Research Article ISSN Online: 2783-4859 Computational Mathematics and Computer Modeling with Applications

Received 14 May 2023
Accepted 24 July 2023

DOI: 10.52547/CMCMA.1.1.105
AMS Subject Classification: 65F10; 65F50

Solving linear systems of equations by two-step diagonal and off-diagonal multisplitting methods

Maryam Bashirizadeh

In the realm of solving large linear systems of equations, multisplitting methods emerge as a prominent class of iterative techniques. This paper introduces two-step diagonal and off-diagonal multisplitting methods and evaluates their effectiveness in comparison to symmetric successive overrelaxation multisplitting and quasi-Chebyshev accelerated multisplitting techniques for solving linear systems of equations. Additionally, this study investigates convergence theorems when the system matrix is an H-matrix and demonstrates the effectiveness of the proposed methods by presenting numerical results. Copyright © 2022 Shahid Beheshti University.

Keywords: Iterative methods; Multisplitting; Linear system.

1. Introduction

The system of linear equations, represented as

$$Ax = b, \qquad A \in R^{n\times n},\ x \in R^n,\ b \in R^n, \tag{1}$$

is encountered in various fields such as industrial applications, engineering sciences, and economics. The growing interest in studying effective techniques for solving these systems has led to the emergence of a range of powerful methods, including direct, iterative, and multisplitting methods [2, 4, 11–17, 21, 27, 29, 31, 35, 40]. Iterative methods are particularly suited for solving systems with a sparse coefficient matrix. The basic idea behind iterative methods stems from the single splitting of the coefficient matrix A, which can be expressed as follows:

$$A = F - G, \tag{2}$$

$$x^{(k+1)} = F^{-1}Gx^{(k)} + F^{-1}b, \qquad k = 0, 1, 2, \dots, \tag{3}$$

where F is a nonsingular matrix. For any initial vectors $x^{(0)}$, the iterative scheme (3) converges to the unique solution if and only if $\rho(F^{-1}G) < 1$ (where ρ represents the spectral radius).
Jacobi, Gauss-Seidel, and successive overrelaxation methods are classical iterative methods commonly used for solving linear systems [2, 27, 35]. An effective numerical iterative method is the two-step diagonal and off-diagonal method [17]. The Krylov subspace methods, such as conjugate gradient (CG), minimal residual (MINRES), and general minimal residual (GMRES), have proven to be highly effective for positive definite, symmetric indefinite, and general sparse systems, respectively [7, 20, 22, 34]. Chebyshev iterative methods, including Chebyshev semi-iterative and quasi-Chebyshev accelerated methods, also exhibit strong performance [26, 36, 42]. Among the various iterative techniques, multisplitting methods are recognized as powerful tools for solving large linear systems of equations [1, 30]. These methods, based on multiple splittings of the coefficient matrix, were first introduced by O'Leary and White in 1985 [32]. An ordered triple (F_i, G_i, E_i) is referred to as a multisplitting of A if

1. $A = F_i - G_i$, where the matrices F_i are invertible $(i = 1, 2, \dots, l)$.
2. $\sum\limits_i E_i = I$, where the matrices E_i are diagonal and $E_i \geq 0$.

Department Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran. E-mail: m_bashirizadeh@sbu.ac.ir
** Correspondence to: M. Bashirizadeh.*

By partitioning the system matrix in several ways

$$A = F_i - G_i, \qquad i = 1, 2, ..., l,$$

and by utilizing weighting matrices E_i $(i = 1, 2, ..., l)$, a multisplitting method can be defined as follows:

Method 1 *Multisplitting iteration method* [32]
Let (F_i, G_i, E_i) $(i = 1, 2, ..., l)$ *be a multisplitting of the matrix* $A \in R^{n\times n}$.

1. *Select an initial guess* $x^{(0)} \in R^n$.
2. *For* $k = 0, 1, ...$ *until convergence.*
3. *For* $i = 0, 1, ..., l$
$$x^{(k+1,i)} = F_i^{-1}G_i x^{(k)} + F_i^{-1}b. \tag{4}$$
4. *Compute* $x^{(k+1)} = \sum_{i=1}^{l} E_i x^{(k+1,i)}$.

It has been observed in [32] that the use of a relaxed parameter significantly enhances the convergence of multisplitting methods. Deren [38] introduced a class of relaxed parallel multisplitting methods.

Method 2 *Relaxed multisplitting iteration method* [32]
Let (F_i, G_i, E_i) $(i = 1, 2, ..., l)$ *be a multisplitting of the matrix* $A \in R^{n\times n}$.

1. *Select an initial guess* $x^{(0)} \in R^n$.
2. *For* $k = 0, 1, ...$ *until convergence.*
3. *For* $i = 0, 1, ..., l$
$$x^{(k+1,i)} = F_i^{-1}G_i x^{(k)} + F_i^{-1}b. \tag{5}$$
4. *Compute* $x^{(k+1)} = \theta \sum_{i=1}^{l} E_i x^{(k+1,i)} + (1-\theta)x^{(k)}$, *where* θ *is the relaxation parameter.*

The multisplitting AOR and TOR methods were proposed for the case when the system matrix is an H-matrix [9, 10]. Subsequently, Frommer and Mayer studied the convergence properties of parallel multisplitting methods [23, 25]. In [45], Yun analyzed the convergence of both the multisplitting and relaxed multisplitting methods associated with the SSOR method. Additionally, Bai examined the convergence of two-stage multisplitting methods [3, 6].
With the multisplitting method, a task is divided into multiple subtasks, solved iteratively, then combined by weighting matrices to obtain the final result. This technique is also used to solve problems other than linear equations, such as nonlinear multisplitting methods, multisplitting for linear complementarity problems, and multisplitting preconditioned methods [5, 18, 19, 28, 39].
For solving linear systems of equations, the two-step diagonal and off-diagonal multisplitting methods, as well as the relaxed DOM (RDOM) are proposed.This paper presents numerical results to demonstrate the effectiveness of these new methods by comparing them with the symmetric successive overrelaxation multisplitting method (SSORM) and the quasi-Chebyshev accelerated multisplitting method (QCAM) [41–43, 45]. The numerical results demonstrate the efficiency and effectiveness of the proposed methods. The following is a list of the next sections.

