Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

Online ISSN 3007-3197

Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Improved Generalized Classical Iterative Technique for Solving Linear System

Sumaira Kanwal
Institute of Mathematics & Computer Science University of Sindh, Jamshoro, Allama I.I. Kazi Campus, Jamshoro 76080, Pakistan

Zubair Ahmed Kalhoro (Corresponding Author)
Institute of Mathematics & Computer Science University of Sindh, Jamshoro, Allama I.I. Kazi Campus, Jamshoro 76080, Pakistan
Email: zubair.kalhoro@usindh.edu.pk

Sanaullah Jamali
University of Sindh, Laar Campus, Badin Sindh, Pakistan

Zohaib Ali
Institute of Mathematics & Computer Science University of Sindh, Jamshoro, Allama I.I. Kazi Campus, Jamshoro 76080, Pakistan

Syeda Hira Fatima Naqvi
Institute of Mathematics & Computer Science University of Sindh, Jamshoro, Allama I.I. Kazi Campus, Jamshoro 76080, Pakistan

Tooba Rana
Institute of Mathematics & Computer Science University of Sindh, Jamshoro, Allama I.I. Kazi Campus, Jamshoro 76080, Pakistan

Muzaffar Hussain Laghari
GBHSS Bhansinghabad,Mirpurkhas, Sindh Education and Literacy Department, Government of Sindh

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

ABSTRACT

One of the major problems of linear algebra is to find the solution of a system of linear equations (SOLEs). Among iterative techniques, one of the most important topics of interest is SOLEs. This type of systems can be found in a wide range of disciplines, such as natural and social sciences, engineering, medicine, and business. Iterative methods are typically used to solve sparse SOLEs. A better method- Improved Generalized Classical Iterative Techniques (IGCT) has been put forward in this study. This procedure can be used when the coefficient matrix is diagonally dominant (SDD), irreducibly diagonally dominant (IDD), an M-matrix, symmetric positive definite under some specifications or an H-matrix. Such systems can frequently be an outcome of ordinary differential equations (ODEs) and partial differential equations (PDEs). The suggested approach has a good spectral radius and convergence rate and fewer iterations. Its action has been confirmed by numerical experiments, such as comparison to classical methods on a number of nonlinear problems of literature. The execution was done with the help of MATLAB (R2014b), a high-level computational programming language.

Keywords: System of Linear Equations, Generalized Gauss–Jacobi (GGJ), Generalized Gauss–Seidel (GGS), Composite Refinement of Jacobi–Gauss–Seidel (CRJGS), Composite Refinement of Gauss–Seidel–Jacobi (CRGSJ).

Introduction

Solving large-scale systems of linear equations is a central problem in many modern modeling and computational applications. These systems are commonly expressed in matrix form as:

$$Ax = b \tag{1}$$

where A is an invertible $n \times n$ matrix, b is a known vector in $\mathbb{R}^n$,and x is the vector of unknowns to be determined. Their wide occurrence across various disciplines underscores their importance.[1]

Solutions to such systems are usually categorized into two. classifications: direct methods and iterative methods. For large-scale

iterative methods tend to be preferred over direct ones, problems. Iterative methods start off with a rough approximation and then improve upon this approximation. stop when the accuracy or predetermined criterion is achieved. These methods normally cause minor errors in numbers, and the amount of numbers are many. The number of iterations to be made is unknown.

Direct methods on the other hand give an exact solution in a. determined number of steps and therefore very reliable. However, they are not efficient to sparse systems since they fail to utilize sparsity, usually leading to increase in the computation cost. For sparse systems, iterative methods can significantly reduce both memory usage and the number of floating-point operations. Gaussian elimination, LU factorization, Gauss-Jordan elimination, and Cholesky decomposition all are examples of direct methods. There are the Jacobi (J) method, the Gauss-Seidel (GS) method, and the Successive Over-Relaxation (SOR) method which are classical iterative methods. The major advantage of the iterative methods is that they have less memory, and computation needs than the direct methods in the case of sparse systems. [2-6].

