are the classical methods which are used for the diagonally dominant systems by spiting the coefficient matrix into three matrices. In Jacobi method, the iterative scheme [2] can be expressed as:

$$x^{(k)} = D^{-1}(L + U)x^{(k-1)} + D^{-1}b, \tag{1.2}$$

and similarly, for Gauss-Seidel method [2], the iterative scheme is suggested as:

$$x^{(k)} = (D - L)^{-1}Ux^{(k-1)} + (D - L)^{-1}b. \tag{1.3}$$

If the coefficient matrix A is strictly diagonally dominant, the Jacobi and Gauss-Seidel methods converge for any x_0. However Gauss-Seidel method converges rapidly as compare to Jacobi method [2, 4].

The Successive Over-Relaxation (SOR) techniques

$$x^{(k)} = (D - wL)^{-1}((1 - w)D + wU)x^{(k-1)} + w(D - wL)^{-1}b, \tag{1.4}$$

are nicely addressed in literature [2, 5, 6]. Requirement for the parameter w for SOR is that it lies between zero and two and for each particular matrix the optimal value of w is discussed very comprehensively [7].

In 1978, the accelerated over-relaxation (AOR) method was initially presented by Hadjidimos as a modification of the successive over-relaxation (SOR) method with two parameters [8]. In mostly cases, the AOR technique improves the Jacobi, Gauss-Seidel, and SOR methods [8–11]. Significance of AOR method can be seen in [9, 12–14]. For the convergence of AOR method sufficient conditions are discussed [15–19]. Various aspects of applications of AOR method can also be studied in [20–23]. We also see in literature the preconditioned AOR technique to improve the convergence rate of AOR method [24–29]. While Krylov subspace techniques [3, 30–32] are recognized as one of the most significant and effective iterative approaches to solve the sparse linear systems because they are inexpensive to be implemented and are able to fully exploit the sparsity of the coefficient matrix. Krylov subspace techniques are extremely slow or fail to converge when the coefficient matrix of the system is ill-conditioned and excessively indefinite which is the drawback of these schemes.

The purpose of this paper is to present a new iterative method for solving the systems of linear equations (1.1), which is the generalization of existing methods and fast convergent than the Jacobi, Gauss-Seidel, SOR, and AOR methods. In Section 2, generalized iterative scheme is developed for the best approximate solution. In Section 3, convergence of the proposed iterative scheme is discussed. Numerical and graphical results are discussed in Section 4.

2. Development of iterative scheme

In this section, we construct a generalized iterative scheme for solving the system of linear equations (1.1). Jacobi method, Gauss-Seidel method, SOR method, and AOR method are the special cases for this presented scheme.

System (1.1) can be written as:

$$wAx = wb, \tag{2.1}$$

where $0 < w < 2$ and

$$w(D - L - U)x = bw. \tag{2.2}$$

We split matrix A as sum of three matrices D, L and U. Here, D is a diagonal matrix, L is the strictly lower triangular matrix, and U is the strictly upper triangular matrix.

Above Eq (2.2) can be re-written as:

$$(D - rL - tU)x = [(1 - w)D + (w - r)L + (w - t)U]x + bw. \tag{2.3}$$

Now (2.3) can be expressed as:

$$x = (D - rL - tU)^{-1}[(1 - w)D + (w - r)L + (w - t)U]x + (D - rL - tU)^{-1}bw, \tag{2.4}$$

where $0 < t < w < r < 2$.

Relation (2.4) is a fixed point formulation which allows us to suggest the following iterative scheme.

Algorithm 2.1. For a given initial vector $x^{(0)}$, find the approximate solution $x^{(k)}$ from the following iterative scheme:

$$x^{(k)} = (D - rL - tU)^{-1}[(1 - w)D + (w - r)L + (w - t)U]x^{(k-1)} + (D - rL - tU)^{-1}bw, k = 1, 2, 3, ...$$

Algorithm 2.1 is the main iterative scheme that converges to the solution rapidly as compared with other methods. This is the generalized scheme for obtaining the solution of a system of linear equations. We present some special cases.

If $t = 0$, Algorithm 2.1 reduces to the following iterative scheme.

Algorithm 2.2. For a given initial vector $x^{(0)}$, find the approximate solution $x^{(k)}$ from the following technique:

$$x^{(k)} = (D - rL)^{-1}[(1 - w)D + (w - r)L + wU]x^{(k-1)} + (D - rL)^{-1}bw, k = 1, 2, 3, ...$$

which is well-known AOR method [2, 3].

If $t = 0$ and $w = r$, the Algorithm 2.1 reduces to the following SOR method [2, 3].

