AN EFFICIENT CHEBYSHEV SEMI-ITERATIVE METHOD FOR THE SOLUTION OF LARGE SYSTEMS

F. SHARIFFAR1, A. H. refahi SHEIKHANI2, and H. Saberi NAJAFI3

In this paper, we propose a new method for solving large sparse symmetric positive definite linear systems based on a special case of the Richardson iteration process. Our algorithm is easy to implement and computationally attractive. The convergence analysis and error bounds of our method have been proved under suitable restrictions on iteration parameters. Finally, a number of numerical computations are presented based on some particular linear systems.

**Keywords**: Semi-iterative methods, Chebyshev method, Error bounds, Large sparse system, Richardson iteration process.

**MSC2010**: 65F10, 65F08

1. **Introduction**

Consider the following linear system

\[ Ax = b \]  

where \( A \in \mathbb{R}^{n \times n}, b, x \in \mathbb{R}^n \). Such systems often occur in a wide variety of areas, including numerical differential equations [1, 10 and 19], eigenvalue problems [16, 17, and 19], design and computer analysis of circuits [2], and physical models [15, 18]. There are various iterative methods for solving the linear system (1) namely as the SSOR iteration method [9, 14]. Also, as one other resource in this regard, we should name the Chebyshev semi-iterative method [13, 12], which is considered as a nonstationary iterative method. A large family of iterative methods for solving (1) take the splitting form. For any splitting \( A=M-N \), where \( M \) is nonsingular, the iterative method for solving the linear system of (1) is as:

\[ x^{(i+1)} = M^{-1}N x^{(i)} + M^{-1}b, \quad i = 0,1,\ldots,k. \]  

This iterative process converges to the unique solution of system (1) for initial vector \( x^{(0)} \in \mathbb{R}^n \) if and only if \( \rho(M^{-1}N) < 1 \), where \( \rho(A) \) shows the spectral radius of \( A \). There are many iterative methods based on splitting [7, 11]. For

---

1 Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran e-mail: farhadshariffar@yahoo.com
2 Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran e-mail: ah_refahi@liau.ac.ir (Corresponding author)
3 Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran e-mail: hnajafi@guilan.ac.ir
example, suppose $D = \text{diag}(A)$ and $A = D - L - U$, where $L$ and $U$ are the strictly lower and strictly upper triangular part of $A$. Set $N = M - A$, such that for the classical Jacobi iterative method $M = D$, for the Gauss-Seidel $M = D - L$, for the Richardson method $M = \omega I, (\omega \in \mathbb{R})$, and for the SOR method $M = \frac{1}{\omega} (I - \omega L), (\omega \in \mathbb{R})$. In this paper, we focus on such nonstationary one-stage iterative methods. Here, we use $\| \cdot \|_2$ to denote the Euclidean norm and define $Tr(A) = \sum_{i=1}^{n} a_{ii}$. Let $\| \cdot \|_#$ be an arbitrary vector norm on $\mathbb{R}^n$. For a nonsingular matrix $A \in \mathbb{R}^{n \times n}$, $\| x \|_{A, #} = \| Ax \|_#$ defines a vector norm on $\mathbb{R}^n$ for all $x \in \mathbb{R}^n$ [6].

This paper is organized as follows. In Section 2, we recall a special case of the Richardson Iteration Method and its convergence analysis. Section 3 contains our new method based on the Chebyshev Semi-iterative Process. The convergence analysis and error bounds of our method will be presented in this Section. In Section 4, we examine the advantages of our method by carrying out a number of numerical computations. Finally, the conclusions are presented in Section 5.

2. A special case of the Richardson iteration method

Let $M = \frac{\text{Tr}(A)}{\omega} I$ where $I$ is the identity matrix and $\omega \in \mathbb{R}$. Then the Richardson iteration method for the solution of the system (1) is as follows:

Algorithm 1.

Step 1. Choose an initial vector $x^0 \in \mathbb{R}^n$ and a parameter $\omega$.

Step 2. For $i = 0, 1, 2, \ldots$ do

$$x^{(i+1)} = \left( I - \frac{\omega}{\text{Tr}(A)} A \right) x^{(i)} + \frac{\omega}{\text{Tr}(A)} b,$$

(3)

Step 3. If $\| b - Ax^{(i+1)} \| < \text{tol}$, then stop; otherwise, set $i = i + 1$, and go to Step 2.

