Japan Journal of Industrial and Applied Mathematics (2022) 39:145–163
https://doi.org/10.1007/s13160-021-00479-7

ORIGINAL PAPER

A new two-parameter iteration method for indefinite complex symmetric linear systems

Maeddeh Pourbagher[1] · **Davod Khojasteh Salkuyeh**[1,2]

Received: 14 April 2021 / Revised: 26 July 2021 / Accepted: 2 August 2021 /
Published online: 9 August 2021

Abstract

We present a new iteration method, namely symmetric positive definite and negative stable splitting (SNSS) method for solving complex symmetric indefinite linear systems. Theoretical analysis shows that the proposed method is convergent under suitable conditions. In each iteration of the method two subsystems should be solved. One of them can be solved inexactly using the conjugate gradient method, and the second one by the Chebyshev acceleration method in conjunction with the well-known PRESB preconditioner. Numerical experiments are reported to indicate efficiency of the SNSS method.

Keywords Complex · Symmetric · Symmetric positive definite · Negative stable · Indefinite

Mathematics Subject Classification 65F10 · 65F50 · 65F08

1 Introduction

We consider the system of linear equations

$$Ax \equiv (W + iT)x = b, \tag{1}$$

where $W = -W_1 + W_2$, W_1, W_2 and $T \in \mathbb{R}^{n\times n}$, and $i = \sqrt{-1}$ denotes the imaginary unit. We assume that the matrices W_1, W_2 and T are symmetric positive definite (SPD). This type of complex symmetric linear systems come from many problems

✉ Davod Khojasteh Salkuyeh
khojasteh@guilan.ac.ir

Maeddeh Pourbagher
mapourbagher@gmail.com

[1] Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

[2] Center of Excellence for Mathematical Modelling, Optimization and Combinational Computing (MMOCC), University of Guilan, Rasht, Iran

in scientific computing and engineering applications, such as structural dynamics [7, 8, 15] and Helmholtz equations [9, 17, 33, 34].

Since the coefficient matrix of the system (1) is often of large size, iterative methods are preferred for solving the system. Based on the Hermitian and skew-Hermitian splitting (HSS) method [6], Bai et al. proposed the modified HSS method in [7]. This method for solving Eq. (1) can be written as following.

The MHSS method Let $x^{(0)} \in \mathbb{C}^n$ be an initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x^{(k)}\}_{k=1}^{\infty}$ converges, compute the next iterate $x^{(k+1)}$ via the following procedure:

$$\begin{cases} (\alpha I + W)x^{\left(k+\frac{1}{2}\right)} = (\alpha I - iT)x^{(k)} + b, \\ (\alpha I + T)x^{(k+1)} = (\alpha I + iW))x^{\left(k+\frac{1}{2}\right)} - ib, \end{cases} \tag{2}$$

where α is a given positive constant and I is the identity matrix.

There are several other methods for solving the system (1). For example in [8] Bai et al. constructed the preconditioned modified HSS (PMHSS) iteration method. Salkuyeh et al. applied the successive overrelaxation (SOR) method for solving this system [28] (see also [13, 14, 18, 19]). Hezari et al. in [19] presented the scale-splitting (SCSP) method (see also [20, 21, 27, 29] for some other versions of SCSP). The C-to-R method was presented by Axelsson and Kucherov in [2]. The transformed matrix iteration method was presented by Axelsson and Salkuyeh in [5]. Xie and Li proposed the preconditioned Euler-extrapolated single-step HSS splitting method in [36]. A parameterized splitting iteration method for complex symmetric system of linear equations was presented by Zhang and Zheng [37].

The authors of [7] proved that the MHSS method is unconditionally convergent to the unique solution of the system (1), when both of the matrices W and T are symmetric positive semidefinite (SPSD) with at least one of them being positive definite. However, when the matrices W_1 and W_2 are symmetric positive definite (SPD), the matrix W may be symmetric indefinite. In this case, the matrices $\alpha I + W$ and $\alpha W + T$ (or $\alpha T + W$) may be indefinite or singular and then the MHSS and PMHSS methods may fail to converge. This is the case for all the aforementioned methods.

Li and Wu in [35] proposed the modified positive/negative stable splitting (MPNS) method when W is symmetric indefinite. A preconditioned version of MPNS was presented by Cui et al. in [12]. In this paper, we present a two-parameter iteration method which is called the symmetric positive definite and negative stable splitting (SNSS) method for solving complex symmetric linear system (1) in the case that matrix W is symmetric indefinite. We prove that the method is convergent under some conditions. Numerical results show that the SNSS method is more effective than the MPNS method.

Throughout the paper we use the following notation. $\|.\|_2$ denotes the Euclidean norm. Spectrum and spectral radius of a square matrix are denoted by $\sigma(.)$ and $\rho(.)$, respectively. The imaginary unit is shown by i $(i = \sqrt{-1})$. The real and imaginary parts of a complex number (or vector) z are denoted by $\Im(z)$ and $\Re(z)$, respectively. We use $\|.\|_F$ for the Frobenius norm of a matrix. The Kronecker product is denoted by $\otimes$.

This paper is organized as following. In Sect. 2, we preset a brief review of the MPNS method. In Sect. 3, we design the SNSS method and discuss its convergence properties. In Sect. 4, we present inexact version of the SNSS method. The SNSS preconditioner is introduced in Sect. 5. Section 7 is devoted to some numerical experiments. Eventually, we present some conclusions in Sect. 8.

2 The MPNS method

In this section, we briefly study the MPNS method. Li and Wu [35] split the coefficient matrix A in (1) as

$$A = P + N,$$

where

$$P = W_2 \quad \text{and} \quad N = -W_1 + iT.$$

This is a positive/negative stable splitting (PNS) because P and N are positive-stable and negative stable, respectively. Therefore, the complex symmetric linear system (27) can be written as

$$\big(\alpha I + W_2\big)x = \big(\alpha I + W_1 - iT\big)x + b,$$

By multiplying both sides of the system (1) by $-i$ gives

$$\big(iW_1 - iW_2 + T\big)x = -ib.$$

Hence, the system (1) is expressed as following fixed point equation

$$\big(\alpha I + iW_1 + T\big)x = \big(\alpha I + iW_2\big)x - ib.$$

This yields the modified positive/negative stable splitting (MPNS) method which can be written as following.

The MPNS method Let $x^{(0)} \in \mathbb{C}^n$ be an initial guess. For $k = 0, 1, 2, \dots$ until the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ converges, compute the next iterate $x^{(k+1)}$ via:

$$\begin{cases} \big(\alpha I + W_2\big)x^{\left(k+\frac{1}{2}\right)} = \big(\alpha I + W_1 - iT\big)x^{(k)} + b, \\ \big(\alpha I + iW_1 + T\big)x^{(k+1)} = \big(\alpha I + iW_2\big)x^{\left(k+\frac{1}{2}\right)} - ib, \end{cases} \tag{3}$$

where α is a given positive constant.