Algorithm 2.3. For a given initial vector $x^{(0)}$, find the approximate solution $x^{(k)}$ from the following technique:

$$x^{(k)} = (D - wL)^{-1}[(1 - w)D + wU]x^{(k-1)} + (D - wL)^{-1}bw, k = 1, 2, 3, ...$$

If $t = 0$ and $w = r = 1$, the Algorithm 2.1 reduces to the following scheme.

Algorithm 2.4. For a given initial vector $x^{(0)}$, find the approximate solution $x^{(k)}$ from the following technique:

$$x^{(k)} = (D - L)^{-1}Ux^{(k-1)} + (D - L)^{-1}b, k = 1, 2, 3, ...$$

Algorithm 2.4 is Gauss-Seidel method [2, 3].

If $t = r = 0$ and $w = 1$, the Algorithm 2.1 reduces to the following scheme.

Algorithm 2.5. For a given initial vector $x^{(0)}$, find the approximate solution $x^{(k)}$ from the following technique:

$$x^{(k)} = D^{-1}(L + U)x^{(k-1)} + D^{-1}b, k = 1, 2, 3, ...$$

Algorithm 2.5 is well-known Jacobi method [2, 3].

3. Convergence analysis

In this section, we consider the convergence analysis of the newly developed iterative scheme mentioned as Algorithm 2.1.

$$x^{(k)} = (D - rL - tU)^{-1}[(1 - w)D + (w - r)L + (w - t)U]x^{(k-1)} + (D - rL - tU)^{-1}bw.$$

Lemma 3.1. [2] *If the spectral radius satisfies*

$$\rho[(D - rL - tU)^{-1}(1 - w)D + (w - r)L + (w - t)U)] \leq 1,$$

then

$$\left[I - (D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]^{-1}$$

exists and

$$\begin{aligned}
&\left[I - (D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]^{-1} \\
&= I + \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] \\
&\quad + \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]^{2} + \cdots \\
&= \sum_{j=0}^{\infty} \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]^{j}.
\end{aligned} \tag{3.1}$$

Theorem 3.2. *For a given any $x^{(0)} \in R^n$, the sequence $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ defined by*

$$x^{(k)} = \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] x^{(k-1)} + (D - rL - tU)^{-1}bw,$$

for each $k \geq 1$, converges to the unique solution

$$x = \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] x + (D - rL - tU)^{-1}bw,$$

if and only if

$$\rho\left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] < 1.$$

Proof. For the proof of the statement, it is enough to show that spectral radius of iteration matrix < 1. For this, let us consider the iterative scheme suggested in Algorithm 2.1.

$$x^{(k)} = \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] x^{(k-1)} + (D - rL - tU)^{-1}bw,$$

which can be rewritten as:

$$\begin{aligned}
x^{(k)} &= \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]\Big[((D - rL - tU)^{-1}((1 - w)D \\
&\quad +(w - r)L + (w - t)U))x^{(k-2)} + (D - rL - tU)^{-1}bw\Big] + (D - rL - tU)^{-1}bw \\
&\ \vdots \\
&= \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]^{k} x^{(0)} \\
&\quad + \Big[\left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] x^{(k-1)} + \cdots \\
&\quad + \left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right] + I\Big](D - rL - tU)^{-1}bw.
\end{aligned} \tag{3.2}$$

Since

$$\rho\left(\left[(D - rL - tU)^{-1}((1 - w)D + (w - r)L + (w - t)U)\right]\right) \leq 1,$$

the matrix converges and

$$\lim_{k\to\infty}\left[(D-rL-tU)^{-1}((1-w)D+(w-r)L+(w-t)U)\right]^k x^{(0)}=0,$$

and Lemma 3.1 implies that

$$\begin{aligned}\lim_{k\to\infty}x^{(k)}&=0+\lim_{k\to\infty}\left[\sum_{j=0}^{k-1}\left[(D-rL-tU)^{-1}((1-w)D+(w-r)L+(w-t)U)\right]^j\right](D-rL-tU)^{-1}bw\\&=\left[I-\left[(D-rL-tU)^{-1}((1-w)D+(w-r)L+(w-t)U)\right]\right]^{-1}(D-rL-tU)^{-1}bw.\end{aligned}$$

As a result, the sequence $x^{(k)}$ converges to the vector

$$x=\left[I-(D-rL-tU)^{-1}((1-w)D+(w-r)L+(w-t)U)\right]^{-1}(D-rL-tU)^{-1}bw.$$

□

We can also view the convergence criteria of the purposed method as an application of the Banach fixed point theorem [33]. System of linear equations can be described with the relations of parameters in the equation form as:

$$\begin{cases}x_1=(1-wa_{11})x_1-(w-2t)a_{12}x_2-\ldots-(w-2t)a_{1n}x_n+wb_1\\x_2=-(w-2r)a_{21}x_1+(1-wa_{22})x_2-\ldots-(w-2t)a_{2n}x_n+wb_2\\\vdots\\x_n=-(w-2r)a_{n1}x_1-(w-2r)a_{n2}x_2-\ldots+(1-wa_{nn})x_n+wb_n.\end{cases}\tag{3.3}$$

