IMA Journal of Numerical Analysis (2018) **00**, 1–36
doi:10.1093/imanum/dry046

Regularized HSS iteration methods for stabilized saddle-point problems

Zhong-Zhi Bai
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, P. R. China
bzz@lsec.cc.ac.cn

[Received on 18 August 2017; revised on 14 June 2018]

We extend the regularized Hermitian and skew-Hermitian splitting (RHSS) iteration methods for standard saddle-point problems to stabilized saddle-point problems and establish the corresponding unconditional convergence theory for the resulting methods. Besides being used as stationary iterative solvers, this class of RHSS methods can also be used as preconditioners for Krylov subspace methods. It is shown that the eigenvalues of the corresponding preconditioned matrix are clustered at a small number of points in the interval (0, 2) when the iteration parameter is close to 0 and, furthermore, they can be clustered near 0 and 2 when the regularization matrix is appropriately chosen. Numerical results on stabilized saddle-point problems arising from finite element discretizations of an optimal boundary control problem and of a Cahn–Hilliard image inpainting problem, as well as from the Gauss–Newton linearization of a nonlinear image restoration problem, show that the RHSS iteration method significantly outperforms the Hermitian and skew-Hermitian splitting iteration method in iteration counts and computing times when they are used either as linear iterative solvers or as matrix splitting preconditioners for Krylov subspace methods, and optimal convergence behavior can be achieved when using inexact variants of the proposed RHSS preconditioners.

Keywords: stabilized saddle-point problem; Hermitian and skew-Hermitian splitting; stationary iteration method; inexact implementation; preconditioning; convergence.

1. Introduction

Consider the stabilized saddle-point problem

$$Ax \equiv \begin{pmatrix} B & E \\ -E^* & C \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b, \tag{1.1}$$

where $B \in \mathbb{C}^{p\times p}$ is a Hermitian positive definite matrix, $C \in \mathbb{C}^{q\times q}$ is a Hermitian positive semidefinite matrix, $E \in \mathbb{C}^{p\times q}$ is a rectangular matrix, $E^* \in \mathbb{C}^{q\times p}$ is the conjugate transpose of E and $f \in \mathbb{C}^p$, $g \in \mathbb{C}^q$, with p and q being two given positive integers such that $p \geq q$. Under these assumptions if, in addition, the null spaces of the matrices C and E do not overlap, i.e., $\mathrm{null}(C) \cap \mathrm{null}(E) = \{0\}$ then, in accordance with Wathen & Silvester (1993), we know that the stabilized saddle-point problem (1.1) admits a unique solution; see also Silvester & Wathen (1994), Benzi *et al.* (2005), Wathen (2015) and the references therein. Here and in the sequel we indicate by $(\cdot)^*$ the conjugate transpose of either a vector or a matrix of suitable dimension and we let $n = p + q$. If, in particular, $C = O$, i.e., the zero matrix, then the system of linear equations (1.1) is termed a standard saddle-point problem. This class of stabilized saddle-point problems has plentiful background in scientific computing and engineering applications. For example, it frequently arises in stabilized mixed finite element methods, regularized and weighted least-squares problems and certain interior point methods in optimization. For more details we refer to Fortin & Glowinski (1983), Silvester & Kechkar (1990), Brezzi & Fortin (1991), Elman *et al.* (2002, 2014), Benzi *et al.* (2005), Bai (2006), Wathen (2015) and the references therein.

Because the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$ in (1.1) admits the *Hermitian and skew-Hermitian* (HS) splitting

$$A = \begin{pmatrix} B & O \\ O & C \end{pmatrix} + \begin{pmatrix} O & E \\ -E^* & O \end{pmatrix} = H + S, \tag{1.2}$$