To obtain the fixed point form of the MPNS method, one can compute vector $x^{(k+\frac{1}{2})}$ from the first equation in (3) and substitute in the second equation to obtain

$$x^{(k+1)} = P_\alpha x^{(k)} + Q_\alpha b,$$

where

$$P_\alpha = \left(\alpha I + iW_1 + T\right)^{-1}\left(\alpha I + iW_2\right)\left(\alpha I + W_2\right)^{-1}\left(\alpha I + W_1 - iT\right),$$

and

$$Q_\alpha = (1-i)\alpha\left(\alpha I + iW_1 + T\right)^{-1}\left(\alpha I + W_2\right)^{-1}.$$

Here P_α is the iteration matrix of the MPNS method. In addition, if we introduce

$$E_\alpha = \tfrac{1+i}{2\alpha}\left(\alpha I + W_2\right)\left(\alpha I + iW_1 + T\right) \quad \text{and} \quad F_\alpha = \tfrac{1+i}{2\alpha}\left(\alpha I + iW_2\right)\left(\alpha I + W_1 - iT\right),$$

then

$$A = E_\alpha - F_\alpha, \quad \text{and} \quad P_\alpha = E_\alpha^{-1}F_\alpha.$$

As a result we have $P_\alpha = I - E_\alpha^{-1}A$. Therefore, the matrix E_α can be used as a preconditioner for the system (1). If the matrix $T - W_1$ is positive semi-definite, then the MPNS method is convergent for every $\alpha > 0$ (See [35, Theorem 3.1]).

3 The SNSS method

Multiplying both sides of Eq. (1) by -1 and then adding αT gives

$$\left(\alpha T + W_1\right)x = \left((\alpha + i)T + W_2\right)x - b. \tag{4}$$

On the other hand, by adding $i\beta T$ to the both sides of (1) results in the equation

$$\left(i(\beta + 1)T + W_2\right)x = \left(i\beta T + W_1\right)x + b. \tag{5}$$

Now using Eqs. (4) and (5) we state the SNSS iteration methods as follows.

The SNSS method Let $x^{(0)} \in \mathbb{C}^n$ be an initial guess. For $k = 0, 1, 2, \dots$ until the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ converges, compute the next iterate $x^{(k+1)}$ via:

$$\begin{cases} (\alpha T + W_1)x^{\left(k+\frac{1}{2}\right)} = \left((\alpha + i)T + W_2\right)x^{(k)} - b, \\ \left(i(\beta + 1)T + W_2\right)x^{(k+1)} = (i\beta T + W_1)x^{\left(k+\frac{1}{2}\right)} + b, \end{cases} \tag{6}$$

where α is a given positive constant.

In each iteration of the SNSS method two linear systems with the coefficient matrices $S_1 = \alpha T + W_1$ and $S_2 = i(\beta + 1)T + W_2$ should be solved. Obviously, the matrix S_1 is SPD. Therefore, the first subsystem in the SNSS method can be solved exactly using the Cholesky factorization or inexactly using the conjugate gradient (CG) method. On the other hand, the matrix S_2 is complex symmetric with real and imaginary parts being SPD. So the second subsystem can be solved exactly using the LU factorization or inexactly using an iteration method. There are several Krylov subspace methods for solving the second subsystem (see [16, 31, 32]). We will shortly see that this subsystem can be efficiently solved using the generalized minimal residual (GMRES) [24, 25] or Chebyshev acceleration [24] methods in conjunction with preconditioned square block (PRESB) matrix as a preconditioner [4, 5].

By eliminating the vector $x^{(k+\frac{1}{2})}$ from Eq. (6) we get

$$x^{(k+1)} = G_{\alpha,\beta} x^{(k)} + H_{\alpha,\beta} b,$$

where

$$G_{\alpha,\beta} = \left(i(\beta+1)T + W_2\right)^{-1}\left(i\beta T + W_1\right)\left(\alpha T + W_1\right)^{-1}\left((\alpha+i)T + W_2\right),$$

and

$$H_{\alpha,\beta} = (\alpha - i\beta)\left(i(\beta+1)T + W_2\right)^{-1} T\left(\alpha T + W_1\right)^{-1},$$

in which $G_{\alpha,\beta}$ is the iteration matrix of the SNSS method. The next theorem investigates the convergence of the SNSS iteration method.

Theorem 1 *Let $A = -W_1 + W_2 + iT$ such that W_1, W_2 and T be real symmetric and positive definite matrices. Suppose that $\lambda_{\max}$ is the largest eigenvalue of $\tilde{W}_1 = T^{-\frac{1}{2}} W_1 T^{-\frac{1}{2}}$ and $\mu_{\min}$ is the smallest eigenvalue of $\tilde{W}_2 = T^{-\frac{1}{2}} W_2 T^{-\frac{1}{2}}$. If α is large enough and*

$$\beta > \max\left\{0, \frac{1}{2}\left(\lambda_{\max}^2 - \mu_{\min}^2 - 1\right)\right\},$$

then $\rho(G_{\alpha,\beta}) < 1$, where $G_{\alpha,\beta}$ is the iteration matrix of the SNSS method. This means that the SNSS iteration method converges to the unique solution of complex symmetric linear system (1).

Proof First of all, we see that the matrix $G_{\alpha,\beta}$ is similar to

$$\tilde{G}_{\alpha,\beta} = \left(i(\beta+1)T + W_2\right) G_{\alpha,\beta} \left(i(\beta+1)T + W_2\right)^{-1}.$$

Therefore,

$$\begin{aligned}\rho(G_{\alpha,\beta})^2 &= \rho(\tilde{G}_{\alpha,\beta})^2 \\ &= \rho\Big((i\beta T + W_1)(\alpha T + W_1)^{-1}\big((\alpha+i)T + W_2\big)\big(i(\beta+1)T + W_2\big)^{-1}\Big)^2 \\ &= \rho\Big(T^{\frac{1}{2}}(i\beta I + \tilde{W}_1)(\alpha I + \tilde{W}_1)^{-1}\big((\alpha+i)I + \tilde{W}_2\big)\big(i(\beta+1)I + \tilde{W}_2\big)^{-1}T^{-\frac{1}{2}}\Big)^2 \\ &= \rho\Big((i\beta I + \tilde{W}_1)(\alpha I + \tilde{W}_1)^{-1}\big((\alpha+i)I + \tilde{W}_2\big)\big(i(\beta+1)I + \tilde{W}_2\big)^{-1}\Big)^2 \\ &\le \left\|(i\beta I + \tilde{W}_1)(\alpha I + \tilde{W}_1)^{-1}\big((\alpha+i)I + \tilde{W}_2\big)\big(i(\beta+1)I + \tilde{W}_2\big)^{-1}\right\|_2^2 \\ &\le \left\|(i\beta I + \tilde{W}_1)(\alpha I + \tilde{W}_1)^{-1}\right\|_2^2\left\|\big((\alpha+i)I + \tilde{W}_2\big)\big(i(\beta+1)I + \tilde{W}_2\big)^{-1}\right\|_2^2 \\ &= \max_{\lambda_i\in\sigma(\tilde{W}_1)}\frac{\beta^2+\lambda_i^2}{(\alpha+\lambda_i)^2}\max_{\mu_j\in\sigma(\tilde{W}_2)}\frac{(\alpha+\mu_j)^2+1}{(\beta+1)^2+\mu_j^2},\end{aligned} \tag{7}$$

