Extrapolation Accelerated PRESB Method for Solving a Class of Block Two-By-Two Linear Systems

Li-Dan Liao\(^1\), Guo-Feng Zhang\(^2\) and Xiang Wang\(^{1,*}\)

\(^1\)Department of Mathematics, Nanchang University, Nanchang 330031, PR. China.
\(^2\)School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, PR. China.

Received 28 May 2019; Accepted (in revised version) 18 January 2020.

Abstract. We use extrapolation acceleration technique to speed up the preconditioned square block matrix splitting iteration method for two-by-two block linear systems. It is shown that for relaxation parameter \(\omega = 4/3\), the convergence factor of the iteration method under consideration is \(1/3\). This yields the robustness and efficiency of the method. Numerical examples confirm the theoretical results and demonstrate the effectiveness of the approach developed.

AMS subject classifications: 65F10, 65F50, 65W05

Key words: PRESB preconditioner, two-by-two block matrix, spectral radius, iteration method, convergence factor.

1. Introduction

We consider the block two-by-two linear systems

\[
\mathbf{A} \mathbf{z} = \begin{pmatrix} W & -T \\ T & W \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b,
\]

where \(W, T \in \mathbb{R}^{n \times n}\) are symmetric positive semi-definite matrices such that

\[
\text{null}(W) \cap \text{null}(T) = \{0\}.
\]

Various applications involving systems (1.1) include time-dependent PDEs, distributed control problems, structural dynamics, quantum mechanics, molecular scattering and the solution of complex-valued systems

\[(W + iT)(x + iy) = f + ig\]
in real arithmetics — cf. Refs. [1–6, 8, 13, 19, 20, 22, 23, 26, 30, 34]. Therefore, such linear systems attracted considerable attention in recent years. A variety of efficient iteration methods and preconditioning techniques for solving systems (1.1) have been developed [2, 3, 5–13, 17, 18, 24, 25, 27, 28, 31–33, 35, 37–41]. Among those, preconditioned MHSS (PMHSS) iterations [12], preconditioned GSOR (PGSOR) iterations [24] and a preconditioned square block (PRESB) method [5, 28] proved to be very efficient and robust. In particular, Bai et al. [14] used Hermitian and skew-Hermitian splitting of the coefficient matrix \( A \) to prove that the spectral radius of the PMHSS iteration matrix is bounded by \( \sigma(\alpha) = \sqrt{\alpha^2 + 1}/(\alpha + 1) \) and \( \sigma(\alpha) \) attains the minimum \( \sqrt{2}/2 \approx 0.707 \) at \( \alpha = 1 \). However, it was noted that the PMHSS iteration matrix has complex eigenvalues. Hezari et al. [24] estimated the convergence factor of the PGSOR method by \( (\sqrt{2} - 1)/(\sqrt{2} + 1) \). The PGSOR method involves computing optimal parameters \( \alpha_{opt} \) and \( \omega^* \) related to the smallest and largest eigenvalues of \( W^{-1}T \). This method can be regarded as a special case of the GSOR method from [16]. In addition, Axelsson [5] proposed a PRESB preconditioner — viz

\[
P(\alpha) = \begin{pmatrix} W & -T \\ T & \alpha^2W + 2aT \end{pmatrix}.
\]

It is shown that \( P(\alpha)^{-1}A \) has only real valued eigenvalues, an optimal parameter \( \alpha^* \) is obtained and the eigenvalues of the PRESB preconditioned matrix are bounded by \([1/2, 1]\) with \( \alpha = 1 \). However, this optimal parameter \( \alpha^* \) is related to restrictive eigenvalues of the matrix \( W^{-1}T \), that may not be available for large scale problems.

It is well known that extrapolation can be used to improve the convergence of stationary iteration methods [15, 37]. Nevertheless, it is difficult to apply since a relaxation parameter should be determined and the spectral properties of the iteration matrix can be analysed under stronger conditions. In other words, after the extrapolation the iteration method may require a more complex theoretical analysis. For example, in order to find a quasi-optimal relaxation parameter for the SSOR iteration method, the coefficient matrix should be symmetric and positive definite. Therefore, not all stationary methods can be extrapolated.

