Applications of Mathematics

Alireza Fakharzadeh Jahromi; Nafiseh Nasseri Shams

A new optimized iterative method for solving M-matrix linear systems

Applications of Mathematics, Vol. 67 (2022), No. 3, 251–272

Persistent URL: http://dml.cz/dmlcz/150314

A NEW OPTIMIZED ITERATIVE METHOD FOR SOLVING M-MATRIX LINEAR SYSTEMS

Alireza Fakharzadeh Jahromi, Nafiseh Nasseri Shams, Shiraz

Received September 1, 2020. Published online November 22, 2021.

Abstract. In this paper, we present a new iterative method for solving a linear system, whose coefficient matrix is an M-matrix. This method includes four parameters that are obtained by the accelerated overrelaxation (AOR) splitting and using the Taylor approximation. First, under some standard assumptions, we establish the convergence properties of the new method. Then, by minimizing the Frobenius norm of the iteration matrix, we find the optimal parameters. Meanwhile, numerical results on test examples show the efficiency of the new proposed method in contrast with the Hermitian and skew-Hermitian splitting (HSS), AOR methods and a modified version of the AOR (QAOR) iteration.

Keywords: linear system; M-matrix; optimal parameter; Taylor approximation; optimization

MSC 2020: 90C99, 65F10

1. Introduction

Lately, the iterative solution methods of a linear system of equations

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

have been regarded in many scientific calculations and engineering problems, where the coefficient matrix $A \in \mathbb{R}^{n\times n}$ is nonsingular, $b \in \mathbb{R}^n$ is a given right-hand side vector and $x \in \mathbb{R}^n$ is an unknown vector. A lot of linear systems appear in most branches of science and engineering such as applied mathematics, biology, chemistry, physics, electrical engineering, mechanics, transportation, buildings, vibrations and so on. When the coefficient matrix of the linear system (1.1) is large and sparse, iterative methods are offered instead of direct methods. In order to solve (1.1) more efficiently by using the iterative methods, usually effective splittings of the coefficient matrix A are necessary. For example, the classical Jacobi and Gauss-Seidel

DOI: 10.21136/AM.2021.0246-20

iterations are gained by splitting the matrix A into its diagonal and off-diagonal parts. Hadjidimos [13] first offered the accelerated overrelaxation (AOR) method, which has been verified to be a potent device for solving the linear system (1.1), and discussed its convergence under the conditions that the coefficient matrix is an irreducible diagonally dominant matrix, L-matrix, or consistently ordered matrix. In certain cases the AOR method has better convergence rate than the Jacobi, JOR, Gauss-Seidel, or SOR methods [13], [1]. Bai and Chi [2] drafted the asymptotically optimal successive overrelaxation method by choosing the optimal factor in a dynamic fashion according to the known information in the current iteration step. Moreover, Meng in [16] introduced a method for determining the optimal parameter of the successive overrelaxation (SOR) method. Wu and Liu [22] presented a new version of the AOR method called the quasi accelerated overrelaxation (QAOR) method, when the coefficient matrix is an irreducible diagonally dominant matrix, H-matrix, symmetric positive definite matrix, or L-matrix. In [3], Bai et al. introduced the Hermitian/skew-Hermitian splitting (HSS) method to solve non-Hermitian positive definite linear systems of equations. Afterwards, this method received much attention and different versions of it were presented. Benzi and Golub in [8] applied the HSS method to solve a saddle point problem and Bai et al. in [5] used it as a preconditioner. The normal/skew-Hermitian splitting (NSS) method has been proposed by Bai et al. [4]; in addition, Bai et al. in [3] offered the positive definite and skew-Hermitian splitting (PSS) method to solve positive definite linear systems of equations. Also, a lopsided version of the HSS (LHSS) method has been introduced by Li et al. in [15]. Moreover, lately Benzi in [7] presented a generalization of the HSS method to solve positive definite linear systems of equations.

Mostly, the methods mentioned are connected with some parameters that usually are determined experimentally or randomly. But to have a faster rate of convergence it is very useful if they are determined optimally. Based on this idea, in the sequel, we build a new iterative method for solving the linear system of equations (1.1); first we split the matrix A into two matrices M and N by using the AOR idea and applying the Taylor approximation. Then we calculate the optimal value of the parameters included in this splitting. We show that this modified method is more stable and effective than the HSS and AOR methods and the method which was presented in [22].

This paper is organized as follows: In Section 2, we present some required preliminary concepts including symbols, definitions, theorems etc.; also, a brief review on the HSS, AOR and QAOR methods and their properties is given. Section 3 is devoted to describing the structure of our new proposed method and proving its convergence under some standard assumptions. Then, we present a new

algorithm for solving the linear system of equations (1.1). Numerical results of applying the new method to some standard test problems taken from the literature are given in Section 4. We end up the paper by some concluding remarks in Section 5.

2. Preliminaries

In what follows, we recall some definitions and results which are utilized in the paper. Throughout this paper, we denote by $\|A\|$ the spectral norm of the matrix A that is defined by $\|A\| = \{\|Ax\|\colon\ x \in \mathbb{R}^n,\ \|x\| = 1\}$ where $\|x\|^2 = x^\top x$. For a square matrix A, the spectral radius of A is denoted by $\varrho(A)$. For a given matrix $A \in \mathbb{R}^{n\times n}$ of the linear equation system (1.1), the decomposition $A = M - N$ is named a splitting if M and N belong to $\mathbb{R}^{n\times n}$ and M is nonsingular. For an arbitrary given splitting $A = M - N$, a basic stationary iterative method for solving $Ax = b$ has the form

$$x^{(k+1)} = Tx^{(k)} + M^{-1}b, \quad k = 0, 1, 2, \ldots, \tag{2.1}$$

where the initial vector $x^{(0)}$ is given and $T = M^{-1}N$ is named the iteration matrix. The (asymptotic) rate of convergence is defined by $R_\infty = -\ln(\varrho)$. The convergence analysis of the iterative method (2.1), based on the spectral radius of the iteration matrix T, is presented in [20]. For large values of k, at each step, the corresponding error reduces in magnitude approximately by the factor of $\varrho(T)$. That is, when the spectral radius is smaller, the convergence is faster.

In this study, we work with special kinds of matrices which are introduced in the following definitions.

Definition 2.1 (Berman and Plemmons [9])**.** The matrix $A \in \mathbb{R}^{n\times n}$ is called a Z-matrix if $a_{ij} \leqslant 0$ for $i, j = 1, 2, 3, \ldots, n$ $(i \neq j)$. A Z-matrix with positive diagonal elements is called an L-matrix.

Definition 2.2 (Berman and Plemmons [9])**.** Let A be an L-matrix. Then the matrix A is said to be an M-matrix if A is nonsingular and $A^{-1} \geqslant 0$.

The next definition expounds different types of splittings employed in this study.

Definition 2.3 (Woznicki [21])**.** The splitting $A = M - N$ is called

(i) a regular splitting of A if $M^{-1} \geqslant 0$ and $N \geqslant 0$,
(ii) a nonnegative splitting of A if $M^{-1} \geqslant 0$, $M^{-1}N \geqslant 0$ and $NM^{-1} \geqslant 0$,
(iii) a weak nonnegative splitting of A if $M^{-1} \geqslant 0$ and either $M^{-1}N \geqslant 0$ (the first type) or $NM^{-1} \geqslant 0$ (the second type),
(iv) a convergent splitting of A if $\varrho(M^{-1}N) < 1$.

Theorem 2.4. *The series*

$$\sum_{k=0}^{\infty} A^k$$

converges if and only if $\varrho(A) < 1$. *Under this condition,* $I - A$ *is nonsingular and the sum of the series is equal to* $(I - A)^{-1}$.