where

$$\tilde{W}_1 = T^{-\frac{1}{2}}W_1T^{-\frac{1}{2}},$$

and

$$\tilde{W}_2 = T^{-\frac{1}{2}}W_2T^{-\frac{1}{2}}.$$

Clearly, the matrices $\tilde{W}_1$ and $\tilde{W}_2$ are SPD. Therefore, for every $\lambda_i \in \sigma(\tilde{W}_1)$ and $\mu_i \in \sigma(\tilde{W}_2)$ we have $\lambda_i > 0$ and $\mu_i > 0$. There exist $1 \le k \le n$ and $1 \le r \le n$ such that

$$\max_{\lambda_i\in\sigma(\tilde{W}_1)}\frac{\beta^2+\lambda_i^2}{(\alpha+\lambda_i)^2} = \frac{\beta^2+\lambda_k^2}{(\alpha+\lambda_k)^2},$$

and

$$\max_{\mu_j\in\sigma(\tilde{W}_2)}\frac{(\alpha+\mu_j)^2+1}{(\beta+1)^2+\mu_j^2} = \frac{(\alpha+\mu_r)^2+1}{(\beta+1)^2+\mu_r^2}.$$

Therefore, it follows from Eq. (7) that

$$\begin{aligned}\rho(G_{\alpha,\beta})^2 &\le \frac{\beta^2+\lambda_k^2}{(\alpha+\lambda_k)^2}\frac{(\alpha+\mu_r)^2+1}{(\beta+1)^2+\mu_r^2} \\ &= \frac{\beta^2+\lambda_k^2}{(\beta+1)^2+\mu_r^2}\frac{(\alpha+\mu_r)^2+1}{(\alpha+\lambda_k)^2} \\ &= f(\beta)g(\alpha),\end{aligned} \tag{8}$$

where

$$f(\beta) = \frac{\beta^2 + \lambda_k^2}{(\beta+1)^2 + \mu_r^2}, \tag{9}$$

and

$$0 < g(\alpha) = \frac{(\alpha + \mu_r)^2 + 1}{(\alpha + \lambda_k)^2}. \tag{10}$$

It is easy to see that $f(\beta) < 1$ is equivalent to

$$\beta > \frac{1}{2}\left(\lambda_k^2 - \mu_r^2 - 1\right).$$

Hence to achieve $f(\beta) < 1$ it is enough to have

$$\beta > \max\left\{0, \frac{1}{2}\left(\lambda_{\max}^2 - \mu_{\min}^2 - 1\right)\right\}. \tag{11}$$

On the other hand,

$$\lim_{\alpha \to +\infty} \frac{(\alpha + \mu_r)^2 + 1}{(\alpha + \lambda_k)^2} = 1.$$

Thus, for every $\epsilon > 0$, there exists a $P > 0$ such that for all $\alpha > P$,

$$\left| \frac{(\alpha + \mu_r)^2 + 1}{(\alpha + \lambda_k)^2} - 1 \right| < \epsilon.$$

So for $\alpha > P$, we have

$$0 \leq g(\alpha) = \frac{(\alpha + \mu_r)^2 + 1}{(\alpha + \lambda_k)^2} < 1 + \epsilon.$$

Therefore, from (8) we deduce that

$$\rho(G_{\alpha,\beta})^2 \leq f(\beta) g(\alpha) < f(\beta)(1 + \epsilon).$$

Hence to attain the convergence we need to have $f(\beta)(1 + \epsilon) < 1$ and this can be achieved if the parameter ϵ satisfies $\epsilon < \frac{1}{f(\beta)} - 1$. □

4 Inexact version of SNSS

To improve the computational efficiency of the SNSS iteration method, we can employ iteration methods for solving the two subsystems. This results in the inexact version of the SNSS (ISNSS) iteration method.

Set $\gamma^{(k)} = x^{(k+\frac{1}{2})} - x^{(k)}$. So, we have

$$x^{(k+\frac{1}{2})} = x^{(k)} + \gamma^{(k)}.$$

Substituting $x^{(k+\frac{1}{2})}$ in the first half-step of (6), gives

$$\begin{aligned}(\alpha T + W_1)\gamma^{(k)} &= \big(-W_1 + W_2 + iT\big)x^{(k)} - b\\ &= (W + iT)x^{(k)} - b\\ &= Ax^{(k)} - b =: -r^{(k)}.\end{aligned} \tag{12}$$

For computing $\gamma^{(k)}$, since the coefficient matrix of system (12) is SPD, we can use the CG method or its preconditioned version. Similarly, by setting $\gamma^{(k+\frac{1}{2})} = x^{(k+1)} - x^{(k+\frac{1}{2})}$ and substituting

$$x^{(k+1)} = \gamma^{\left(k+\frac{1}{2}\right)} + x^{\left(k+\frac{1}{2}\right)},$$

in the second half-step in (6), we get

$$\begin{aligned}\big(i(\beta + 1)T + W_2\big)\gamma^{(k+\frac{1}{2})} &= (W_1 - W_2 - iT)x^{(k+\frac{1}{2})} + b\\ &= (-W - iT)x^{\left(k+\frac{1}{2}\right)} + b\\ &= b - Ax^{\left(k+\frac{1}{2}\right)} =: r^{(k+\frac{1}{2})}.\end{aligned} \tag{13}$$

