Journal of Indian Acad. Math. ISSN: 0970-5120
Vol. 46, No. 2 (2024) pp. 225-246

*Urboshi Hazarika*[1]
*and*
*Samira Behera*[2]

# ACCELERATING SOLUTIONS OF SYSTEMS OF LINEAR EQUATIONS: THE SECOND REFINEMENT OF GENERALIZED GAUSS-SEIDEL (SRGGS) METHOD

**Abstract:** In this paper, we introduce a Second Refinement of Generalized Gauss-Seidel (SRGGS) method, an enhancement of the classical Gauss-Seidel (GS) method. We discuss the convergence properties of the SRGGS method for various matrices, including Strictly Diagonally Dominant (SDD), Symmetric Positive Definite (SPD), and M-matrices. To assess its effectiveness, numerical experiments have been conducted. Comparison of the SRGGS method with several established techniques, including Gauss-Seidel (GS), Refinement of Gauss-Seidel (RGS), Second Refinement of Gauss-Seidel (SRGS), Generalized Gauss-Seidel (GGS), and Refinement of Generalized Gauss-Seidel (RGGS) methods has also been made. The outcomes of these numerical experiments affirm that the SRGGS method reduces the number of iterations, minimizes the spectral radius, and increases the convergence rate.

**Keywords:** Iterative Methods, Linear Systems, Generalized Gauss-Seidel (GGS), Refinement of Generalized Gauss-Seidel (RGGS), Second Refinement of Generalized Gauss-Seidel (SRGGS).

**Mathematics Subject Classification (2020) No.:** 65F10, 65F15, 65N22, 65F50, 97N40.

## 1. Introduction

Linear equations are commonly solved using direct or iterative methods. Direct methods become challenging when handling large, sparse systems. In contrast,

iterative methods offer approximate solutions for systems of any size and prove efficient in terms of computer storage and time requirements. For a considerable time, researchers have been deeply intrigued by the efficiency and convergence behavior of iterative techniques in this pursuit. Carl Gustav Jacob Jacobi (1804-1851) introduced the first iterative technique to solve linear equations, later it was named after him. In the mid-1820s, Gauss, and later in 1874, Philipp Ludwig von Seidel (1821-1896) jointly developed the Gauss-Seidel iteration method. Gauss-Seidel method converges for matrices that satisfy the conditions of being Strictly Diagonally Dominant (SDD) or Irreducibly Diagonally Dominant [15], Symmetric Positive Definite (SPD) [8], and M-matrices [15].

In 2007, D. K. Salkuyeh [14] introduced a generalization of Gauss-Seidel (GGS) method by incorporating the concept of a banded matrix, which is convergent for SDD and M-matrices. However, the GGS method is not convergent for SPD matrices [16]. In 2014, Laskar and Behera [13] presented the Refinement of Generalized Gauss-Seidel (RGGS) method and it is convergent for the SDD matrix [13] and M-matrix [10].

Recent advancements have introduced further improvements to the Gauss-Seidel method. For example, Bai *et al*. (2021) proposed a randomized Gauss-Seidel method that accelerates convergence for specific matrices [3]. The Parallel Jacobi-Embedded Gauss-Seidel (PJG) method, developed by Ahmadi *et al*. (2021), enhances computational efficiency through parallelism, particularly on modern hardware [1]. Xie and Miao (2023) introduced a preconditioner for the Gauss-Seidel method, significantly improving convergence in multi-linear systems with M-tensors [17]. Additionally, Ding *et al*. (2024) developed structure-preserving algorithms for Gauss-Seidel and SOR methods, which improved computational accuracy and efficiency for quaternion linear systems [7].

Further refinements to the Gauss-Seidel method include the Generalized Refinement of Gauss-Seidel method, improving stability and convergence when applied to consistently ordered 2-cyclic matrices (Dessalew *et al*., 2021) [6]. Moreover, Ali (2022) developed a generalized version of the Gauss-Seidel method to handle non-square linear systems, broadening its applicability and improving convergence for complex problems [12].

The Gauss-Seidel method has been successfully applied to various real-life problems. For instance, Calvin and Choi (2024) used the Chebyshev Rational Approximation Method (CRAM) to solve nuclear fuel depletion systems, yielding faster performance and greater accuracy [5]. It has also been applied to image reconstruction in Diffuse Optical Tomography (DOT), where it improved computational efficiency and regularization parameter estimation (Hussain *et al*.,

2024) [11]. Additionally, the method has demonstrated effectiveness in solving complex linear systems in fields such as computational physics and engineering, improving iterative efficiency and enhancing convergence [6].

In this paper, we introduce the Second Refinement of Generalized Gauss-Seidel (SRGGS) method designed for solving linear systems,

$$A x = b, \tag{1.1}$$

Here $A = (a_{ij})_{n\times n}$ represents a non-singular matrix with real entries, while $b$ and $x$ are column vectors.