P r o o f. See [18]. □

Theorem 2.5. *Assume that* A *is a* Z*-matrix. Moreover, suppose that* $A = M - N$ *is a weak nonnegative splitting of the first type. Then* $\varrho(M^{-1}N) < 1$ *if and only if* A *is an* M*-matrix.*

P r o o f. See [6]. □

2.1. HSS iterative method. The HSS method is one of the efficient methods for solving positive definite systems of linear equations which has been presented by Bai, Golub and Ng [3], that we are going to introduce briefly here. Suppose $A \in \mathbb{R}^{n\times n}$ is a positive definite matrix. Consider the Hermitian and skew-Hermitian splitting $A = H + S$ where $H = \frac{1}{2}(A + A^\top)$ and $S = \frac{1}{2}(A - A^\top)$. Given an initial guess $x^{(0)} \in \mathbb{R}^n$, compute $x^{(k+1)}$ by the following iterative scheme for $k = 0, 1, 2, \ldots$ until the convergence,

$$\begin{cases} (\alpha I + H)x^{(k+1/2)} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+1/2)} + b, \end{cases} \tag{2.2}$$

where α is a given positive constant. The matrix-vector form of the HSS iteration can be equivalently rewritten as

$$x^{(k+1)} = T(\alpha)x^{(k)} + G(\alpha)b, \quad k = 0, 1, 2, \ldots,$$

where

$$T(\alpha) = (\alpha I + S)^{-1}(\alpha I - H)(\alpha I + H)^{-1}(\alpha I - S),$$

and

$$G(\alpha) = 2\alpha(\alpha I + S)^{-1}(\alpha I + H)^{-1}.$$

Bai et al. in [3] showed that for any positive constant α we have $\varrho(T(\alpha)) < 1$. This proves that the HSS iteration unconditionally converges to the exact solution of linear system of equations for any initial guess $x^{(0)} \in \mathbb{R}^n$.

Since our new method for solving (1.1) is related to the used idea of the AOR method, we give a brief review on this method in the next subsection.

2.2. AOR method and its properties. For solving (1.1), Hadjidimos [13] proposed the splitting $A = M_{\gamma,\omega} - N_{\gamma,\omega}$ with two parameters for the coefficient matrix A where

$$M_{\gamma,\omega} = \frac{1}{\omega}(A_D - \gamma A_L), \quad N_{\gamma,\omega} = \frac{1}{\omega}[(1-\omega)A_D + (\omega-\gamma)A_L + \omega A_U] \tag{2.3}$$

and $\gamma, \omega \neq 0$ are parameters, A_D is the diagonal part of A, and $-A_L$ and $-A_U$ are strictly lower and strictly upper triangular parts of A, respectively. The iteration format of the AOR method for solving the linear system (1.1) is

$$x^{(k+1)} = T^{\mathrm{AOR}}_{\gamma,\omega} x^{(k)} + g_{\gamma,\omega}, \quad k = 0, 1, 2, \ldots, \tag{2.4}$$

where

$$T^{\mathrm{AOR}}_{\gamma,\omega} = (A_D - \gamma A_L)^{-1}[(1-\omega)A_D + (\omega-\gamma)A_L + \omega A_U]; \quad g_{\gamma,\omega} = \omega(A_D - \gamma A_L)^{-1} b.$$

We show the particular values of the parameters γ and ω (see [13]) when the AOR method can be simplified into

- ▷ the Jacobi method if $\omega = 1$, $\gamma = 0$;
- ▷ the simultaneous overrelaxation method if $\gamma = 0$;
- ▷ the Gauss-Seidel method if $\omega = \gamma = 1$;
- ▷ the successive overrelaxation method if $\omega = \gamma$.

In [22], a version of the AOR method is presented under the title QAOR method, which we introduce in the following subsection.

2.3. QAOR method. Consider the linear system of equations (1.1). Wu and Liu proposed the following splitting for the coefficient matrix A:

$$M_1 = (1+\omega)A_D - \gamma A_L, \quad N_1 = A_D + (\omega-\gamma)A_L + \omega A_U. \tag{2.5}$$

Then

$$A = \frac{1}{\omega}(M_1 - N_1). \tag{2.6}$$

Based on the above matrix splitting, the QAOR method is defined as

$$((1+\omega)A_D - \gamma A_L)x^{(k+1)} = [A_D + (\omega-\gamma)A_L + \omega A_U]x^{(k)} + \omega b, \quad k = 0, 1, 2, \ldots, \tag{2.7}$$

and its iteration matrix is

$$\begin{aligned} T^{\mathrm{QAOR}}_{\gamma,\omega} &= ((1+\omega)A_D - \gamma A_L)^{-1}[A_D + (\omega-\gamma)A_L + \omega A_U] \\ &= ((1+\omega)I - \gamma L)^{-1}[I + (\omega-\gamma)L + \omega U], \end{aligned} \tag{2.8}$$

where $L = A_D^{-1}A_L$ and $U = A_D^{-1}A_U$.

Comparing the QAOR method with the AOR method, it is easy to see that the iteration matrix of the QAOR method is similar to that of the AOR method. Based on this fact, the QAOR method may retain all the privileges of the AOR method [22]. Wu and Liu in [22] showed that under some conditions, the QAOR method converges when the coefficient matrix is an irreducible diagonally dominant matrix, H-matrix, symmetric positive definite matrix, or L-matrix; also they demonstrated that under some assumptions, the QAOR method is a simplified AOR method. If $w = \gamma$, the QAOR is reduced to the QSOR method [22].

3. A new iterative method based on the AOR iteration

We know that $(A_D - \gamma A_L)^{-1} = (I - \gamma L)^{-1} A_D^{-1}$, where $L = A_D^{-1} A_L$. Since L is a strictly lower triangular matrix, we have $L^n = 0$ (the zero matrix) and $\varrho(\gamma L) < 1$; also according to Theorem 2.4, $(I - \gamma L)^{-1}$ can be written in the form of the Taylor expansion as

$$(I - \gamma L)^{-1} = \sum_{k=0}^{n-1} (\gamma L)^k.$$

Thus $M_{\gamma,\omega}^{-1}$ can be expressed as

$$M_{\gamma,\omega}^{-1} = \omega (A_D - \gamma A_L)^{-1} = \omega (I - \gamma L)^{-1} A_D^{-1} = \omega \sum_{k=0}^{n-1} (\gamma L)^k A_D^{-1}. \tag{3.1}$$

Obviously, $M_{\gamma,\omega}^{-1}$ can be approximated by a lower-order truncation of the matrix series on the right-hand side of (3.1); generally, $M_{\gamma,\omega}^{-1}$ can be expressed approximately by

$$M_{\gamma,\omega}^{-1} \simeq \omega (I + \alpha\gamma L + \beta^2 \gamma^2 L^2) A_D^{-1}, \tag{3.2}$$

where α and β are two real parameters. Now, by applying this idea, we present a splitting for the matrix A and then we propose a new method for solving the linear system of equations $Ax = b$. Then we prove that our method converges for a certain type of matrices.

Now, the new iterative algorithm is proposed as

$$\begin{aligned} &\frac{1}{\omega} A_D (I + \alpha\gamma L + \beta^2 \gamma^2 L^2)^{-1} x^{(k+1)} \\ &\qquad = \Big(\frac{1}{\omega} A_D (I + \alpha\gamma L + \beta^2 \gamma^2 L^2)^{-1} - A_D + A_L + A_U \Big) x^{(k)} + b, \end{aligned} \tag{3.3}$$

its iteration matrix is

$$T = I + \omega (I + \alpha\gamma L + \beta^2 \gamma^2 L^2)(-I + L + A_D^{-1} A_U), \tag{3.4}$$

where A_D is the diagonal part of A, $-A_L$ and $-A_U$ are strictly lower and strictly upper triangular parts of A, respectively, $L = A_D^{-1}A_L$, and α, γ, ω and β are real parameters with $\omega \neq 0$. Since we split the matrix A into two matrices, using the AOR idea and applying the Taylor approximation, we call this new method as TAOR method. Now we show that when the coefficient matrix A is an M-matrix, the TAOR iterative method is convergent.