This system is equivalent to

$$\mathcal{A}u \equiv \begin{pmatrix} W_2 & -(\beta+1)T \\ (\beta+1)T & W_2 \end{pmatrix}\begin{pmatrix} \gamma_1^{(k+\frac{1}{2})} \\ \gamma_2^{(k+\frac{1}{2})} \end{pmatrix} = \begin{pmatrix} r_1^{(k+\frac{1}{2})} \\ r_2^{(k+\frac{1}{2})} \end{pmatrix} \equiv c, \tag{14}$$

where $\gamma_1^{(k+\frac{1}{2})} = \mathfrak{R}(\gamma^{(k+\frac{1}{2})})$, $\gamma_2^{(k+\frac{1}{2})} = \mathfrak{I}(\gamma^{(k+\frac{1}{2})})$, $r_1^{(k+\frac{1}{2})} = \mathfrak{R}(r^{(k+\frac{1}{2})})$, and $r_2^{(k+\frac{1}{2})} = \mathfrak{I}(r^{(k+\frac{1}{2})})$. To solve the above system, we can use the following PRESB matrix as a preconditioner (see [1, 3–5])

$$\mathcal{P}_{PRESB} = \begin{pmatrix} W_2 & -(\beta+1)T \\ (\beta+1)T & W_2 + 2(\beta+1)T \end{pmatrix}. \tag{15}$$

In [4], Axelsson et al. showed that the eigenvalues of the preconditioned matrix $\mathcal{P}_{PRESB}^{-1}\mathcal{A}$ are contained in the interval $[\frac{1}{2}, 1]$. So the Chebyshev acceleration method can be used for solving the system

$$\mathcal{P}_{PRESB}^{-1}\mathcal{A}u = \mathcal{P}_{PRESB}^{-1}c. \tag{16}$$

In the implementation of the preconditioner $\mathcal{P}$ in each iteration of the Chebyshev acceleration method a linear system of form

$$\mathcal{P}_{PRESB}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix},$$

should be solved, which is written as

$$\begin{cases} W_2 w_1 - (\beta+1)Tw_2 = r_1, \\ (\beta+1)Tw_1 + (W_2 + 2(\beta+1)T)w_2 = r_2. \end{cases} \tag{17}$$

By adding the second equation in (17) to the first one, gives

$$Qs = r_1 + r_2, \tag{18}$$

where $Q = W_2 + (\beta+1)T$ and $s = w_1 + w_2$. Since, the matrix Q is SPD, we can solve the system (18) exactly using the Cholesky factorization or inexactly using the PCG mehtod. On the other side, from the second equation in (17) we obtain

$$Qw_2 = r_2 - (\beta+1)Ts, \tag{19}$$

and it can be solved similar to Eq. (18). Note that we obtain the vectors w_2 and s from Eqs. (18) and (19), respectively. Finally, we can calculate the vector w_1 using $w_1 = s - w_2$. According to above notes, one can use the following steps for computing the vector $(w_1;w_2)$.

```
Solve $Qs = r_1 + r_2$ for $s$.
$r = r_2 - (\beta+1)Ts$.
$Qw_2 = r$ for $w_2$.
$w_1 = s - w_2$.
```

As we mentioned the eigenvalues of the preconditioned matrix $\mathcal{P}_{PRESB}^{-1}\mathcal{A}$ are contained in the interval $[\frac{1}{2}, 1]$. In this case, the lower and upper bounds for the eigenvalues $\mathcal{P}_{PRESB}^{-1}\mathcal{A}$ are $\eta = \frac{1}{2}$ and $\zeta = 1$, respectively. Therefore we can solve the inner systems by the Chebyshev acceleration method [24]. The Chebyshev acceleration algorithm for solving the preconditioned system (16) is as following.

Algorithm 1. The Chebyshev acceleration algorithm for $\mathcal{P}_{PRESB}^{-1}\mathcal{A}x = \mathcal{P}_{PRESB}^{-1}b$.

```
Choose an initial guess $x^{(0)}$ and $\theta := (\eta+\zeta)/2$ and $\delta := (\zeta-\eta)/2$;
$r^{(0)} := b - \mathcal{A}x^{(0)}$, $z^{(0)} := \mathcal{P}_{PRESB}^{-1} r^{(0)}$ and $\sigma_1 := \theta/\delta$;
$\rho^{(0)} := 1/\sigma_1$ and $d^{(0)} := z^0/\theta$;
For $k = 0, 1, \dots$, until convergence, Do
    $x^{(k+1)} = x^{(k)} + d^{(k)}$;
    $r^{(k+1)} = r^{(k)} - \mathcal{A}d^{(k)}$;
    $z^{(k+1)} = \mathcal{P}_{PRESB}^{-1} r^{(k+1)}$;
    $\rho^{(k+1)} = (2\sigma_1 - \rho^{(k)})^{-1}$;
    $d^{(k+1)} = \rho^{(k+1)}\rho^{(k)}d^{(k)} + \frac{2\rho^{(k+1)}}{\delta} z^{k+1}$;
EndDo
```

Now, the ISNSS iteration scheme is described in the following algorithm.

Algorithm 2. The ISNSS iteration method

```
1. Choose an initial guess $x^{(0)}$;
2. For $k = 0, 1, 2, \dots,$ until convergence, Do
3.     Compute $r^{(k)} = b - Ax^{(k)}$;
4.     Solve system (12) approximately for $\gamma^{(k)}$ using PCG;
5.     $x^{(k+\frac{1}{2})} = x^{(k)} + \gamma^{(k)}$;
6.     Compute $r^{(k+\frac{1}{2})} = b - Ax^{(k+\frac{1}{2})}$;
7.     Solve $\mathcal{A}u = c$ using Algor. 1 in with preconditioner $\mathcal{P}_{PRESB}$;
8.     $x^{(k+1)} = \gamma^{(k+\frac{1}{2})} + x^{(k+\frac{1}{2})}$;
9. EndDo
```

5 SNSS preconditioner and its implementation issues

In the SNSS iteration method, if we introduce

$$B_{\alpha,\beta} = \frac{1}{(\alpha - i\beta)}\big(\alpha T + W_1\big)T^{-1}\big(i(\beta+1)T + W_2\big),$$

and

$$C_{\alpha,\beta} = \frac{1}{(\alpha - i\beta)}(i\beta T + W_1)T^{-1}\big((\alpha + i)T + W_2\big), \tag{20}$$

then

$$A = B_{\alpha,\beta} - C_{\alpha,\beta}, \quad \text{and} \quad G_{\alpha,\beta} = B_{\alpha,\beta}^{-1}C_{\alpha,\beta}. \tag{21}$$

It follows from Eq. (21) that

$$B_{\alpha,\beta}^{-1}A = I - G_{\alpha,\beta}.$$

From the latter equation we conclude that if the SNSS method is convergent, then the eigenvalues of the matrix $B_{\alpha,\beta}^{-1}A$ are clustered in a circle with radius 1 centered at (1, 0). In this case, a Krylov subspace method like GMRES is quite suitable for solving the preconditioned system (see [10])

$$B_{\alpha,\beta}^{-1}Ax = B_{\alpha,\beta}^{-1}b. \tag{22}$$

In each iteration of a Krylov subspace method like GMRES for solving the system (22) a vector of the form

$$w = B_{\alpha,\beta}^{-1}z = (\alpha - i\beta)\big(i(\beta+1)T + W_2\big)^{-1}T(\alpha T + W_1)^{-1}z,$$

should be computed. To do so, we need solving a system with the coefficient matrix $\alpha T + W_1$ which can be accomplished exactly using the Cholesky factorization or inexactly using the CG method. We also need to solve a linear system with the coefficient matrix $i(\beta + 1)T + W_2$ which can be solved using the GMRES or the

Chebyshev acceleration methods with the PRESB preconditioner presented in the previous section.