Overall [7,9], iterative solvers bring the original equation Ax = b into a similar form of the fixed-point form:

$$x = Tx + c$$

Where T is a constant matrix and c is a constant vector. Starting from an initial guess $x^{(0)}$, a sequence of approximate solutions is generated as:

$$x^{(k)} = Tx^{(k-1)} + c, \text{ for each k=1,2,3…}$$

We shall give in this paper the composite refinement strategy,[10-15] in which we take into account two different iterative procedures in succession. The refining treatments in the first step have a dominant role in the increase in the cost of computing as dictated by the attained speed of convergence. The usage of a virtual step $x^{(vir)}$ is the fundamental concept in the refinement therapy. Various iterative strategies are taken into consideration in the successive sweeps of the composite refining. We apply this idea to the Generalized Gauss-Seidel and Jacobi methods, two straightforward iterative techniques.

Material and Methods

In this work, we consider the numerical solution of systems of linear equations of the form

$$Ax = b,$$

where $A \in \mathbb{R}^{n\times n}$is a nonsingular matrix and $b \in \mathbb{R}^{n}$. The coefficient matrix Ais is broken down into diagonal, strictly lower triangular and strictly upper triangular parts. On this basis matrix splitting, the Generalized Gauss-Jacobi (GGJ) and Generalized Gauss-Seidel (GGS) iterative schemes are developed. These universal techniques are extensions of the classical Jacobi and Gauss-Seidel techniques, based on altered iteration forms, with no relaxation or weighting parameters. The resulting iteration matrices are explicitly obtained and are taken as reference methods to be refined by further refinement.

More convergence efficiency is achieved with composite refinement strategies, which are built by the systematic combination of Jacobi and Gauss-Seidel updates as a single iterative method. Specifically, the Composite Refinement of Jacobi-Gauss-Seidel (CRJGS) and the Composite Refinement of Gauss-Seidel-Jacobi (CRGSJ) algorithms are designed through the generation of successive correction steps, which utilize the complementary convergence nature of the underlying schemes. An enhanced version, dubbed as Improved Generalized Classical Iterative Techniques (IGCT) is then developed by optimizing the organization of the iteration process. All methods are analyzed in terms of the convergence behavior by spectral analysis of the relevant iteration matrices, and their computational performance are analyzed with the help of a number of numerical test problems. All the numerical simulations are done in MATLAB.

Generalized Methods

Let [16-20] $A = (a_{ij})$ be an $n \times n$ matrix, and define $T_m = (t_{ij})$ as a banded matrix with bandwidth $2m + 1$ where,

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

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

Matrix A can be decomposed as; $A = T_m - E_m - F_m$, in which $E_m$corresponds to the strictly lower triangular part and $F_m$to the strictly upper triangular part. Explicitly, the matrices are defined by:

$$T_m = \begin{bmatrix} a_{1,1} & \text{L} & a_{1,m+1} & & \\ \text{M} & \text{O} & & \text{O} & \\ a_{m+1,1} & & \text{O} & & a_{n-m,n} \\ & \text{O} & & \text{O} & \text{M} \\ & & a_{n,n-m} & \text{L} & a_{n,n} \end{bmatrix},$$

$$E_m = \begin{bmatrix} & & & \\ -a_{m+2,1} & & & \\ \text{M} & \text{O} & & \\ -a_{n,1} & \text{L} & -a_{n,n-m-1} & \end{bmatrix}$$

$$F_m = \begin{bmatrix} & -a_{1,m+2} & \text{L} & -a_{1,n} \\ & & \text{O} & \text{M} \\ & & & -a_{n-m-1,n} \\ & & & \end{bmatrix}$$

Generalized Jacobi method

In the Generalized Jacobi method, the matrix $A$is decomposed as $A = T_m - E_m - F_m$.

The system $Ax = b$can then be written as:

$(T_m - E_m - F_m)x = b$ or equivalently $T_m x = (E_m + F_m)x + b$.

If $T_m$is invertible, requiring $a_{ii} \neq 0$ for each i, we can express the solution as:

$$x = T_m^{-1}(E_m + F_m)x + T_m^{-1}b,$$

which defines the iterative formula of the Generalized Jacobi method:

$$x^{(k)} = {T_m}^{-1}(E_m + F_m)x^{(k-1)} + {T_m}^{-1}b, \qquad \text{k=1,2,…} \qquad (2)$$

Introducing the notation $T_{GJ} = {T_m}^{-1}(E_m + F_m)$ and $C_{GJ} = {T_m}^{-1}b$

The iteration can be expressed compactly as $x^{(k)} = T_{GJ}x^{(k-1)} + C_{GJ}$

Generalized Gauss Seidel method.