This paper is structured as follows: Section 2, introduces key preliminary definitions, theorems, and results. In Section 3, we thoroughly examine the Generalized Gauss-Seidel method, focusing on its convergence properties. Section 4, provides an in-depth examination of the Refinement of Generalized Gauss-Seidel method, complete with the presentation of its convergence theorems. Section 5, introduces the novel Second Refinement of Generalized Gauss-Seidel method, exploring its convergence properties. Finally, Section 6, showcases a series of numerical examples to illustrate the practical effectiveness of this newly introduced method.

## 2. Preliminaries

Let $A = (a_{ij})$ be an $n \times n$ matrix then $D_m = (d_{ij})$ is the banded matrix of bandwidth $2m + 1$, if

$$d_{ij} = \begin{cases} a_{ij}, & |i - j| \le m, \\ 0, & otherwise. \end{cases}$$

Let us consider the splitting of matrix $A$ as

$$A = D_m + L_m + U_m \tag{2.1}$$

where $L_m$ and $U_m$ are strictly lower and strictly upper part of matrix $A - D_m$, respectively.

That is,

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

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

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

**Definition 2.1** Spectral radius of a matrix ([4]): *Let $A$ be an $n \times n$ matrix. The spectral radius $\rho(A)$ of a matrix $A$ is defined as,*

$$\rho(A) = \max_{1 \le i \le n} |\lambda_i|,$$

*where $\lambda_i$'s are the eigenvalues of a matrix $A$.*

**Lemma 2.1** ([9]): *The Spectral radius satisfies the following rule*:

$$\rho(A^k) = (\rho(A))^k \quad \text{for all } k \in \mathbb{N} \text{ and } A \in \mathbb{C}^{n \times n}.$$

**Definition 2.2** Strictly Diagonally Dominant (SDD) matrix ([4]): *A square matrix is said to be Strictly Diagonally Dominant (SDD) if*

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

**Definition 2.3** Symmetric Positive Definite (SPD) matrix ([9]): *An* $n \times n$ *matrix* $A = (a_{ij})$ *is said to be Symmetric Positive Definite if A is symmetric* $(A = A^T)$ *and positive definite that is* $x^T A x > 0$ *for all* $x \neq 0$.

**Definition 2.4** M-matrix ([9]): *A matrix is classified as an M-matrix if it adheres to the following criteria*:

- $a_{ii} > 0$ *for* $i = 1, 2, \ldots, n$.
- $a_{ij} \leq 0$ *for* $i \neq j$ *and* $i, j = 1, 2, \ldots, n$.
- $A$ *is non-singular.*
- $A^{-1} \geq O$ .

**Theorem 2.1** ([4]): *For any initial approximation* $x^{(0)} \in R$ , *the sequence* $\{x^{(k)}\}_{k=0}^{\infty}$ *of approximations defined by* $x^{(k+1)} = (Tx^{(k)} + c)$, *for each* $k \geq 0$, *and* $c \neq 0$ *converges to the unique solution of* $x = Tx + c$ , *if and only if* $\rho(T) < 1$. *Also, since* $\rho(T) \leq ||T||$ *for any natural norm, so* $\rho(T) < 1$ *is equivalent to* $||T|| < 1$.

## 3. Generalized Gauss-Seidel (GGS) Method

Davod K. Salkuyeh [14] enhanced the Gauss-Seidel method by introducing the concept of a banded matrix, as outlined earlier in the introduction section. For system equation (1.1), the matrix $A$ is decomposed as shown in equation (2.1). Then the Generalized Gauss-Seidel method is,

$$x^{(k+1)} = -(D_m + L_m)^{-1} U_m x^{(k)} + (D_m + L_m)^{-1} b \ , \qquad (3.1)$$

where, $k = 0, 1, 2, \ldots$ and $m = 0, 1, 2, \ldots, n-1$.

If $m = 0$ , the above method reduces to the classical Gauss-Seidel method.

**Theorem 3.1** ([14]): *Let A represent an SDD matrix. Then, for any natural number* $m \leq n$, *both the Generalized Jacobi* (*GJ*) *and Generalized Gauss-Seidel* (*GGS*) *methods exhibit convergence with an initial guess* $x^{(0)}$.

**Theorem 3.2** ([14]): *Let A be an M-matrix. Then for a given natural number* $m \leq n$, *both of the GJ and GGS methods are convergent for initial guess* $x^{(0)}$.

**Theorem 3.3:** *Let A and* $2(D_m + L_m) - A$ *are symmetric positive definite matrices, then Generalized Gauss-Seidel* (*GGS*) *method converges for any initial approximation* $x^{(0)}$.

**Proof:** Let $A$ and $2(D_m + L_m) - A$ are SPD matrix.

So, $x^T A x > 0$ and $x^T (2(D_m + L_m) - A)x > 0$.

Now, $\lambda$ be the eigenvalue of $-(D_m + L_m)^{-1} U_m$.