Remark 1 The structure of subsystems in the MPNS method are the same as the ones in the SNSS iteration method. So the subsystems in the MPNS method, as well as the MPNS preconditioner, can be treated similar to the SNSS iteration.

6 Parameters estimation

In this section, using the idea of Ren and Cao [23] we present a strategy for estimating the iteration parameters α and β of the SNSS preconditioner. We rewrite the matrix in (20) as

$$\begin{aligned} C_{\alpha,\beta} =& \frac{-i^2}{(\alpha - i\beta)}\big(i\beta T + W_1\big)T^{-1}\big((\alpha + i)T + W_2\big) \\ =& \frac{1}{(i\beta - \alpha)}(-\beta T + iW_1)T^{-1}\big(\alpha iT + iW_2 - T\big), \end{aligned} \tag{23}$$

and define the function ψ as

$$\begin{aligned} \psi(\alpha,\beta) =& \big(-\beta\|T\|_F + \|iW_1\|_F\big)\|T^{-1}\|_F\big(\alpha\|iT\|_F + \|iW_2\|_F - \|T\|_F\big) \\ =& \big(-\beta\|T\|_F + \|W_1\|_F\big)\|T^{-1}\|_F\big(\alpha\|T\|_F + \|W_2\|_F - \|T\|_F\big). \end{aligned}$$

To estimate the iteration parameters α and β, we set

$$-\beta\|T\|_F + \|W_1\|_F = 0,$$

and

$$\alpha\|T\|_F + \|W_2\|_F - \|T\|_F = 0.$$

Since the matrix T is SPD, then $\|T\|_F$, $\|T^{-1}\|_F \neq 0$. Then we obtain the following estimation formulas for the iteration parameters α and β of the SNSS preconditioner

$$\beta_{est} = \frac{\|W_1\|_F}{\|T\|_F}, \tag{24}$$

$$\alpha_{est} = 1 - \frac{\|W_2\|_F}{\|T\|_F}. \tag{25}$$

Note that if $\alpha_{est} < 0$, then we apply the above strategy for the function $-\psi$. In this case, we get

$$\alpha_{est} = \frac{\|W_2\|_F}{\|T\|_F} - 1.$$

Summarizing the above discussion results in the the following estimation parameters:

$$(\alpha_{est}, \beta_{est}) = \left(\left| 1 - \frac{\|W_2\|_F}{\|T\|_F} \right|, \frac{\|W_1\|_F}{\|T\|_F} \right). \tag{26}$$

In the next section we see that the above strategy gives often suitable results.

7 Numerical experiments

In this section, we consider the following two examples for our numerical tests.

Example 1 We consider the complex symmetric linear system of equations [7, 8, 35]

$$\left[\left(-\omega^2 M + K\right) + i\omega C\right]x = b, \tag{27}$$

where M and K are the inertia and stiffness matrices, respectively. We take $C = \omega C_V + C_H$ where C_V and C_H are the viscous and hysteretic damping matrices, respectively; and ω is the driving circular frequency. In our numerical experiments, we set $M = I$, $C_V = 5M$ and $C_H = \mu K$ with a damping coefficient $\mu = 0.02$ and K the five-point centered difference matrix approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0,1] \times [0,1]$ with the mesh size $h = 1/(m+1)$. In this case, we have

$$K = \left(I \otimes V_m + V_m \otimes I\right) \in \mathbb{R}^{n \times n},$$

with $V_m = h^{-2}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$. Hence, the total number of variables is $n = m^2$. In addition, the right-hand side vector f is to be adjusted such that $b = (1+i)Ae$ where $e = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. Note that in both of the SNSS and MPNS methods, we consider $W_1 = \omega^2 M$, $W_2 = K$ and $T = \omega C$.

Example 2 We consider the complex Helmholtz equation in 2-D of the form [7, 8, 30, 35]

$$\begin{cases} -\Delta u + \sigma_1 u + i\sigma_2 u &= f, \quad in \quad \Omega, \\ \qquad\qquad\qquad\quad u &= g, \quad on \quad \partial\Omega, \end{cases} \tag{28}$$

where

$$\Delta = \sum_{j=1}^{2} \frac{\partial^2}{\partial x_j^2},$$

$\sigma_1 \in \mathbb{R}$, $\sigma_2 \geq 0$ and $\Omega = [0,1]^2$, $i = \sqrt{-1}$. The discretization of the equation above in 2-D, using the second order central difference scheme on an $(m+2) \times (m+2)$ grid of Ω with mesh-size $h = 1/(m+1)$ leads to a system of linear equations with the coefficient matrix $A = W + iT \in \mathbb{C}^{n \times n}$, such that $n = m^2$, and

$$W = K + \sigma_1 h^2 (I_m \otimes I_m) \quad \text{and} \quad T = \sigma_2 h^2 (I_m \otimes I_m),$$

with $K = I_m \otimes V_m + V_m \otimes I_m$ and $V_m = tridiag(-1, 2, -1) \in \mathbb{R}^{m \times m}$. In addition, the right-hand side vector b is to be adjusted such that $b = (1 + i)Ae$ where $e = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. In our numerical experiments, we set $(\sigma_1, \sigma_2) = (-100, 100)$ and $(\sigma_1, \sigma_2) = (-1000, 10)$. Note that in both of the SNSS and MPNS methods, we consider $W_1 = -\sigma_1 h^2 (I_m \otimes I_m)$, $W_2 = K$.

We present our numerical results in two parts. In the first part, we compare the numerical results of the ISNSS with the incomplete version of MPNS (IMPNS) method. In both of the ISNSS and IMPNS methods, the first subsystem is solved using the preconditioned CG (PCG) method incorporated with the incomplete Cholesky factorization with dropping tolerance 10^{-2} as a preconditioner. In the Matlab notation the following command can be used for computing the incomplete Cholesky factor

$$\texttt{L} = \texttt{ichol(Z,struct('type','ict','droptol',1e}-\texttt{2));}$$

where Z is a given SPD matrix. The second subsystem is solved by the Chebyshev acceleration method in conjunction with the PRESB preconditioner. Note that in all of the methods we solve the subsystems (18) and (19) exactly using the sparse Cholesky factorization incorporated with the symmetric approximate minimum degree permutation. To do so, the `symamd.m` command of Matlab is applied. We use a null vector as an initial guess. The outer iteration is stopped as soon as the residual 2-norm is reduced by a factor of 10^6 and the the inner iteration by a factor of 10^2. The maximum number of inner and outer iterations are set to be 10000, respectively. In the tables, a dagger (†) shows that the method has not converged in 10000 iterations. All of the numerical experiments are performed in Matlab R2018b by using a Laptop with 2.50 GHz central processing unit (Intel(R) Core(TM) i5-7200U), 6 GB RAM and Windows 10.