This system is equivalent to

$$x=cx+d\tag{3.4}$$

with $d=wb$ and $c_{ij}=\begin{cases}(1-wa_{ij}) & if\quad i=j\\-(w-2t)a_{ij} & if\quad i<j.\\-(w-2r)a_{ij} & if\quad i>j\end{cases}$

The solution can be obtained by

$$x^{(k+1)}=cx^{(k)}+d.\tag{3.5}$$

The iteration method is defined by

$$x_j^{(k+1)}=\frac{1}{c_{jj}}(\gamma-\sum_{k=1,k\neq j}^{n}c_{jk}x^{(k)}).\tag{3.6}$$

Assuming that $c_{jj}\neq 0$ for $j=1,...n$. This iteration is suggested for the jth equation of the system. It is not difficult to verify that (3.6) can be written in the form of

$$c=(D-rL-tU)^{-1}[(1-w)D+(w-r)L+(w-t)U],\tag{3.7}$$

and

$$d = (D - rL - tU)^{-1} wb. \tag{3.8}$$

Here D=diag(c_{jj}) is the diagonal matrix whose non-zero elements are of those of the principle diagonal of A. Condition of diagonally dominant applied to c is sufficient for the convergence of Algorithm 2.1. We can express directly in terms of the elements of A. The result is the row sum criteria for the convergence will be

$$\sum_{k=1,k\neq j}^{n} \left|\frac{a_{jk}}{a_{jj}}\right| < 1, \tag{3.9}$$

or

$$\sum_{k=1,k\neq j}^{n} |a_{jk}| < |a_{jj}|. \tag{3.10}$$

This shows that convergence is guaranteed, if the elements in principle diagonal of A are sufficiently large.

Note that all the components of a new approximation are introduced simultaneously at the end of an iteration cycle.

4. Numerical results

In this section, we provide few numerical applications to clarify the efficiency of new developed three parameter iterative scheme Algorithm 2.1, on some system of linear equations for $0 < t < w < r < 2$, whose coefficient matrices satisfy

$$\max_{1\leq i\leq n-1} u_i = \alpha \text{ and } \max_{1\leq i\leq n-1} l_i = \beta, \quad \alpha + \beta \leq 1,$$

where

$$l_i = \max \sum_{j=1}^{i-1} \mid \beta_{ij} \mid, \text{ for } i = 2, 3, \dots, n,$$

and

$$u_i = \max \sum_{j=i+1}^{n} \mid \alpha_{ij} \mid, \text{ for } i = 1, 2, \dots, n-1.$$

In this part, we will compare our developed scheme with previous techniques as namely AOR method, SOR method, Jacobi method and Gauss-Seidel method. All computations are calculated by using computer programming by MATLAB. We use $\varepsilon = 10^{-15}$ and the following stopping criteria is used for computer programs as:

$$\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|} \leq \varepsilon.$$

This stopping criteria is deduced from relative error and the infinite sequence generated by the computer code will be chopped at the stage when this criteria is satisfied. We assume the following examples to compare the new developed method Algorithm 2.1 (Alg 2.1) with various iterative methods AOR (Alg 2.2), SOR (Alg 2.3), Gauss-Seidel (Alg 2.4) and Jacobi (Alg 2.5), to analyze the new iterative scheme's feasibility and effectiveness.

For the numerical and graphical comparison of methods, we select some examples from the literature.

Example 4.1. *[3] We consider a problem where the loop-current approach is combined with Ohm's law and Kirchhoff's voltage law. Each loop in the network is supposed to be circulated by a loop current. Thus, the loop current I_1 cycles the closed-loops $a, b, c,$ and d in the network shown in Figure 1. As a result, the current $I_1 - I_2$ passes via the link joining b and c.*

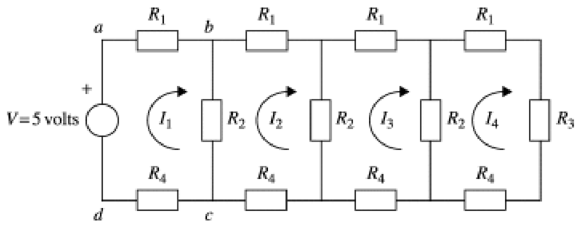

Figure 1. Network of loop-current.