Then,

$$(-(D_m + L_m)^{-1} U_m)x = \lambda x$$

$$\Rightarrow (D_m + L_m - A)x = \lambda (D_m + L_m)x$$

$$\Rightarrow x^T (D_m + L_m - A)x = \lambda x^T (D_m + L_m)x$$

$$\Rightarrow x^T A x = (1 - \lambda)\, x^T (D_m + L_m)x$$

$$\Rightarrow 1 - \lambda > 0$$

$$\Rightarrow \lambda < 1 \qquad (3.2)$$

Again

$$x^T (2(D_m + L_m) - A)x > 0$$

$$\Rightarrow x^T A x < 2x^T (D_m + L_m)x$$

$$\Rightarrow (1-\lambda)x^T (D_m + L_m)x < 2x^T (D_m + L_m)x$$

$$\Rightarrow 1 - \lambda < 2$$

$$\Rightarrow \lambda > -1. \tag{3.3}$$

From (3.2) and (3.3) $\Rightarrow -1 < \lambda < 1 \Longleftrightarrow \rho(-(D_m + L_m)^{-1}U_m) < 1.$

Hence, the GGS method is convergent. ,

**4. Refinement of Generalized Gauss-Seidel (RGGS) Method**

In 2014, Laskar and Behera [13] introduced the Refinement of the Generalized Gauss-Seidel (RGGS) method. Consider the system (1.1) and the matrix splitting as in (2.1).

Now,

$$Ax = b$$

$$\Rightarrow (D_m + L_m + U_m)x = b$$

$$\Rightarrow (D_m + L_m)x = b - U_m x$$

$$\Rightarrow (D_m + L_m)x = b + (D_m + L_m - A)x$$

$$\Rightarrow x = x + (D_m + L_m)^{-1}(b - Ax).$$

So, the refinement iterative form will be,

$$x^{(k+1)} = \tilde{x}^{(k+1)} + (D_m + L_m)^{-1}(b - A\tilde{x}^{(k+1)}) \tag{4.1}$$

where, $\tilde{x}^{(k+1)} = -(D_m + L_m)^{-1}Um^{x^{(k)}} + (D_m + L_m)^{-1}b$ , (From (3.1))

Using (3.1) in (4.1), we get the RGGS method which is given by,

$$x^{(k+1)} = (-(D_m + L_m)^{-1}U_m)^2 x^{(k)} + (I - (D_m + L_m)^{-1}U_m)(D_m + L_m)^{-1}b, \quad (4.2)$$

where, $k = 0, 1, 2, \ldots$ and $m = 0, 1, 2, \ldots, n-1$.

**Theorem 4.1** ([13]): *If A is SDD matrix, then the RGGS method converges for any arbitrary choice of the initial approximation* $x^{(0)}$.

**Theorem 4.2** ([10]): *If* $A = (a_{ij})$ *represents an M-matrix, then the Refinement of Generalized Gauss-Seidel* (*RGGS*) *method converges for any initial choice of approximation* $x^{(0)}$.

## 5. Second Refinement of Generalized Gauss-Seidel (SRGGS) Method

Here, we introduce the SRGGS method by taking equation (4.1) which is

$$x^{(k+1)} = \tilde{x}^{(k+1)} + (D_m + L_m)^{-1}(b - A\tilde{x}^{(k+1)})$$

Now, substituting the value of $\tilde{x}^{(k+1)}$ from equation (4.2) treating $\tilde{x}^{(k+1)}$ as $x^{(k+1)}$ we have,

$$x^{(k+1)} = (-(D_m + L_m)^{-1}U_m)^2 x^{(k)} + (D_m + L_m)^{-1}b[I - (D_m + L_m)^{-1}U_m]$$

$$+ (D_m + L_m)^{-1}[b - A((-(D_m + L_m)^{-1}U_m)^2 x^{(k)}$$

$$+ (D_m + L_m)^{-1}b(I - (D_m + L_m)^{-1}U_m))].$$

After simplification, we get,

$$x^{(k+1)} = (-(D_m + L_m)^{-1}U_m)^3 x^{(k)} + (D_m + L_m)^{-1}b[I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2] \quad (5.1)$$

where, $k = 0, 1, 2, \ldots$ and $m = 0, 1, 2, \ldots, n-1$. When $m = 0$ it will be the SRGS method.

Next, we prove some theorems that will imply the convergence and efficiency of this new proposed method.

**Theorem 5.1:** *If A is an SDD matrix, then the Second Refinement of Generalized Gauss-Seidel method converges for any choice* $x^{(0)}$.

**Proof:** *Let* $Ax = b$, *be the system of linear equations and let x be its true solution.*

Given that, $A$ is SDD.

By using Theorem 4.1, the Refinement of Generalized Gauss-Seidel methods is convergent.

So,

$$\tilde{x}^{(k+1)} \to x$$

Now,

$$||x^{(k+1)} - x|| = ||\tilde{x}^{(k+1)} + (D_m + L_m)^{-1}(b - A\tilde{x}^{(k+1)}) - x||$$

$$\Rightarrow ||x^{(k+1)} - x|| \le ||\tilde{x}^{(k+1)} - x|| + ||(D_m + L_m)^{-1}(b - A\tilde{x}^{(k+1)})||$$

$$\Rightarrow ||x^{(k+1)} - x|| \le 0 + ||(D_m + L_m)^{-1}(b - Ax)|| = 0.$$

So, the SRGGS method is convergent, when the coefficient matrix is SDD.,

**Theorem 5.2:** *If A is an M-matrix, the Second Refinement of Generalized Gauss- Seidel method is convergent for any initial guess* $x^{(0)}$.