We call this algorithm the Trace Iterative Method (TIM). In the following Theorem we perform the convergence analysis of the TIM.

Theorem 2.1. Let $A$ be a symmetric positive definite matrix with eigenvalues $\mu_i, (i = 1, \ldots, n)$ and $0 < \omega < \left( \frac{\sqrt{\mu}}{\bar{\mu}} \right) \text{Tr}(A)$, $\bar{\mu} = \max(\mu_i)$. Then Algorithm 1
converges to the solution of the system (1) for any choice of initial vector $x^{(0)}$.

**Proof.** Let $\mu_i$ ($i = 1, 2, \ldots, n$) be the eigenvalues of $A$. Since $A$ is SPD, then all eigenvalues of $A$ are positive and $\frac{\mu_i}{\overline{\mu}} < 1$. Moreover, we know that $\sum_{i=1}^{n} \mu_i = Tr(A)$, and $1 - \frac{\omega \mu_i}{Tr(A)}$ ($i = 1, \ldots, n$) are the eigenvalues of $I - \frac{\omega A}{Tr(A)}$. On the other hand, from $\omega \in \left[0, \frac{2Tr(A)}{\overline{\mu}}\right]$, we have $0 < \frac{\omega \mu_i}{Tr(A)} < \frac{2\mu_i}{\overline{\mu}}$, so $-1 < 1 - \frac{2\mu_i}{\overline{\mu}} < 1 - \frac{\omega \mu_i}{Tr(A)} < 1$, or $\left|1 - \frac{\omega \mu_i}{Tr(A)}\right| < 1$. Therefore, $\rho \left(1 - \frac{\omega A}{Tr(A)}\right) < 1$, and the proof is completed.

**3. The Chebyshev-TIM method**

In this section, first we discuss the Chebyshev semi-iterative method [3, 8]. Then we will describe our new iterative method on the basis of combining the TIM algorithm and the Chebyshev semi-iterative method. Also, we will discuss the convergence theorem and the error bound analysis. Given the results produced by the iterative formula in (2) be as $x^{(0)}, \ldots, x^{(k)}$, and let $e^{(k)}$ be the error vector at the $k$th iteration; then we have:

$$e^{(k)} = x^{(k)} - x = M^{-1}N\left(x^{(k-1)} - x\right) = \ldots = \left(M^{-1}N\right)^{k} e^{(0)}.$$  

(4)

We would like to obtain a better result from their linear combinations, so we have

$$y^{(k)} = \sum_{j=0}^{k} v_{j,k} x^{(j)},$$  

(5)

in which $v_{j,k}$ are the blending coefficients to be determined. If the results are good already: $x^{(0)} = x^{(1)} = \ldots = x^{(k)}$, we must have $y^{(k)} = x$. So $\sum_{j=0}^{k} v_{j,k} = 1$. The question is how to reduce the error of $y^{(k)}$. Using (4) and (5), we have

$$y^{(k)} - x = \sum_{j=0}^{k} v_{j,k} \left(x^{(j)} - x\right) = \sum_{j=0}^{k} v_{j,k} \left(M^{-1}N\right)^{j} e^{(0)} = p_k \left(M^{-1}N\right)^{k} e^{(0)},$$  

(6)
such that \( p_k(x) = \sum_{j=0}^{k} v_{j,k} x^{(j)} \) is a polynomial function. So to reduce the error, we must reduce \( \|p_k(M^{-1}N)\|_2 = \max_{\lambda_i} |p_k(\lambda_i)| \) in which \( \lambda_i \) can be any eigenvalue of \( M^{-1}N \). Suppose that all of the eigenvalues are real. If we know all of the eigenvalues, and if \( k \) is sufficiently large, we can construct the polynomial function in a way that \( p_k(\lambda_i) = 0 \), for any \( \lambda_i \). Unfortunately, it is difficult to know the eigenvalues when the linear system is large and varying. Instead, if we know the spectral radius \( \rho \) such that \(-1 < -\rho \leq \lambda_n \leq \ldots \leq \lambda_1 \leq \rho < 1\), we let \( p_k(x) = \arg\min \{ \max |p_k(x)| : -\rho \leq x \leq \rho \} \). The unique solution of (6) is given by \( p_k(x) = \frac{C_k(\rho^{-1}x)}{C_k(\rho^{-1})} \) in which \( C_k(x) \) is the Chebyshev polynomial with the recurrence relation \( C_{k+1}(x) = 2xC_k(x) - C_{k-1}(x) \) with \( C_0(x) = 1 \) and \( C_1(x) = x \).