- Section 2 contains some definitions, lemmas, and theorems used throughout the article.
- Section 3 presents new multisplitting and relaxed new multisplitting methods.
- Section 4 proposes theorems that justify the convergence of new methods under certain conditions.
- Section 5 provides examples to show that new algorithms can reduce CPU time and iteration steps (IT) compared with the SSORM and QCAM methods.

2. Preliminaries

This section aims to provide a concise introduction to some essential definitions, notations, and lemmas that are utilized throughout this paper. The matrix $A = (a_{ij}) \in R^{n\times n}$ is nonnegative or $A \geq 0$ if $a_{ij} \geq 0$ $(1 \leqslant i, j \leqslant n)$. The absolute value of matrix A is defined by $|A| = (|a_{ij}|)$. For any two matrices A and B of compatible sizes $|AB| \leq |A||B|$. In [37], it was shown that if $|A| \leq B$, then $\rho(A) \leq \rho(B)$.

The square matrix $A = (a_{ij})$ is classified as a Z-matrix if all its off-diagonal entries (entries not on the main diagonal) are non-positive. In the context of linear algebra, a matrix $A \in \mathbb{R}^{n\times n}$ is considered an M-matrix if A is a Z-matrix and nonsingular with $A^{-1} \geq 0$. Furthermore, an H-matrix is defined as a matrix $A \in \mathbb{R}^{n\times n}$ for which its comparison matrix $\langle A \rangle$ is an M-matrix. The comparison matrix $\langle A \rangle$ is

$$\langle A \rangle = (\langle a_{ij} \rangle) = \begin{cases} |a_{ij}|, i = j, \\ -|a_{ij}|, i \neq j. \end{cases} \tag{6}$$

Also, a matrix A is monotone if A is nonsingular and $A^{-1} \geq 0$.
The decomposition $A = F - G$ is called [8]

1. A regular splitting of A if F is a nonsingular matrix with $F^{-1} \geq 0$ and $G \geq 0$.
2. Weak regular splitting if $F^{-1} \geq 0$ and $F^{-1}G \geq 0$.
3. A M-splitting if F is an M-matrix and $G \geqslant 0$.

Lemma 1 *[23, 44] Let $A \in R^{n\times n}$ be an H-matrix, and $A = D - B$, where D is the diagonal part of the matrix A, then the following statements hold:*

1. *A is nonsingular and $|A^{-1}| \leq \langle A\rangle^{-1}$;*
2. *$|D|$ is nonsingular and $\rho(|D|^{-1}|B|) < 1$.*

Lemma 2 *[37] Suppose A is a nonnegative irreducible matrix, then $\rho(A)$ is an eigenvalue of A and the eigenvector x corresponding to $\rho(A)$ such that $x > 0$.*

Lemma 3 *[33] Let A be a Z-matrix, then the following statements are equivalent:*

1. *A is an M-matrix;*
2. *There exists a positive vector x, such that $Ax > 0$;*
3. *Let $A = F - G$ be a splitting of A and $F^{-1} \geq 0$, $G \geq 0$, then $\rho(F^{-1}G) < 1$.*

Lemma 4 *[37] If $|A| \leq B$, then $\rho(A) \leq \rho(B)$.*

Lemma 5 *[29] Let $A = F - G$ be a regular splitting of the matrix A. Then $\rho(F^{-1}G) < 1$ if and only if A is nonsingular and $A^{-1} \geq 0$.*

Lemma 6 *[24, 44] Assume that there exists a positive vector $(u > 0,\ u \in R^n)$ such that $|A|u < u$. Then there exists a constant $\alpha \in [0, 1)$ such that $\rho(A) \leqslant \alpha$.*

Theorem 1 *[32]*

1. *If (F_i, G_i), $\quad i = 1, 2, \ldots, l$ is a weak regular splitting of a matrix A satisfying $A^{-1} \geq 0$, then the multisplitting Method 1 is convergent;*
2. *If $\|F_i^{-1}G_i\|_\infty < 1$, then the multisplitting Method 1 is convergent.*

3. Convergence theory

By applying the multisplitting (F_i, G_i, E_i) on the DOS method [17], this section presents the DOM method and its relaxed variant (RDOM) for solving the linear system (1). Then, the convergence of these methods is discussed.

Method 3 *DOM method for solving (1)*
Given an initial vector $x^{(0)}$;
For $k = 0, 1, 2, \ldots$ until convergence do
For $i = 1, 2, \ldots, l$ do

$$\begin{aligned} x^{(k+\frac{1}{2},i)} &= D^{-1}[\alpha D + (1-\alpha)B]x^{(k)} + D^{-1}(1-\alpha)b,\\ x^{(k+1,i)} &= (D-\beta L_i)^{-1}[(1-\beta)D + \beta U_i]x^{(k+\frac{1}{2},i)} + (D-\beta L_i)^{-1}\beta b.\\ x^{(k+1)} &= \sum_{i=1}^{l} E_i x^{(k+1,i)}. \end{aligned} \tag{7}$$

The RDOM method is an extension of Method 3, where incorporates a relaxation parameter to enhance the convergence behavior.