Benzi & Golub (2004) proposed the *Hermitian and skew-Hermitian splitting* (HSS) iteration method to solve the stabilized saddle-point problem (1.1). This stationary iteration method is a straightforward generalization of the HSS iteration method initially proposed in Bai *et al.* (2003) for solving general non-Hermitian positive definite linear systems, with the saddle-point structure and the non-Hermitian positive semidefiniteness of the linear system (1.1) being paid particular attention. Just like the HSS iteration method in Bai *et al.* (2003) it also converges unconditionally to the unique solution of the stabilized saddle-point problem (1.1); see also Bai *et al.* (2004), Simoncini & Benzi (2004), Bai & Golub (2007), Bai *et al.* (2007) and the references therein. Recently, specifically focused on the standard saddle-point problem, i.e., $C = O$ in (1.1), Bai & Benzi (2017) proposed the *regularized HSS* (RHSS) iteration method by introducing an extra Hermitian positive semidefinite matrix, called the *regularization matrix*, in the HS splitting in (1.2). The RHSS iteration method is a valuable development and quality improvement of the HSS iteration method discussed in Benzi & Golub (2004). In Bai & Benzi (2017) it was proved that the RHSS iteration method converges unconditionally to the unique solution of the standard saddle-point problem and that the eigenvalues of the RHSS-preconditioned matrix are clustered at 0_+ and 2_- (i.e., to the right of 0 and to the left of 2) when the iteration parameter α is close to 0, which implies possibly fast convergence of the corresponding preconditioned Krylov subspace methods; see, e.g., Eisenstat *et al.* (1983), Saad & Schultz (1986), Greenbaum (1997), Saad (2003) and Bai (2015). With numerical experiments it was shown in Bai & Benzi (2017) that the RHSS and its inexact variant, the *inexact RHSS* (IRHSS), can be more efficient and robust than the HSS and the *inexact HSS* (IHSS), respectively, both as stationary linear solvers and as matrix splitting preconditioners for (flexible) GMRES methods (Saad & Schultz, 1986; Saad, 2003) when they are employed to solve certain types of standard saddle-point problems.

In this paper we are going to extend the RHSS iteration method for the standard saddle-point problem to the stabilized saddle-point problem (1.1). At each step, owing to the specific regularization strategy and to the appropriate normalization parameter used, we need to solve only two linear subsystems in the RHSS iteration method, which is distinct from the HSS iteration method in Benzi & Golub (2004) which requires solving three linear subsystems. This significantly reduces the computational complexity and executing time and, hence, considerably improves the computational efficiency of the RHSS iteration method. In addition, the regularization and parametrization strategies definitely improve the conditioning of the inner linear subsystems, so that the corresponding IRHSS preconditioner can be expected to be more effective and robust when applied to compute the solution of the stabilized saddle-point problem (1.1).

In theory, we prove that the RHSS iteration method converges unconditionally to the unique solution of the stabilized saddle-point problem (1.1) and that the eigenvalues of the RHSS-preconditioned matrix are clustered at 0_+, 2_- and a small number ($< q$) of points in the interval (0, 2) when the iteration parameter α is close to 0. In addition, these interior clustering points can be eliminated by appropriately choosing the regularization matrix. And in computations we show that the RHSS iteration method and the RHSS-preconditioned GMRES method outperform the HSS iteration method and the HSS-preconditioned GMRES method in terms of both iteration counts and computing times and that the IRHSS iteration method and the IRHSS-preconditioned (flexible) GMRES method have significantly higher computing efficiency than their exact counterparts in terms of computing times. Moreover, besides its convergence behavior being independent of the problem size, the IRHSS-preconditioned flexible GMRES method is competitive with and even superior to the preconditioned MINRES method implemented with certain optimal block-diagonal preconditioners, as well as the preconditioned GMRES method implemented with certain best block-triangular preconditioners, in the aspects of iteration counts and computing times. Hence, with the experiments we show that the RHSS and the IRHSS methods can be efficient and robust for solving certain types of the stabilized saddle-point problem (1.1) when they are used to precondition Krylov subspace methods such as the (flexible) GMRES.

The organization of the paper is as follows. In Section 2 we present the algorithmic description of the RHSS iteration method. In Section 3 we prove the unconditional convergence of the RHSS iteration method and analyse clustering properties of the eigenvalues of the RHSS-preconditioned matrix. Numerical results are given in Section 4. Finally, in Section 5 we end the paper with brief conclusions and remarks.