It is trivial to see that \( p_k(1) = 1 \), satisfying \( \sum_{j=0}^{k} v_{j,k} = 1 \). For any \( x \in [-1,1] \), \( |C_k(x)| \leq 1 \) but for any \( x \not\in [-1,1] \), \( |C_k(x)| \) grows rapidly when \( k \to \infty \). So \( p_k(x) \) diminishes quickly for any \( x \in [-\rho, \rho] \) when \( k \to \infty \). To reduce the computational and memory cost, we can avoid calculating \( y^{(k)} \) by its definition in (4). Instead, we use (6) to formulate the recurrence relation of \( p_k(x) \) as:

\[
p_{k+1}(x)C_{k+1}(\rho^{-1}) = 2\rho^{-1}xp_k(x)C_k(\rho^{-1}) - p_{k-1}(x)\left[2\rho^{-1}C_k(\rho^{-1}) - C_{k+1}(\rho^{-1})\right],
\]

which can be reorganized into:

\[
C_{k+1}(\rho^{-1})(p_{k+1}(x) - p_{k-1}(x)) = 2\rho^{-1}C_k(\rho^{-1})(xp_k(x) - p_{k-1}(x)).
\]

After replacing \( x \) by \( M^{-1}N \) and multiplying both sides of (8) by \( e^{(0)} \), we get:

\[
C_{k+1}(\rho^{-1})(y^{(k+1)} - y^{(k-1)}) = 2\rho^{-1}C_k(\rho^{-1})\left(M^{-1}N\left(y^{(k)} - x\right) - y^{(k-1)} + x\right).
\]

Using the fact that \(-M^{-1}Nx + x = M^{-1}(M - N)x = M^{-1}b\), we can obtain the following update function called the Chebyshev semi-iterative method:
An efficient Chebyshev semi-iterative version

\[ y^{(i+1)} = \omega_{i+1} \left( M^{-1}N y^{(i)} + M^{-1}b - y^{(i-1)} \right) + y^{(i-1)}, \quad \text{for } i = 1, 2, 3, \ldots, \]

\[ \omega_{k+1} = \frac{2\sigma C_k(\sigma)}{C_{k+1}(\sigma)}, \sigma = \rho \left( M^{-1}N \right)^{-1}, k \geq 1, \omega_1 = 1, y^{(0)} \in \mathbb{R}^n, y^{(1)} = M^{-1}Ny^{(0)} + M^{-1}b. \]

Now we can present our new iterative method on the basis of combining the TIM algorithm and the Chebyshev semi-iterative method. By applying (3), the Chebyshev–TIM method is as follows:

\[ y^{(i+1)} = \omega_{i+1} \left( I - \frac{\omega}{\text{Tr}(A)} A \right) y^{(i)} + \frac{\omega}{\text{Tr}(A)} b - y^{(i-1)} + y^{(i)}. \]

**Algorithm 2.**

**Step 1.** Choose an initial vector \( y^{(0)} \in \mathbb{R}^n \) and a parameter \( \omega \), and splitting \( M - N \).

**Step 2.** Set \( y^{(1)} = M^{-1}Ny^{(0)} + M^{-1}b \) and \( \sigma = \rho \left( M^{-1}N \right)^{-1} \).

**Step 3.** For \( i = 1, 2, 3, \ldots \) do

\[ \omega_{i+1} = \frac{2\sigma C_i(\sigma)}{C_{i+1}(\sigma)}, \quad y^{(i+1)} = \omega_{i+1} \left( I - \frac{\omega}{\text{Tr}(A)} A \right) y^{(i)} + \frac{\omega}{\text{Tr}(A)} b - y^{(i-1)} + y^{(i)} \]