**Proof:** *Given that A is an M-matrix.*

By using Theorem 4.2, the GGS method is convergent.

$$\rho(-(D_m + L_m)^{-1}U_m) < 1 .$$

Now, using the Lemma 2.1,

$$\rho((-(D_m + L_m)^{-1}Um)^3) = (\rho(-(D_m + L_m)^{-1}Um))^3 < 1.$$

Hence, SRGGS is convergent for M-matrix. ,

**Theorem 5.3:** *If $A$ and $(2(D_m + L_m) - A)$ are symmetric and positive definite matrices, then the Second Refinement of Generalized Gauss-Seidel method converges for any choice of initial approximation $x^{(0)}$.*

**Proof:** *Let $Ax = b$, be the system of linear equations and let x be its true solution. The generalized Gauss-Seidel (GGS) method is,*

$$x^{(k+1)} = -(D_m + L_m)^{-1}U_m x^{(k)} + (D_m + L_m)^{-1}b,$$

Using Theorem 3.3

$$\rho(-(D_m + L_m)^{-1}U_m) < 1. \tag{5.2}$$

If $x^{(k+1)} \to x$,

Then, the GGS method can also be written as,

$$x = (I + (D_m + L_m)^{-1}U_m)^{-1}(D_m + L_m)^{-1}b.$$

Now, the Second Refinement of Generalized Gauss-Seidel method is,

$$x^{(k+1)} = (-(D_m + L_m)^{-1}U_m)^3 x^{(k)} + (D_m + L_m)^{-1}b[I - (D_m + L_m)^{-1}U_m + ((Dm + Lm)^{-1}Um)^2]. \tag{5.3}$$

Now, using equation (5.3), and exact solution $x$, we have,

$$x = (-(D_m + L_m)^{-1}U_m)^3 x + (D_m + L_m)^{-1}b[I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2]$$

$$\Rightarrow x = (I + ((D_m + L_m)^{-1}U_m)^3)^{-1}[I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2](D_m + L_m)^{-1}b$$

$$\Rightarrow x = [I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2 - ((D_m + L_m)^{-1}U_m)^3 + ((D_m + L_m)^{-1}U_m)^4 - \ldots](D_m + L_m)^{-1}b$$

$$\Rightarrow x = (I + (U_m + L_m)^{-1}U_m)^{-1}(D_m + L_m)^{-1}b$$

So, $x = (I + (D_m + L_m)^{-1}U_m)^{-1}(D_m + L_m)^{-1}b$ is consistent for $Ax = b$ and Generalized Gauss-Seidel method.

On the other hand,

$$x^{(k+1)} = (-(D_m + L_m)^{-1}U_m)^3 x^{(k)} + (D_m + L_m)^{-1}b[I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2][I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2]$$

$$\Rightarrow x^{(k+1)} = ((D_m + L_m)^{-1}U_m)^6 x^{(k-1)} + [I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2((D_m + L_m)^{-1}U_m)^3 + ((D_m + L_m)^{-1}U_m)^4 - ((D_m + L_m)^{-1}U_m)^5](D_m + L_m)^{-1}b$$

$$\vdots$$

$$x^{(k+1)\dagger} = ((D_m + L_m)^{-1}U_m)^{3k+3} x^{(0)} + [I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2 + ((D_m + L_m)^{-1}U_m)^3 + \ldots + ((D_m + L_m)^{-1}U_m)^{3k+2}] (D_m + L_m)^{-1}b.$$

As, $A$ is SPD, so by using Theorem 3.3,

$$\lim_{k\to\infty}(-(D_m + L_m)^{-1}U_m)^{3k+3} = 0.$$

Taking limit on both sides of the above equation we have,

$$\lim_{k\to\infty} x^{(k+1)} = \lim_{k\to\infty}(-(D_m + L_m)^{-1}U_m)^{3k+3})x^{(0)} + \sum_{k=0}^{\infty}(-(D_m + L_m)^{-1}U_m)^k (D_m + L_m)^{-1}b$$
$$= 0 + (I - (D_m + L_m)^{-1}U_m)^{-1}(D_m + L_m)^{-1}b$$
$$= (I - (D_m + L_m)^{-1}U_m)^{-1}(D_m + L_m)^{-1}b \to x$$

$$\rho(((D_m + L_m)^{-1}U_m)^3) = (\rho(D_m + L_m)^{-1}U_m)^3 < 1.$$

Hence, the second Refinement of Generalized Gauss-Seidel (SRGGS) method is convergent. ,

**Theorem 5.4:** *The Second Refinement of Generalized Gauss-Seidel* (*SRGGS*) *method converges faster than the both Generalized Gauss-Seidel* (*GGS*) *method, and Refinement of Generalized Gauss-Seidel* (*RGGS*) *method, when the Generalized Gauss-Seidel* (*GGS*) *method is convergent.*