Method 4 *RDOM method for solving (1)*
Given an initial vector $x^{(0)}$;
For $k = 0, 1, 2, \ldots$ until convergence do
For $i = 1, 2, \ldots, l$ do

$$\begin{aligned} x^{(k+\frac{1}{2},i)} &= D^{-1}[\alpha D + (1-\alpha)B]x^{(k)} + D^{-1}(1-\alpha)b,\\ x^{(k+1,i)} &= (D-\beta L_i)^{-1}[(1-\beta)D + \beta U_i]x^{(k+\frac{1}{2},i)} + (D-\beta L_i)^{-1}\beta b.\\ x^{(k+1)} &= (\theta)\sum_{i=1}^{l} E_i x^{(k+1,i)} + (1-\theta)x^{(k)}. \end{aligned} \tag{8}$$

In both methods, suppose the coefficient matrix A can be decomposed into $A = D - L_i - U_i$, where D is a diagonal matrix, L_i's are strictly lower triangular matrices, and U_i's are general matrices. By decomposing the matrix this way, diagonal and off-diagonal components can be separated, which can help with iterative solving techniques. The next step is to discuss the convergence of the proposed methods.

Theorem 2 *Let $A \in R^{n\times n}$ be an H-matrix, $b \in R^n$, $D = diag(A)$, (F_i, G_i, E_i), $(i = 1, 2, ..., l)$ be a multisplitting of A and $\langle A\rangle = |D| - |L_i| - |U_i|$. Then, the multisplitting Method 3 converges to the exact solution of $Ax = b$ for any initial vector $x^{(0)}$, provided that $0 \le \alpha \le 1$ and $0 < \beta \le 1$.*

Proof. The DOM Algorithm 3 can be written as

$$x^{(k+1)} = \sum_{i=1}^{l} E_i F_i^{-1} G_i x^{(k)} + \sum_{i=1}^{l} E_i F_i^{-1} b, \tag{9}$$

where

$$\begin{aligned} F_i =& \frac{1}{1-\alpha+\alpha\beta} D[D + \frac{(1-\alpha)\beta}{1-\alpha+\alpha\beta} U_i]^{-1} (D - \beta L_i),\\ G_i =& \frac{1}{1-\alpha+\alpha\beta} D[D + \frac{(1-\alpha)\beta}{1-\alpha+\alpha\beta} U_i]^{-1} [(1-\beta)D + \beta U_i] D^{-1} [\alpha D + (1-\alpha)B]. \end{aligned} \tag{10}$$

The iteration matrix is $T = \sum_{i=1}^{l} E_i F_i^{-1} G_i$. It is sufficient to prove that $\rho(T) < 1$.
By applying the inequality $\rho(T) \le \rho(|T|)$, it is sufficient to demonstrate that $\rho(|T|) < 1$. As

$$\begin{aligned} |T| = |\sum_{i=1}^{l} E_i F_i^{-1} G_i| &\le \sum_{i=1}^{l} E_i |F_i^{-1} G_i| \\ &\le \sum_{i=1}^{l} E_i |F_i^{-1}||G_i| \\ &\le \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} \tilde{G}_i, \end{aligned} \tag{11}$$

where

$$\begin{aligned} \tilde{F}_i =& |D|[|D| + \frac{(1-\alpha)\beta}{1-\alpha+\alpha\beta} |U_i|]^{-1} (|D| - \beta |L_i|),\\ \tilde{G}_i =& |D|[|D| + \frac{(1-\alpha)\beta}{1-\alpha+\alpha\beta} |U_i|]^{-1} [(1-\beta)|D| + \beta |U_i|] |D|^{-1} [\alpha |D| + (1-\alpha)|B|]. \end{aligned} \tag{12}$$

For $0 \le \alpha \le 1$ and $0 < \beta \le 1$

$$\tilde{F}_i - \tilde{G}_i = (1-\alpha+\alpha\beta)\langle A\rangle.$$

As $\tilde{F}_i^{-1} \ge 0$ and $\tilde{G}_i \ge 0$, the splitting $\tilde{F}_i - \tilde{G}_i$ is a regular splitting of $(1-\alpha+\alpha\theta)\langle A\rangle$ for $i = 1, 2, ..., l$. Additionally, since A is an H-matrix, thus $\langle A\rangle^{-1} \ge 0$. Therefore, based on Lemma 3 and Theorem 1,

$$\rho(\sum_{i=1}^{l} E_i \tilde{F}_i^{-1} \tilde{G}_i) < 1. \tag{13}$$

This implies that

$$\rho(\sum_{i=1}^{l} E_i F_i^{-1} G_i) < 1.$$

□

The following theorem presents the convergence of the RDOM Method 4 for solving equation (1).

Theorem 3 *Assume that $A \in R^{n\times n}$ is an H-matrix, (F_i, G_i, E_i) $(i = 1, 2, ..., l)$ is a multisplitting of A, $\langle A\rangle = |D| - |L_k| - |U_k|$. When $\lambda = \rho(|D|^{-1}|B|)$ and $\gamma = 1 - \alpha + \alpha\beta$, the RDOM method is convergent for any initial vector if the parameters satisfy $0 \le \alpha \le 1$, $0 < \beta \le 1$ and $0 < \theta < \frac{2}{2-(1-\lambda)\gamma}$.*

Proof. In RDOM method,

$$x^{(k+1)} = \theta \sum_{i=1}^{l} E_i F_i^{-1} G_i x^{(k)} + \theta \sum_{i=1}^{l} E_i F_i^{-1} b + (1-\theta) x^{(k)},$$

that F_i's and G_i's are defined by (10).
Let $T_\theta = \theta \sum_{i=1}^{l} E_i F_i^{-1} G_i + (1-\theta)$, Since $\rho(T_\theta) \leq \rho(|T_\theta|)$ it is suffices to show $\rho(|T_\theta|) < 1$.

$$|T_\theta| = |\theta \sum_{i=1}^{l} E_i F_i^{-1} G_i + (1-\theta) I| \leqslant \theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} \tilde{G}_i + (1-\theta) I,$$

where $\tilde{F}_i$'s and $\tilde{G}_i$'s are defined in (12). Thus

$$\tilde{G}_i = \tilde{F}_i - (1-\alpha+\alpha\beta)\langle A \rangle = \tilde{F}_i - \gamma(|D|-|B|).$$