2. The RHSS iteration method

In this section we derive the RHSS iteration method for solving the stabilized saddle-point problem (1.1) and the RHSS preconditioning matrix for transforming the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$. Also, we briefly discuss inexact implementations of the stationary and the preconditioned iteration methods.

Let $Q \in \mathbb{C}^{q\times q}$ be a given Hermitian positive semidefinite matrix and ω be a prescribed non-negative parameter. Then we can split the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$ in (1.1), obtaining the *regularized Hermitian and skew-Hermitian* (RHS) splitting:

$$\begin{aligned} A &= \begin{pmatrix} B & O \\ O & Q+\omega C \end{pmatrix} + \begin{pmatrix} O & E \\ -E^* & -Q+(1-\omega)C \end{pmatrix} = H_+(\omega) + S_-(\omega) \\ &= \begin{pmatrix} O & E \\ -E^* & Q+(1+\omega)C \end{pmatrix} + \begin{pmatrix} B & O \\ O & -Q-\omega C \end{pmatrix} = S_+(\omega) + H_-(\omega). \end{aligned} \tag{2.1}$$

Here the matrix Q plays a regularization role in the HS splitting in (1.2), so it is called a regularization matrix. Moreover, we call ω a normalization parameter. For $C = O$ the RHS splitting in (2.1) reduces to the RHS splitting in Bai & Benzi (2017) and if, in addition, $Q = O$ then it becomes the HS splitting in (1.2), for the standard saddle-point matrix. With the aid of a shift constant $\alpha > 0$, called the iteration parameter, the RHS splitting in (2.1) of the matrix A naturally leads to equivalent reformulations of the stabilized saddle-point problem (1.1) into two systems of fixed-point equations:

$$\begin{cases} \big(\alpha I + H_+(\omega)\big)x = \big(\alpha I - S_-(\omega)\big)x + b, \\ \big(\alpha I + S_+(\omega)\big)x = \big(\alpha I - H_-(\omega)\big)x + b. \end{cases}$$

By iterating alternatively between these two fixed-point systems as

$$\big(\alpha I + H_{+}(\omega)\big)x^{(k+1/2)} = \big(\alpha I - S_{-}(\omega)\big)x^{(k)} + b \tag{2.2}$$

and

$$\big(\alpha I + S_{+}(\omega)\big)x^{(k+1)} = \big(\alpha I - H_{-}(\omega)\big)x^{(k+1/2)} + b, \tag{2.3}$$

or in their blockwise forms

$$\begin{pmatrix} \alpha I + B & O \\ O & \alpha I + Q + \omega C \end{pmatrix} \begin{pmatrix} y^{(k+1/2)} \\ z^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} \alpha I & -E \\ E^{*} & \alpha I + Q - (1-\omega)C \end{pmatrix} \begin{pmatrix} y^{(k)} \\ z^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix}$$

and

$$\begin{pmatrix} \alpha I & E \\ -E^{*} & \alpha I + Q + (1+\omega)C \end{pmatrix} \begin{pmatrix} y^{(k+1)} \\ z^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha I - B & O \\ O & \alpha I + Q + \omega C \end{pmatrix} \begin{pmatrix} y^{(k+1/2)} \\ z^{(k+1/2)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix},$$

we obtain the RHSS iteration method for solving the stabilized saddle-point problem (1.1) as follows.

The RHSS iteration method. Let α be a positive constant, ω be a non-negative constant and $Q \in \mathbb{C}^{q\times q}$ be a Hermitian positive semidefinite matrix. Given an initial guess $x^{(0)} = (y^{(0)^{*}}, z^{(0)^{*}})^{*} \in \mathbb{C}^{n}$, for $k = 0, 1, 2, \ldots$ until the iteration sequence $\{x^{(k)}\} = \{(y^{(k)^{*}}, z^{(k)^{*}})^{*}\} \subset \mathbb{C}^{n}$ converges, compute the next iterate $x^{(k+1)} = (y^{(k+1)^{*}}, z^{(k+1)^{*}})^{*} \in \mathbb{C}^{n}$ according to the following procedure:

(i) solve for $y^{(k+1/2)} \in \mathbb{C}^{p}$ from the linear subsystem

$$(\alpha I + B)y^{(k+1/2)} = \alpha y^{(k)} - Ez^{(k)} + f;$$

(ii) compute

$$f^{(k+1/2)} = (\alpha I - B)y^{(k+1/2)} + f$$

and

$$g^{(k+1/2)} = E^{*}y^{(k)} + \big[\alpha I + Q + (\omega - 1)C\big]z^{(k)} + 2g;$$

(iii) solve for $z^{(k+1)} \in \mathbb{C}^{q}$ from the linear subsystem

$$\left(\alpha I + Q + (1+\omega)C + \frac{1}{\alpha}E^{*}E\right) z^{(k+1)} = \frac{1}{\alpha}E^{*}f^{(k+1/2)} + g^{(k+1/2)};$$

(iv) compute

$$y^{(k+1)} = \frac{1}{\alpha}\left(-Ez^{(k+1)} + f^{(k+1/2)}\right).$$

We remark that when $C = O$ the RHSS iteration method is mathematically equivalent to the one discussed in Bai & Benzi (2017) and when $C \neq O$ different choices of ω and Q yield a whole family of matrix splitting iteration methods. Furthermore, the iteration parameter α, the normalization parameter ω and the regularization matrix Q can be judiciously adjusted to speed up the overall convergence rate of the RHSS iteration method, and the regularization matrix Q can also be chosen to improve the conditioning of the linear subsystem in step (iii).

The main costs at each step of the RHSS iteration method are solving two linear subsystems with respect to the Hermitian positive definite matrices

$$\alpha I + B \quad \text{and} \quad \alpha I + Q + (1+\omega)C + \frac{1}{\alpha}E^*E. \tag{2.4}$$

This is in contrast to the HSS iteration method in Benzi & Golub (2004), which requires solving three linear subsystems with respect to the Hermitian positive definite matrices

$$\alpha I + B, \quad \alpha I + C \quad \text{and} \quad \alpha I + \frac{1}{\alpha}E^*E.$$

Moreover, the conditioning of the matrix $\alpha I + \frac{1}{\alpha}E^*E$ could be worse than that of the matrix $\alpha I + Q + (1+\omega)C + \frac{1}{\alpha}E^*E$, especially when E is almost rank-deficient. Therefore, the RHSS iteration method possesses reasonably faster convergence speed and has significantly less computational complexity, and as a result can exhibit substantially higher computational efficiency than the HSS iteration method. A brief explanation of the implementation of the RHSS iteration method is given at the end of this section.

Using the iterations in (2.2) and (2.3) we can rewrite the RHSS iteration method as a standard stationary iteration scheme as

$$x^{(k+1)} = M(\alpha,\omega)^{-1}N(\alpha,\omega)\,x^{(k)} + M(\alpha,\omega)^{-1}b, \quad k = 0,1,2,\ldots,$$

where

$$\begin{aligned} M(\alpha,\omega) &= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}I & O \\ O & (\alpha I + Q + \omega C)^{-1} \end{pmatrix} \big(\alpha I + H_+(\omega)\big)\big(\alpha I + S_+(\omega)\big) \\ &= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}(\alpha I + B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & E \\ -E^* & \alpha I + Q + (1+\omega)C \end{pmatrix} \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} N(\alpha,\omega) &= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}I & O \\ O & (\alpha I + Q + \omega C)^{-1} \end{pmatrix} \big(\alpha I - H_-(\omega)\big)\big(\alpha I - S_-(\omega)\big) \\ &= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}(\alpha I - B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & -E \\ E^* & \alpha I + Q + (\omega-1)C \end{pmatrix}. \end{aligned} \tag{2.6}$$

Note that $A = M(\alpha,\omega) - N(\alpha,\omega)$ forms a splitting of the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$. So, the RHSS iteration method can also be regarded as a stationary iteration method induced by this splitting, with its iteration matrix being given by

$$L(\alpha,\omega) = M(\alpha,\omega)^{-1}N(\alpha,\omega).$$

The splitting matrix $M(\alpha,\omega)$ can be employed to precondition the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$ and will be referred to as the RHSS preconditioner.