Based on the earlier definitions of T_m, E_m, and F_m, the Gauss-Seidel method for solving $Ax = b$ can be written in the following form:

$$(T_m - E_m - F_m)x = b$$
$$(T_m - E_m)x = F_m x + b$$
$$x^{(k)} = (T_m - E_m)^{-1}F_m x^{(k-1)} + (T_m - E_m)^{-1}b \quad k = 1,2,\ldots \qquad (3)$$

Let $T_{GGS} = (T_m - E_m)^{-1}F_m$ and $C_{GGS} = (T_m - E_m)^{-1}b$

The Generalized Gauss Seidel technique can be written as,

$$x^{(k)} = T_{GGS}x^{(k-1)} + C_{GGS} \qquad k = 1,2,\ldots$$

Derivation of proposed Algorithm

The proposed improved Generalized Classical Iterative Techniques for solving linear system (IGCT) is proposed by using eq (2) and eq (3) as follows.

We have eq (2), $x^{(k)} = {T_m}^{-1}(E_m + F_m)x^{(k-1)} + {T_m}^{-1}b$

And eq (3), $x^{(k)} = (T_m - E_m)^{-1}F_m x^{(k-1)} + (T_m - E_m)^{-1}b$

By applying Composite Refinement Technique we get,

$$x^{(vir)} = (T_m - E_m)^{-1}F_m x^{(k-1)} + (T_m - E_m)^{-1}b \qquad (i)$$
$$x^{(k)} = {T_m}^{-1}(E_m + F_m)x^{(vir)} + (T_m)^{-1}b \qquad (ii)$$

Replacing the relevant terms from equation (i) into equation (ii) allows us to simplify the expression.

$$x^{(k)} = {T_m}^{-1}(E_m + F_m)\begin{bmatrix}(T_m - E_m)^{-1}F_m x^{(k-1)} + \\ (T_m - E_m)^{-1}b\end{bmatrix} + (T_m)^{-1}b$$
$$x^{(k)} = {T_m}^{-1}(E_m + F_m)\,(T_m - E_m)^{-1}F_m x^{(k-1)} + {T_m}^{-1}(E_m + F_m)(T_m - E_m)^{-1}b + (T_m)^{-1}b$$
$$x^{(k)} = T_m^{-1}(E_m + F_m)(T_m - E_m)^{-1}F_m\, x^{(k-1)} + T_m^{-1}[I + (E_m + F_m)(T_m - E_m)^{-1}]b \qquad (4)$$

Equation (2.1) corresponds to the proposed IGCT algorithm, which can be transformed into the iterative form:

$$x^{(k)} = T_{\text{IGCT1}}x^{(k-1)} + C_{\text{IGCT1}}, k = 1,2,3,\ldots,$$

where the matrices are defined as:

$T_{IGCT} = T_m^{-1}(E_m + F_m)(T_m - E_m)^{-1}F_m$ and $C_{IGCT} = T_m^{-1}[I + (E_m + F_m)(T_m - E_m)^{-1}]b$

Convergence Theory

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

Theorem 1: For a square matrix A, the following statements are equivalent:

1. A is a convergent matrix.
2. There exists at least one matrix norm such that $\| A^n \| \to 0$ as $n \to \infty$.
3. The sequence $\| A^n \|$converges to zero for all matrix norms.
4. The spectral radius of $A$is less than unity, i.e., $\rho(A) < 1$.
5. The matrix power A^n converges to the zero matrix as $n \to \infty$. [21]

Theorem 2: Suppose $A = M - N$ is regular splitting. Then A is invertible and nonnegative if and only if $\rho(M^{-1}N) < 1$. [21]

Theorem 3: Let $A = (a_{ij})$ be an M-matrix. If another matrix $B = (b_{ij})$ satisfies $A \leq B$and $b_{ij} \leq 0$ for all $i \neq j$, then B is also an M-matrix. [21]

***Theorem 4*:** For any choice of initial vector $x^{(0)}$, the IGCT method converges if the coefficient matrix $A$is strictly diagonally dominant.

Proof.