As a result,

$$\begin{aligned}
|T_\theta| &\leqslant \theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} \tilde{G}_i + |1-\theta| I \\
&\leqslant \theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} \\
&= \theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} (\tilde{F}_i - \gamma(|D|-|B|)) + |1-\theta| I \\
&= \theta \sum_{i=1}^{l} E_i (I - \gamma(\tilde{F}_i^{-1}(|D|-|B|))) + |1-\theta| I \\
&= \theta I + |1-\theta| I - \gamma\theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} |D| (I - |D|^{-1}|B|)
\end{aligned} \tag{14}$$

According to the given information, for any $\varepsilon > 0$, the matrix $|D|^{-1}|B| + \varepsilon ee^T$ has only positive entries and is irreducible, where $e = (1, 1, ..., 1)^T \in R^n$ [23]. By Lemma 2, for any $\varepsilon > 0$ there exist a positive vector $u > 0$ such that $(|D|^{-1}|B| + \varepsilon ee^T)u = \lambda_\varepsilon u$, where $\lambda_\varepsilon = \rho(|D|^{-1}|B| + \varepsilon ee^T)$.

$$\begin{aligned}
|T_\theta| &\leqslant \theta I + |1-\theta| I - \gamma\theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} |D| (I - |D|^{-1}|B|) \\
&\leqslant \theta I + |1-\theta| I - \gamma\theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} |D| (I - |D|^{-1}|B| - \varepsilon ee^T)
\end{aligned}$$

then

$$\begin{aligned}
|T_\theta| u &\leqslant (\theta I + |1-\theta| I - \gamma\theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} |D| (I - |D|^{-1}|B| - \varepsilon ee^T)) u \\
&= (\theta + |1-\theta|) u - \gamma\theta \sum_{i=1}^{l} E_i \tilde{F}_i^{-1} |D| (I - |D|^{-1}|B| - \varepsilon ee^T) u \\
&= \theta u + |1-\theta| u - \gamma\theta (u - \lambda_\varepsilon u) \\
&= \begin{cases} (1-(1-\lambda_\varepsilon)\gamma\theta) u < u & 0 < \theta \leq 1 \\ (2\theta - 1 - (1-\lambda_\varepsilon)\gamma\theta) u < u & 1 < \theta < \dfrac{2}{2-(1-\lambda)\gamma}. \end{cases}
\end{aligned} \tag{15}$$

By considering the specified constraints $0 \leq \alpha \leq 1$ and $0 < \beta \leq 1$, it can be inferred that $0 < \gamma \leq 1$ and $1 - \gamma + \lambda\gamma = 1 - (\lambda - 1)\gamma < 1$.
By the continuity of the spectral radius, one can deduce $\lambda_\varepsilon < 1$ and $1 - \gamma + \lambda_\varepsilon \gamma < 1$ for sufficiently small $\varepsilon > 0$. As per Lemma 6, it follows that $\rho(T_\theta) < 1$ □

Remark 1 *Theorems 2 and 3 satisfies for M-matrix $A \in R^{n\times n}$. Since A is an M-matrix conclude that A is an H-matrix ($A = \langle A \rangle$).*

4. Numerical results

The purpose of this section is to demonstrate the effectiveness of DOM and RDOM methods (Method 3 and Method 4) using three numerical examples. DOM and RDOM methods are compared with the quasi-Chebyshev accelerated multisplitting method (QCAM) [41, 42] and numerical results are reported. Also, as DOM and RDOM methods are two-step multisplitting methods, these are compared with symmetric successive overrelaxation multisplitting multisplitting (SSORM) [45] method which is a two-step mutisplitting method too. In this way reported numerical results show that presented methods are performed efficiently. Numerical computations are conducted under the following assumptions:

- The matrix $A \in R^{n\times n}$ is an H-matrix and is decomposed as $A = D - L_i - U_i$ for $i = 1, 2, ..., l$, where $D = \text{diag}(A)$.
- The initial guess for the solution is $x^{(0)} = (0, 0, ..., 0)^T \in R^n$.
- The iteration process stops when $||Ax - b||_2 < 10^{-5}$.
- The weight matrices are defined as $E_i = Diag(0, 0, ..., I_{S_i}, 0, 0, ..., 0) \in R^{n\times n}$, where the size S_i is determined as follows: $S_i = \begin{cases} \psi_q + 1 & i \leqslant \psi_r, \\ \psi_q & otherwise. \end{cases}$
  Note that $n = \psi_q l + \psi_r$ and $0 \leqslant \psi_r < l$.
- In the QCAM method, M_i represents the lower-triangular part of $E_i A E_i$, and $N_i = M_i - A$.

It is worth mentioning that all the examples are implemented in MATLAB 2020 on a personal computer (16 GB RAM).
The following examples illustrate the application of the DOM, RDOM, SSORM, and QCAM algorithms to solve of linear systems.