When the RHSS preconditioner is employed to accelerate a Krylov subspace iteration method, at each step we need to solve a generalized residual equation of the form

$$M(\alpha, \omega)\, w = r, \tag{2.7}$$

where $w = (w_a^*, w_b^*)^* \in \mathbb{C}^n$, with $w_a \in \mathbb{C}^p$ and $w_b \in \mathbb{C}^q$, is the generalized residual, and $r = (r_a^*, r_b^*)^* \in \mathbb{C}^n$, with $r_a \in \mathbb{C}^p$ and $r_b \in \mathbb{C}^q$, is the current residual. In actual implementations this generalized residual equation can be solved according to the following procedure:

(i) solve for $u_a \in \mathbb{C}^p$ from the linear subsystem

$$(\alpha I + B) u_a = 2\alpha r_a;$$

(ii) solve for $w_b \in \mathbb{C}^q$ from the linear subsystem

$$\left(\alpha I + Q + (1+\omega) C + \frac{1}{\alpha} E^* E\right) w_b = \frac{1}{\alpha} E^* u_a + 2 r_b;$$

(iii) compute $w_a \in \mathbb{C}^p$ from the formula

$$w_a = \frac{1}{\alpha}(u_a - E w_b).$$

Hence, analogously to the implementation of the RHSS iteration method, the action of the RHSS preconditioning matrix $M(\alpha, \omega)$ also requires solving two linear subsystems with the Hermitian positive definite coefficient matrices given in (2.4).

In actual computations the two Hermitian positive definite linear subsystems with respect to the coefficient matrices in (2.4) may be solved either exactly by sparse Cholesky factorization when the matrix sizes are moderate, or inexactly by the *preconditioned conjugate gradient* (PCG) method when the matrix sizes are very large; see Axelsson (1996), Golub & Van Loan (1996) and Greenbaum (1997). With this approach the linear subsystems are solved inexactly, we obtain the IRHSS iteration method for solving the stabilized saddle-point problem (1.1) and the IRHSS preconditioner for the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$ in the linear system (1.1). Of course, in the case of IRHSS preconditioning we may need to use a flexible Krylov subspace method, such as FGMRES (Saad, 2003). In addition, the choice of preconditioner to be used in the PCG method will be in general problem dependent. Standard options are the *incomplete Cholesky* factorization or the *algebraic multigrid* (AMG) approximation; see, e.g., Axelsson (1996), Golub & Van Loan (1996), Bai *et al.* (2003) and Saad (2003).

3. Convergence and preconditioning properties

In this section we prove the unconditional convergence of the RHSS iteration method and discuss the eigenvalue distribution of the preconditioned matrix $M(\alpha, \omega)^{-1} A$ with respect to the RHSS preconditioner.

As is known, the RHSS iteration method is convergent if and only if the spectral radius of its iteration matrix

$$L(\alpha, \omega) = M(\alpha, \omega)^{-1} N(\alpha, \omega)$$

is less than 1, i.e., $\rho(L(\alpha,\omega)) < 1$, where $M(\alpha,\omega)$ and $N(\alpha,\omega)$ are defined in (2.5) and (2.6), respectively; see Varga (1962), Axelsson (1996) and Golub & Van Loan (1996). The following theorem establishes the asymptotic convergence property of the RHSS iteration method.

THEOREM 3.1 For the stabilized saddle-point problem (1.1), assume that $B \in \mathbb{C}^{p\times p}$ is Hermitian positive definite, $C \in \mathbb{C}^{q\times q}$ is Hermitian positive semidefinite and $E \in \mathbb{C}^{p\times q}$ satisfies $\mathrm{null}(C) \cap \mathrm{null}(E) = \{0\}$. Let α be a prescribed positive constant, ω be a given non-negative constant and $Q \in \mathbb{C}^{q\times q}$ be a given Hermitian positive semidefinite matrix. Then it holds that $\rho(L(\alpha,\omega)) < 1$, i.e., the RHSS iteration method converges unconditionally to the exact solution of the stabilized saddle-point problem (1.1).