It is well established that $x$is the solution of $Ax = b$. If the coefficient matrix $A$is strictly diagonally dominant, then the GGJ and RJGS iterative methods are convergent, leading to the convergence of $\overline{x^{(n+1)}}$ to $x$as $n$grows large.

$$\overline{x^{(n+1)}} = {T_m}^{-1}(E_m + F_m)x^{(n)} + {T_m}^{-1}b$$

So, $\overline{x^{(n+1)}} \to x$, we can write IGCT as,

$$x^{(n+1)} = \overline{x^{(n+1)}} + (T_m)^{-1}(b - A\,\overline{x^{(n+1)}})$$

$$x^{(n+1)} - x = \overline{x^{(n+1)}} - x + (T_m)^{-1}(b - A\,\overline{x^{(n+1)}})$$

Taking norm on both sides,

$$\| x^{(n+1)} - x\| = \|\overline{x^{(n+1)}} - x + (T_m)^{-1}(b - A\,\overline{x^{(n+1)}})\|$$

$$\| x^{(n+1)} - x\| \leq \|\overline{x^{(n+1)}} - x\| + \|(T_m)^{-1}\|\|(b - A\,\overline{x^{(n+1)}})\|$$

$$\| x^{(n+1)} - x\| \leq \|x - x\| + \|(T_m)^{-1}\|\|(b - b)\|$$

$$\| x^{(n+1)} - x\| \leq 0 + \|(T_m)^{-1}\| \times 0$$

$$\| x^{(n+1)} - x\| \leq 0$$

Consequently,

$$\| x^{(n+1)} - x \| \rightarrow 0$$

Then by using theorem 1.

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

Consequently, for every strictly diagonally dominant matrix, the IGCT-1 method exhibits convergence.

***Theorem 5*:** The IGCT method is guaranteed to converge for all initial vectors $x^{(0)}$provided that $A$is an M-matrix.

Proof:

Let $A = M - N$

But $A = T_m - E_m - F_m$

By IGCT Method,

$$M = (T_m^{-1}(E_m + F_m)(T_m - E_m)^{-1}F_m)$$
$$N = T_m^{-1}[I + (E_m + F_m)(T_m - E_m)^{-1}]b$$

$A \leq M$ is indicated by theorem 3,

$\Longrightarrow M\ is\ an\ M$ −Matrix

$\Longrightarrow\ M^{-1} > 0$ Likewise, $N^{-1} \geq 0$

Thus, A admits a regular splitting $A = M - N$. Since the inverse of $A$is nonnegative, an application of Theorem 2 yields that

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

As a result, the IGCT algorithm is convergent for any matrix belonging to the class of M-matrices.

Theorem 6: Provided that GGJ converges starting from $x^{(0)}$, the IGCT algorithm achieves faster convergence than the GGJ and RJGS methods.

Proof:

Let,$W_m = {T_m}^{-1}(E_m + F_m)$

$$Y_m = {T_m}^{-1}b$$
$$P_m = D^{-1}(L + U)(D - L)^{-1}U$$
$$Q_m = D^{-1}[I + (L + U)(D - L)^{-1}]b$$
$$R_m = T_m^{-1}(E_m + F_m)(T_m - E_m)^{-1}F_m$$
$$S_m = T_m^{-1}[I + (E_m + F_m)(T_m - E_m)^{-1}]b$$

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

Therefore, the iterative expressions associated with GGJ, RJGS, and IGCT are presented, respectively, as: $x^{(n+1)} = W_m x^{(n)} + Y_m$, $x^{(n+1)} = P_m x^{(n)} + Q_m$, and $x^{(n+1)} = R_m x^{(n)} + S_m$

The convergence criterion for GGJ is given by $\| W_m \| < 1$. Taking into account that $x$is the solution of $Ax = b$, we first focus on the GGJ method and write:

$$x^{(n+1)} = W_m x^{(n)} + Y_m$$
$$x^{(n+1)} - x = W_m x^{(n)} - x + Y_m$$

Incorporating both the addition and subtraction of $W_m x$on the right-hand side, and using the identity $x = W_m x^{(n)} + Y_m$, we arrive at:

$$x^{(n+1)} - x = W_m x^{(n)} - W_m x + W_m x + Y_m - x$$
$$x^{(n+1)} - x = W_m (x^{(n)} - x) + x - x$$
$$x^{(n+1)} - x = W_m (x^{(n)} - x)$$

Taking norm on both sides

$$\left\| x^{(n+1)} - x \right\| = \left\| W_m (x^{(n)} - x) \right\|$$
$$\left\| x^{(n+1)} - x \right\| \leq \| W_m \| \left\| (x^{(n)} - x) \right\| \quad \text{(a)}$$