Example 1 *[4, 17] The system of linear equations (1) is presented as follows in this example.*

$$(wC_V + C_H)x = b,$$

where

- *C_V and C_H are the viscous and hysteretic damping matrices, respectively, and w is the driving circular frequency.*
- *$C_V = 10I$ and $C_H = \mu H$ with a damping coefficient μ, and $H \in R^{n\times n}$ is the five-point centered difference matrix approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0, 1] \times [0, 1]$ with the mesh-size $r = \frac{1}{m+1}$.*
- *$H = I \otimes W_m + W_m \otimes I$, with $W_m = r^{-2} tridiag(-1, 2, -1) \in R^{m\times m}$. Then H is an $n \times n$ block-tridiagonal matrix, with $n = m^2$.*
- *$w = \pi$, $\mu = 0.02$, and $b = (-w^2 I + H + wC_V + C_H)B$, with B being the vector of all entries equal to 1.*

Example 2 *[4, 17] The system of linear equations*

$$(I \otimes W + W \otimes I)x = q,$$

where

- *$W = tridiag(-1, 2, -1) \in R^{m\times m}$,*
- *$q = [10(I \otimes W_C + W_C \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I - (I \otimes W + W \otimes I)]B$,*
- *$W_C = W - e_1 e_m^T + e_m e_1^T \in R^{m\times m}$, e_1 and e_m are unit vectors,*
- *B is the vector that all entries are one.*

Example 3 *[4, 17] A system of linear equations is represented by the following:*

$$(H + \frac{(3 + \sqrt{3})I}{\tau})x = b,$$

Here are the details:

- *The time step-size is denoted by τ.*
- *The matrix K is a five-point centered difference matrix that approximates the negative Laplacian operator $L = -\Delta$ with homogeneous Dirichlet boundary conditions. It is defined on a uniform mesh in the domain $[0, 1] \times [0, 1]$ with a mesh size of $r = \frac{1}{m+1}$.*
- *The matrix $H \in R^{n\times n}$ is defined as $H = I \otimes W_m + W_m \otimes I$, where I is the identity matrix and $W_m = r^{-2}$ tridiag $(-1, 2, -1) \in R^{m\times m}$.*
- *The matrix $H \in R^{n\times n}$ is a block-tridiagonal matrix with size $n = m^2$.*
- *In the numerical computations, $\tau = r$, and the entries of vector b are defined as $b_j = \frac{j}{\tau(j+1)^2}$ for $j = 1, 2, \ldots, n$. These specific choices of parameters and matrices define the system of linear equations.*

Table 1. Numerical results for examples with $\alpha = 0,\ \beta = 1$

			Ex 1			Ex 2			Ex 3	
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.01015	14	9.3796e-06	0.1977	1177	9.9624e-06	0.035612	226	9.6784e-06
30	DOM	0.011293	20	6.7773e-06	0.22549	1414	9.9296e-06	0.039753	272	9.8892e-06
30	QCAM	0.0073556	44	7.0581e-06	0.33806	2090	9.9948e-06	0.10576	681	9.8875e-06
30	SSORM	0.28931	143	9.3089e-06				2.5851	1850	9.9512e-06
50	RDOM	0.020519	38	6.2426e-06	1.2841	3057	9.9941e-06	0.16085	426	9.9879e-06
50	DOM	0.025147	47	9.175e-06	1.5531	3670	9.9803e-06	0.18858	513	9.8566e-06
50	QCAM	0.05054	109	9.7182e-06	2.3486	5515	9.9972e-06	0.51974	1311	9.9793e-06
50	SSORM	5.0108	324	9.6978e-06				15.8979	3479	9.9831e-06
70	RDOM	0.072625	72	7.8765e-06	5.1269	5758	9.9862e-06	0.49233	634	9.999e-06
70	DOM	0.089581	88	8.9547e-06	6.0701	6910	9.9981e-06	0.49233	763	9.8453e-06
70	QCAM	0.21233	209	9.497e-06				1.8856	1960	9.9631e-06
70	SSORM	19.4244	598	9.7495e-06				50.6818	5168	9.9882e-06

Table 2. Numerical results for examples with $\alpha = 0.25,\ \beta = 1$

			Ex 1			Ex 2			Ex 3	
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.0038226	16	9.9099e-06	0.3196	1386	9.9277e-06	0.055384	247	9.5136e-06
30	DOM	0.0051694	22	9.2366e-06	0.38129	1664	9.9577e-06	0.073149	297	9.785e-06
30	QCAM	0.01749	44	7.0581e-06	0.35256	2090	9.9948e-06	0.14223	681	9.8875e-06
30	SSORM	0.090516	143	7.7415e-06				3.3864	1850	9.9512e-06
50	RDOM	0.02412	42	6.5746e-06	2.0975	3598	9.9724e-06	0.23117	465	9.936e-06
50	DOM	0.028639	52	8.5205e-06	2.5658	4318	9.9929e-06	0.28079	559	9.9922e-06
50	QCAM	0.063078	109	9.7182e-06	2.2163	5515	9.9972e-06	0.69906	1311	9.9793e-06
50	SSORM	4.888	324	9.6978e-06				19.6991	3479	9.9831e-06
70	RDOM	0.08909	79	8.229e-06	7.7513	6775	9.9854e-06	0.72537	692	9.9419e-06
70	DOM	0.11093	97	8.1446e-06	9.2484	8131	9.9872e-06	0.88691	832	9.89e-06
70	QCAM	0.9262	209	9.497e-06				2.3514	1960	9.9631e-06
70	SSORM	47.9849	598	9.7495e-06				63.0641	5168	9.9882e-06

Table 3. Numerical results for examples with $\alpha = 0.5,\ \beta = 1$

			Ex 1			Ex 2			Ex 3	
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.0047245	19	4.7584e-06	0.28384	1683	9.9976e-06	0.042523	271	9.8264e-06
30	DOM	0.0055472	25	7.0014e-06	0.32785	2021	9.9864e-06	0.055066	327	9.6275e-06
30	QCAM	0.0099598	44	7.0581e-06	0.35229	2090	9.9948e-06	0.14542	681	9.8875e-06
30	SSORM	0.26405	143	9.3089e-06				2.6091	1850	9.9512e-06
50	RDOM	0.026523	46	9.3099e-06	1.8926	4369	9.9972e-06	0.19539	512	9.8066e-06
50	DOM	0.032858	58	7.6432e-06	2.2523	5244	9.9964e-06	0.23563	615	9.9371e-06
50	QCAM	0.066155	109	9.7182e-06	2.986	5515	9.9972e-06	0.74652	1311	9.9793e-06
50	SSORM	5.0893	324	9.6978e-06				14.8037	3479	9.9831e-06
70	RDOM	0.32	87	9.242e-06	7.3496	8227	9.9949e-06	0.63956	761	9.9617e-06
70	DOM	0.39788	107	8.4574e-06	8.809	9874	9.9895e-06	0.78244	915	9.8874e-06
70	QCAM	0.2841	209	9.497e-06				1.8451	1960	9.9631e-06
70	SSORM	41.9837	598	9.7495e-06				49.422	5168	9.9882e-06