Theorem 3.1. *Let $A = A_D - A_L - A_U$ be an M-matrix, where A_D, $-A_L$ and $-A_U$ are defined as above, $0 < \omega \leqslant 1$ and $\beta^2\gamma^2 \leqslant \alpha\gamma \leqslant 1$. Then, the iterative method defined by* (3.3) *converges to the exact solution of the linear system* (1.1).

P r o o f. By Theorem 2.5, it is sufficient to prove that $A = M - N$ is a weak nonnegative splitting of the first type, where $M = \omega^{-1}A_D(I+\alpha\gamma L+\beta^2\gamma^2L^2)^{-1}$ and $N = \omega^{-1}A_D(I+\alpha\gamma L+\beta^2\gamma^2L^2)^{-1} - A_D + A_L + A_U$. To this end, we need to show that $M^{-1} \geqslant 0$ and $M^{-1}N \geqslant 0$. Because A is an M-matrix, therefore $L = A_D^{-1}A_L \geqslant 0$. On the other hand, since $\alpha\gamma \geqslant 0$ we have $\alpha\gamma L \geqslant 0$. Hence $\omega > 0$ implies that $M^{-1} \geqslant 0$. Also, we have

$$\begin{aligned}
T = M^{-1}N &= \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)A_D^{-1} \\
&\quad \times \Big(\frac{1}{\omega}A_D(I + \alpha\gamma L + \beta^2\gamma^2L^2)^{-1} - A_D + A_L + A_U\Big) \\
&= I - \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2) + \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)L \\
&\quad + \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)A_D^{-1}A_U \\
&= I - \omega I - \omega\alpha\gamma L - \omega\beta^2\gamma^2L^2 + \omega L + \omega\alpha\gamma L^2 + \omega\beta^2\gamma^2L^3 \\
&\quad + \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)A_D^{-1}A_U \\
&= (1-\omega)I + \omega(1-\alpha\gamma)L + \omega\gamma(\alpha - \beta^2\gamma)L^2 + \omega\beta^2\gamma^2L^3 \\
&\quad + \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)A_D^{-1}A_U \geqslant 0.
\end{aligned}$$

In this regard, $A = M - N$ is a weak nonnegative splitting of the first type and thus A is an M-matrix. Now, Theorem 2.5 concludes that the iterative method defined by (3.3) is convergent. □

Selecting the parameters for applying (3.3) can be randomly, but to have more suitable results, it is better to choose them optimally. In this regard, and also to have a better approximation in (3.3), and also a fast rate of convergence to reach the best solution of (1.1) and also the least possible time-consumption, especially for large scale problems, we prefer to find the parameters optimally.

Ren et al. presented an optimization technique to find the optimal parameters of the AOR iteration by minimizing the 2-norm of the residual vector [17]. But in general, a small residual does not guarantee that the computed solution is accu-

rate [10], Subsection 2.2. Therefore, minimizing the residual vector is not always the suitable criterion for examining the convergence rate. In addition, the authors have used the Taylor expansion of the matrix M^{-1} in calculations, thus two parameters also are added to the AOR iteration parameters. But during optimization, the two parameters obtained from the Taylor expansion are not optimized and only the real parameters are considered. To achieve better results, we calculate the optimal value of each of the four parameters. So our optimization technique is much more robust than the technique used at [17]. In the following, we present an optimization technique to find the optimal parameters of the proposed method which just needs to minimize the Frobenius norm of the iteration matrix and avoids calculating the 2-norm of the iteration matrix of the TAOR method.

The TAOR iterative method can be rewriten as

$$x^{(k+1)} = Tx^{(k)} + Gb, \quad k = 0, 1, 2, \ldots, \tag{3.5}$$

where

$$T = (1-\omega)I + \omega(1-\alpha\gamma)L + \omega\gamma(\alpha-\beta^2\gamma)L^2 + \omega\beta^2\gamma^2L^3 + \omega(I+\alpha\gamma L+\beta^2\gamma^2L^2)A_D^{-1}A_U$$

and

$$G = \omega(I + \alpha\gamma L + \beta^2\gamma^2L^2)A_D^{-1}b.$$

Let x^* be the exact solution of the linear system (1.1) and $x^{(k)}$ be the kth approximated solution obtained by the iterative method (3.5). We define the error vector as

$$e(k) = x^* - x^{(k)}.$$

Lemma 3.2. *Suppose that $T = M^{-1}N \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are given. Denote by $e(k+1) = x^* - x^{(k+1)}$ the error vector in the kth iteration. Then the inequality*

$$\|e(k+1)\| \leqslant \|T\|\|e(k)\| \tag{3.6}$$

holds for $k = 0, 1, 2, \ldots$

P r o o f. By (3.5) we have

$$e(k+1) = x^* - x^{(k+1)} = T(x^* - x^{(k)}) = Te(k). \tag{3.7}$$

By taking the 2-norm of the sides of the above relationship, the desired result is obtained. □

According to (3.6) the 2-norm of the iteration matrix plays an important role in the rate of convergence; that is, the smaller $\|T\|$, the faster rate of convergence.

R e m a r k 3.3. In order to compute the optimal parameters, the determination of $\|T\|$ is quite expensive and may greatly decrease the efficiency of the TAOR iterative method. So, we use the Frobenius norm of the iteration matrix, which is the upper bound for $\|T\|$.

We know that $\|T\|_F^2 = \mathrm{tr}(T^\top T)$. Put $U = A_D^{-1}A_U$, then from (3.5) we have:

$$\begin{aligned}
T^\top T = {} & (1-\omega)^2 I + \omega(1-\omega)(1-\alpha\gamma)L + \omega\gamma(1-\omega)(\alpha-\beta^2\gamma)L^2 \\
& + \omega\beta^2\gamma^2(1-\omega)L^3 + \omega(1-\omega)(I+\alpha\gamma L+\beta^2\gamma^2L^2)U \\
& + \omega(1-\omega)(1-\alpha\gamma)L^\top + \omega^2(1-\alpha\gamma)^2L^\top L \\
& + \omega^2\gamma(1-\alpha\gamma)(\alpha-\beta^2\gamma)L^\top L^2 + \omega^2\beta^2\gamma^2(1-\alpha\gamma)L^\top L^3 \\
& + \omega^2(1-\alpha\gamma)L^\top(I+\alpha\gamma L+\beta^2\gamma^2L^2)U \\
& + \omega\gamma(1-\omega)(\alpha-\beta^2\gamma)(L^2)^\top + \omega^2\gamma(\alpha-\beta^2\gamma)(1-\alpha\gamma)(L^2)^\top L \\
& + \omega^2\gamma^2(\alpha-\beta^2\gamma)^2(L^2)^\top L^2 + \omega^2\beta^2\gamma^3(\alpha-\beta^2\gamma)(L^2)^\top L^3 \\
& + \omega^2\gamma(\alpha-\beta^2\gamma)(L^2)^\top(I+\alpha\gamma L+\beta^2\gamma^2L^2)U \\
& + \omega\beta^2\gamma^2(1-\omega)(L^3)^\top + \omega^2\beta^2\gamma^2(1-\alpha\gamma)(L^3)^\top L \\
& + \omega^2\beta^2\gamma^3(\alpha-\beta^2\gamma)(L^3)^\top L^2 + \omega^2\beta^4\gamma^4(L^3)^\top L^3 \\
& + \omega^2\beta^2\gamma^2(L^3)^\top(I+\alpha\gamma L+\beta^2\gamma^2L^2)U \\
& + \omega(1-\omega)U^\top(I+\alpha\gamma L^\top+\beta^2\gamma^2(L^2)^\top) \\
& + \omega^2(1-\alpha\gamma)U^\top(I+\alpha\gamma L^\top+\beta^2\gamma^2(L^2)^\top)L \\
& + \omega^2\gamma(\alpha-\beta^2\gamma)U^\top(I+\alpha\gamma L^\top+\beta^2\gamma^2(L^2)^\top)L^2 \\
& + \omega^2\beta^2\gamma^2U^\top(I+\alpha\gamma L^\top+\beta^2\gamma^2(L^2)^\top)L^3 \\
& + \omega^2U^\top(I+\alpha\gamma L^\top+\beta^2\gamma^2(L^2)^\top)(I+\alpha\gamma L+\beta^2\gamma^2L^2)U.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{tr}(T^\top T) = {} & (1-\omega)^2 n + 2\omega(1-\omega)\alpha\gamma c_1 + 2\omega(1-\omega)\beta^2\gamma^2c_2 \\
& + \omega^2(1-\alpha\gamma)^2c_3 + 2\omega^2\gamma(1-\alpha\gamma)(\alpha-\beta^2\gamma)c_4 \\
& + 2\omega^2\beta^2\gamma^2(1-\alpha\gamma)c_5 + 2\omega^2(1-\alpha\gamma)c_6 \\
& + 2\omega^2(1-\alpha\gamma)\alpha\gamma c_7 + 2\omega^2\beta^2\gamma^2(1-\alpha\gamma)c_8 \\
& + \omega^2\gamma^2(\alpha-\beta^2\gamma)^2c_9 + 2\omega^2\beta^2\gamma^3(\alpha-\beta^2\gamma)c_{10} \\
& + 2\omega^2\gamma(\alpha-\beta^2\gamma)c_{11} + 2\alpha\omega^2\gamma^2(\alpha-\beta^2\gamma)c_{12} \\
& + 2\omega^2\beta^2\gamma^3(\alpha-\beta^2\gamma)c_{13} + \omega^2\beta^4\gamma^4c_{14} + 2\omega^2\beta^2\gamma^2c_{15} \\
& + 2\alpha\omega^2\beta^2\gamma^3c_{16} + 2\omega^2\beta^4\gamma^4c_{17} + \omega^2c_{18} + 2\alpha\omega^2\gamma c_{19} \\
& + 2\omega^2\beta^2\gamma^2c_{20} + \alpha^2\omega^2\gamma^2c_{21} + 2\alpha\omega^2\beta^2\gamma^3c_{22} + \omega^2\beta^4\gamma^4c_{23},
\end{aligned}$$

where

$$
\begin{aligned}
(3.8)\quad & c_1 = \operatorname{tr}(LU), && c_2 = \operatorname{tr}(L^2U), && c_3 = \operatorname{tr}(L^\top L),\\
& c_4 = \operatorname{tr}(L^\top L^2), && c_5 = \operatorname{tr}(L^\top L^3), && c_6 = \operatorname{tr}(L^\top U),\\
& c_7 = \operatorname{tr}(L^\top LU), && c_8 = \operatorname{tr}(L^\top L^2U), && c_9 = \operatorname{tr}((L^2)^\top L^2),\\
& c_{10} = \operatorname{tr}((L^2)^\top L^3), && c_{11} = \operatorname{tr}((L^2)^\top U), && c_{12} = \operatorname{tr}((L^2)^\top LU),\\
& c_{13} = \operatorname{tr}((L^2)^\top L^2U), && c_{14} = \operatorname{tr}((L^3)^\top L^3), && c_{15} = \operatorname{tr}((L^3)^\top U),\\
& c_{16} = \operatorname{tr}((L^3)^\top LU), && c_{17} = \operatorname{tr}((L^3)^\top L^2U), && c_{18} = \operatorname{tr}(U^\top U),\\
& c_{19} = \operatorname{tr}(U^\top LU), && c_{20} = \operatorname{tr}(U^\top L^2U), && c_{21} = \operatorname{tr}(U^\top L^\top LU),\\
& c_{22} = \operatorname{tr}(U^\top L^\top L^2U), && c_{23} = \operatorname{tr}(U^\top (L^2)^\top L^2U).
\end{aligned}
$$

Now put $f = \operatorname{tr}(T^\top T)$. Thus we have

$$
\begin{aligned}
\frac{\partial f}{\partial \omega} = {} & -2(1-\omega)n + 2(1-2\omega)\alpha\gamma c_1 + 2(1-2\omega)\beta^2\gamma^2 c_2 + 2\omega(1-\alpha\gamma)^2 c_3\\
& + 4\omega\gamma(1-\alpha\gamma)(\alpha-\beta^2\gamma)c_4 + 4\omega\beta^2\gamma^2(1-\alpha\gamma)c_5 + 4\omega(1-\alpha\gamma)c_6\\
& + 4\alpha\gamma\omega(1-\alpha\gamma)c_7 + 4\omega\beta^2\gamma^2(1-\alpha\gamma)c_8 + 2\omega\gamma^2(\alpha-\beta^2\gamma)^2 c_9\\
& + 4\omega\beta^2\gamma^3(\alpha-\beta^2\gamma)c_{10} + 4\omega\gamma(\alpha-\beta^2\gamma)c_{11} + 4\alpha\omega\gamma^2(\alpha-\beta^2\gamma)c_{12}\\
& + 4\omega\beta^2\gamma^3(\alpha-\beta^2\gamma)c_{13} + 2\omega\beta^4\gamma^4 c_{14} + 4\omega\beta^2\gamma^2 c_{15} + 4\alpha\omega\beta^2\gamma^3 c_{16}\\
& + 4\omega\beta^4\gamma^4 c_{17} + 2\omega c_{18} + 4\alpha\omega\gamma c_{19} + 4\omega\beta^2\gamma^2 c_{20}\\
& + 2\alpha^2\omega\gamma^2 c_{21} + 4\alpha\omega\beta^2\gamma^3 c_{22} + 2\omega\beta^4\gamma^4 c_{23},\\
\frac{\partial f}{\partial \gamma} = {} & 2\alpha\omega(1-\omega)c_1 + 4\omega\gamma\beta^2(1-\omega)c_2 - 2\alpha\omega^2(1-\alpha\gamma)c_3\\
& + 2\omega^2(\alpha - 2\beta^2\gamma - 2\alpha^2\gamma + 3\alpha\gamma^2\beta^2)c_4 + 2\omega^2\beta^2(2\gamma - 3\alpha\gamma^2)c_5\\
& - 2\alpha\omega^2 c_6 + 2\alpha\omega^2(1-2\alpha\gamma)c_7 + 2\omega^2\beta^2(2\gamma - 3\alpha\gamma^2)c_8\\
& + 2\omega^2\gamma(\alpha^2 - 3\alpha\gamma\beta^2 + 2\gamma^2\beta^4)c_9 + 2\omega^2\beta^2\gamma^2(3\alpha - 4\beta^2\gamma)c_{10}\\
& + 2\omega^2(\alpha - 2\beta^2\gamma)c_{11} + 2\alpha\omega^2\gamma(2\alpha - 3\beta^2\gamma)c_{12}\\
& + 2\omega^2\beta^2\gamma^2(3\alpha - 4\beta^2\gamma)c_{13} + 4\omega^2\beta^4\gamma^3 c_{14} + 4\omega^2\beta^2\gamma c_{15}\\
& + 6\alpha\omega^2\beta^2\gamma^2 c_{16} + 8\omega^2\beta^4\gamma^3 c_{17} + 2\alpha\omega^2 c_{19} + 4\omega^2\beta^2\gamma c_{20}\\
& + 2\alpha^2\omega^2\gamma c_{21} + 6\alpha\omega^2\beta^2\gamma^2 c_{22} + 4\omega^2\beta^4\gamma^3 c_{23},\\
\frac{\partial f}{\partial \alpha} = {} & 2\omega\gamma(1-\omega)c_1 - 2\omega^2\gamma(1-\alpha\gamma)c_3 + 2\omega^2\gamma(-2\alpha\gamma + \beta^2\gamma^2 + 1)c_4\\
& - 2\omega^2\beta^2\gamma^3 c_5 - 2\omega^2\gamma c_6 + 2\omega^2\gamma(1-2\alpha\gamma)c_7\\
& - 2\omega^2\beta^2\gamma^3 c_8 + 2\omega^2\gamma^2(\alpha - \beta^2\gamma)c_9 + 2\omega^2\beta^2\gamma^3 c_{10}\\
& + 2\omega^2\gamma c_{11} + 2\omega^2\gamma^2(2\alpha - \beta^2\gamma)c_{12} + 2\omega^2\beta^2\gamma^3 c_{13}\\
& + 2\omega^2\beta^2\gamma^3 c_{16} + 2\omega^2\gamma c_{19} + 2\alpha\omega^2\gamma^2 c_{21} + 2\omega^2\beta^2\gamma^3 c_{22},
\end{aligned}
$$