**Step 4.** If \( \|b - Ay^{(i+1)}\| \leq \text{tol} \), then stop; otherwise, set \( i = i + 1 \), and go to Step 3.

We call Algorithm 2 the **Chebyshev-Trace Iterative Method** (Chebyshev-TIM). However, since the spectral radius of the iterative matrix is not known in advance, \( \rho \left( M^{-1}N \right) \) is usually replaced by the lower and upper bounds [3], that is,

\[ y^{(i+1)} = \omega_{i+1} \left( \gamma z^{(i)} + y^{(i)} - y^{(i-1)} \right) + y^{(i-1)}, \]  

where \( \lambda \) is the eigenvalue of \( M^{-1}N \), and

\[ M^2 = b - Ay^{(i)}, y^{(0)} \in \mathbb{R}^n, y^{(1)} = M^{-1}Ny^{(0)} + M^{-1}b, \]

\[ \omega_{i+1} = 2\nu \frac{C_i(\nu)}{C_{i+1}(\nu)}, \gamma = \frac{2}{2-(\beta+\alpha)}, \nu = \frac{2-(\beta+\alpha)}{(\beta-\alpha)}, -1 \leq \alpha \leq \lambda \leq \beta \leq 1, \beta > \alpha. \]

By substituting the above equations in (5), we have:

\[ y^{(i+1)} = \frac{\rho_{i+1}}{2-(\alpha+\beta)} \left( 2M^{-1}N - (\alpha + \beta) I \right) y^{(i)} + 2M^{-1}b \right) + (1-\rho_{i+1}) y^{(i-1)}, \]  

\[
\rho_1 = 1, \rho_2 = 2\nu^2 \left( 2\nu^2 - 1 \right), \text{for } n \geq 2; \rho_{i+1} = \left( 1 - \rho_k / 4\nu^2 \right)^{-1}. \]

Moreover, from (2) we set \( \varepsilon^{(i)} = x^{(i)} - x \), and from (13) let \( \xi^{(i)} = y^{(i)} - x \), where
Therefore, we have the following modified algorithm (See [3, 12]).

**Algorithm 3.**

Step 1. Choose an initial vector \( y^{(0)} \in \mathbb{R}^n \) and parameter \( \omega \).

Step 2. Set:

\[
\begin{align*}
\mathbf{y}_I &= A \mathbf{y} + b \mathbf{b} \mathbf{A} \mathbf{T} \mathbf{A} \mathbf{T} \\
\mathbf{y}^{(0)} &= \left( I - \frac{\omega}{\text{Tr}(A)} A \right) \mathbf{y}^{(0)} + \frac{\omega}{\text{Tr}(A)} b.
\end{align*}
\]

Step 3. For \( i = 1, 2, \ldots \), do

\[
y^{(i+1)} = \frac{\rho_i + 1}{2 - (\alpha + \beta)} \left( \left( 2M^{-1}N - (\alpha + \beta)I \right) y^{(i)} + 2M^{-1} b \right) + (1 - \rho_i) y^{(i-1)}.
\]

Step 4. If \( \| b - A y^{(i+1)} \| \leq \text{tol} \), then stop; otherwise, set \( i = i + 1 \), and go to Step 3.

**Theorem 3.1.** Let \( T \) be the iteration matrix of the TIM, and \( P_1(T) = \frac{2T - \alpha - \beta}{2 - \alpha - \beta} \).

If \( \rho(P_1(T)) < 1 \), then the Chebyshev–TIM method will be convergent.

**Proof.** Set \( q = 2 - (\alpha + \beta) \), from (13) we have:

\[
y^{(i)} = q^{-1} \rho_i \left( 2M^{-1}N - (\alpha + \beta)I \right) y^{(i)} + 2M^{-1} b \right) + (1 - \rho_i) y^{(i-1)}, i \geq 1. \tag{16}
\]

Introducing the notations:

\[
d_i = 1 - \rho_i, G_i = q^{-1} \rho_i \left( 2M^{-1}N - (\alpha + \beta)I \right), h_i = q^{-1} \rho_i \left[ 0 \quad 2M^{-1} b \right]^T \text{ and}
\]

\[
u^{(i)} = \begin{bmatrix} y^{(i-1)} \\ y^{(i)} \end{bmatrix}^T, G_i^* = \begin{bmatrix} 0 & I \\ -d_i I & G_i \end{bmatrix},
\]

then equation (13) becomes \( u^{(i)} = G_i^* u^{(i)} + h_i \); since this is a non-stationary first degree iterative method, we can rewrite it into \( u^{(i)} = \phi_i u^{(0)} + \tau_i \), where \( \phi_i = \prod_{r=1}^{i} G_r^* \) and \( \tau_i = h_i + G_i^* h_{i-1} + \ldots + G_i^* G_{i-1} \ldots G_2 h_1 \). Since \( \rho_1 = 1 \), then \( d_i G_i = G_i - G_i \).