Let us note that the extrapolation accelerated technique does not work for PMHSS and PGSOR methods since the corresponding iteration matrices have complex eigenvalues. On the other hand, the eigenvalues of the PRESB iteration matrix are real valued and are restricted to the interval \([0, 1/2]\) with \( \alpha = 1 \). This motivates us to utilise the extrapolation accelerated technique to speed up the PRESB method. The new method, called EAPRESB, is faster than PRESB and PMHSS methods. We also can use the quasi-optimal parameter \( \omega = 4/3 \) to make the convergence factor 1/3 to be exploited in all examples. Another advantage of the EAPRESB method is that the matrices \( W \) and \( T \) should not be positive definite simultaneously.

The rest of this paper is organised as follows. In Section 2, we review the PRESB method, introduce the extrapolation accelerated PRESB iteration method and discuss its convergence. Section 3 contains the results of numerical experiments. Finally, our brief conclusions are given in Section 4.
2. The EAPRESB Iteration Method and its Convergence

Let us recall an auxiliary result.

**Lemma 2.1** (cf. Wang et al. [36]). If $W, T \in \mathbb{R}^{n \times n}$ are symmetric positive semi-definite matrices satisfying the condition (1.2), then $W + T$ is a symmetric positive definite matrix.

Considering the coefficient matrix $\mathcal{A}$, we split it as

$$\mathcal{A} = P - R,$$

where

$$P = \begin{pmatrix} W & -T \\ T & W + 2T \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 \\ 0 & 2T \end{pmatrix}.$$

Then the PRESB iteration method can be described as follows:

$$P_{\tilde{z}}(k+1) = R_{\tilde{z}}(k) + b.$$

Introducing the matrix $H = W + T$, we represent $P$ in the form

$$P = \begin{pmatrix} I & -I \\ 0 & I \end{pmatrix} \begin{pmatrix} H & 0 \\ T & H \end{pmatrix} \begin{pmatrix} I & I \\ 0 & I \end{pmatrix}.$$  \hspace{1cm} (2.1)

According to Lemma 2.1 and (2.1), we have

$$P^{-1} \mathcal{A} = \begin{pmatrix} I & -2I - 2H^{-1}WH^{-1}W + 4H^{-1}W \\ 0 & M \end{pmatrix},$$  \hspace{1cm} (2.2)

where $M = I + 2H^{-1}WH^{-1}W - 2H^{-1}W$.

Set $G = P^{-1}R$. It follows from (2.2) that

$$G = I - P^{-1} \mathcal{A} = \begin{pmatrix} 0 & 2I + 2H^{-1}WH^{-1}W - 4H^{-1}W \\ 0 & I - M \end{pmatrix}. $$  \hspace{1cm} (2.3)

It was shown in [29] that all eigenvalues of $M$ are real and

$$\text{sp}(M) \subseteq [\lambda_n, 1] \subseteq \left[\frac{1}{2}, 1\right],$$  \hspace{1cm} (2.4)

where $\lambda_n$ is the minimum eigenvalue of $M$. We also have

$$\lambda(G) \in [0, 1 - \lambda_n] \subseteq \left[0, \frac{1}{2}\right].$$

Recalling the extrapolation accelerated technique, we introduce the following splitting iteration method

$$\tilde{z}^{(k+1)} = (1 - \omega)\tilde{z}^{(k)} + \omega(G\tilde{z}^{(k)} + P^{-1}b).$$  \hspace{1cm} (2.5)

It is easily seen that

$$G_\omega = (1 - \omega)I + \omega G$$  \hspace{1cm} (2.6)

is the iteration matrix of the method (2.5).

Let us discuss the convergence of the method (2.5).
**Theorem 2.1.** Let $W, T \in \mathbb{R}^{n \times n}$ be symmetric positive semi-definite matrices satisfying the condition (1.2). If

$$0 < \omega < 2,$$

then the iterative method (2.5) converges. Moreover, the convergence factor of the iteration method (2.5) is $1/3$ with $\omega = 4/3$.