$$\begin{aligned}\frac{\partial f}{\partial \beta} &= 4\omega\beta\gamma^2(1-\omega)c_2 - 4\omega^2\beta\gamma^2(1-\alpha\gamma)c_4 + 4\omega^2\beta\gamma^2(1-\alpha\gamma)c_5 \\ &\quad + 4\omega^2\beta\gamma^2(1-\alpha\gamma)c_8 - 4\omega^2\beta\gamma^3(\alpha-\beta^2\gamma)c_9 + 4\omega^2\beta\gamma^3(\alpha-2\beta^2\gamma)c_{10} \\ &\quad - 4\omega^2\beta\gamma^2c_{11} - 4\alpha\omega^2\beta\gamma^3c_{12} + 4\omega^2\beta\gamma^3(\alpha-2\beta^2\gamma)c_{13} + 4\omega^2\beta^3\gamma^4c_{14} \\ &\quad + 4\omega^2\beta\gamma^2c_{15} + 4\alpha\omega^2\beta\gamma^3c_{16} + 8\omega^2\beta^3\gamma^4c_{17} \\ &\quad + 4\omega^2\beta\gamma^2c_{20} + 4\alpha\omega^2\beta\gamma^3c_{22} + 4\omega^2\beta^3\gamma^4c_{23}.\end{aligned}$$

In view of the above, the optimal parameters of the new method can be calculated using the following theorem.

Theorem 3.4. *Let $T = M^{-1}N \in \mathbb{R}^{n\times n}$ be the iteration matrix of the TAOR method. Then the optimal parameters of the new method providing* $\arg\min \|T\|_F$ *are given by solving the nonlinear system*

$$\begin{cases} \dfrac{\partial f}{\partial \omega} = 0, \\ \dfrac{\partial f}{\partial \gamma} = 0, \\ \dfrac{\partial f}{\partial \alpha} = 0, \\ \dfrac{\partial f}{\partial \beta} = 0. \end{cases} \tag{3.8}$$

Now, by applying a suitable optimization method, the following algorithm is proposed to solve (1.1) iteratively.

Proposed Algorithm.
Algorithm 3.5.

Step 0. Given an initial vector $x^{(0)} \in \mathbb{R}^n$, a tolerance δ_k. Set $k = 0$.

Step 1. Compute (3.8).

Step 2. Solve (3.9) by the Newton method to obtain γ^*, ω^*, α^* and β^*.

Step 3. Compute $r^{(k)} = b - Ax^{(k)}$.

Step 4. Compute $x^{(k+1)} = x^{(k)} + \omega^*(I + \alpha^*\gamma^*L + (\beta^*)^2(\gamma^*)^2L^2)A_D^{-1}r^{(k)}$.

Step 5. If the stopping condition is satisfied, stop and $x^{(k+1)}$ is the solution.

If not, set $k \leftarrow k+1$ and go to Step 3.

We remind that the stopping condition can be a fixed number of iterations, a residual threshold and so on; we consider δ_k as a residual threshold in the above algorithm.

4. Numerical results

In order to elucidate the credibility of the established results in this paper, we present some numerical results in this part. All computations have been carried out on a computer with an Intel Core i7-4770K CPU @ 3.50GHz processor and 24GB RAM using Matlab R2018b. Further, the right-hand side b in (1.1) was selected such that $b = Ae$ where $e = (1, \dots, 1)^\top$. In all of the following experiments, the initial guess is taken to be the zero vector and the iterations are terminated if the current iteration satisfies either the residual condition

$$\delta_k \equiv \|r^{(k)}\|_2 \leqslant 10^{-6}\|r^{(0)}\|_2, \tag{4.1}$$

or the number of iterations exceeds 20 000 [17]. Here $x^{(k)}$ refers to the kth approximate solution. Notation "Fail" in tables means that the corresponding iterative method was stopped after 20 000 iterations while the computed approximate solution did not satisfy (4.1). In this section, we report some numerical results to compare the performance of the proposed method with the iterative schemes (2.2), (2.4) and (2.7).

Let $H = \frac{1}{2}(A+A^\top)$ and $S = \frac{1}{2}(A-A^\top)$ be the Hermitian and the skew-Hermitian parts of the matrix A. Let also $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and smallest eigenvalues of H, respectively. It was shown in [11] that the optimal parameter value of the HSS iterative method can be calculated by

$$\alpha_{\min} = \arg\min(\varrho(S(\alpha)^{-1}H(\alpha))) = \frac{\lambda_{\max} + \lambda_{\min}}{2},$$

where $S(\alpha) = \alpha I + S$ and $H(\alpha) = H - \alpha I$. It should be noted that in the implementation of the HSS iterative method we considered the optimal value of the parameter α.

We mainly work with three test problems that have previously been examined in literature. Their results are shown in Tables 1–13 in which they report the optimal parameters of the TAOR method, the number of iterations (denoted by Iter), the error value (denoted by Err), the spectral radius of the iteration matrix and the CPU time for the convergence (denoted by CPU). It should be noted that the calculated time is in seconds. Also, the optimal values of parameters for the proposed iterative method ω^*, γ^*, α^* and β^* are presented.

Example 4.1 ([14], [19]). Consider the two-dimensional convection diffusion equation

$$\begin{gathered} -(u_{xx} + u_{yy}) + q(u_x + u_y) + pu = f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) = 0, \quad u(x,y) \in \partial\Omega, \end{gathered}$$

where $\Omega = (0,1)\times(0,1)$, $\partial\Omega$ denotes the boundary of Ω, q is a positive constant and p is a real number. By applying the five-point finite difference scheme to the diffusive terms and the central difference scheme to the convective terms with the equidistant step size $h = 1/(m+1)$, we arrive at the system of linear equations $Bx = d$, where B is a matrix of order $n = m^2$ of the form

$$B = T_x \otimes I_m + I_m \otimes T_y + pI_n,$$

where

$$T_x = \text{tridiag}(-1-r, 4, -1+r) \quad \text{and} \quad T_y = \text{tridiag}(-1-r, 0, -1+r)$$

and $r = (qh)/2$ is the mesh Reynolds number. We point out that for $q = 0$, the problem reduces to the test example in [14]. We present the results for the following two cases:

Case I. Set $q = 0$. Let $p = 10$ and $A = B$.

n	ω^*	γ^*	α^*	β^*
100	0.9998	0.9917	1.0068	0.9914
400	0.9998	0.9944	1.0043	0.9882
900	0.9998	0.9935	1.0052	0.9890
2500	0.9998	0.9928	1.0059	0.9895
10000	0.9998	0.9191	1.0867	1.0688
40000	0.9998	0.9190	1.0868	1.0688
90000	0.9998	0.9190	1.0868	1.0688

Table 1. The calculated optimal parameters of the TAOR method for Case I from Example 4.1.