Numerical results are presented for Examples 1 and 2 in Tables 1, 2, 3 and 4. In these tables we report the number of outer iterations ("Iters"), elapsed CPU time in seconds ("CPU") and the following values

$$R_k = \frac{\|b - Ax^{(k)}\|_2}{\|b\|_2}, \quad E_k = \frac{\|x^* - x^{(k)}\|_2}{\|x^*\|_2},$$

to demonstrate the accuracy of the computed solutions, where $x^{(k)}$ and x^* are the computed solution at iteration k and the exact solution, respectively. We also present the minimum eigenvalue of the matrices W and T to see that whether the matrix is indefinite or not. For both the ISNSS and IMPNS methods the the optimal values of the parameters (the one with minimum number of iterations) are computed experimentally.

As the numerical results show the ISNSS outperforms the IMPNS method from both the CPU time and the iterations. We observe that ISNSS iteration method is h-independent. However, by increasing the parameter ω, the number of iterations is increased moderately.

Table 1 Numerical results of IMPNS and ISNSS for Example 1 and $m = 128$ ($n = 16384$)

Method		$\omega = 1$	$\omega = 3$	$\omega = 5$	$\omega = 7$	$\omega = 9$
	$\lambda_{\min}(W)$	18.738233	10.738233	– 5.261766	– 29.261766	– 61.261766
	$\lambda_{\min}(T)$	5.394766	46.184293	126.973823	247.763352	408.552881
	Iter	2999	1851	890	514	329
GMRES(10)	CPU	6.70	4.14	2.05	1.22	0.78
	R_k	9.99e–07	9.97e–07	9.97e–07	9.99e–07	9.84e–07
	E_k	1.51e–04	6.13e–05	2.20e–05	1.15e–05	6.47e–06
	α_{opt}	300	650	1100	1700	2290
IMPNS	Iter	117	119	112	106	96
	CPU	2.60	2.51	2.34	2.15	1.85
	R_k	9.48e–07	9.94e–07	9.40e–07	9.43e07	9.48e–07
	E_k	1.20e–04	4.54e–05	1.29e–05	7.11e–06	5.45e–06
	α_{opt}	30	5	10	15	10
	β_{opt}	0.001	0.1	0.1	0.1	0.1
ISNSS	Iter	3	6	7	7	8
	CPU	0.22	0.28	0.29	0.31	0.34
	R_k	8.39e–07	6.85e–07	6.12e–07	7.91e07	2.54e–07
	E_k	1.11e–04	4.28e–05	1.43e–05	9.46e-06	1.86e–06

Table 2 Numerical results of IMPNS and ISNSS for Example 1 and $m = 256$ ($n = 65536$)

Method		$\omega = 1$	$\omega = 3$	$\omega = 5$	$\omega = 7$	$\omega = 9$
	$\lambda_{\min}(W)$	18.738963	10.738963	– 5.261036	– 29.261036	– 61.261036
	$\lambda_{\min}(T)$	5.394779	46.184337	126.973896	247.763454	408.553013
	Iter	†	6423	3068	1740	1119
GMRES(10)	CPU	109.24	65.90	31.58	18.35	11.52
	R_k	1.30e–06	9.99e–07	9.98e–07	9.97e–07	9.94e–07
	E_k	5.51e–04	1.75e-04	6.48e–05	3.25e–05	1.96e–05
	α_{opt}	720	1400	2410	3570	4780
IMPNS	Iter	240	225	203	182	174
	CPU	28.52	25.55	20.18	18.04	17.04
	R_k	9.96e–07	9.86e–07	9.78e–07	9.92e07	9.65e–07
	E_k	3.30e–04	1.09e–04	4.60e–05	2.50e–05	7.88e–06
	α_{opt}	30	20	20	10	10
	β_{opt}	0.001	0.1	0.1	0.1	0.1
ISNSS	Iter	3	5	7	7	7
	CPU	0.79	0.99	1.17	1.27	1.29
	R_k	4.71e–07	9.90e–07	2.07e–07	6.48e07	4.55e–07
	E_k	1.07e–04	1.73e–04	1.36e–05	2.17e–05	9.21e–06

Table 3 Numerical results of IMPNS, ISNSS for Example 2 with $\sigma_1 = -100$ and $\sigma_2 = 100$

Method		$m = 128$	$m = 192$	$m = 256$
	$\lambda_{\min}(W)$	– 0.004823	– 0.002154	– 0.001215
	$\lambda_{\min}(T)$	0.006009	0.002684	0.001514
	Iter	1647	3406	5739
GMRES(10)	CPU	4.25	17.37	62.93
	R_k	9.96e–07	9.99e–07	9.99e–07
	E_k	2.20e–05	4.13e–05	6.36e–05
	α_{opt}	0.067	0.048	0.048
IMPNS	Iter	603	629	975
	CPU	7.89	16.71	54.49
	R_k	9.93e–07	9.56e–07	9.91e–07
	E_k	2.06e–05	1.45e–07	4.52e–05
	α_{opt}	10	30	40
	β_{opt}	1	1	1
ISNSS	Iter	26	27	27
	CPU	0.67	1.58	2.80
	R_k	7.66e–07	9.41e–07	8.72e–07
	E_k	1.77e–05	3.96e–05	5.63e–05

Table 4 Numerical results of IMPNS, ISNSS for Example 2 with $\sigma_1 = -1000$ and $\sigma_2 = 10$

Method		$m = 128$	$m = 192$	$m = 256$
	$\lambda_{\min}(W)$	– 0.058906	– 0.026316	– 0.014841
	$\lambda_{\min}(T)$	0.000609	0.000268	0.0001514
	Iter	18400	35990	67438
GMRES(10)	CPU	41.74	172.90	727.47
	R_k	1.00e–06	1.00e–06	1.00e–06
	E_k	4.04e–05	2.88e-05	3.97e—05
	α_{opt}	–	–	–
IMPNS	Iter	†	†	†
	CPU	–	–	–
	R_k	–	–	–
	E_k	–	–	–
	α_{opt}	111.5	111.5	190
	β_{opt}	104.5	104.5	130
ISNSS	Iter	1200	1099	1738
	CPU	23.75	48.63	158.07
	R_k	9.96e–07	9.98e–07	9.98e–07
	E_k	6.29e–05	1.07e–04	6.92e–05