**Proof.** It follows from (2.6) and (2.3) that

$$\lambda(G_\omega) = 1 - \omega + \omega \lambda(M) \in [1 - \omega, 1 - \omega \lambda_n],$$

where $\lambda$ denotes an eigenvalue of the corresponding matrix and $\lambda_n$ is the minimum eigenvalue of $M$. Therefore,

$$\rho(G_\omega) = \max\{|1 - \omega|, |1 - \omega \lambda_n|\},$$

and $\rho(G_\omega) < 1$ if and only if

$$|1 - \omega| < 1, \quad |1 - \omega \lambda_n| < 1. \quad (2.7)$$

Simplifying the inequalities (2.7) and using (2.4), we obtain that if $0 < \omega < 2$, then $\rho(G_\omega) < 1$.

Observations show that the spectral radius $\rho(G_\omega)$ attains its minimum when $|1 - \omega \lambda_n| = |1 - \omega|$, i.e.

$$\min_{\omega} \rho(G_\omega) = \rho(G_{\omega_{opt}}) = \frac{1 - \lambda_n}{1 + \lambda_n},$$

where $\omega_{opt} = 2/(1 + \lambda_n)$.

If $\omega = \omega_{s} = 4/3$ and $\lambda_n \geq 1/2$, one can easily check that $|1 - \omega \lambda_n| \leq 1/3$, which indicates that

$$\rho(G_{\omega_s}) = \frac{1}{3}.$$

**Remark 2.1.** Theorem 2.1 shows that the quasi-optimal convergence factor of the EAPRESB method is $1/3 \approx 0.333$. It is smaller than the factors $\sqrt{2} - 1 \approx 0.414$ and $\sqrt{2}/2 \approx 0.707$ derived in [29] and in [12], respectively.

**Remark 2.2.** Theorem 2.1 also shows that the EAPRESB method converges for the mesh of any size, implying the robustness of the method. Moreover, the existence of the quasi-optimal parameter $4/3$ makes EAPRESB method more practical than many classical methods — cf. [10, 11, 17, 24, 39], since the convergence does not rely on eigenvalues.

In actual implementation of the EAPRESB method we set

$$\tilde{x}^{(k+1)} = \tilde{x}^{(k)} + \omega P^{-1} r, \quad (2.8)$$

where $r = (r_1^T, r_2^T)^T$, $\tilde{x} = (x^T, y^T)^T$, and proceed as follows:

1. solve $(W + T) u_1 = r_1 + r_2;$
(2) solve \((W + T)u_2 = (r_2 - Tu_1);\)
(3) compute \(x = x + \omega u_2, y_1 = y + \omega (r_1 - u_2).\)

If \(\omega = 1,\) the algorithm (2.8) turns into the PRESB method \([5]\). We note that the main computational cost of the algorithm is related to the first two steps, where two linear sub-systems with the matrix \(W + T\) have to be solved. Since the matrix \(W + T\) is symmetric and positive definite, a direct method like the Cholesky decomposition or a Krylov subspace method like conjugate gradient method with suitable preconditioners can be used. Besides, we also discovered that at each iteration step, the EAPRESB requires the same amount of calculations as PMHSS and PRESB methods. However, EAPRESB has a smaller convergence factor, so that it is faster than the methods mentioned.

3. Numerical Experiments

We want to illustrate the effectiveness and robustness of the EAPRESB method with respect to iteration steps and elapsed CPU time in seconds referred to as 'IT' and 'CPU', respectively. All computations are conducted in MATLAB R2016a with machine precision \(2.2 \times 10^{-16}\) on a personal computer with 3.20 GHz CPU (Intel(R) Core(TM) i5-3470), 8.00 GB RAM.

Starting with the zero initial guess, the EAPRESB is compared with the PMHSS \([12]\), PGSOR \([24]\) and PRESB \([5]\) methods. The iterations are terminated once the current residual in the first stage satisfies
\[
\text{RES} := \frac{\|r_k\|}{\|r_0\|} < 10^{-6}
\]
or if the number of iterations exceeds \(k_{\text{max}} = 500.\)

Two types of examples from different applications are used to examine the performance of the methods — viz. complex symmetric linear systems with positive definite or indefinite blocks and PDE-constrained optimisation problems.

3.1. Complex symmetric linear system

The results of simulations here are listed in Table 1. In addition, Fig. 1 demonstrates the spectral radii of the EAPRESB iteration matrices in Examples 3.1-3.3.