From the above network as shown in Figure 1, we get a four-variable linear equations system by letting $R_1 = R_4 = 1\Omega,\ R_2 = 2\Omega,\ R_3 = 4\Omega$ and $V = 5 volts$. We get the following system of the form

$$\begin{aligned} 4I_1 - 2I_2 &= 5,\\ -2I_1 + 6I_2 - 2I_3 &= 0,\\ -2I_2 + 6I_3 - 2I_4 &= 0,\\ -2I_3 + 8I_4 &= 0. \end{aligned}$$

Table 1 displays the numerical results for Example 4.1, which indicate that Alg 2.1 is more efficient than the other methods.

Table 1. Tabular comparison.

Methods	Parameters	Iterations	Relative error
Alg 2.1	$w = 1.02, r = 1.05, t = 0.88$	14	$8.9966e^{-18}$
Alg 2.2	$w = 1.02, r = 1.05$	28	$2.8789e^{-16}$
Alg 2.3	$w = 1.02$	30	$2.8789e^{-16}$
Alg 2.4	...	32	$4.3184e^{-16}$
Alg 2.5	...	61	$7.1973e^{-16}$

In Figure 2 the residual fall of different methods shows that new method is faster convergent than the other methods. Figure 3 is the comparison of iterations of different algorithms that shows our new iterative method which described in Alg 2.1 is more efficient than other methods described in Alg 2.2–2.5.

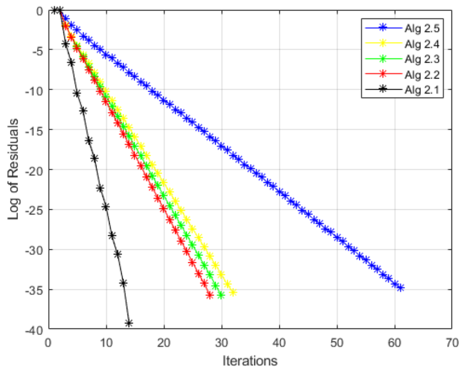

Figure 2. Log of residual.

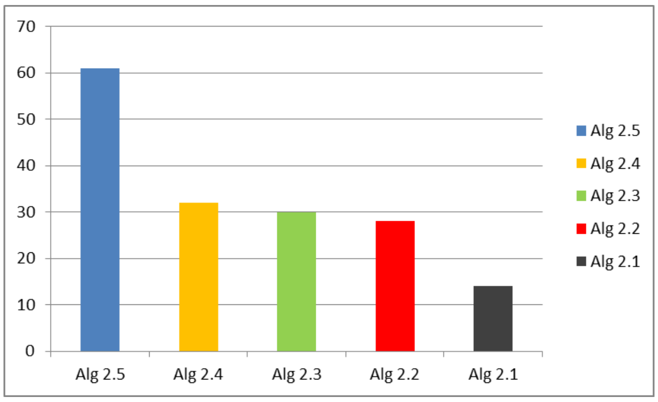

Figure 3. Comparison of iterations.

Example 4.2. [34] *Consider the following system of the form*

$$\begin{aligned} x_1 + 0.250x_2 &= 0.75,\\ 0.250x_1 + x_2 + 0.250x_3 &= 1.50,\\ 0.250x_2 + x_3 + 0.250x_4 &= 1.50,\\ 0.250x_3 + x_4 + 0.250x_5 &= 1.50,\\ 0.250x_4 + x_5 &= 1.25. \end{aligned}$$

Table 2 displays the numerical results which indicate that Alg 2.1 is more efficient than the other techniques.

Table 2. Tabular comparison.

Methods	Parameters	Iterations	Relative error
Alg 2.1	$w = 1.01, r = 1.06, t = 0.86$	13	$5.8249e^{-16}$
Alg 2.2	$w = 1.01, r = 1.06$	19	$4.8541e^{-16}$
Alg 2.3	$w = 1.01$	23	$2.4271e^{-16}$
Alg 2.4	...	24	$2.4271e^{-16}$
Alg 2.5	...	44	$7.7666e^{-16}$

The residual fall of different technique can be seen in Figure 4 which illustrate that the new method is rapidly convergent than the other methods. Figure 5 is the comparison of iterations of different algorithms that shows our new iterative method which described in Alg 2.1 is more efficient than other methods described in Alg 2.2–2.5.

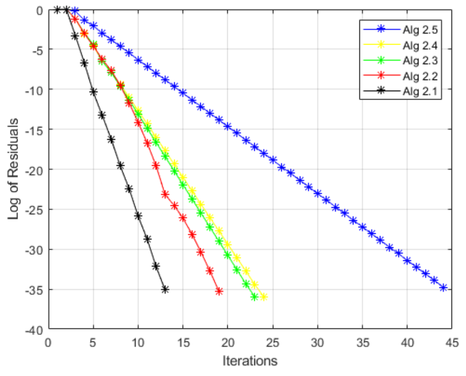

Figure 4. Log of residual.