Therefore, by choosing \( V_i = d_i \begin{bmatrix} 0 & 0 \\ -I & G_i \end{bmatrix} \), we get \( G_i^* = G_i^* + V_i \). Also, by the Ostrowsky theorem [9, pp. 141], there exists a matrix \( M \) such that:
An efficient Chebyshev semi-iterative version

\[ \rho \left( R \left( G_1^* \right) \right) \leq MR \left( G_1^* \right) M^{-1} \leq \rho \left( R \left( G_1^* \right) \right) + \varepsilon, \]

and a constant \( \mu_1 \) where \( \mu_1 = \varepsilon / \left( 2M_{\infty}M_{-1, \infty} \right) \). It is obvious that the parameter \( \rho_i \) appearing in (16) satisfies \( 1 \leq \rho_i < 2 \), so \( d_i < 1 \). Now, by choosing \( \|G_1\|_{\infty} = L \), we get \( \|V_i\|_{\infty} = d_i(1 + L) \). On the other hand, because \( \rho_i \rightarrow 1 \), \( d_i \rightarrow 1 \) when \( i \rightarrow \infty \); then, there exists \( m \in \mathbb{R} \) such that \( \|V_n\|_{\infty} \leq \mu_1, (n > m) \).

Introducing a convenient index \( k \), we can write \( \|V_k\|_{\infty} \leq \mu_1, k = 1, 2, \ldots \). By hypothesis, \( \rho \left( R \left( G_1 \right) \right) = \rho \left( G_1 \right) < 1 - \varepsilon, \varepsilon > 0 \). Hence, \( \rho \left( G_1^* \right) + \varepsilon < 1 \). So

\[ \phi_i = \prod_{r=1}^{i} G_r^* \rightarrow 0 \text{ when } i \rightarrow \infty, \] and the proof is completed. ■

Next, we determine the error bounds for the \( \|z^{(i)}\|_{A, \#} \) of the error vector at the \( i \)th iteration of the Chebyshev-TIM method in terms of \( \rho \left( M^{-1}N \right) \).

**Theorem 3.2.** Let \( \xi^{(i)} \) be the error vector at the \( i \)th iteration of the Chebyshev-TIM method, \( \rho \) be the spectral radius of iterative matrix in (3), and due to this, the conditions of Theorem 2.1 are satisfied. Then we have:

\[ \frac{\|\xi^{(i)}\|_{A, 2}}{\|\xi^{(0)}\|_{A, 2}} \leq \frac{2(\beta - \alpha)^i}{\left( \sqrt{1 - \alpha} + \sqrt{1 - \beta} \right)^{2i} + \left( \sqrt{1 - \alpha} - \sqrt{1 - \beta} \right)^{2i}}. \]

**Proof.** From the TIM scheme, we know that \( T = M^{-1}N = I - \frac{\omega}{Tr \left( A \right)} A \).

According to the Theorem 2.1, for \( \omega \in \left( 0, \frac{2Tr \left( A \right)}{\rho} \right) \), all eigenvalues \( \lambda_i \) of \( T \) lie in \(-1 \leq \alpha \leq \lambda_i \leq \beta \leq 1 \). In addition,

\[ \frac{2T - (\beta + \alpha)}{(\beta - \alpha)} = \frac{2T - (\beta + \alpha)}{(\beta - \alpha)} I - \frac{2\omega}{(\beta - \alpha)Tr \left( A \right)} A = v I - \frac{2}{(\beta - \alpha)} A \frac{\omega A}{Tr \left( A \right)} \] (17)

\[ \xi = C_i \left( v \right), \text{ and since } \xi^{(0)} = \varepsilon^{(0)}, \text{ from (14) we get} \]

\[ \xi^{(i)} = \frac{1}{C_i \left( v \right)} C_i \left( \frac{2T - (\alpha + \beta)}{(\beta - \alpha)} I \right) \xi^{(0)} = \frac{1}{C_i} \left( v I - \frac{2}{(\beta - \alpha)} A \frac{\omega A}{Tr \left( A \right)} \right) \xi^{(0)} \] (18)
Since \( \frac{\omega}{\text{Tr}(A)} > 0 \), then \( A' = \frac{\omega}{\text{Tr}(A)} A \) is also SPD, and we can obtain \( A' = UDU^T \).

where \( UU^T = U^TU, D = \text{diag}(\mu_i') \) and \( \mu_i' \) \( (i = 1, \ldots, n) \) are the eigenvalues of \( A' \).