For the second set of our numerical experiments, we present the numerical results for solving Examples 1 and 2 by means of the flexible version of GMRES (FGMRES) [24, 26] incorporated with the SNSS and MPNS preconditioners (hereafter, are denoted by FGMRES-SNSS and FGMRES-MPNS, respectively). We use a zero

Table 5 Numerical results of FGMRES, and FGMRES with the SNSS and MPNS preconditioners for Example 1 and $m = 128$ ($n = 16384$)

Method		$\omega = 1$	$\omega = 3$	$\omega = 5$	$\omega = 7$	$\omega = 9$
	$\lambda_{\min}(W)$	18.738233	10.738233	– 5.261766	– 29.261766	– 61.261766
	$\lambda_{\min}(T)$	5.394766	46.184293	126.973823	247.763352	408.552881
	Iter	303	308	301	290	279
FGMRES	CPU	7.82	8.05	7.70	7.19	6.81
	R_k	9.45e–07	8.80e–07	7.48e–07	7.09e–07	5.95e–07
	E_k	1.54e–05	8.15e–06	3.63e–06	2.44e–06	1.77e–06
	α_{opt}	300	800	1000	2000	3000
FGMRES-MPNS	Iter	23	26	31	34	38
	CPU	1.18	1.21	1.36	1.34	1.41
	R_k	7.55e–07	7.30e–07	7.07e–07	8.12e–07	6.98e–07
	E_k	1.70e–05	6.24e–06	7.34e–06	6.19e–06	4.38e–06
	α_{est}	48.8499	15.5174	8.8515	5.9951	4.4086
	β_{est}	0.0007	0.0020	0.0033	0.0046	0.0059
FGMRES-SNSS	Iter	5	7	7	7	8
	CPU	0.75	0.80	0.80	0.87	0.85
	R_k	6.69e–07	8.87e–08	7.56e–07	7.66e–07	3.94e–07
	E_k	7.53e–05	1.16e–06	1.39e–05	6.18e–06	1.58e–06

vector as an initial guess and the iteration of FGMRES is stopped as soon as the residual 2-norm is reduced by a factor of 10^6. The maximum number of iterations is set to be 10,000. In the implementation of the preconditioners the subsystems are solved as mentioned in the first set of the numerical experiments. Numerical results are presented in Tables 5, 6, 7 and 8. To demonstrate the efficiency of the preconditioners we also present the numerical results of FGMRES without preconditioning. We use the optimal value of the parameter in the MPNS method, however the estimated parameters given in Eq. (26) are used in the SNSS method.

As we see both of the preconditioners expedite greatly the convergence speed of the FGMRES method. Numerical results show that the SNSS preconditioner is more efficient than the MPNS preconditioner in terms of both the number of iterations and the elapsed CPU time. We also see that as the mesh size is refined the number of iterations of FGMRES-SNSS remains almost constant. However, the number of iterations increased moderately as the value of the parameter ω is increased.

8 Conclusion

In this paper, we have presented the symmetric positive definite and negative stable splitting (SNSS) method to solve the complex symmetric indefinite linear systems. Theoretical analysis show that the SNSS iteration method is convergent under suitable condition. Numerical results that the ISNSS-Cheb and SNSS-FGMRES methods

Table 6 Numerical results of FGMRES, and FGMRES with the SNSS and MPNS preconditioners for Example 1 and $m = 256$ ($n = 65536$)

Method		$\omega = 1$	$\omega = 3$	$\omega = 5$	$\omega = 7$	$\omega = 9$
	$\lambda_{\min}(W)$	18.738963	10.738963	– 5.261036	– 29.261036	– 61.261036
	$\lambda_{\min}(T)$	5.394779	46.184337	126.973896	247.763454	408.553013
	Iter	568	584	570	553	523
FGMRES	CPU	238.97	253.92	239.52	226.52	199.97
	R_k	9.77e–07	9.05e–07	7.40e–07	6.83e–07	6.47e–07
	E_k	5.40e–05	3.19e–05	1.23e–05	7.46e–06	5.12e–06
	α_{opt}	900	1800	2200	3000	4000
FGMRES-MPNS	Iter	31	35	40	44	48
	CPU	7.05	7.31	8.44	8.75	9.32
	R_k	7.71e–07	9.68e–07	8.50e–07	8.76e–07	8.76e–07
	E_k	1.73e–05	1.84e–05	1.50e–05	1.13e–05	6.74e–06
	α_{est}	48.9621	15.6289	8.9622	6.1052	4.5179
	β_{est}	0.0002	0.0005	0.0008	0.0012	0.0015
FGMRES-SNSS	Iter	5	7	7	7	7
	CPU	3.15	3.29	3.27	3.17	3.31
	R_k	5.50e–07	1.04e–07	3.66e–07	3.83e–07	5.26e–07
	E_k	1.78e–04	6.85e–06	1.82e–05	1.05e–05	6.62e–06

Table 7 Numerical results of FGMRES, and FGMRES with the SNSS and MPNS preconditioners for Example 2 with $\sigma_1 = -100$ and $\sigma_2 = 100$

Method		$m = 128$	$m = 192$	$m = 256$
	$\lambda_{\min}(W)$	– 0.004823	– 0.002154	– 0.001215
	$\lambda_{\min}(T)$	0.006009	0.002684	0.001514
	Iter	337	493	648
FGMRES	CPU	9.43	40.88	297.88
	R_k	8.93e–07	9.19e–07	9.24e–07
	E_k	3.95e–06	8.44e–06	1.43e–05
	α_{opt}	0.005	0.003	0.003
FGMRES-MPNS	Iter	21	19	18
	CPU	1.22	2.70	5.40
	R_k	8.58e–07	9.30e–07	4.50e–07
	E_k	1.74e–05	4.02e–05	1.77e–05
	α_{est}	742.6265	1663.9581	2951.6470
	β_{est}	1	1	1
FGMRES-SNSS	Iter	13	12	12
	CPU	0.90	1.71	3.37
	R_k	2.62e–07	7.03e–07	4.62e–07
	E_k	3.83e–06	2.48e–05	2.49e–05

Table 8 Numerical results of FGMRES, and FGMRES with the SNSS and MPNS preconditioners for Example 2 with $\sigma_1 = -1000$ and $\sigma_2 = 10$

Method		$m = 128$	$m = 192$	$m = 256$
FGMRES	$\lambda_{\min}(W)$	– 0.058906	– 0.026316	– 0.014841
	$\lambda_{\min}(T)$	0.000609	0.000268	0.0001514
	Iter	714	1064	7925
	CPU	39.18	161.33	5634.40
	R_k	9.85e–07	9.67e–07	9.71e–07
	E_k	4.22e–06	2.02e–05	1.90e–04
	α_{opt}	0.002	0.002	0.0021
FGMRES-MPNS	Iter	216	176	178
	CPU	9.58	17.93	54.27
	R_k	8.23e–07	8.31e–07	8.87e–07
	E_k	1.22e–04	2.11e–04	4.73e–04
	α_{est}	7435.2650	16648.5808	29525.4702
	β_{est}	100	100	100
FGMRES-SNSS	Iter	116	107	104
	CPU	3.78	8.05	19.60
	R_k	4.89e–07	7.58e–07	8.06e–07
	E_k	2.25e–06	1.01e–05	1.10e–05

are efficient when the real part of the coefficient matrix of the systems is symmetric indefinite.