Example 3.1 (cf. Refs. \([5, 11]\)). Consider the linear algebraic system
\[
\left[ \left( K + \frac{(3 - \sqrt{3})}{\tau} I \right) + i \left( K + \frac{(3 + \sqrt{3})}{\tau} I \right) \right] u = c,
\]
where \(\tau\) is a positive parameter, \(K\) is the five-point centred difference matrix approximating the negative Laplacian operator \(L = -\Delta\) with homogeneous Dirichlet boundary conditions on the uniform mesh of the size \(h = 1/(m+1)\) on the unit square \([0, 1] \times [0, 1]\). The matrix \(K \in \mathbb{R}^{n \times n}\) can be represented in the tensor-product form
\[
K = I \otimes V_m + V_m \otimes I
\]
with

\[ V_m = h^{-2} \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}. \]

Therefore, \( K \) is an \( n \times n \) matrix with \( n = m^2 \). In actual implementations, we set \( m = 64, 128, 256, 512, 1024 \) for five grids, respectively. The right-hand side vector \( c \) in (3.1) has the entries

\[ c_j = \frac{(1 - i)j}{(\tau(j + 1)^2)}, \quad j = 1, 2, \ldots, n, \]

and the parameter \( \tau = h \).

**Example 3.2** (cf. Refs. [5, 11]). Consider the complex linear system \((W + iT)u = c\) with the matrices

\[ T = I \otimes V + V \otimes I, \quad W = 10(I \otimes V_c + V_c \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I, \]

where

\[ V = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}, \quad V_c = V - e_1 e_m^T - e_m e_1^T \]

and \( e_1 \) and \( e_m \) are the first and last unit vectors in \( \mathbb{R}^m \), respectively. The size of \( W \) is \( n = m^2 \).

In actual implementation, we set \( m = 64, 128, 256, 512, 1024 \), respectively, the right-hand side vector \( c = (1 + i)Al \), where \( I \) is the vector with all entries 1.

**Example 3.3** (cf. Bai et al. [12]). On the square \( D = [0, 1] \times [0, 1] \), we consider the complex Helmholtz equation

\[ -\Delta p + \sigma_1 p + i \sigma_2 p = f, \]

where \( \sigma_1 \) and \( \sigma_2 \) are real coefficient functions and \( p \) satisfies the Dirichlet boundary conditions. The problem is discretised by finite differences on a \( m \times m \) grid with mesh size \( h = 1/(m + 1) \). This leads to the following system of linear equations:

\[ ((K + \sigma_1 I) + i \sigma_2 I)u = c, \]

where \( K = I \otimes V_m + V_m \otimes I \) is the discretisation of \(-\Delta \) by centered differences and \( V_m = h^{-2} \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m} \). Besides, \( \sigma_1 = 100, \sigma_2 = 100 \) and \( c = (1 + i)Al \) with the vector \( I \) all entries of which are 1. Here, we take \( m = 64, 128, 256, 512, 1024 \).

Table 1 demonstrates the full robustness and effectiveness of all methods. We also note that the EAPRESB and PRESB methods outperform the PMHSS and PGSOR methods, requiring much less CPU time. The EAPRESB is more efficient than the PRESB method, since it needs less CPU time and the iteration steps. Moreover, the number of iteration steps is almost the same for different \( m \), which shows the robustness of the method. Considering Fig. 1, we note that the EAPRESB method is very stable. The convergence factor is the same 1/3 in all examples and for all \( m \), consistent with the theoretical results.
| Method   | $\alpha$ | $m$ | $\omega$ | Iterations | CPU Time |
|----------|----------|-----|----------|------------|----------|
| **Example 3.1** |         |     |          |            |          |
| PMHSS    | 1        | 64 | 1        | 21         | 0.0401   |
|          |          | 128| 1        | 21         | 0.1939   |
|          |          | 256| 1        | 21         | 1.3018   |
|          |          | 512| 1        | 20         | 7.6140   |
|          |          | 1024| 1      | 20        | 47.8706  |
| PGSOR    | 1        | 8  | 0.8280   | 8          | 8.8280   |
|          |          | 8  | 0.8280   | 8          | 0.8280   |
|          |          | 8  | 0.8280   | 8          | 0.8280   |
|          |          | 8  | 0.8280   | 8          | 0.8280   |
| PRESB    | 1        | 21| 0.8280   | 21         | 1.1180   |
|          |          | 21| 0.8280   | 21         | 7.1696   |
|          |          | 21| 0.8280   | 21         | 45.6517  |
| EAPRESB  | 0.9999   | 0.9999| 0.9999 | 0.9999 | 0.9999 |
|          | 0.9999   | 0.9999| 0.9999 | 0.9999 | 0.9999 |
|          | 0.9999   | 0.9999| 0.9999 | 0.9999 | 0.9999 |
|          | 0.9999   | 0.9999| 0.9999 | 0.9999 | 0.9999 |