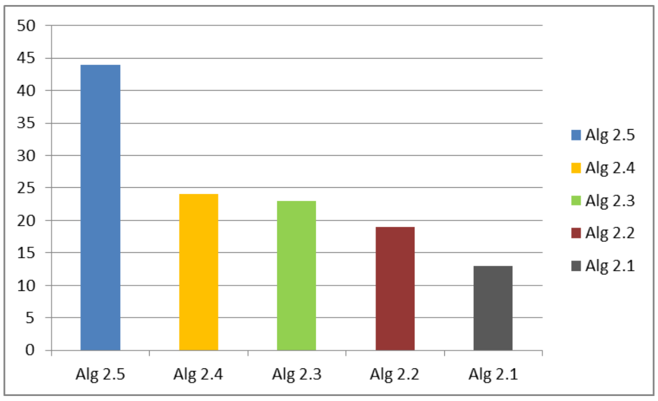

Figure 5. Comparison of iterations.

Example 4.3. [2] *Consider the following system of linear equations of the form*

$$\begin{aligned} 4x_1 - x_2 - x_3 &= 1, \\ -x_1 + 4x_2 - x_4 &= 1, \\ -x_1 + 4x_3 - x_4 &= 1, \\ -x_2 - x_3 + 4x_4 &= 1. \end{aligned}$$

Table 3 displays the numerical results for Example 4.5, which indicate that Alg 2.1 is more efficient than the other methods.

Table 3. Tabular comparison.

Methods	Parameters	Iterations	Relative error
Alg 2.1	$w = 1.05; r = 1.07; t = 0.9$	15	$5.5511e^{-16}$
Alg 2.2	$w = 1.05; r = 1.07$	19	$9.9920e^{-16}$
Alg 2.3	$w = 1.05$	22	$5.5511e^{-16}$
Alg 2.4	....	28	$4.4409e^{-16}$
Alg 2.5	....	51	$8.8818e^{-16}$

In Figure 6 the residual fall of different methods shows that New method is faster convergent than the other methods. Figure 7 is the comparison of iterations of different algorithms that shows our new iterative method which described in Alg 2.1 is more efficient than other methods described in Alg 2.2–2.5.

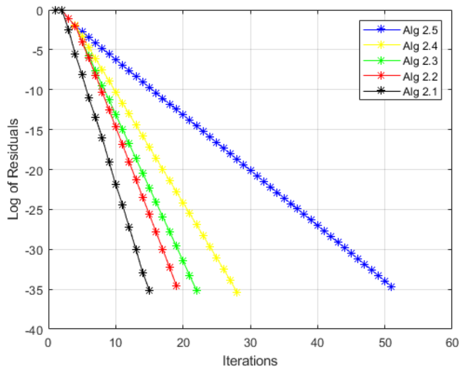

Figure 6. Log of residual.

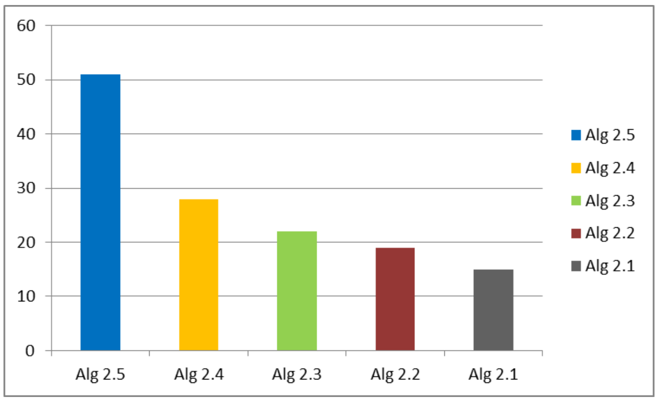

Figure 7. Comparison of iterations.

Example 4.4. [35] *Let the matrix A be given by*

$$a_{i,j} = \begin{cases} 8, & \text{if } j = i; \\ -1, & \text{if } \begin{cases} j = i+1, \text{ for } i = 1, 2, \dots, n-1; \\ j = i-1, \text{ for } i = 2, 3, \dots, n; \end{cases} \\ 0, & \text{otherwise.} \end{cases}$$

Let $b = (6, 5, 5, \dots, 5, 6)^T$, we take $n = 100$.

Table 4 displays the numerical results for Example 4.4, which indicate that Alg 2.1 is more efficient than the other techniques.

Table 4. Tabular comparison.