Now for RJGS,

$$x^{(n+1)} = P_m x^{(n)} + Q_m$$
$$x^{(n+1)} - x = P_m x^{(n)} - x + Q_m$$

By introducing and removing $P_m x$ on the right-hand side and recalling that $x = P_m x^{(n)} + Q_m$, the equation can be rewritten as:

$$x^{(n+1)} - x = P_m x^{(n)} - P_m x + P_m x + Q_m - x$$
$$x^{(n+1)} - x = P_m (x^{(n)} - x) + x - x$$
$$x^{(n+1)} - x = P_m (x^{(n)} - x)$$

Taking norm on both sides

$$\left\| x^{(n+1)} - x \right\| = \left\| P_m (x^{(n)} - x) \right\|$$
$$\left\| x^{(n+1)} - x \right\| \leq \| P_m \| \left\| (x^{(n)} - x) \right\| \quad \text{(b)}$$

Now let us consider for IGCT ,

$$x^{(n+1)} = R_m x^{(n)} + S_m$$
$$x^{(n+1)} - x = R_m x^{(n)} - x + S_m$$

Adding and subtracting $R_m x$ on R.H.S keeping in view that $x = R_m x^{(n)} + S_m$

$$x^{(n+1)} - x = R_m x^{(n)} - R_m x + R_m x + S_m - x$$
$$x^{(n+1)} - x = R_m (x^{(n)} - x) + x - x$$
$$x^{(n+1)} - x = R_m (x^{(n)} - x)$$

Taking norm on both sides

$$\left\|x^{(n+1)} - x\right\| = \left\|R_m(x^{(n)} - x)\right\|$$

$$\left\|x^{(n+1)} - x\right\| \leq \|R_m\|\left\|(x^{(n)} - x)\right\| \quad \text{(c)}$$

Combining equations (a), (b), and (c), we obtain:

$$\|R_m\| < \|P_m\| < \|W_m\| \qquad Since\ \|W_m\| < 1$$

Thus, the IGCT method demonstrates superior convergence speed compared to the GGJ and RJGS algorithms.

Results and Discussions

A variety of linear systems, such as symmetric positive definite (SPD), strictly diagonally dominant (SDD), and M-matrix are considered to rigorously determine the efficiency of the proposed algorithm against the RJGS and RGSJ methods. Using this detailed analysis, a fair comparison between systems of different structural and spectral features can be conducted, which will support the relevance and generality of the suggested method. Three basic performance measures are used in the evaluation of the performance, i.e. the spectral radius of the iteration matrix, rate of convergence, and the overall count of the iterations taken to achieve a desired tolerance.

The relative numerical characteristics are presented in tabular form in order to allow them to be quantitatively analyzed with precision and bar charts are used in order to improve the visual decoding of the differences in the observed performances. All these graphical illustrations are a clear indication of the steady decrease in the spectral radius of the proposed method which directly translates to a rapid convergence behavior and a lower number of iterations when compared with RJGS and RGSJ. In general, the findings prove that the suggested algorithm is more efficient and stable when it comes to all the classes of linear systems under test, which makes it an effective and reliable solver through the iterative approach to large-scale problems that occur in the context of scientific and engineering calculations.

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 4, Issue 1 (2026)

Example 1: M-Matrix [22]

$$A = \begin{bmatrix} 2 & -3 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & 2 & -3 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}, \quad b = \begin{bmatrix} -\frac{5}{3} \\ \frac{2}{3} \\ 3 \\ -\frac{4}{3} \\ -\frac{1}{3} \\ \frac{5}{3} \end{bmatrix}$$

Table 1. Shows Comparison of No. of iteration, Spectral radius & rate of convergence problem 1, by Proposed Method & their Counterparts.

Methods	No. of Iterations	Spectral Radius	Rate of Convergence
RJGS	15	0.4317	0.3648
RGSJ	21	0.6106	0.2142
IGCT	09	0.2500	0.6021

Example 2: **SDD Matrix** [23]

$$\mathrm{A} = \begin{bmatrix} 4 & 1 & -1 & 0 \\ 1 & 3 & -1 & 0 \\ -1 & -1 & 5 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 8 \\ -4 \\ 6 \end{bmatrix}$$

Table 2. Shows Comparison of No. of iteration, Spectral radius & rate of convergence problem 2, by Proposed Method & their Counterparts.