Multiplying the both sides of (18) by \( U^TA' \), we have:

\[
U^T A' \hat{\xi}(i) = U^T A' \left( \frac{vI - \left( \frac{2}{(\beta - \alpha)} A' \right)}{c} \right) \xi(0) = \left( \frac{vI - \left( \frac{2}{(\beta - \alpha)} D \right)}{c} \right) U^T A' \xi(0). \tag{19}
\]

By choosing \( \hat{\xi}(i) = U^T A' \xi(i) = DU^T \xi(i), i = 0, 1, \ldots, n \), we have:

\[
\hat{\xi}(i) = \left( \frac{vI - \left( \frac{2}{(\beta - \alpha)} D \right)}{c} \right) \xi(0), \tag{20}
\]

so we obtain:

\[
\left\| \hat{\xi}(i) \right\|_{A,2}^2 = \left\| U^T A' \hat{\xi}(i) \right\|_{l_2}^2 = \left\| \hat{\xi}(i) \right\|_{l_2}^2 = \frac{1}{c^2} \sum_{i=1}^{n} \left( C_i \left( v - \left( \frac{2}{(\beta - \alpha)} \right) \mu_i' \right) \right)^2 \left( \hat{\xi}(0) \right)^2,
\]

where \( \hat{\xi}(0) \) is the \( l \)th component of the vector \( \hat{\xi}(0) \) defined in (20). Assume that \( \phi_l = v - \frac{2}{(\beta - \alpha)} \mu_i' \), since \( \lambda_i = 1 - \mu_i' \), then we have \( \phi_l = \frac{2}{\beta - \alpha} \lambda_i - \frac{\beta + \alpha}{\beta - \alpha} \). From Theorem 2.1 we have \(-1 \leq -\frac{2}{\beta - \alpha} \lambda_i - \frac{\beta + \alpha}{\beta - \alpha} \leq 1 \), so \(-1 \leq \phi_i \leq 1 \), and by (11) we have:

\[
\left\| \hat{\xi}(i) \right\|_{A,2}^2 = \frac{1}{c^2} \sum_{i=1}^{n} \left( C_i(\phi_i) \right)^2 \left( \hat{\xi}(0) \right)^2 \leq \frac{1}{c^2} \max_{i} \left( C_i(\phi_i) \right)^2 \sum_{l=1}^{n} \left( \hat{\xi}(0) \right)^2. \tag{21}
\]

Since \( \max_{-1 \leq \phi_i \leq 1} \left( C_i(\phi_i) \right) = 1 \), we have:

\[
\left\| \hat{\xi}(i) \right\|_{A,2}^2 \leq \frac{1}{c^2} \Rightarrow \left\| \hat{\xi}(i) \right\|_{A,2} \leq \frac{1}{c} \Rightarrow \left\| \hat{\xi}(0) \right\|_{A,2} \leq C_i \left( \frac{2 - (\beta + \alpha)}{\beta - \alpha} \right). \tag{22}
\]
It can be shown by mathematical induction that
\[
C_i \left( \frac{2-(\beta+\alpha)}{\beta-\alpha} \right) = \frac{1}{2(\beta-\alpha)^i} \left( \left( \sqrt{1-\alpha} + \sqrt{1-\beta} \right)^{2i} + \left( \sqrt{1-\alpha} - \sqrt{1-\beta} \right)^{2i} \right);
\]
for more details see [8]. Hence, from (22) and the above equality, the inequality of theorem 3.2 is hold.

This theorem shows that if we use the Chebyshev –TIM method for solving the SPD linear systems, then this scheme can be terminated after the \(i\)th iteration, where \(i\) satisfies the inequality in the Theorem 3.2.

4. Numerical Experiments

In this Section, we present a number of numerical experiments to illustrate the results obtained in the previous Sections. The initial guess was always zero vector, and the right hand side was selected such that the exact solution of the augmented system (1) is \((1, 2, \ldots, n)^T \in \mathbb{R}^n\). The stopping criterion, when the current iteration satisfies, is \(tol = 10^{-6}\). The number of iterations and CPU time are denoted by the \texttt{Iter}\ and the \texttt{CPU}, respectively.