Methods	Parameters	Iterations	Relative error
Alg 2.1	$w = 1.02; r = 0.97; t = 0.50$	15	$3.8978e^{-16}$
Alg 2.2	$w = 1.02; r = 0.97$	19	$7.7956e^{-16}$
Alg 2.3	$w = 1.02$	19	$3.8978e^{-16}$
Alg 2.4	....	20	$5.1970e^{-16}$
Alg 2.5	....	27	$6.4963e^{-16}$

The residual fall of different methods can be seen in Figure 8 which illustrate that the new method is rapidly convergent than the other methods. Figure 9 is the comparison of iterations of different algorithms that shows our new iterative method which described in Alg 2.1 is more efficient than other methods described in Alg 2.2–2.5.

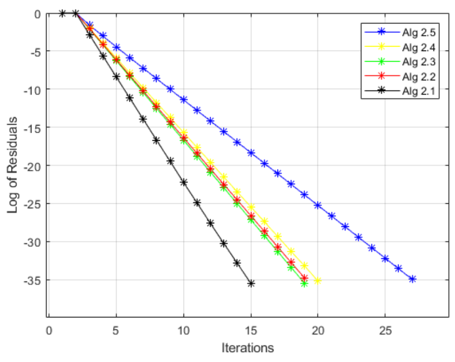

Figure 8. Log of residual.

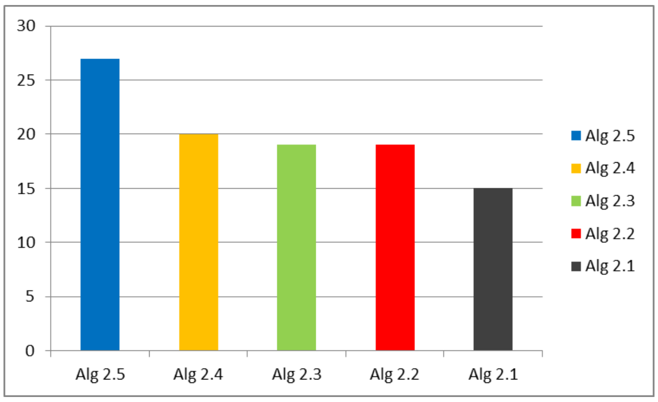

Figure 9. Comparison of iterations.

Example 4.5. [2,36] *Consider the system* (1.1), *having co-efficient matrix A is given by*

$$a_{ij} = \begin{cases} 2i, & \text{if } i = j \text{ and } i = 1, 2, \ldots, 1000; \\ -1, & \text{if } \begin{cases} j = i + 1, & \text{for } i = 1, 2, \ldots, 999; \\ j = i - 1, & \text{for } i = 2, 3, \ldots, 1000; \end{cases} \\ 0, & \text{otherwise.} \end{cases}$$

and $b_i = 1.5i - 6$ *for each* $i = 1, 2, \ldots, 1000$.

Table 5 shows the numerical results for Example 4.3, which indicate that Alg 2.1 is much more efficient than the other techniques.

Table 5. Tabular comparison.

Methods	Parameters	Iterations	Relative error
Alg 2.1	$w = 1.021; r = 1.079; t = 0.98$	13	$3.6092e^{-17}$
Alg 2.2	$w = 1.021; r = 1.079$	18	$4.3310e^{-16}$
Alg 2.3	$w = 1.0219$	20	$2.8873e^{-16}$
Alg 2.4	....	22	$5.7747e^{-16}$
Alg 2.5	....	41	$5.7747e^{-16}$

The residual fall of different techniques can be seen in Figure 10 which illustrate that the new method is rapidly convergent than the other methods. Figure 11 is the comparison of iterations of different algorithms that shows our new iterative method which described in Alg 2.1 is more efficient than other methods described in Alg 2.2–2.5.

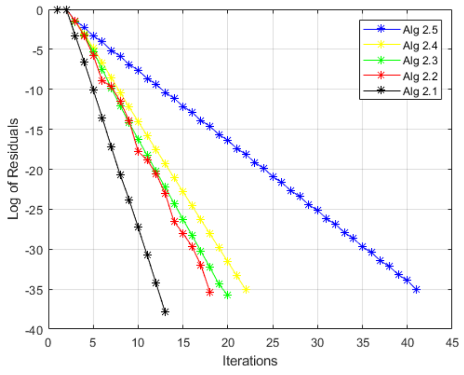

Figure 10. Log of residual.