**Proof:** We can write, GGS method as,

$$x^{(k+1)} = Gx^{(k)} + C .$$

RGGS method as,

$$x^{(k+1)} = G^{2}x^{(k)} + B .$$

SRGGS method as,

$$x^{(k+1)} = G^{3}x^{(k)} + K .$$

Where,

$$G = -(D_m + L_m)^{-1}U_m,$$
$$C = (D_m + L_m)^{-1}b,$$
$$B = (I - (D_m + L_m)^{-1}U_m)(D_m + L_m)^{-1}b,$$

$$K = (I - (D_m + L_m)^{-1}U_m + ((D_m + L_m)^{-1}U_m)^2)(D_m + L_m)^{-1}b.$$

Given that, the GGS method is convergent.

So, $$||G|| < 1 \qquad (5.4)$$

Let, $x$ be the exact solution of $Ax = b$, then

$$x = Gx + C , \qquad x = G^{2}x + B , \qquad x = G^{3}x + K .$$

Now, consider the GGS method,

$$x^{(k+1)} = Gx^{(k)\dagger} + C$$

$$\Rightarrow x^{(k+1)} - x = Gx^{(k)} - x + C$$

$$\Rightarrow x^{(k+1)} - x = G(x^{(k)} - x) + Gx + C \quad \text{ñ } x$$

$$\Rightarrow ||x^{(k+1)} - x|| = ||G(x^{(k)} - x)||$$

$$\Rightarrow ||x^{(k+1)} - x|| \le ||G\ ||\ ||(x^{(k)} - x)||$$

$$\le ||G^{\,2}||\ ||x^{(k-1)} - x||$$

$$\le \ldots \le ||G^{\,k+1}||\ \ ||(x^{(0)} - x)||$$

$$= ||G\ ||^{k+1} ||(x^{(0)} - x)||.$$

For the RGGS method,

$$x^{(k+1)} = G^{\,2}x^{(k)} + B$$

$$\Rightarrow ||x^{(k+1)} - x|| \le ||G^{\,2}||\,||(x^{(k)} - x)||$$

$$= ||G\ ||^{2k+2}\,||(x^{(0)} - x)||$$

Similarly, considering SRGGS,

$$x^{(k+1)} = G^{\,3}x^{(k)} + K$$

$$\Rightarrow ||x^{(k+1)} - x|| \le ||G^{\,3}||\,||(x^{(k)} - x)||$$

$$= ||G\ ||^{3k+3}\,||(x^{(0)} - x)||$$

According to the coefficient of the above inequalities, we have,

$$||G\ ||^{3k+3} \le ||G\ ||^{2k+2} \le ||G\ ||^{k+1} \qquad \text{(From (3.1))}$$

Hence, the SRGGS method accelerates convergence more rapidly than the

RGGS and GGS methods when the GSS method converges. ,

## 6. Numerical Examples

In this section, we will examine numerical examples provided by T. K. Enyew, G. Awgichew, H. Haile, G. D. Abie in [8], and K. J. Audu, J. N. Essien, A. B. Zahiri, A. R. Taiwo in [9]. These examples will be utilized to conduct a comparative analysis of spectral radius, number of iterations, and computational time for various iterative methods implemented using Python code. The analysis has/have been performed on a system featuring an AMD Ryzen 5 5625U processor with Radeon Graphics running at 2.30 GHz, and 16.0 GB of RAM and Windows 11 Home.

**Example 6.1:** Examine the subsequent system of linear equations, characterized by a coefficient matrix that is Strictly Diagonally Dominant (SDD) yet not Symmetric Positive Definite (SPD), and set with an error tolerance of $10^{-4}$.

$$6x_1 + 4x_2 - x_3 = 9$$

$$3x_1 + 7x_2 + 2x_3 = 12$$

$$-4x_1 + 3x_2 + 8x_3 = 7$$

Now, solving this system with the error tolerance $10^{-4}$ and $(0, 0, 0)^T$ as an initial approximation, we compare the number of iterations, their spectral radii, and computational time for different methods as shown in the tables below.

Table 6.1
Comparison with No. of Iterations, Spectral Radius, and Computational Time

| Methods | GS | RGS | SRGS | GGS | RGGS | SRGGS |
|---|---|---|---|---|---|---|
| No. of Iterations | 15 | 9 | 7 | 7 | 4 | 3 |
| Spectral Radius | 0.6496 | 0.4220 | 0.2741 | 0.2151 | 0.0463 | 0.0100 |
| Computational Time(In Seconds) | 0.0064 | 0.0029 | 0.0023 | 0.0026 | 0.0022 | 0.0018 |