Table 4. Numerical results for examples with $\alpha = 0.75,\ \beta = 1$

		Ex 1			Ex 2			Ex 3		
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.003761	21	9.0596e-06	0.53547	2143	9.995e-06	4.2988	1850	9.9512e-06
30	DOM	0.0054731	28	7.4192e-06	0.56104	2573	9.9862e-06	0.095181	362	9.9532e-06
30	QCAM	0.029164	44	7.0581e-06	0.33128	2090	9.9948e-06	0.14803	681	9.8875e-06
30	SSORM	0.26912	143	9.3089e-06				4.2988	1850	9.9512e-06
50	RDOM	0.021068	52	8.1887e-06	2.4537	5562	9.986e-06	0.29372	568	9.9481e-06
50	DOM	0.027769	64	9.8371e-06	2.8819	6675	9.9957e-06	0.35372	683	9.9213e-06
50	QCAM	0.063551	109	9.7182e-06	2.986	5515	9.9972e-06	0.52747	1311	9.9793e-06
50	SSORM	3.9336	324	9.6978e-06				19.7413	3479	9.9831e-06
70	RDOM	0.083615	97	9.6837e-06	9.2166	8333	9.9975e-06	0.87146	846	9.856e-06
70	DOM	0.10342	119	8.9141e-06	10.9394	9009	9.9980e-06	1.0559	1016	9.9103e-06
70	QCAM	0.27784	209	9.497e-06				2.3016	1960	9.963e-06
70	SSORM	17.4414	598	9.7495e-06				64.0521	5168	9.9882e-06

Table 5. Numerical results for examples with $\alpha = 0,\ \beta = 0.25$

		Ex 1			Ex 2			Ex 3		
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.0097478	41	8.4529e-06	0.37057	1604	9.9869e-06	0.10806	529	9.8507e-06
30	DOM	0.0099201	44	6.487e-06	0.44684	1926	9.9877e-06	0.1284	636	9.8622e-06
30	QCAM	0.01749	44	7.0581e-06	0.35256	2090	9.9948e-06	0.14223	681	9.8875e-06
30	SSORM	0.090516	143	7.7415e-06				3.3864	1850	9.9512e-06
50	RDOM	0.049363	87	8.736e-06	2.7401	4168	9.9936e-06	0.51713	998	9.9566e-06
50	DOM	0.059115	106	9.5432e-06	2.9104	5003	9.9881e-06	0.6235	1199	9.9412e-06
50	QCAM	0.063078	109	9.7182e-06	2.2163	5515	9.9972e-06	0.69906	1311	9.9793e-06
50	SSORM	4.888	324	9.6978e-06				19.6991	3479	9.9831e-06
70	RDOM	0.18627	166	9.1805e-06	9.2233	7851	9.9888e-06	1.5659	1485	9.9284e-06
70	DOM	0.22337	201	9.4063e-06	16.2836	9422	9.9927e-06	1.8719	1783	9.9457e-06
70	QCAM	0.9262	209	9.497e-06				2.3514	1960	9.9631e-06
70	SSORM	47.9849	598	9.7495e-06				63.0641	5168	9.9882e-06

Table 6. Numerical results for examples with $\alpha = 0,\ \beta = 0.5$

		Ex 1			Ex 2			Ex 3		
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.0065586	27	6.9975e-06	0.31866	1401	9.9522e-06	0.08933	408	9.8883e-06
30	DOM	0.008244	35	6.2202e-06	0.38138	1682	9.9819e-06	0.10436	491	9.8534e-06
30	QCAM	0.01749	44	7.0581e-06	0.35256	2090	9.9948e-06	0.14223	681	9.8875e-06
30	SSORM	0.090516	143	7.7415e-06				3.3864	1850	9.9512e-06
50	RDOM	0.03668	68	7.6513e-06	2.0939	3639	9.9936e-06	0.40111	770	9.9159e-06
50	DOM	0.044844	83	9.0989e-06	2.5175	4368	9.9926e-06	0.48167	925	9.9497e-06
50	QCAM	0.063078	109	9.7182e-06	2.2163	5515	9.9972e-06	0.69906	1311	9.9793e-06
50	SSORM	4.888	324	9.6978e-06				19.6991	3479	9.9831e-06
70	RDOM	0.14666	128	9.9616e-06	7.7764	6854	9.9939e-06	1.189	1145	9.9596e-06
70	DOM	0.17186	156	9.5452e-06	9.3439	8226	9.9929e-06	1.435	1375	9.9821e-06
70	QCAM	0.9262	209	9.497e-06				2.3514	1960	9.9631e-06
70	SSORM	47.9849	598	9.7495e-06				63.0641	5168	9.9882e-06

Note that:

- In Tables 1-4, results are presented for Examples 1, 2 and 3, in which $\beta = 1$ is fixed and α is variable.