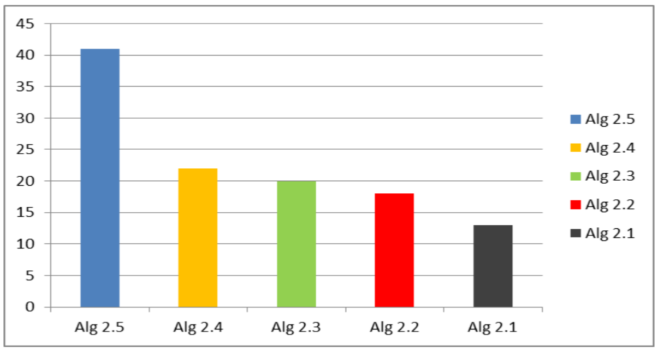

Figure 11. Comparison of iterations.

In Table 6, IT stands for the number of iterations in above tabular comparison which shows that our new iterative method work much effectively.

Table 6. Comparison table for Algorithm 2.1 with various combinations of parameters.

Parameters			Example 4.1	Example 4.2	Example 4.3	Example 4.4	Example 4.5
w	r	t	IT	IT	IT	IT	IT
0.2	0.7	0.9	186	160	181	160	169
0.4	0.5	0.8	102	82	96	77	86
0.6	0.5	0.8	63	50	58	46	52
0.3	0.8	0.5	138	111	132	108	120
0.2	0.8	0.3	229	186	217	174	195
0.3	0.8	1.2	96	97	97	97	95
0.3	0.8	0.2	155	124	145	114	129
0.5	0.8	0.3	85	67	79	61	70
0.5	0.8	0.5	77	61	73	59	66
0.8	0.4	0.4	57	40	50	33	42
0.8	0.5	0.7	46	34	41	30	36
0.9	0.5	0.8	36	27	32	23	28
0.9	1.04	0.5	24	21	22	21	21
1.02	1.08	0.8	15	14	14	12	14
1.03	1.09	0.9	14	13	14	12	13

5. Conclusions

In this article, a new generalized iterative scheme is suggested for solving systems of linear equations. We have studied the convergence criteria of this iterative scheme. This scheme is not only the generalized one but also give good results as compared to the existing schemes. This iterative scheme is also suitable for sparse matrices. Numerical results show that this scheme is more effective than the conventional schemes. We would also like to purpose that the given scheme can be extended for the absolute value problems of the type $Ax + B|x| = b$.

Conflict of interest

All authors declare no conflicts of interest in this paper.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their constructive comments and suggestions, which improved the quality of this paper.

References

1. G. Cramer, Introduction a l'analyse des lignes courbes algébriques, *A Geneva: Fréres Cramer and Cl. Philibert,* 1730.
2. R. L. Burden, J. D. Faires, *Numerical analysis*, Boston: PWS, 1980.
3. Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003. https://doi.org/10.1137/1.9780898718003
4. D. K. Salkuyeh, Generalized Jacobi and Gauss-Seidel methods for solving linear system of equations, *Numer. Math. J. Chin. Univ.*, **16** (2007), 164–170.
5. R. S. Varga, *Iterative analysis*, Berlin: Springer, 1962.
6. D. M. Young, *Iterative Solution of Large Linear Systems*, Elsevier, 2014.
7. C. E. Froberg, *Numerical Mathematics: Theory and computer applications*, Basic Books, 1985.
8. A. Hadjidimos, Accelerated overrelaxation method, *Math. Comput.*, **32** (1978), 149–157. http://doi.org/10.2307/2006264
9. G. Avdelas, A. Hadjidimos, A. Yeyios, Some theoretical and computational results concerning the accelerated overrelaxation (AOR) method, *Math. Rev. Anal. Numér. Théor. Approximation*, **9** (1980), 5–10.
10. A. I. Faruk, A. Ndanusa, Improvements of successive overrelaxation iterative (SOR) method for L-matrices, *SJBAS*, **1** (2020), 218–223.
11. Z. Mayaki, A. Ndanusa, Modified successive overrelaxation (SOR) type methods for M-matrices, *Sci. World J.*, **14** (2019), 1–5.
12. K. Audu, Y. Yahaya, K. Adeboye, U. Abubakar, A. Ndanusa, Triple accelerated over-relaxation method for system of linear equations, *Int. J. Math. Educ. Sci. Technol.*, **16** (2020), 137–146.
13. Z. Z. Bai, The monotone convergence rate of the parallel nonlinear AOR method, *Comput. Math. Appl.*, **31** (1996), 1–8. https://doi.org/10.1016/0898-1221(96)00013-2
14. Z. Z. Bai, Asynchronous multisplitting AOR methods for a class of systems of weakly nonlinear equations, *Appl. Math. Comput.*, **98** (1999), 49–59. https://doi.org/10.1016/S0096-3003(97)10154-0
15. R. Ali, I. Khan, A. Ali, A. Mohamed, Two new generalized iteration methods for solving absolute value equations using m-matrix, *AIMS Mathematics*, **7** (2022), 8176–8187. https://doi.org/10.3934/math.2022455