Example 4.1. Consider the following three-dimensional convection-diffusion equation
\[
\begin{aligned}
-\left( u_{xx} + u_{yy} + u_{zz} \right) + 2u_x + u_y + u_z &= f(x, y, z),
\end{aligned}
\]
on the unit cube domain \(\Omega=[0,1] \times [0,1] \times [0,1]\) with Dirichlet boundary conditions. When the seven-point finite difference discretization (e.g., the centered differences approximations to the diffusive terms and the convective terms) are applied into the above model of convection-diffusion equation, we get the system of linear equations with the coefficient matrix
\[
A = T \otimes I \otimes I + I \otimes T \otimes I + I \otimes I \otimes T,
\]
where \(\otimes\) denotes the Kronecker product, and the equidistant step-size \(h=1/n+1\) is used in the discretization on all of the three directions, and the natural lexicographic ordering is employed to the unknowns. In addition
\[
T_x = \text{tridiagonal} \left[ -\left( \frac{2+2h}{12} \right), 1, -\left( \frac{2-2h}{12} \right) \right], T_y = T_z = \text{tridiagonal} \left[ -\left( \frac{2+2h}{12} \right), 1, -\left( \frac{2-2h}{12} \right) \right].
\]
For details, see [4]. Then, we solve the \(n^3 \times n^3\) matrix yielded by the TIM scheme and the Chebyshev–TIM method. In Table 1, we report the CPU time and the number of iterations for the corresponding TIM scheme and Chebyshev–TIM method with different parameters.

Example 4.2. (The Poisson’s equation) The Poisson’s matrix is the block tridiagonal matrix of order \(n^2\) resulting from the discretizing the Poisson's equation with the 5-point operator on an n-by-n mesh [5]. To produce a Poisson
matrix of dimension \( n \), one may use the MATLAB command
\( A = \text{gallery}(\text{'poisson'},n) \). In Tables 2-5,

\[
\begin{array}{|c|c|c|c|c|}
\hline
n^2 & w & \text{TIM} & \text{Chebyshev –TIM} \\
\hline
125 & w \in (0,134.6041), w = 120 & 88 & 29 & 0.016167 & 29 & 0.008668 \\
343 & w \in (0,357.4503), w = 340 & 150 & 38 & 0.127688 & 38 & 0.062618 \\
729 & w \in (0,748.4160), w = 740 & 228 & 59 & 0.697578 & 59 & 0.216027 \\
\hline
\end{array}
\]

we report the CPU time and the number of iterations for different iterative methods with different parameters. We can see that the SOR method performs much better than the other existing methods. Moreover, for \( \omega \in [90,110] \) and \( n^2 = 100 \), the results of using the Chebyshev–TIM method with 500 iterations, based on the Gauss-Seidel splitting, are shown in Figure 1. From Figure 1, we can find that the optimal value of \( \omega \) lies in (99,101). Also, Figure 2 shows that the optimal value of \( \omega \) lies in (100.18,100.22); therefore, by setting \( \omega = 100.2 \), we have a fast convergence.

**Example 4.3.** In this example we consider system (1) with matrix \( A \) as a Toeplitz matrix. Such a system is called the Toeplitz linear system [7]. Suppose that \( A \) is a large sparse non-Hermitian positive definite Toeplitz matrix as

\[
A = \begin{pmatrix}
a_1 & a_2 & a_3 & \cdots & a_n \\
a_{-2} & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{-3} & \ddots & \ddots & \ddots & a_3 \\
a_{-n} & \cdots & a_{-3} & a_{-2} & a_1
\end{pmatrix}
\]

One of the latest iterative methods for solving such a system is the HSS method. This method uses the splitting \( (\beta I + H)y = (\beta I - S)x^{(k)} + b \), \( (\beta I + S)x^{(k+1)} = (\beta I - H)y + b \) where \( \beta \) is a given positive constant.
Table 2

Results of the example 4.2 for the Jacobi method

<table>
<thead>
<tr>
<th>$N = n^2$</th>
<th>Iter</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>34</td>
<td>0.002987</td>
</tr>
<tr>
<td>100</td>
<td>85</td>
<td>0.003971</td>
</tr>
<tr>
<td>1024</td>
<td>297</td>
<td>0.046571</td>
</tr>
</tbody>
</table>