| **Example 3.2** |         |     |          |            |          |
| PMHSS    | 1        | 32 | 0.8280   | 32         | 0.0797   |
|          |          | 32| 0.8280   | 32         | 0.4010   |
|          |          | 32| 0.8280   | 32         | 2.7777   |
|          |          | 32| 0.8280   | 32         | 21.0868  |
|          |          | 32| 0.8280   | 32         | 168.9692 |
| PGSOR    | 1        | 10| 0.8280   | 10         | 9        |
|          |          | 10| 0.8280   | 10         | 9        |
|          |          | 10| 0.8280   | 10         | 10       |
|          |          | 10| 0.8280   | 10         | 10       |
| PRESB    | 1        | 16| 0.8280   | 16         | 15       |
|          |          | 16| 0.8280   | 16         | 15       |
|          |          | 16| 0.8280   | 16         | 15       |
|          |          | 16| 0.8280   | 16         | 15       |
| EAPRESB  | 0.5      | 0.5| 0.5     | 0.5        | 0.5     |
|          | 0.5      | 0.5| 0.5     | 0.5        | 0.5     |
|          | 0.5      | 0.5| 0.5     | 0.5        | 0.5     |
|          | 0.5      | 0.5| 0.5     | 0.5        | 0.5     |

| **Example 3.3** |         |     |          |            |          |
| PMHSS    | 1        | 39 | 0.8280   | 39         | 0.0589   |
|          |          | 40| 0.8280   | 40         | 0.2658   |
|          |          | 40| 0.8280   | 40         | 1.6132   |
|          |          | 40| 0.8280   | 40         | 10.4895  |
|          |          | 40| 0.8280   | 40         | 61.2325  |
| PGSOR    | 1        | 8  | 0.8280   | 8          | 8        |
|          |          | 8  | 0.8280   | 8          | 8        |
|          |          | 8  | 0.8280   | 8          | 9        |
|          |          | 8  | 0.8280   | 8          | 9        |
| PRESB    | 1        | 17 | 0.8280   | 17         | 14       |
|          |          | 16| 0.8280   | 16         | 14       |
|          |          | 14| 0.8280   | 14         | 13       |
|          |          | 13| 0.8280   | 13         | 13       |
| EAPRESB  | 1/3      | 1/3| 1/3     | 1/3        | 1/3     |
|          | 1/3      | 1/3| 1/3     | 1/3        | 1/3     |
|          | 1/3      | 1/3| 1/3     | 1/3        | 1/3     |
|          | 1/3      | 1/3| 1/3     | 1/3        | 1/3     |

CPU times are given in seconds.
3.2. PDE-constrained optimisation problems

Example 3.4 (cf. Refs. [8, 30, 37]). Let $\Omega = [0, 1]^2$ and $\partial \Omega$ be the boundary of $\Omega$. We consider the distributed control problem

$$\min_{u,f} \frac{1}{2} \|u - u_\ast\|^2_2 + \beta \|f\|^2_2,$$

subject to $-\nabla^2 u = f$ in $\Omega$,

with $u = g$ on $\partial \Omega$,

(3.2)

where $\beta \in \mathbb{R}^+$ is a regularisation parameter, $g$ a given function and

$$u_\ast = \begin{cases} (2x-1)^2(2y-1)^2, & \text{if } (x,y) \in [0,1/2]^2, \\ 0, & \text{otherwise}. \end{cases}$$

We are looking for a function $u$ which satisfies the PDE and is as close as possible to $u_\ast$ in a norm.