Proof. Denote

$$\hat{C} = I + \frac{1}{\alpha}(Q + \omega C), \quad \tilde{C} = \hat{C}^{-1/2} C \hat{C}^{-1/2}, \quad \tilde{E} = E\hat{C}^{-1/2} \tag{3.1}$$

and

$$\tilde{D} = \begin{pmatrix} I & O \\ O & \hat{C}^{1/2} \end{pmatrix}, \quad \tilde{G} = \begin{pmatrix} O & \tilde{E} \\ -\tilde{E}^* & \tilde{C} \end{pmatrix}, \quad \tilde{W} = \begin{pmatrix} (\alpha I + B)^{-1}(\alpha I - B) & O \\ O & I \end{pmatrix}. \tag{3.2}$$

Then the splitting matrices $M(\alpha,\omega)$ and $N(\alpha,\omega)$ in (2.5) and (2.6) can be rewritten as

$$M(\alpha,\omega) = \frac{1}{2}\tilde{D}\begin{pmatrix} \frac{1}{\alpha}(\alpha I + B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & \tilde{E} \\ -\tilde{E}^* & \alpha I + \tilde{C} \end{pmatrix}\tilde{D} \tag{3.3}$$

and

$$N(\alpha,\omega) = \frac{1}{2}\tilde{D}\begin{pmatrix} \frac{1}{\alpha}(\alpha I - B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & -\tilde{E} \\ \tilde{E}^* & \alpha I - \tilde{C} \end{pmatrix}\tilde{D}.$$

Consequently, the iteration matrix $L(\alpha,\omega)$ of the RHSS iteration method can be equivalently reformulated as

$$L(\alpha,\omega) = \tilde{D}^{-1}(\alpha I + \tilde{G})^{-1}\tilde{W}(\alpha I - \tilde{G})\tilde{D},$$

which is similar to the matrix

$$\tilde{L}(\alpha,\omega) = \tilde{W}(\alpha I + \tilde{G})^{-1}(\alpha I - \tilde{G}).$$

Applying the matrix identity

$$\tilde{B} := (\alpha I + B)^{-1}(\alpha I - B) = (\alpha I + B)^{-1/2}(\alpha I - B)(\alpha I + B)^{-1/2}$$

and the assumption that the matrix B is Hermitian positive definite we find that $\|\tilde{B}\| < 1$ and, thereby, $\|\tilde{W}\| \leq 1$ is valid for all $\alpha > 0$. Here and in the sequel we use $\|\cdot\|$ to indicate the Euclidean norm of either a vector or a matrix. In addition, as the matrix $\tilde{C}$ is Hermitian positive semidefinite and the matrix $\tilde{G}$ is non-Hermitian positive semidefinite, from Kellogg (1963) and Bai & Hadjidimos (2014) we know that the Cayley transform $(\alpha I + \tilde{G})^{-1}(\alpha I - \tilde{G})$ satisfies $\|(\alpha I + \tilde{G})^{-1}(\alpha I - \tilde{G})\| \leq 1$ for all $\alpha > 0$ and $\omega \geq 0$. As a result, it holds that

$$\left\|\tilde{L}(\alpha,\omega)\right\| \leq \|\tilde{W}\| \left\|(\alpha I + \tilde{G})^{-1}(\alpha I - \tilde{G})\right\| \leq 1 \quad \text{for any} \quad \alpha > 0 \quad \text{and} \quad \omega \geq 0.$$

Therefore, for all $\alpha > 0$ and $\omega \geq 0$ we have

$$\rho\big(L(\alpha,\omega)\big) = \rho\big(\tilde{L}(\alpha,\omega)\big) \leq \big\|\tilde{L}(\alpha,\omega)\big\| \leq 1$$

due to the similarity of the matrices $\tilde{L}(\alpha,\omega)$ and $L(\alpha,\omega)$.