16. L. Cvetkovic, V. Kostic, A note on the convergence of the AOR method, *Appl. Math. Comput.*, **194** (2007), 394–399. https://doi.org/10.1016/j.amc.2007.04.030

17. M. Fallah, S. Edalatpanah, On the some new preconditioned generalized AOR methods for solving weighted linear least squares problems, *IEEE*, **8** (2020), 33196–33201. https://doi.org/10.1007/s40314-016-0350-8

18. Z. X. Gao, T. Z. Huang, Convergence of AOR method, *Appl. Math. Comput.*, **176** (2006), 134–140. https://doi.org/10.1016/j.amc.2005.09.020

19. F. Hailu, G. G. Gonfa, H. M. Chemeda, Second degree generalized successive over relaxation method for solving system of linear equations, *MEJS*, **2** (2020), 60–71. https://doi.org/10.4314/mejs.v12i1.4

20. V. Kumar Vatti, G. Chinna Rao, S. S. Pai, Parametric Accelerated Over Relaxation (PAOR) method, *Adv. Intell. Syst. Comput.*, **979** (2020), 283–288. https://doi.org/10.1007/978-981-15-3215-3-27

21. W. Li, W. Sun, Comparison results for parallel multisplitting methods with applications to AOR methods, *Linear Algebra Appl.*, **331** (2001), 131–144. https://doi.org/10.1016/S0024-3795(01)00276-2

22. A. Yeyios, A necessary condition for the convergence of the accelerated overrelaxation (AOR) method, *J. Comput. Appl. Math.*, **26** (1989), 371–373. https://doi.org/10.1016/0377-0427(89)90309-9

23. J. Y. Yuan, X. Q. Jin, Convergence of the generalized AOR method, *Appl. Math. Comput.*, **99** (1999), 35–46. https://doi.org/10.1016/S0096-3003(97)10175-8

24. Y. T. Li, C. X. Li, S. L. Wu, Improvements of preconditioned AOR iterative method for L-matrices, *J. Comput. Appl. Math.*, **206** (2007), 656–665. https://doi.org/10.1016/j.cam.2006.08.019

25. Y. T. Li, C. X. Li, S. L. Wu, Improving AOR method for consistent linear systems, *Appl. Math. Comput.*, **186** (2007), 379–388. https://doi.org/10.1016/j.amc.2006.07.097

26. Z. Q. Wang, Optimization of the parameterized Uzawa preconditioners for saddle point matrices, *J. Comput. Appl. Math.*, **226** (2009), 136–154. https://doi.org/10.1016/j.cam.2008.05.019

27. M. Wu, L. Wang, Y. Song, Preconditioned AOR iterative method for linear systems, *Appl. Numer. Math.*, **57** (2007), 672–685. https://doi.org/10.1016/j.apnum.2006.07.029

28. S. Wu, T. Huang, A modified AOR-type iterative method for L-matrix linear systems, *ANZIAM*, **49** (2007), 281–292. https://doi.org/10.1017/S1446181100012840

29. J. H. Yun, Comparison results of the preconditioned AOR methods for L-matrices, *Appl. Math. Comput.*, **218** (2011), 3399–3413. https://doi.org/10.1016/j.amc.2011.08.085

30. J. W. Pearson, J. Pestana, Preconditioners for Krylov subspace methods: An overview, *GAMM-Mitt.*, **43** (2020), e202000015. https://doi.org/10.1002/gamm.202000015

31. Z. Z. Bai, Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems, *Appl. Math. Comput.*, **109** (2000), 273–285.

32. R. Kehl, R. Nabben, D. B. Szyld, Adaptive multilevel Krylov methods, *Electron. Trans. Numer. Anal.*, **51** (2019).

33. E. Kreyszig, *Introductory Functional analysis with applications*, Wiley, 1991.

34. M. Darivishi, The best values of parameters in accelerated successive overrelaxation methods, *WSEAS Trans. Math.*, **3** (2004), 505–510.

35. M. A. Noor, J. Iqbal, K. I. Noor, E. Al-Said, On an iterative method for solving absolute value equations, *Optim. Lett.*, **6** (2012), 1027–1033. https://doi.org/10.1007/s11590-011-0332-0

36. M. A. Noor, K. I. Noor, M. Waseem, A new decomposition technique for solving a system of linear equations, *J. Assoc. Arab Univ. Basic Appl. Sci.*, **16** (2014), 27–33. http://doi.org/10.1016/j.jaubas.2013.07.001