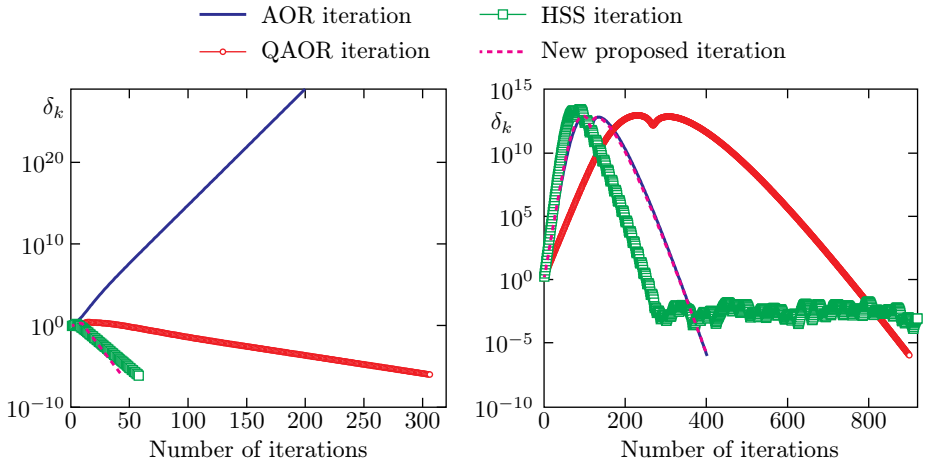
n	ω_{rand}	γ_{rand}
100	0.5033	0.2465
400	0.9559	0.8190
900	0.7035	0.0434
2500	0.0928	0.0587
10000	0.0172	0.0159
40000	0.7939	0.5119
90000	0.6174	0.0547

Table 2. The considered random parameters of the AOR and QAOR methods for Case I from Example 4.1.

Case II. Set $p = -1$. Let $q = 20$, 30, 40, 50 and $A = B + 0.5 * (B_L - B_L^\top)$ where B_L is the strictly lower part of B.

n	Method	AOR	QAOR	HSS	**TAOR**
100	Iter	29	48	8	8
	CPU	0.0013	0.0016	0.0024	0.0011
	Err	$9.078e-07$	$9.620e-07$	$3.494e-07$	$3.642e-07$
400	Iter	10	30	8	8
	CPU	0.0015	0.0024	0.0037	0.0012
	Err	$2.792e-07$	$7.531e-07$	$4.702e-07$	$5.268e-07$
900	Iter	20	40	8	8
	CPU	0.0040	0.0071	0.0065	0.0012
	Err	$6.621e-07$	$7.052e-07$	$5.114e-07$	$5.821e-07$
2500	Iter	199	219	8	8
	CPU	0.3472	0.4096	0.0186	0.0016
	Err	$9.834e-07$	$9.398e-07$	$5.447e-07$	$6.269e-07$
10000	Iter	1113	1132	8	8
	CPU	49.3042	54.4283	0.0817	0.0029
	Err	$9.937e-07$	$9.974e-07$	$5.699e-07$	$6.602e-07$
40000	Iter	15	35	8	8
	CPU	9.0873	29.4522	0.4045	0.0092
	Err	$6.717e-07$	$8.234e-07$	$5.826e-07$	$6.773e-07$
90000	Iter	24	44	8	8
	CPU	58.8077	94.7962	1.2531	0.0208
	Err	$7.369e-07$	$7.526e-07$	$5.868e-07$	$6.830e-07$

Table 3. Numerical results of Case I from Example 4.1 for different values of n.



(a) $m = 10$ (b) $m = 50$

Figure 1. Example 4.1 Case II with $q = 20$.

The first case has been also examined in [19]. In Table 3, we compare the TAOR method with the AOR, QAOR and HSS iterative methods from the points of view of the number of iterations, CPU time and error for this case. As Table 3 states, all of the iterative methods mentioned are convergent. The results reported in Table 3 show that the TAOR method is superior to other iterative methods and, also, that for the larger matrix dimension, this superiority can be seen better.

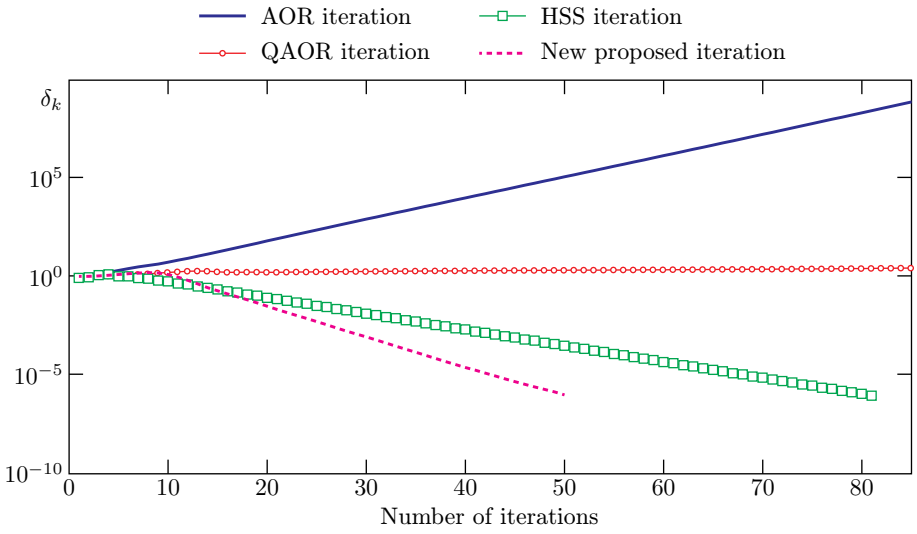
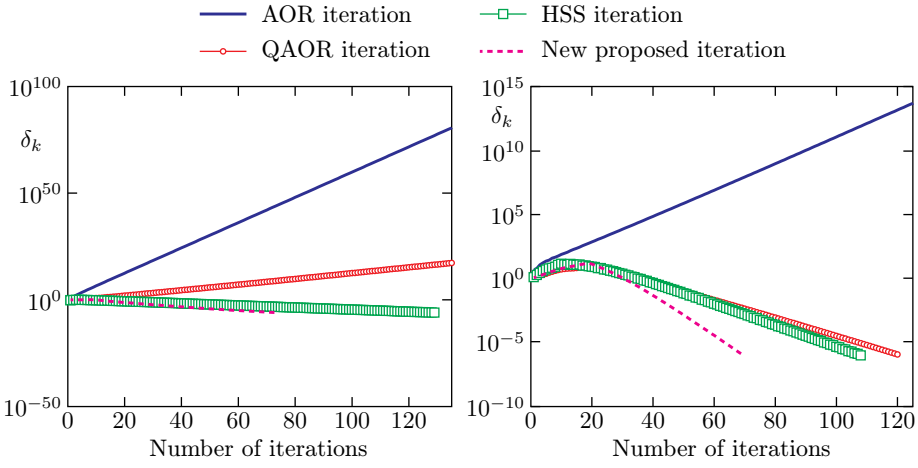


Figure 2. Example 4.1 Case II with $q = 30$ and $m = 10$.



(a) $m = 10$ (b) $m = 20$

Figure 3. Example 4.1 Case II with $q = 40$.