Table 6.2
Numerical results and comparison between GGS, RGGS, and SRGGS

| | GGS | RGGS | SRGGS |
|---|---|---|---|

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.7093 | 1.1860 | 0.7849 | 0.9375 | 1.0400 | 0.9537 | 0.9865 | 1.0086 | 0.9900 |
| 2 | 0.9375 | 1.0400 | 0.9537 | 0.9971 | 1.0019 | 0.9979 | 0.9999 | 1.0001 | 0.9999 |
| 3 | 0.9865 | 1.0086 | 0.9900 | 0.9999 | 1.0001 | 0.9999 | 1.0000 | 1.0000 | 1.0000 |
| 4 | 0.9971 | 1.0019 | 0.9979 | 1.0000 | 1.0000 | 1.0000 | | | |
| 5 | 0.9994 | 1.0004 | 0.9995 | | | | | | |
| 6 | 0.9999 | 1.0001 | 0.9999 | | | | | | |
| 7 | 1.0000 | 1.0000 | 1.0000 | | | | | | |

Table 6.1 shows that the SRGGS method has a smaller spectral radius than all the other methods and takes less time than others. Table 6.2 shows that the SRGGS (3 iterations) method is better than the GGS (7 iterations) and RGGS (4 iterations) method in terms of the number of iterations.

**Example 6.2:** Consider the 2-cycle matrix, which also qualifies as an M-matrix, originating from the discretization of Poisson's equations $\frac{\partial^2 T}{\partial^2 x} + \frac{\partial^2 T}{\partial^2 y} = f$ on the unit square. This system can be expressed in the form of $A x = b$ that is,

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

we solve this system using different iterative methods, setting an error tolerance of $10^{-5}$ and initializing with the vector $(0, 0, 0, 0, 0, 0)^T$. A comparative analysis showcases each method's iteration count, spectral radii, and computational time. The summarized results are presented in the subsequent Tables.

Table 6.3
Comparison with No. of Iterations, Spectral Radius, and Computational Time

| Methods | GS | RGS | SRGS | GGS | RGGS | SRGGS |
|---|---|---|---|---|---|---|
| No. of Iterations | 9 | 5 | 3 | 6 | 3 | 2 |
| Spectral Radius | 0.3645 | 0.1327 | 0.0483 | 0.1496 | 0.0224 | 0.0033 |
| Computational Time(In Seconds) | 0.0050 | 0.0028 | 0.0027 | 0.0031 | 0.0022 | 0.0015 |

Table 6.4
Numerical results for GGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.2679 | 0.0714 | 0.0179 | 0.0772 | 0.0408 | 0.0147 |
| 2 | 0.2917 | 0.0897 | 0.0261 | 0.0850 | 0.0483 | 0.0186 |
| 3 | 0.2944 | 0.0926 | 0.0278 | 0.0860 | 0.0495 | 0.0193 |
| 4 | 0.2948 | 0.0931 | 0.0281 | 0.0861 | 0.0497 | 0.0194 |
| 5 | 0.2948 | 0.0932 | 0.0281 | 0.0861 | 0.0497 | 0.0195 |
| 6 | 0.2948 | 0.0932 | 0.0282 | 0.0861 | 0.0497 | 0.0195 |

Table 6.5
Numerical results for RGGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.2917 | 0.0897 | 0.0216 | 0.0850 | 0.0483 | 0.0186 |
| 2 | 0.2948 | 0.0931 | 0.0281 | 0.0861 | 0.0497 | 0.0194 |
| 3 | 0.2948 | 0.0932 | 0.0282 | 0.0861 | 0.0497 | 0.0195 |

Table 6.6
Numerical results for SRGGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.2944 | 0.0926 | 0.0278 | 0.0860 | 0.0495 | 0.0193 |
| 2 | 0.2948 | 0.0932 | 0.0282 | 0.0861 | 0.0497 | 0.0195 |

From above Table 6.3, we can see that the SRGGS method takes only 2 iterations to tackle this framework, and the spectral radius of the iteration matrix is also diminished in this case than all other existing methods. Also, Tables 6.4, 6.5, and 6.6 present the solution for GGS, RGGS, and SRGGS methods. All imply that our new method is convergent faster than other iterative methods.

**Example 6.3:** Here, we consider an example of a linear system as given below with an error tolerance $10^{-5}$. The coefficient matrix of the system is SDD and also M-matrix of size $9 \times 9$.

$$
\begin{aligned}
4x_1 - x_2 &= 20 \\
-x_1 + 4x_2 - x_3 - x_5 &= 20 \\
-x_2 + 4x_3 - x_5 &= 0 \\
-x_2 + 4x_4 - x_5 - x_7 &= 20 \\
-x_3 - x_4 + 4x_5 - x_6 - x_8 &= 0 \\
-x_5 + 4x_6 - x_9 &= 0 \\
-x_4 + 4x_7 - x_8 &= 30 \\
-x_5 - x_7 + 4x_8 - x_9 &= 10 \\
-x_6 - x_8 + 4x_9 &= 10.
\end{aligned}
$$

Now, resolving this system with an error tolerance of $10^{-5}$ and initiating the process with the vector $(0, 0, 0, 0, 0, 0, 0, 0, 0)^T$, the compiled results are displayed in the subsequent tables.