Using the approach [8,30,37], we reduce the problem (3.2) to the linear system

$$Bx \equiv \begin{pmatrix} 2\beta M & 0 & -M \\ 0 & M & K \\ -M & K & 0 \end{pmatrix} \begin{pmatrix} f \\ u \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ b \\ d \end{pmatrix} \equiv \tilde{g},$$

or equivalently,

$$Bx \equiv \begin{pmatrix} M \\ \sqrt{2\beta K} \\ \sqrt{2\beta K} \end{pmatrix} \begin{pmatrix} \frac{\lambda}{\sqrt{2\beta}} \\ 1 \\ \frac{1}{\sqrt{2\beta}} \end{pmatrix} = \begin{pmatrix} \frac{-\tilde{d}}{\sqrt{2\beta}} \\ 1 \\ \frac{1}{\sqrt{2\beta}} \end{pmatrix} \equiv \tilde{g},$$

(3.3)
Table 2: The results for Example 3.4.

| method | $\nu = 10^{-2}$ | $\nu = 10^{-4}$ | $\nu = 10^{-6}$ | $\nu = 10^{-8}$ | $\nu = 10^{-10}$ |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|
| PGSOR  | $2^6$ 9 0.0581 | 9 0.0425 | 9 0.0416 | 9 0.0367 | 9 0.0347 |
|        | $2^7$ 9 0.4853 | 9 0.5327 | 9 0.4779 | 9 0.4755 | 10 0.4987 |
|        | $2^8$ 9 3.3082 | 9 3.1582 | 9 3.1218 | 9 3.1051 | 10 3.2317 |
|        | $2^9$ 9 25.3419 | 9 25.5047 | 9 25.7915 | 9 25.7203 | 10 26.1311 |
| PMHSS  | $2^6$ 39 0.0701 | 40 0.0536 | 40 0.0560 | 39 0.0512 | 39 0.0537 |
|        | $2^7$ 39 0.5657 | 40 0.5705 | 40 0.6930 | 39 0.5707 | 39 0.5555 |
|        | $2^8$ 39 3.6991 | 40 3.7603 | 40 3.7054 | 40 3.7723 | 39 3.7464 |
|        | $2^9$ 39 29.7287 | 40 30.0386 | 40 29.8575 | 40 30.2443 | 39 29.1483 |
| EAPRESB| $2^6$ 14 0.0619 | 14 0.0444 | 14 0.0413 | 13 0.0357 | 13 0.0323 |
|        | $2^7$ 13 0.5804 | 13 0.5578 | 14 0.5319 | 13 0.4974 | 13 0.4378 |
|        | $2^8$ 14 3.1123 | 14 3.1696 | 14 2.0966 | 14 2.9384 | 14 2.6063 |
|        | $2^9$ 14 14.6780 | 14 14.6392 | 14 14.7099 | 14 13.6448 | 14 13.6069 |
| PRESB  | $2^6$ 9 0.0307 | 11 0.0320 | 16 0.0388 | 18 0.0413 | 19 0.0436 |
|        | $2^7$ 9 0.3167 | 13 0.4356 | 16 0.5231 | 18 0.5884 | 19 0.6158 |
|        | $2^8$ 9 1.7596 | 12 2.2630 | 15 3.0674 | 17 3.1106 | 19 3.4710 |
|        | $2^9$ 9 10.4556 | 10 11.4705 | 13 14.5333 | 16 17.5274 | 18 19.5537 |

where $M \in \mathbb{R}^{n \times n}$ is the mass matrix, $K \in \mathbb{R}^{n \times n}$ the stiffness matrix, $b \in \mathbb{R}^n$ the Galerkin projection of the discrete state $u_\ast$, $\lambda$ the vector of Lagrange multipliers, and $d \in \mathbb{R}^n$ represents the boundary data. In this example, we let $g = 0$ and discretise the distributed control problem by $Q_1$ finite element method on the $m \times m$ uniform grids with the mesh size $h = 1/(m+1)$, so that $M$ is the matrix of order $n = m^2$. In the implementation we use the IFISS software package [21].

The linear system (3.3) has the form (1.1) with $W = M$, $T = \sqrt{2\beta K}$ and can be solved by EAPRESB, PGSOR, PMHSS and PRESB iteration methods. Table 2 provides numerical results obtained by these methods for various mesh sizes and regularisation parameters. For the sake of convenience, $\omega = 1$ in the EAPRESB method, $\alpha = 1$ in the PMHSS method, $\alpha = 1$ and $\omega = 0.828$ in the PGSOR method.