Table 3

Results of the example 4.2 for the Gauss Seidel and SOR methods

<table>
<thead>
<tr>
<th>$N = n^2$</th>
<th>$\omega \in (0,2)$</th>
<th>Iter</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.0 (Gauss-Seidel)</td>
<td>137</td>
<td>0.010002</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>44</td>
<td>0.003181</td>
</tr>
<tr>
<td></td>
<td><strong>1.5604 (opt)</strong></td>
<td>32</td>
<td><strong>0.002441</strong></td>
</tr>
<tr>
<td>1024</td>
<td>1.9</td>
<td>133</td>
<td>0.124887</td>
</tr>
<tr>
<td></td>
<td>1.0 (Gauss-Seidel)</td>
<td>975</td>
<td>3.279529</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>118</td>
<td>0.392415</td>
</tr>
<tr>
<td></td>
<td><strong>1.8264 (opt)</strong></td>
<td><strong>88</strong></td>
<td><strong>0.307156</strong></td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>134</td>
<td>0.521695</td>
</tr>
</tbody>
</table>

Table 4

Results of the example 4.2 for the TIM method

<table>
<thead>
<tr>
<th>$N = n^2$</th>
<th>$\omega$</th>
<th>Iter</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>100, $\omega \in (0,0.102.0672)$</td>
<td>97</td>
<td>298</td>
<td>0.021206</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>292</td>
<td>0.020938</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>286</td>
<td>0.020428</td>
</tr>
<tr>
<td>102</td>
<td><strong>102</strong></td>
<td><strong>283</strong></td>
<td><strong>0.020291</strong></td>
</tr>
<tr>
<td>1024, $\omega \in (0,0.1026.3)$</td>
<td>1022</td>
<td>2288</td>
<td>11.034147</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>2284</td>
<td>10.323143</td>
</tr>
<tr>
<td></td>
<td><strong>1026</strong></td>
<td><strong>2280</strong></td>
<td><strong>10.043827</strong></td>
</tr>
</tbody>
</table>

Fig. 1. Generated errors in example 4.2 for $\omega \in [90,110]$.

Fig. 2. Generated errors in example 4.2 for $\omega \in [99,101]$.
In this example, we consider a Toeplitz linear system with symmetric positive definite Toeplitz matrix $A$ with $a(i, i+j-1) = \frac{1}{\sqrt{j+1}}$, for $i = 1, \ldots, 64$ and $j = 1, \ldots, 64 - i + 1$. Fig. 3 shows the errors generated by the HSS method with $\beta = 25$, the Richardson iteration method and the Chebyshev –TIM method with $\beta = 6.98$. From Figure 3 we find that in this example the convergence of the Chebyshev–TIM method is much faster than the HSS and Richardson iteration methods. Moreover, notice that the HSS method is a method for the Toeplitz linear system, but the Chebyshev-TIM method can be used for a variety of linear systems.

![Graph comparing numerical errors of Chebyshev –TIM, Richardson iteration method, and HSS method](image)

**Table 5**

<table>
<thead>
<tr>
<th>$N = n^2$</th>
<th>$\omega$</th>
<th>Iter</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,</td>
<td>99</td>
<td>53</td>
<td>0.004145</td>
</tr>
<tr>
<td>$\omega \in (0, 102.0672)$</td>
<td>100</td>
<td>52</td>
<td><strong>0.004064</strong></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td>60</td>
<td>0.004722</td>
</tr>
<tr>
<td>1000</td>
<td>145</td>
<td>0.560800</td>
<td></td>
</tr>
<tr>
<td>$\omega \in (0, 1026.3)$</td>
<td>1010</td>
<td>144</td>
<td>0.584371</td>
</tr>
<tr>
<td>1024</td>
<td>143</td>
<td>0.521024</td>
<td></td>
</tr>
</tbody>
</table>

**5. Conclusions**

In this paper, a special case of the Richardson iterative method for solving large sparse linear systems has been developed. We have shown that the error bounds of this method are smaller than the TIM algorithm (Theorem 3.2). The numerical results also proved our claim concerning this issue, and in the most
cases the performance of these methods is much better in comparison with the existing methods. The results showed that the new algorithm converges fast and works with high accuracy for a variety of linear systems.

REFERENCES