Table 7. Numerical results for examples with $\alpha = 0,\ \beta = 0.75$

			Ex 1			Ex 2			Ex 3	
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.0048088	20	9.2545e-06	0.29468	1269	9.9813e-06	0.068021	309	9.7821e-06
30	DOM	0.0060748	27	6.6341e-06	0.34602	1524	9.9824e-06	0.082215	372	9.8034e-06
30	QCAM	0.01749	44	7.0581e-06	0.35256	2090	9.9948e-06	0.14223	681	9.8875e-06
30	SSORM	0.090516	143	7.7415e-06				3.3864	1850	9.9512e-06
50	RDOM	0.02849	51	9.5379e-06	1.9372	3297	9.9819e-06	0.30478	583	9.858e-06
50	DOM	0.036185	64	7.8906e-06	2.3252	3957	9.9983e-06	0.46312	701	9.8325e-06
50	QCAM	0.063078	109	9.7182e-06	2.2163	5515	9.9972e-06	0.69906	1311	9.9793e-06
50	SSORM	4.888	324	9.6978e-06				19.6991	3479	9.9831e-06
70	RDOM	0.12552	98	8.3207e-06	7.0736	6209	9.9953e-06	0.88752	867	9.9049e-06
70	DOM	0.13211	119	9.3644e-06	8.553	7452	9.9942e-06	1.0718	1041	9.9822e-06
70	QCAM	0.9262	209	9.497e-06				2.3514	1960	9.9631e-06
70	SSORM	47.9849	598	9.7495e-06				63.0641	5168	9.9882e-06

Table 8. Numerical results for examples with $\alpha = 0.1,\ \beta = 0.9$

			Ex 1			Ex 2			Ex 3	
m	method	CPU	IT	error	CPU	IT	error	CPU	IT	error
30	RDOM	0.005064	18	4.6228e-06	0.25443	1290	9.9895e-06	0.056361	268	9.6315e-06
30	DOM	0.0061406	24	6.1046e-06	0.27217	1550	9.9283e-06	0.069226	322	9.9604e-06
30	QCAM	0.01749	44	7.0581e-06	0.35256	2090	9.9948e-06	0.14223	681	9.8875e-06
30	SSORM	0.090516	143	7.7415e-06				3.3864	1850	9.9512e-06
50	RDOM	0.026595	45	7.4556e-06	1.4808	3351	9.9852e-06	0.27769	505	9.8882e-06
50	DOM	0.032598	56	8.1305e-06	1.7957	4022	9.9957e-06	0.32521	607	9.9399e-06
50	QCAM	0.063078	109	9.7182e-06	2.2163	5515	9.9972e-06	0.69906	1311	9.9793e-06
50	SSORM	4.888	324	9.6978e-06				19.6991	3479	9.9831e-06
70	RDOM	0.14619	85	8.8737e-06	6.2893	6311	9.9865e-06	0.77935	751	9.9535e-06
70	DOM	0.12361	104	9.0912e-06	8.6406	7574	9.9914e-06	0.95427	903	9.8783e-06
70	QCAM	0.9262	209	9.497e-06				2.3514	1960	9.9631e-06
70	SSORM	47.9849	598	9.7495e-06				63.0641	5168	9.9882e-06

- In Tables 5-7, numerical results are shown for three examples in which $\alpha = 0$ is fixed and β is variable.
- In Table 8, the results are shown for three examples in which α and β are not one and zero.
- The parameters involving QCAM and RDOM were chosen to be the experimentally found optimal parameters. Additionally, in the SSORM, $w = 0.2$.

As a result of the provided tables, it is possible to evaluate the performance of the methods as follows.

1. CPU (time in seconds): For various parameters of α and β, the presented methods remain superior as the problem size increases. As a result, new methods for solving examples converge faster than SSORM and QCAM in all cases. Using $\beta = 1$ as a fixed value and α as a variable, Tables 1-4 illustrate that results are improved when α is decreased. In Tables 5-7, it is shown that by increasing β, the results are improved if α is set to 0.
2. IT (iteration steps): As in the previous item, the new method has fewer iteration steps than the old method, and this advantage is maintained by increasing the size of the problem.
3. Error ($||Ax - b||_2 < 10^{-5}$): It has been observed that the performance of the methods is almost similar with regard to error.
4. As far as convergence results are concerned, the numerical findings in Ex 2 demonstrate that the new methods are convergent, whereas the old methods are divergent.

Overall, it can be concluded that the new methods outperform the mentioned techniques in terms of CPU time and iteration steps under certain conditions.

5. Conclusions

For solving a linear system of equations $Ax = b$, two-step diagonal and off-diagonal multisplitting (DOM) methods, as well as relaxed DOM (RDOM) methods are presented. Under the assumption that the coefficient matrix is an H-matrix, the convergence properties of these methods have been discussed. A comparison of the DOM and RDOM methods to existing approaches (QCAM and SSORM) is demonstrated through numerical examples. The applicability of these approaches in different contexts can also be examined, and their performance can be improved through algorithmic improvements.

6. Acknowledgments

The author sincerely appreciates the reviewer's valuable comments, which greatly improved the original manuscript of this paper.