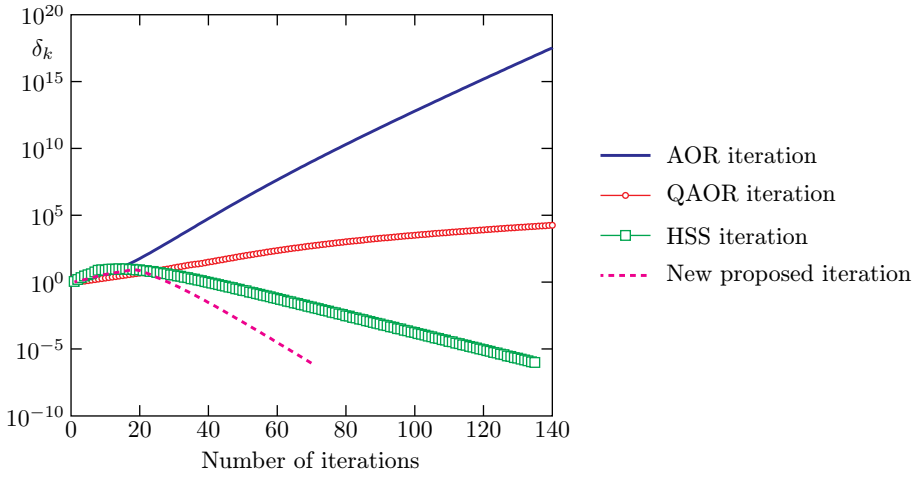


Figure 4. Example 4.1 Case II with $q = 50$ and $m = 20$.

n	Optimal parameter	$q = 20$	$q = 30$	$q = 40$	$q = 50$
100	ω^*	0.4578	0.3156	0.2138	0.1845
	γ^*	0.8253	0.5747	0.5551	0.2022
	α^*	0.5331	0.6894	0.7192	1.7352
	β^*	0.4748	0.5777	0.5188	1.2288
400	ω^*	0.6553	0.5269	0.4231	0.3377
	γ^*	0.8226	0.9562	0.8191	0.7676
	α^*	0.6359	0.4823	0.5177	0.5303
	β^*	0.5543	0.4298	0.4574	0.4487
900	ω^*	0.7357	0.6449	0.5481	0.4831
	γ^*	0.8499	0.8167	1.3253	0.7668
	α^*	0.6741	0.6308	0.3573	0.5723
	β^*	0.5722	0.5507	0.3165	0.5114
2500	ω^*	0.8003	0.7468	0.6914	0.6365
	γ^*	0.8802	0.8166	0.6463	0.8347
	α^*	0.7046	0.7095	0.8383	0.6099
	β^*	0.5838	0.5990	0.7209	0.5330

Table 4. The calculated optimal parameters of the TAOR method for Case II from Example 4.1.

The results of the second case of Example 4.1 are reported in Tables 4–6. As can be seen from Table 6, there are cases where each of the methods (2.2), (2.4) and (2.7) does not converge with respect to the stopping criterion (4.1), while in all these cases

our proposed method is convergent. Furthermore, the TAOR method has the least CPU time among other methods. For further details, we plotted the convergence histories of iterations in Figures 1–4.

n		$q = 20$	$q = 30$	$q = 40$	$q = 50$
100	$(\omega_{\text{rand}}, \gamma_{\text{rand}})$	(0.7867,0.1953)	(0.5296,0.2243)	(0.8425,0.4311)	(0.0334,0.0064)
400	$(\omega_{\text{rand}}, \gamma_{\text{rand}})$	(0.9574,0.2061)	(0.5406,0.1644)	(0.9202,0.5358)	(0.5128,0.1277)
900	$(\omega_{\text{rand}}, \gamma_{\text{rand}})$	(0.4331,0.2129)	(0.2075,0.1970)	(0.7576,0.1954)	(0.4819,0.1509)
2500	$(\omega_{\text{rand}}, \gamma_{\text{rand}})$	(0.8969,0.3820)	(0.0037,0.0016)	(0.3231,0.3136)	(0.2782,0.2663)

Table 5. The considered random parameters of the AOR and QAOR methods for Case II from Example 4.1.

n	q	Method	AOR	QAOR	HSS	**TAOR**
100	20	Iter	Fail	307	58	44
		CPU	-	0.0022	0.0183	0.0011
		Err	-	9.773e − 07	7.266e − 07	6.860e − 07
	30	Iter	Fail	Fail	81	50
		CPU	-	-	0.0245	0.0011
		Err	-	-	8.157e − 07	8.878e − 07
	40	Iter	Fail	Fail	129	74
		CPU	-	-	0.0379	0.0012
		Err	-	-	9.990e − 07	8.604e − 07
	50	Iter	548	563	268	85
		CPU	0.0032	0.0031	0.0775	0.0013
		Err	9.790e − 07	9.756e − 07	9.908e − 07	9.536e − 07
400	20	Iter	155	100	88	49
		CPU	0.0071	0.0043	0.0933	0.0014
		Err	7.508e − 07	7.409e − 07	9.647e − 07	6.251e − 07
	30	Iter	172	132	93	60
		CPU	0.0071	0.0054	0.0978	0.0014
		Err	8.822e − 07	9.891e − 07	7.817e − 07	9.010e − 07
	40	Iter	Fail	120	108	70
		CPU	-	0.0055	0.1157	0.0015
		Err	-	9.985e − 07	8.433e − 07	7.163e − 07
	50	Iter	Fail	Fail	135	70
		CPU	-	-	0.1412	0.0015
		Err	-	-	9.909e − 07	8.611e − 07

Table 6. Numerical results of Case II from Example 4.1 for different values of n and q. (First part.)

n	q	Method	AOR	QAOR	HSS	**TAOR**
900	20	Iter	241	375	140	100
		CPU	0.0439	0.0660	0.3222	0.0025
		Err	9.480e − 07	8.391e − 07	9.836e − 07	6.425e − 07
	30	Iter	299	375	126	69
		CPU	0.0471	0.0646	0.2843	0.0021
		Err	7.111e − 07	7.728e − 07	7.146e − 07	5.581e − 07
	40	Iter	1020	171	127	77
		CPU	0.1560	0.0279	0.2858	0.0021
		Err	9.993e − 07	9.365e − 07	7.818e − 07	8.877e − 07
	50	Iter	315	203	137	90
		CPU	0.0527	0.0342	0.3067	0.0024
		Err	9.447e − 07	9.020e − 07	9.598e − 07	8.654e − 07
2500	20	Iter	402	902	Fail	402
		CPU	0.7996	1.8080	-	0.0169
		Err	8.928e − 07	9.873e − 07	-	9.463e − 07
	30	Iter	Fail	Fail	243	189
		CPU	-	-	1.7284	0.0080
		Err	-	-	6.659e − 07	9.534e − 07
	40	Iter	369	532	209	116
		CPU	0.7345	1.0901	1.5403	0.0058
		Err	8.073e − 07	8.117e − 07	7.902e − 07	4.404e − 07
	50	Iter	318	436	198	109
		CPU	0.6302	0.8661	1.4156	0.0051
		Err	7.425e − 07	9.419e − 07	9.613e − 07	8.539e − 07

Table 6. Numerical results of Case II from Example 4.1 for different values of n and q. (Continuation.)

In the following example, we consider the case that the coefficient matrix A is randomly constructed. This can be helpful for monitoring the behavior of iterations when A is full. We use `rand('state',i)` (with $i = 72$), so that the data can be reproduced in full accuracy.

Example 4.2. In this example, we work with a randomly generated linear problem with $A = \texttt{rand(n,n)} + \texttt{n} * \texttt{eye(n)}$.