Table 2 indicates that the EAPRESB method is robust with respect to the regularisation parameter $\beta$. The EAPRESB method is generally better than the other tested methods in terms of execution times. For large problems the effect is more obvious. Moreover, the number of iteration steps basically stays at 14, since we proved that for $\omega = 4/3$, the convergence factor of EAPRESB method is stable at $1/3$. This confirms the reliability of the theoretical results to some extent. In addition, comparing the EAPRESB and PRESB iterations, we note the general superiority of the former in the number of iteration steps and CPU time, so that the EAPRESB actually speeds up the PRESB iteration method.
4. Conclusions

We use extrapolation acceleration technique to speed up the PRESB iteration method for linear systems (1.1) with symmetric positive semi-definite matrices $W$ and $T$. It is shown that for relaxation parameter $\omega = 4/3$, the convergence factor of the iteration method under consideration is $1/3$. It is smaller than the convergence factor of the PRESB method and the features of the new method makes the existing parameter selection procedure unnecessary. Numerical examples confirm the theoretical results and demonstrate the effectiveness of the EAPRESB method.

Acknowledgments

We would like to express our sincere thanks to the anonymous referees and the editors for their constructive comments and valuable suggestions, which greatly improved the original manuscript.

This work was supported by the National Natural Science Foundation of China under grants 11901278, 11771193, 11961048 and by NSF of Jiangxi Province under grant 20181ACB20001.

References

[1] O. Axelsson, Iterative Solution Methods, Cambridge University Press (1994).
[2] O. Axelsson, S. Farouq and M. Neytcheva, Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems, Numer. Algorithms 74(1), 1-19 (2015).
[3] O. Axelsson, S. Farouq and M. Neytcheva, A preconditioner for optimal control problems, constrained by Stokes equation with a time-harmonic control, J. Comput. Appl. Math. 310, 5-18 (2016).
[4] O. Axelsson and A. Kucherov, Real valued iterative methods for solving complex symmetric linear algebraic systems, Numer. Linear Algebra Appl. 7, 197-218 (2000).
[5] O. Axelsson, M. Neytcheva and B. Ahmad, A comparison of iterative methods to solve complex valued linear algebraic systems, Numer. Algorithms 66(4), 811-841 (2014).
[6] O. Axelsson and D.K. Salkuyeh, A new version of a preconditioning method for certain two-by-two block matrices with square blocks, Bit Numer Math. 59(2), 321-342 (2019).
[7] Z.-Z. Bai, Structured preconditioners for nonsingular matrices of block two-by-two structures, Math. Comput. 75, 791-815 (2006).
[8] Z.-Z. Bai, Block preconditioners for elliptic PDE-constrained optimization problems, Computing 91(4), 379-395 (2011).
[9] Z.-Z. Bai, On preconditioned iteration methods for complex linear systems, J. Engrg. Math. 93(1), 41-60 (2015).
[10] Z.-Z. Bai, M. Benzi and F. Chen, Modified HSS iteration methods for a class of complex symmetric linear systems, Computing 87, 93-111 (2010).
[11] Z.-Z. Bai, M. Benzi and F. Chen, On preconditioned MHSS iteration methods for complex symmetric linear systems, Numer. Algorithms 56, 297-317 (2011).
[12] Z.-Z. Bai, M. Benzi, F. Chen and Z.-Q. Wang, Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, IMA J. Numer. Anal. 33, 343-369 (2013).

[13] Z.-Z. Bai, F. Chen and Z.-Q. Wang, Additive block diagonal preconditioning for block two-by-two linear systems of skew-Hamiltonian coefficient matrices, Numer. Algorithms. 62, 655-675 (2013).

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

[15] Z.-Z. Bai, M. K. Ng and G. H. Golub, On Successive-Overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, Numer. Linear Algebra Appl. 14(4), 319-335 (2007).

[16] Z.-Z. Bai, B. N. Parlett and Z.-Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, Numer. Math. 102(1), 1-38 (2005).

[17] Y. Cao and Z.-R. Ren, Two variants of the PMHSS iteration method for a class of complex symmetric indefinite linear algebraic systems, Appl. Math. Comput. 264, 61-71 (2015).

[18] S.-M. Cao, W. Feng and Z.-Q. Wang, On a type of matrix splitting preconditioners for a class of block two-by-two linear systems, Appl. Math. Lett., 79, 205-210 (2018).