Table 6.7
Comparison with No. of Iterations, Spectral Radius, and Computational Time

| Methods | GS | RGS | SRGS | GGS | RGGS | SRGGS |
|---|---|---|---|---|---|---|
| No. of Iterations | 16 | 9 | 7 | 11 | 6 | 4 |
| Spectral Radius | 0.4514 | 0.2038 | 0.0920 | 0.2681 | 0.0719 | 0.0193 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Computational Time(In Seconds) | 0.0093 | 0.0063 | 0.0045 | 0.0070 | 0.0044 | 0.0032 |

Table 6.8
Numerical results for GGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ | $x_7^{(n)}$ | $x_8^{(n)}$ | $x_9^{(n)}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 6.7857 | 7.1429 | 1.7857 | 7.3980 | 2.4490 | 0.6122 | 11.0960 | 6.9862 | 4.3996 |
| 2 | 7.0044 | 8.0175 | 2.6166 | 11.2413 | 5.8517 | 2.5628 | 12.4034 | 8.3722 | 5.2338 |
| 3 | 7.3082 | 9.2328 | 3.7711 | 12.1134 | 6.8174 | 3.0128 | 12.7140 | 8.7425 | 5.4388 |
| 4 | 7.3944 | 9.5776 | 4.0988 | 12.3425 | 7.0783 | 3.1293 | 12.7961 | 8.8418 | 5.4928 |
| 5 | 7.4177 | 9.6708 | 4.1873 | 12.4038 | 7.1483 | 3.1603 | 12.8180 | 8.8684 | 5.5072 |
| 6 | 7.4240 | 9.6958 | 4.2110 | 12.4202 | 7.1670 | 3.1685 | 12.8239 | 8.8755 | 5.5110 |
| 7 | 7.4256 | 9.7025 | 4.2174 | 12.4246 | 7.1721 | 3.1708 | 12.8255 | 8.8774 | 5.5120 |
| 8 | 7.4261 | 9.7043 | 4.2191 | 12.4258 | 7.1734 | 3.1714 | 12.8259 | 8.8779 | 5.5123 |
| 9 | 7.4262 | 9.7048 | 4.2196 | 12.4261 | 7.1738 | 3.1715 | 12.8260 | 8.8781 | 5.5124 |
| 10 | 7.4262 | 9.7049 | 4.2197 | 12.4262 | 7.1739 | 3.1716 | 12.8261 | 8.8781 | 5.5124 |
| 11 | 7.4262 | 9.7050 | 4.2197 | 12.4262 | 7.1739 | 3.1716 | 12.8261 | 8.8781 | 5.5124 |

Table 6.9
Numerical results for GGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ | $x_7^{(n)}$ | $x_8^{(n)}$ | $x_9^{(n)}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 7.0044 | 8.0175 | 2.6166 | 11.2413 | 5.8517 | 2.5628 | 12.4034 | 8.3722 | 5.2338 |
| 2 | 7.3944 | 9.5776 | 4.0988 | 12.3425 | 7.0783 | 3.1293 | 12.7961 | 8.8418 | 5.4928 |
| 3 | 7.4240 | 9.6958 | 4.2110 | 12.4202 | 7.1670 | 3.1685 | 12.8239 | 8.8755 | 5.5110 |
| 4 | 7.4261 | 9.7043 | 4.2191 | 12.4258 | 7.1734 | 3.1714 | 12.8259 | 8.8779 | 5.5123 |
| 5 | 7.4262 | 9.7049 | 4.2197 | 12.4262 | 7.1739 | 3.1716 | 12.8261 | 8.8781 | 5.5124 |
| 6 | 7.4262 | 9.7050 | 4.2197 | 12.4262 | 7.1739 | 3.1716 | 12.8261 | 8.8781 | 5.5124 |

Table 6.10

Numerical results for SRGGS with $m = 1$

| $n$ | $x_1^{(n)}$ | $x_2^{(n)}$ | $x_3^{(n)}$ | $x_4^{(n)}$ | $x_5^{(n)}$ | $x_6^{(n)}$ | $x_7^{(n)}$ | $x_8^{(n)}$ | $x_9^{(n)}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 7.3082 | 9.2328 | 3.7711 | 12.1134 | 6.8174 | 3.0128 | 12.7140 | 8.7425 | 5.4388 |
| 2 | 7.4240 | 9.6958 | 4.2110 | 12.4202 | 7.1670 | 3.1685 | 12.8239 | 8.8755 | 5.5110 |
| 3 | 7.4262 | 9.7048 | 4.2196 | 12.4261 | 7.1738 | 3.1715 | 12.8260 | 8.8781 | 5.5124 |
| 4 | 7.4262 | 9.7050 | 4.2197 | 12.4262 | 7.1739 | 3.1716 | 12.8261 | 8.8781 | 5.5124 |

From this example, we observe that even for a large-scale matrix, the number of iterations and the spectral radius of our new method, SRGGS, are lower than those of the other methods and take less time to converge. Therefore, SRGGS demonstrates superior performance compared to the others.

## 7. Conclusion

The SRGGS method, proposed as an enhancement of the classical Gauss-Seidel method, offers a significant advancement in the numerical solution of linear systems.