Methods	No. of	Spectral	Rate Of

	Iterations	Radius	Convergence
RJGS	10	0.2172	0.6631
RGSJ	12	0.3203	0.4944
IGCT	06	0.081	1.0915

Example 3: IDD Matrix [24]

$$A = \begin{bmatrix} 1 & -1/4 & 0 & -1/4 & 0 & 0 & 0 & 0 & 0 \\ -1/4 & 1 & -1/4 & 0 & -1/4 & 0 & 0 & 0 & 0 \\ 0 & -1/4 & 1 & 0 & 0 & -1/4 & 0 & 0 & 0 \\ -1/4 & 0 & 0 & 1 & -1/4 & 0 & -1/4 & 0 & 0 \\ 0 & -1/4 & 0 & -1/4 & 1 & -1/4 & 0 & -1/4 & 0 \\ 0 & 0 & -1/4 & 0 & -1/4 & 1 & 0 & 0 & -1/4 \\ 0 & 0 & 0 & -1/4 & 0 & 0 & 1 & -1/4 & 0 \\ 0 & 0 & 0 & 0 & -1/4 & 0 & -1/4 & 1 & -1/4 \\ 0 & 0 & 0 & 0 & 0 & -1/4 & 0 & -1/4 & 1 \end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}, \quad b = \begin{bmatrix} 375 \\ 250 \\ 250 \\ 500 \\ 0 \\ 0 \\ 375 \\ 250 \\ 250 \end{bmatrix}$$

Table 3. Shows Comparison of No. of iteration, Spectral radius & rate of convergence problem 3, by Proposed Method & their Counterparts.

Methods	No. of Iterations	Spectral Radius	Rate Of Convergence
RJGS	17	0.9615	0.0171
RGSJ	17	0.3603	0.4433
IGCT	12	0.2197	0.6582

Discussion

- All computations were carried out using **MATLAB** Version 8.2.0.701 (R2014b).
- The rate of convergence was calculated using the relation $R(T_h) = -log10\ [\rho(T_h)]$. Where h=RJGS, RGSJ, IGCT.
- The results indicate that the **IGCT** method is a superior iterative approach for solving systems of linear equations when the coefficient matrix belongs to one of the following classes: SPD, SDD, IDD, M-matrix, or H-matrix.

Conclusion

This work presents a generalized composite refinement framework for stationary iterative methods. The newly developed formula, termed Improved Generalized Classical Iterative Techniques (IGCT), is compared against its classical counterparts. It has been proved that the proposed method has good convergence properties, which are validated through both rigorous theoretical analysis and comprehensive numerical experiments. The method effectively modifies the spectral radius of the associated iteration matrices, thereby significantly reducing the number of iterations required to achieve convergence. The IGCT method possesses a well-defined algorithmic structure, and five numerical case studies with distinct convergence characteristics are examined to demonstrate its robustness and efficiency. Mathematical computations were carried out using MATLAB. Overall, the results clearly indicate that IGCT consistently outperforms classical iterative methods in terms of computational efficiency, convergence speed, and numerical stability, making it a reliable and powerful alternative for solving large-scale linear systems arising in scientific and engineering applications.

References

[1] Audu, K., Yahaya, Y., Adeboye, K., & Abubakar, U. (2021). Refinement of extended accelerated over-relaxation method for solution of linear systems. *Nigerian Annals of Pure and Applied Sciences, 4*(1), 53–61. https://doi.org/10.46912/napas.226

[2] Bru, R., Corral, C., Gimenez, I., & Mas, J. (2008). Classes of general H-matrices. *Linear Algebra and Its Applications, 429*, 2358–2366. https://doi.org/10.1016/j.laa.2007.10.030

[3] Cao, Y. C. A. L., & Petzold, L. (2003). A subspace error estimate for linear systems. *SIAM Journal on Matrix Analysis and Applications, 24*(3), 603–626.

[4] Cao, Y., & Petzold, L. (2004). A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM Journal on Scientific Computing, 26*(2), 359–374.