References

1.
2. O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1996.
3. Z.-Z. Bai. Convergence analysis of the two-stage multisplitting method. *Calcolo*, 36(2):63–74, 1999.
4. Z.-Z. Bai, M. Benzi, and F. Chen. Modified HSS iteration methods for a class of complex symmetric linear systems. *Computing*, 87(3-4):93–111, 2010.
5. Z.-Z. Bai and D. J. Evans. Matrix multisplitting methods with applications to linear complementarity problems: Parallel asynchronous methods. *International Journal of Computer Mathematics*, 79(2):205–232, 2002.
6. Z.-Z. Bai and C.-L. Wang. On the convergence of nonstationary multisplitting two-stage iteration methods for hermitian positive definite linear systems. *Journal of Computational and Applied Mathematics*, 138(2):287–296, 2002.
7. F. B. Balani and M. Hajarian. On the generalized AOR and CG iteration methods for a class of block two-by-two linear systems. *Numerical Algorithms*, pages 1–17, 2022.
8. A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.
9. D.-W. Chang. On the interval multisplitting AOR method. *International of Journal of Reliable Computing*, pages 52–53, 1995.
10. D.-W. Chang. Convergence analysis of the parallel multisplitting TOR method. *Journal of Computational and Applied Mathematics*, 72(1):169–177, 1996.
11. A. Chronopoulos. On the squared unsymmetric Lanczos method. *Journal of computational and applied Mathematics*, 54(1):65–78, 1994.
12. A. T. Chronopoulos. S-step iterative methods for (non) symmetric (in) definite linear systems. *SIAM journal on numerical analysis*, 28(6):1776–1789, 1991.
13. A. T. Chronopoulos and D. Kincaid. On the Odir iterative method for non-symmetric indefinite linear systems. *Numerical linear algebra with applications*, 8(2):71–82, 2001.
14. A. T. Chronopoulos and A. B. Kucherov. Block s-step Krylov iterative methods. *Numerical Linear Algebra with Applications*, 17(1):3–15, 2010.
15. A. T. Chronopoulos and C. D. Swanson. Parallel iterative S-step methods for unsymmetric linear systems. *Parallel Computing*, 22(5):623–641, 1996.
16. M. Dehghan, M. Dehghani-Madiseh, and M. Hajarian. A generalized preconditioned MHSS method for a class of complex symmetric linear systems. *Mathematical Modelling and Analysis*, 18(4):561–576, 2013.
17. M. Dehghan, M. Dehghani-Madiseh, and M. Hajarian. A two-step iterative method based on diagonal and off-diagonal splitting for solving linear systems. *Filomat*, 31(5):1441–1452, 2017.
18. M. Dehghan and M. Hajarian. Convergence of SSOR methods for linear complementarity problems. *Operations Research Letters*, 37(3):219–223, 2009.
19. M. Dehghan and M. Hajarian. Asynchronous multisplitting GAOR method and asynchronous multisplitting SSOR method for systems of weakly nonlinear equations. *Mediterranean journal of mathematics*, 7:209–223, 2010.
20. M. Dehghan and M. Hajarian. Modied AOR iterative methods to solve linear systems. *Journal of Vibration and Control*, 20(5):661–669, 2014.
21. J. W. Demmel. *Applied Numerical linear Algebra*. SIAM,Philadelphia, 1997.
22. W. Ford. *Numerical linear algebra with applications: Using MATLAB*. Academic Press, 2014.
23. A. Frommer and G. Mayer. Convergence of relaxed parallel multisplitting methods. *Linear Algebra and its Applications*, 119:141–152, 1989.
24. A. Frommer and H. Schwandt. A unified representation and theory of algebraic additive Schwarz and multisplitting methods. *SIAM Journal on Matrix Analysis and Applications*, 18(4):893–912, 1997.
25. A. Frommer and D. B. Szyld. Asynchronous two-stage iterative methods. *Numerische Mathematik*, 69(2):141–153, 1994.
26. G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.
27. A. Hadjidimos. Successive overrelaxation (SOR) and related methods. *Journal of Computational and Applied Mathematics*, 123(1-2):177–199, 2000.
28. C.-M. Huang and D. P. O'Leary. Preconditioning parallel multisplittings for solving linear systems of equations. In *Proceedings of the 6th international conference on Supercomputing*, pages 478–484, 1992.
29. H. Kotakemori, H. Niki, and N. Okamoto. Accelerated iterative method for Z-matrices. *Journal of Computational and Applied Mathematics*, 75(1):87–97, 1996.

30. W. Li and W. Sun. Comparison results for parallel multisplitting methods with applications to AOR methods. *Linear algebra and its applications*, 331(1-3):131–144, 2001.
31. J. A. Meijerink and H. A. Van Der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31(137):148–162, 1977.
32. D. P. O leary and R. E. White. Multi-splittings of matrices and parallel solution of linear systems. *SIAM Journal on algebraic discrete methods*, 6(4):630–640, 1985.
33. H. Ren, X. Wang, X.-B. Tang, and T. Wang. The general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems. *Computers & Mathematics with Applications*, 77(4):1071–1081, 2019.
34. V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59, 2007.
35. L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
36. R. S. Varga. A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials. *Journal of the Society for Industrial and Applied Mathematics*, 5(2):39–46, 1957.
37. R. S. Varga. *Matrix iterative analysis*, volume 27. Springer Science & Business Media, 1999.
38. D. Wang. On the convergence of the parallel multisplitting AOR algorithm. *Linear algebra and its applications*, 154:473–486, 1991.
39. G. Wang and D. Sun. Preconditioned parallel multisplitting USAOR method for H-matrices linear systems. *Applied Mathematics and Computation*, 275:156–164, 2016.
40. R. Wen and H. Duan. A parallel multisplitting method with self-adaptive weightings for solving H-matrix linear systems. *Journal of Inequalities and Applications*, 2017(1):1–10, 2017.
41. R. Wen, G. Meng, and C. Wang. Parallel quasi-Chebyshev acceleration to nonoverlapping multisplitting iterative methods based on optimization. *Journal of Computational Mathematics*, pages 284–296, 2014.
42. R.-P. Wen, G.-Y. Meng, and C.-L. Wang. Quasi-Chebyshev accelerated iteration methods based on optimization for linear systems. *Computers & Mathematics with Applications*, 66(6):934–942, 2013.
43. R.-P. Wen, F.-J. Ren, and G.-Y. Meng. Modified quasi-Chebyshev acceleration to nonoverlapping parallel multisplitting method. *Numerical Algorithms*, 75:1123–1140, 2017.
44. S.-L. Wu and C.-X. Li. Two-sweep modulus-based matrix splitting iteration methods for linear complementarity problems. *Journal of Computational and Applied Mathematics*, 302:327–339, 2016.
45. J. H. Yun. Convergence of SSOR multisplitting method for an H-matrix. *Journal of Computational and Applied Mathematics*, 217(1):252–258, 2008.