This study confirms the effectiveness of the SRGGS method across a variety of matrix types, including Strictly Diagonally Dominant (SDD), Symmetric Positive Definite (SPD), and M-matrices.

The method exhibits superior efficiency by reducing the number of iterations required for convergence, minimizing the spectral radius, and improving overall convergence rates compared to traditional approaches.

Future research could focus on extending the SRGGS method to incorporate higher-degree refinements, broadening its applicability to more complex and large-scale systems.

Additionally, exploring the integration of advanced optimization techniques and parallel computing strategies could further enhance the scalability and computational performance of the SRGGS method, particularly for real-world scientific and engineering challenges.

## Acknowledgements

We thank the referees for their valuable comments. We also acknowledge the academic help provided by Dr. Arnab Paul.

## REFERENCES

[1] Ahmadi, A., Manganiello, F., Khademi, A. and Smith, M. C. (2021): A parallel Jacobi-embedded Gauss-Seidel method, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 32(6), pp. 1452-1464.

[2] Audu, K. J., Essien, J. N., Zahiri, A. B. and Taiwo, A. R. (2023): A third refinement of Jacobi method for solutions to system of linear equations. *FUDMA Journal of Sciences*, Vol. 7(5), pp. 234-239.

[3] Bai, Z. Z., Wang, L. and Wu, W. T. (2021): On convergence rate of the randomized Gauss-Seidel method, *Linear Algebra and its Applications*, Vol. 611, pp. 237-252.

[4] Butt, R. (2015): An Introduction to Applied Numerical Linear Algebra Using MATLAB. Alpha Science International Limited.

[5] Calvin, O. W. and Choi, N. (2024): Application of the Gauss-Seidel Method to the Chebyshev Rational Approximation Method for Solving Nuclear Fuel Depletion Systems, *Nuclear Science and Engineering*, Vol. 198(6), pp. 1255-1275.

[6] Dessalew, G., Kebede, T., Awgichew, G. and Walelign, A.: Generalized Refinement of Gauss-Seidel Method for Consistently Ordered 2-Cyclic Matrices, *Abstract and Applied Analysis*, Vol. 2021, Article ID 8343207.

[7] Ding, W., Liu, Z., Li, Y., Wei, A. and Zhang, M. (2024): New structure-preserving algorithms of Gauss-Seidel and successive over-relaxation iteration methods for quaternion linear systems, *Numerical Algorithms*, Vol. 95(3), pp. 1309-1323.

[8] Dutta, B. N. (2010): Numerical Linear Algebra and Applications, *Society for Industrial and Applied Mathematics*.

[9] Enyew, T. K., Awgichew, G., Haile, H., and Abie, G. D. (2020): Second refinement of generalized Jacobi iterative method for solving linear system of equations, *Journal of the Nigerian Mathematical Society*, Vol. 39(1), pp. 117-133.

[10] Gonfa, G. G.: Refined iterative method for solving system of linear equations, *American Journal of Computational and Applied Mathematics*, Vol. 6(3), 2016, pp. 144-147.

[11] Hussain, M. D., Rahman, M. H. and Ali, N. M. (2024): Investigation of Gauss-Seidel Method for Load Flow Analysis in Smart Grids, *Sch Journal Eng Tech*, Vol. 5, pp. 169-178.

[12] Ali, R. (2022): The Solution of Absolute Value Equations Using Two New Generalized Gauss-Seidel Iteration Methods, *Computational and Mathematical Methods*, Vol. (1), 4266576.

[13] Laskar, A. H. and Behera, S. (2014): A new refinement of generalized Gauss-Seidel method for solving system of linear equations, *International Journal of Mathematics Archive*, Vol. 5(6), pp. 104-108.

[14] Salkuyeh, D. K. (2007): Generalized Jacobi and Gauss-Seidel methods for solving linear systems of equations. Numerical Mathematics, *A Journal of Chinese Universities* (English Series), Vol. 16(2), pp. 164-170.

[15] Saad, Y. (2003): Iterative Methods for Sparse Linear Systems, *Society for Industrial and Applied Mathematics*.

[16] Saha, M. and Chakravarty, J. (2020): Convergence of generalized SOR, Jacobi, and Gauss-Seidel methods for linear systems, *International Journal of Applied and Computational Mathematics*, Vol. 6, pp. 1-12.

[17] Xie, K. and Miao, S. X. (2023): A new preconditioner for Gauss–Seidel method for solving multi-linear systems, *Japan Journal of Industrial and Applied Mathematics*, Vol. 40(2), pp. 1159-1173.

1. Research Scholar,
Department of Mathematics,
Assam University, 788011, Silchar, Assam, India
E-mail: hzurboshi1@gmail.com
Mobile No: 9101508344

2. Professor,
Department of Mathematics,
Assam University, 788011, Silchar, Assam, India
E-mail: samirabehera@yahoo.in
Mobile No: 9435073898

(*Received*, October 3 2024,)
(*Revised*, October 19, 2024)