[5] Cao, Y., Li, S., Petzold, L., & Serban, R. (2003). Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. *SIAM Journal on Scientific Computing, 24*(3), 1076–1089. https://doi.org/10.1137/S1064827501380630

[6] Congedo, M., Phlypo, R., & Pham, D. T. (2011). Approximate joint singular value decomposition of an asymmetric rectangular matrix set. *IEEE Transactions on Signal Processing, 59*(1), 415–424. https://doi.org/10.1109/TSP.2010.2087018

[7] De Wang, Z., & Huang, T. Z. (2004). Comparison results between Jacobi and other iterative methods. *Journal of Computational and Applied Mathematics, 169*(1), 45–51. https://doi.org/10.1016/j.cam.2003.10.017

[8] Hallett, B. A. J. H. (1986). The convergence of accelerated overrelaxation iterations. *Mathematics of Computation, 47*(175), 219–223.

[9] Kalambi, I. (2010). A comparison of three iterative methods for the solution of linear equations. *Journal of Applied Sciences and Environmental Management, 12*(4). https://doi.org/10.4314/jasem.v12i4.55219

[10] Youssef, I. K., & Arabia, S. (2022). Composite refinement techniques for solving linear systems. *Journal of Mathematics and Computer Science, 12*, 1–14.

[11] Baloch, M. S. R., Kalhoro, Z. A., Khalil, M. S., Shaikh, P. A. W. (2022). A new improved classical iterative algorithm for solving a system of linear equations. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, 58(4), 69–81.

[12] Lu, L., Ahmed, Z., Guan, J. (2016). Numerical methods for a quadratic matrix equation with a nonsingular M-matrix. *Applied Mathematics Letters*, 52, 46–52.

[13] Ahmed, Z., Kalhoro, Z. A., Shaikh, A. W., Baloch, M. S. R., Rajput, O. A. (2022). Animproved iterative scheme using successive over-relaxation for solution of linear system of equations: Improved SOR for linear systems of equations. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, 59(3),35–43.

[14] Kanwal, M., Kalhoro, Z. A., Jamali, S. (2025). Development of Generalized Refinement Strategies in Composite Stationary Iterative Solvers for Linear Systems. *Southern Journal of Research*, 5(2(1)),1–13.

[15] Wen, R., Li, D., Ahmed, Z., Guan, J., Rajput, O. A. (2025). A Flexible-Scalar Splitting Iterative Method for Linear Inverse Problems with Complex Symmetric Matrix. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, 62(4),1–12.

[16] Kanithi, D. S. K., Lakshmi, N. G., Muthuvizhiyan, M., & Sujatha, M. S. (2018). A study on comparison of Jacobi, Gauss-Seidel and SOR methods for the solution in system of linear equations. *International Journal of Mathematics Trends and Technology*, *56*(4), 214–222. https://doi.org/10.14445/22315373/IJMTT-V56P531

[17] Li, C., Ma, F., & Huang, T. (2011). 2-D analysis based iterative learning control for linear discrete-time systems with time delay. *Journal of Industrial Management and Optimization*, *7*(1), 175–181. https://doi.org/10.3934/jimo.2011.7.175

[18] Meligy, S. A., & Youssef, I. K. (2022). Relaxation parameters and composite refinement techniques. *Results in Applied Mathematics*, *15*, 100282. https://doi.org/10.1016/j.rinam.2022.100282

[13] Mohammed, F. D., & Rivaie, M. (2017). Jacobi-Davidson, Gauss-Seidel `and Successive Over-Relaxation for solving systems of linear equations. *Applied Mathematics and Computational Intelligence*, *6*, 41–52.

[19] Saad, Y., & Van Der Vorst, H. A. (2000). Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied*

Mathematics, *123*(1–2), 1–33. https://doi.org/10.1016/S0377-0427(00)00412-X

[20] Saha, M., & Chakravarty, J. (2020). Convergence of generalized SOR, Jacobi and Gauss–Seidel methods for linear systems. *International Journal of Applied and Computational Mathematics, 123*, 1–12. https://doi.org/10.1007/s40819-020-00830-5

[21] Stojnic, M. (2010). l2/l1-Optimization in block-sparse compressed sensing and its strong thresholds. *IEEE Journal of Selected Topics in Signal Processing, 4*(2), 350–357.

[22] Unbehauen, H., et al. (2005). Gradient-based iterative algorithms for solving a class of matrix equations. *IEEE Transactions on Automatic Control, 50*(8), 1216–1221.

[23] Varga, R. S. (2001). *Matrix iterative analysis* (9th ed.). Springer. https://doi.org/10.1007/978-3-540-78862-1

[24] Youssef, I. K., & Arabia, S. (2022). Composite refinement techniques for solving linear systems. *Journal of Mathematics and Computer Science, 12*, 1–14.