We further claim that $\rho(L(\alpha,\omega)) < 1$ holds true for any $\alpha > 0$ and $\omega \geq 0$. Otherwise, if $\rho(L(\alpha,\omega)) = 1$ then $\rho(\tilde{L}(\alpha,\omega)) = 1$, which further implies that corresponding to a $\theta \in [0, 2\pi)$ there exists a nonzero vector $x \in \mathbb{C}^n$ such that

$$\tilde{W}(\alpha I + \tilde{G})^{-1}(\alpha I - \tilde{G})x = e^{\mathrm{i}\theta}x, \tag{3.4}$$

where $\mathrm{i} = \sqrt{-1}$ indicates the imaginary unit. Let

$$\tilde{x} := \begin{pmatrix} \tilde{y} \\ \tilde{z} \end{pmatrix} = (\alpha I + \tilde{G})^{-1}x.$$

Then $\tilde{x} \neq 0$ and the equation in (3.4) can be rewritten as

$$\tilde{W}(\alpha I - \tilde{G})\tilde{x} = e^{\mathrm{i}\theta}(\alpha I + \tilde{G})\tilde{x},$$

which, in the blockwise elements, is equivalent to

$$\begin{cases} \tilde{B}(\alpha\tilde{y} - \tilde{E}\tilde{z}) = e^{\mathrm{i}\theta}(\alpha\tilde{y} + \tilde{E}\tilde{z}), \\ \tilde{E}^*\tilde{y} + (\alpha I - \tilde{C})\tilde{z} = e^{\mathrm{i}\theta}\big[-\tilde{E}^*\tilde{y} + (\alpha I + \tilde{C})\tilde{z}\big]. \end{cases} \tag{3.5}$$

Denote

$$\tilde{\varrho} = \|\tilde{B}\| \quad \text{and} \quad \tilde{\delta} = \tilde{z}^*\tilde{E}^*\tilde{y} + \tilde{y}^*\tilde{E}\tilde{z}.$$

Then it follows from the first equation in (3.5) that

$$\begin{aligned} \|\alpha\tilde{y} + \tilde{E}\tilde{z}\| &= \big\|e^{\mathrm{i}\theta}(\alpha\tilde{y} + \tilde{E}\tilde{z})\big\| = \big\|\tilde{B}(\alpha\tilde{y} - \tilde{E}\tilde{z})\big\| \\ &\leq \|\tilde{B}\|\,\|\alpha\tilde{y} - \tilde{E}\tilde{z}\| = \tilde{\varrho}\,\|\alpha\tilde{y} - \tilde{E}\tilde{z}\|, \end{aligned}$$

which, in turn, implies the inequality

$$\big(1 - \tilde{\varrho}^2\big)\big(\alpha^2\,\tilde{y}^*\tilde{y} + \tilde{z}^*\tilde{E}^*\tilde{E}\tilde{z}\big) + \alpha\big(1 + \tilde{\varrho}^2\big)\tilde{\delta} \leq 0.$$

As a result, we see that

$$\tilde{\delta} \leq \frac{\tilde{\varrho}^2 - 1}{\alpha(\tilde{\varrho}^2 + 1)}\big(\alpha^2\,\tilde{y}^*\tilde{y} + \tilde{z}^*\tilde{E}^*\tilde{E}\tilde{z}\big) \leq 0.$$

We further assert that $\tilde{\delta} \neq 0$. Otherwise, if $\tilde{\delta} = 0$ then

$$\alpha^2\tilde{y}^*\tilde{y} + \tilde{z}^*\tilde{E}^*\tilde{E}\tilde{z} = 0,$$

and it follows that $\tilde{y} = 0$ and $\tilde{E}\tilde{z} = 0$. Now the second equation in (3.5) becomes

$$(\alpha I - \tilde{C})\tilde{z} = e^{\mathrm{i}\theta}(\alpha I + \tilde{C})\tilde{z},$$

which directly gives

$$\|(\alpha I - \tilde{C})\tilde{z}\| = \|(\alpha I + \tilde{C})\tilde{z}\|$$

or, equivalently, $\tilde{C}\tilde{z} = 0$. With the definitions of the matrices $\tilde{E}$ and $\tilde{C}$, we see that

$$E\hat{C}^{-1/2}\tilde{z} = 0 \quad \text{and} \quad \hat{C}^{-1/2}C\hat{C}^{-1/2}\tilde{z} = 0.$$