[19] D. D. Day and M. A. Heroux, Solving complex-valued linear systems via equivalent real reformulations, SIAM J. Sci. Comput. 23, 480-498 (2001).

[20] W. V. Dijk and F. M. Toyama, Accurate numerical solutions of the time-dependent Schrödinger equation, Phys. Rev. E. 75, 1-10 (2007).

[21] H. C. Elman, A. Ramage and D. J. Silvester, IFISS: A Computational Laboratory for Investigating Incompressible Flow Problems, SIAM Rev. 56(2), 261-273 (2014).

[22] A. Ferrani, F. Perotti and V. Simoncini, Iterative system solvers for the frequency analysis of linear mechanical systems, Comput. Methods Appl. Mech. Engrg. 190, 1719-1739 (2000).

[23] A. Frommer, T. Lippert, B. Medeke and K. Schilling, Numerical Challenges in Lattice Quantum Chromodynamics, Springer Lecture Notes Series 15, Springer-Verlag (2000).

[24] D. Hezari, V. Edalatpour and D.K. Salkuyeh, Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations, Numer. Linear Algebra Appl. 22, 761-776 (2015).

[25] C. Lang and Z.R. Ren, Rotated block triangular preconditioners for a class of block two-by-two matrices, J. Eng. Math. 93, 87-98 (2015).

[26] O. Lass, M. Vallejos, A.Borzi and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing 84, 27-48 (2009).

[27] L.-D. Liao and G.-F. Zhang, Efficient preconditioner and iterative method for large complex symmetric linear algebraic systems, East Asian J. Appl. Math. 7, 530-547 (2017).

[28] L.-D. Liao and G.-F. Zhang, A note on block diagonal and block triangular preconditioners for complex symmetric linear systems, Numer. Algorithms 80(4), 1143-1154 (2019).

[29] L.-D. Liao, G.-F. Zhang and R.-X. Li, Optimizing and improving of the C-to-R method for solving complex symmetric linear systems, Appl. Math. Lett. 82, 79-84 (2018).

[30] J.W. Pearson and A.J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numer. Linear Algebra Appl. 19, 816-829 (2012).

[31] H. N. Pour, An alternative lopsided PMHSS iteration method for complex symmetric systems of linear equations, East Asian J. Appl. Math. 8, 313-322 (2018).

[32] Z.-R. Ren, Y. Cao and L.-L. Zhang, On preconditioned MHSS real-valued iteration methods for a class of complex symmetric indefinite linear systems, East Asian J. Appl. Math. 6, 192-210 (2016).

[33] D.K. Salkuyeh, D. Hezari, V Edalatpour, Generalized SOR iterative method for a class of complex symmetric linear algebraic system of equations, Int. J. Comput. Math. 92, 802-815 (2015).
[34] A. Sommerfeld, *Partial Differential Equations in Physics*, Academic Press, New York, NY, USA, (1949).

[35] J. Wang, X.-P Guo and H.-X. Zhong, *Accelerated GPMHSS method for solving complex systems of linear equations*, East Asian J. Appl. Math. 7, 143-155 (2017).

[36] T. Wang, Q.-Q. Zheng and L.-Z. Lu *A new iteration method for a class of complex symmetric linear systems*, J. Comput. Appl. Math. 325, 188-197 (2017).

[37] Z.-Q. Wang, *On a Chebyshev accelerated splitting iteration method with application to two-by-two block linear systems*, Numer. Linear Algebra Appl. 25(5), e2172 (2018).

[38] R.-P. Wen, S.-D. Li and G.-Y. Meng, *SOR-like methods with optimization model for augmented linear systems*, East Asian J. Appl. Math. 7, 101-115 (2017).

[39] X.-Y. Xiao and X. Wang, *A new single-step iteration method for solving complex symmetric linear systems*, Numer. Algorithms, 78, 643-660 (2018).

[40] X.-Y. Xiao, X. Wang and H.-W. Yin, *Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems*, Comput. Math. Appl. 74, 2269-2280 (2017).

[41] Z. Zheng, G.-F. Zhang and M.-Z. Zhu, *A note on preconditioners for complex linear systems arising from PDE-constrained optimization problems*, Appl. Math. Lett. 61, 114-121 (2016).