The performance of the iterative schemes mentioned is reported in Tables 7–10 for Example 4.2. Here, we observe that all of the examined iterative schemes are convergent, but our proposed iterative scheme has a faster rate of convergence. As the numerical results show, the spectral radius of iteration matrix and CPU time of our proposed method is far less than those of the AOR, QAOR and HSS iterative methods.

n	ω^*	γ^*	α^*	β^*
100	0.9996	3.1000	0.3146	0.3064
400	0.9999	3.6789	0.2657	0.2615
900	1	4.4918	0.2177	0.2139
2500	1	4.5095	0.2168	0.2131

Table 7. The calculated optimal parameters of the TAOR method for Example 4.2.

n	ω_{rand}	γ_{rand}
100	0.3741	0.3014
400	0.6192	0.0752
900	0.7260	0.1739
2500	0.4370	0.1213

Table 8. The considered random parameters of the AOR and QAOR methods for Example 4.2.

n	AOR	QAOR	HSS	**TAOR**
100	0.6358	0.7354	0.1219	0.0925
400	0.3901	0.6235	0.1163	0.0947
900	0.2807	0.5834	0.1146	0.0946
2500	0.5655	0.6976	0.1133	0.0948

Table 9. The spectral radius of the iteration matrix for Example 4.2 for different values of n.

n	Method	AOR	QAOR	HSS	**TAOR**
100	Iter	25	36	6	6
	CPU	0.0012	0.0013	0.0084	0.0011
	Err	9.855e − 07	9.463e − 07	9.296e − 07	8.756e − 07
400	Iter	11	21	6	7
	CPU	0.0017	0.0019	0.0348	0.0016
	Err	7.549e − 07	9.830e − 07	7.240e − 07	1.330e − 07
900	Iter	9	20	6	7
	CPU	0.0080	0.0127	0.0940	0.0072
	Err	5.011e − 07	6.129e − 07	6.692e − 07	1.318e − 07
2500	Iter	19	29	6	7
	CPU	0.1187	0.1741	1.3830	0.0452
	Err	6.964e − 07	7.271e − 07	6.263e − 07	1.343e − 07

Table 10. Numerical results of Example 4.2 for different values of n.

E x a m p l e 4.3 ([12], [14]). Let m be a specified positive integer and $n = m^2$. Consider the system (1.1) with $A = M + \mu I_n \in \mathbb{R}^{n\times n}$, where $M = S \otimes I_m + I_m \otimes S \in \mathbb{R}^{n\times n}$ and $S = \text{tridiag}\{-1, 4, -1\} \in \mathbb{R}^{m\times m}$.

The numerical results for Example 4.3 are presented in Tables 11, 12 and 13. The results in Table 13 are reported to illustrate the behavior of the TAOR method in comparison with the other iterative methods.

n	ω^*	γ^*	α^*	β^*
10000	0.9980	0.9978	0.9905	0.9429
40000	0.9980	0.9978	0.9904	0.9427
90000	0.9980	0.9978	0.9904	0.9426

Table 11. The calculated optimal parameters of the TAOR method for Example 4.3.

n	ω_{rand}	γ_{rand}
10000	0.8147	0.7380
40000	0.1270	0.1160
90000	0.6324	0.0617

Table 12. The considered random parameters of the AOR and QAOR methods for Example 4.3.

n	Method	AOR	QAOR	HSS	**TAOR**
10000	Iter	20	48	13	14
	CPU	0.6582	1.5454	0.1615	0.0042
	Err	9.096e − 07	9.543e − 07	5.860e − 07	3.844e − 07
40000	Iter	204	232	13	14
	CPU	181.8223	207.7043	0.6171	0.0151
	Err	9.901e − 07	9.655e − 07	6.065e − 07	4.004e − 07
90000	Iter	36	64	13	14
	CPU	109.6729	168.8760	1.9214	0.0387
	Err	8.605e − 07	8.766e − 07	6.133e − 07	4.058e − 07

Table 13. Numerical results of Example 4.3 for different values of n.

5. Conclusion

A new iterative method for solving linear systems (TAOR) which includes 4 parameters is presented in this paper. We have shown both theoretically and numerically that by choosing appropriate parameters, the method proposed converges properly. Then we have established how one can optimize the parameters of the TAOR method

to reach a better convergence rate. In this article, we compared numerically our new method with the HSS, QAOR and AOR methods from the points of view of the spectral radius, the error, the number of iterations and the CPU time. As the numerical results show, the spectral radius of the new proposed method is smaller than that of these methods. Since the smaller spectral radius leads to faster convergence, our method converges faster than the methods mentioned. Moreover, the CPU time of the TAOR iterative method is remarkably smaller than the time of the other methods for different dimensions. In this regards, the new proposed method is more powerful and efficient than the AOR, QAOR and HSS methods.

A c k n o w l e d g e m e n t. The authors thank Dr. Behnam Hashemi for his useful comments on an earlier version of this paper.

References

[1] *G. Avdelas, A. Hadjidimos, A. Yeyios*: Some theoretical and computational results concerning the accelerated overrelaxation (AOR) method. Math., Rev. Anal. Numér. Théor. Approximation, Anal. Numér. Théor. Approximation *9* (1980), 5–10. zbl MR

[2] *Z. Bai, X. Chi*: Asymptotically optimal successive overrelaxation methods for systems of linear equations. J. Comput. Math. *21* (2003), 503–612. zbl MR

[3] *Z.-Z. Bai, G. H. Golub, M. K. Ng*: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. *24* (2003), 603–626. zbl MR doi

[4] *Z.-Z. Bai, G. H. Golub, M. K. Ng*: On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. Numer. Linear Algebra Appl. *14* (2007), 319–335. zbl MR doi

[5] *Z.-Z. Bai, G. H. Golub, J. Pan*: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. *98* (2004), 1–32. zbl MR doi

[6] *F. P. A. Beik, N. N. Shams*: Preconditioned generalized mixed-type splitting iterative method for solving weighted least-squares problems. Int. J. Comput. Math. *91* (2014), 944–963. zbl MR doi

[7] *M. Benzi*: A generalization of the Hermitian and skew-Hermitian splitting iteration. SIAM. J. Matrix Anal. Appl. *31* (2009), 360–374. zbl MR doi

[8] *M. Benzi, G. H. Golub*: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. *26* (2004), 20–41. zbl MR doi

[9] *A. Berman, R. J. Plemmons*: Nonnegative Matrices in the Mathematical Sciences. Computer Science and Applied Mathematics. Academic Press, New York, 1979. zbl MR doi

[10] *J. W. Demmel*: Applied Numerical Linear Algebra. SIAM, Philadelphia, 1997. zbl MR doi

[11] *G. H. Golub, D. Vanderstraeten*: On the preconditioning of matrices with skew-symmetric splittings. Numer. Algorithms *25* (2000), 223–239. zbl MR doi

[12] *P. Guo, S.-L. Wu, C.-X. Li*: On the SOR-like iteration method for solving absolute value equations. Appl. Math. Lett. *97* (2019), 107–113. zbl MR doi

[13] *A. Hadjidimos*: Accelerated overrelaxation method. Math. Comput. *32* (1978), 149–157. zbl MR doi

[14] *Y. Ke*: The new iteration algorithm for absolute value equation. Appl. Math. Lett. *99* (2020), Article ID 105990, 7 pages. zbl MR doi

[15] *L. Li, T.-Z. Huang, X.-P. Liu*: Modified Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems. Numer. Linear Algebra Appl. *14* (2007), 217–235. zbl MR doi
[16] *G.-Y. Meng*: A practical asymptotical optimal SOR method. Appl. Math. Comput. *242* (2014), 707–715. zbl MR doi
[17] *L. Ren, F. Ren, R. Wen*: A selected method for the optimal parameters of the AOR iteration. J. Inequal. Appl. *2016* (2016), Article ID 279, 14 pages. zbl MR doi
[18] *Y. Saad*: Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia, 2003. zbl MR doi
[19] *D. K. Salkuyeh*: The Picard-HSS iteration method for absolute value equations. Optim. Lett. *8* (2014), 2191–2202. zbl MR doi
[20] *R. S. Varga*: Matrix Iterative Analysis. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, 1962. zbl MR doi
[21] *Z. I. Woźnicki*: Basic comparison theorems for weak and weaker matrix splitting. Electron. J. Linear Algebra *8* (2001), 53–59. zbl MR doi
[22] *S.-L. Wu, Y.-J. Liu*: A new version of the accelerated overrelaxation iterative method. J. Appl. Math. *2014* (2014), Article ID 725360, 6 pages. zbl MR doi

Authors' address: *Alireza Fakharzadeh Jahromi*, *Nafiseh Nasseri Shams* (corresponding author), Department of OR, Faculty of Mathematics, Shiraz University of Technology, 71557-13876 Shiraz, Iran, e-mail: a_fakharzadeh@sutech.ac.ir, n.naseri@sutech.ac.ir.