Acknowledgements The authors would like to thank the referees for their careful reading of the paper and giving several valuable comments and suggestions.

References

1. Axelsson, O., Boyanova, P., Kronbichler, M., Neytcheva, M., Wu, X.: Numerical and computational efficiency of solvers for two-phase problems. Comput. Math. Appl. **65**, 301–314 (2013)
2. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**, 197–218 (2000)
3. Axelsson, O., Neytcheva, M., Ahmad, B.: A comparison of iterative methods to solve complex valued linear algebraic systems. Numer. Algorithms **66**, 811–841 (2014)
4. Axelsson, O., Neytcheva, M., Liang, Z.-Z.: Parallel solution methods and preconditioners for evolution equations. Math. Model. Anal. **23**, 287–308 (2018)
5. Axelsson, O., Salkuyeh, D.K.: A new version of a preconditioning method for certain two-by-two block matrices with square blocks. BIT Numer. Math. **59**, 321–342 (2018)
6. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)
7. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
8. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algorithms **56**, 297–317 (2011)
9. Bayliss, A., Goldstein, C.I., Turkel, E.: An iterative method for Helmholtz equation. J. Comput. Phys. **49**, 443–457 (1983)
10. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 418–477 (2002)

11. Bunse-Gerstner, A., Stöver, R.: On a conjugate gradient-type method for solving complex symmetric linear systems. Linear Algbra Appl. **287**, 105–123 (1999)
12. Cui, L.B., Zhang, X.Q., Zheng, Y.T.: A preconditioner based on a splitting-type iteration method for solving complex symmetric indefinite linear systems. Jpn. J. Ind. Appl. Math. (2021). https://doi.org/10.1007/s13160-021-00471-1
13. Edalatpour, V., Hezari, D., Salkuyeh, D.K.: Two efficient inexact algorithms for a class of large sparse complex linear systems. Mediterr. J. Math. **13**, 2301–2318 (2016)
14. Edalatpour, V., Hezari, D., Salkuyeh, D.K.: Accelerated generalized SOR method for a class of complex systems of linear equations. Math. Commun. **20**, 37–52 (2015)
15. Feriani, A., Perotti, F., Simoncini, V.: Iterative system solvers for the frequency analysis of linear mechanical systems. Comput. Methods Appl. Mech. Eng. **190**, 1719–1739 (2000)
16. Freund, R.W.: Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. SIAM J. Sci. Stat. Comput. **13**(1), 425–448 (1992)
17. Guo, C.-H.: Incomplete block factorization preconditioner for linear systems arising in the numerical solution of the Helmholtz equation. Appl. Numer. Math. **19**, 495–508 (1996)
18. Hezari, D., Salkuyeh, D.K., Edalatpour, V.: Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations. Numer. Linear Algebra Appl. **22**, 761–776 (2015)
19. Hezari, D., Salkuyeh, D.K., Edalatpour, V.: A new iterative method for solving a class of complex symmetric system of linear equations. Numer. Algorithms **73**, 927–955 (2016)
20. Huang, Z.-G.: A new double-step splitting iteration method for certain block two-by-two linear systems. Comput. Appl. Math. **39**, 193 (2020)
21. Huang, Z.-G.: Modified two-step scale-splitting iteration method for solving complex symmetric linear systems. Comput. Appl. Math. **40**, 122 (2021)
22. Li, C.-X., Wu, S.-L.: Modified complex-symmetric and skew-Hermitian splitting iteration method for a class of complex-symmetric indefinite linear systems. Numer. Algorithms **76**, 93–107 (2017)
23. Ren, Z.-R., Cao, Y.: An alternating positive-semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models. IMA J. Numer. Anal. **36**, 922–946 (2016)
24. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
25. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
26. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM J. Sci. Comput. **14**, 461–469 (1993)
27. Salkuyeh, D.K.: Two-step scale-splitting method for solving complex symmetric system of linear equations, math. NA. (2017) arXiv:1705.02468
28. Salkuyeh, D.K., Hezari, D., Edalatpour, V.: Generalized SOR iterative method for a class of complex symmetric linear system of equations. Int. J. Comput. Math. **92**, 802–815 (2015)
29. Salkuyeh, D.K., Siahkolaei, T.S.: Two-parameter TSCSP method for solving complex symmetric system of linear equations. Calcolo **55**, 8 (2018)
30. Siahkolaei, T.S., Salkuyeh, D.K.: A new double-step method for solving complex Helmholtz equation. Hacet. J. Math. Stat. **49**, 1245–1260 (2020)
31. Sogabe, T., Zhang, S.-L.: A COCR method for solving complex symmetric linear systems. J. Comput. Appl. Math. **199**, 297–303 (2007)
32. Van der Vorst, H.A., Melissen, J.B.M.: A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex. IEEE Trans. Mag. **26**, 706–708 (1990)
33. Wu, S.-L., Huang, T.-Z., Li, L., Xiong, L.-L.: Positive stable preconditioners for symmetric indefinite linear systems arising from Helmholtz equations. Phys. Lett. A. **373**, 2401–2407 (2009)
34. Wu, S.-L., Li, C.-X.: A modified SSOR preconditioning strategy for Helmholtz equations. J. Appl. Math. **2012**, 365124 (2012)
35. Wu, S.-L., Li, C.-X.: A splitting method for complex symmetric indefinite linear system. J. Comput. Appl. Math. **313**, 343–354 (2017)
36. Xie, X., Li, H.B.: On preconditioned Euler-extrapolated single-step Hermitian and skew-Hermitian splitting method for complex symmetric linear systems. Jpn .J. Ind. Appl. Math. **38**, 503–518 (2021)
37. Zhang, G.F., Zheng, Z.: A parameterized splitting iteration method for complex symmetric linear systems. Jpn .J. Ind. Appl. Math. **31**, 265–278 (2014)