Recalling that $\text{null}(C) \cap \text{null}(E) = \{0\}$ we know that $\hat{C}^{-1/2}\tilde{z} = 0$ and, hence, $\tilde{z} = 0$. As a consequence, it holds that $\tilde{x} = 0$, which leads to a contradiction. Hence, it holds that $\tilde{\delta} < 0$.

On the other hand, by noticing that the second equation in (3.5) is equivalent to

$$\tilde{E}^*\tilde{y} = \alpha\left(\frac{e^{\mathrm{i}\theta} - 1}{e^{\mathrm{i}\theta} + 1}\tilde{z} + \tilde{C}\tilde{z}\right),$$

we obtain the equalities

$$\begin{cases} \tilde{z}^*\tilde{E}^*\tilde{y} = \alpha\left(\frac{e^{\mathrm{i}\theta}-1}{e^{\mathrm{i}\theta}+1}\tilde{z}^*\tilde{z} + \tilde{z}^*\tilde{C}\tilde{z}\right), \\ \tilde{y}^*\tilde{E}\tilde{z} = \alpha\left(\frac{e^{-\mathrm{i}\theta}-1}{e^{-\mathrm{i}\theta}+1}\tilde{z}^*\tilde{z} + \tilde{z}^*\tilde{C}\tilde{z}\right). \end{cases}$$

The summation of these two equalities straightforwardly leads to

$$\tilde{\delta} = \alpha\left[2\,\tilde{z}^*\tilde{C}\tilde{z} + \left(\frac{e^{\mathrm{i}\theta} - 1}{e^{\mathrm{i}\theta} + 1} + \frac{e^{-\mathrm{i}\theta} - 1}{e^{-\mathrm{i}\theta} + 1}\right)\tilde{z}^*\tilde{z}\right] = 2\alpha\,\tilde{z}^*\tilde{C}\tilde{z} \geq 0,$$

which contradicts the fact $\tilde{\delta} < 0$ that has been verified previously.

In summary, we have demonstrated $\rho(L(\alpha,\omega)) < 1$ (for all $\alpha > 0$ and for all $\omega \geq 0$), which readily shows that the RHSS iteration method converges unconditionally to the exact solution of the stabilized saddle-point problem (1.1). □

From the proof process of Theorem 3.1 we observe that the conditions that the constant ω is non-negative and the matrix Q is Hermitian positive semidefinite can be replaced by the more relaxed one that the matrix $I + \frac{1}{\alpha}(Q + \omega C)$ is Hermitian positive definite. This then allows that ω may be chosen to be a negative constant, and Q may be chosen to be a negative definite or even indefinite Hermitian matrix.

The next result presents a qualitative description of the clustering property of the eigenvalues of the preconditioned matrix $M(\alpha,\omega)^{-1}A$.

Theorem 3.2 For the stabilized saddle-point problem (1.1), assume that $B \in \mathbb{C}^{p\times p}$ is Hermitian positive definite, $C \in \mathbb{C}^{q\times q}$ is Hermitian positive semidefinite and $E \in \mathbb{C}^{p\times q}$ satisfies $\text{null}(C) \cap \text{null}(E) = \{0\}$. Denote the rank of the matrix E by r_E, i.e., $r_E = \text{rank}(E)$. Let α be a prescribed positive constant, ω be a given non-negative constant and $Q \in \mathbb{C}^{q\times q}$ be a given Hermitian positive semidefinite matrix. Then the eigenvalues of the preconditioned matrix $\mathbf{A}(\alpha,\omega) = M(\alpha,\omega)^{-1}A$ are clustered at 0_+

(with multiplicity r_E), 2_- (with multiplicity p) and $q - r_E$ positive points of the form $2(1-\varphi)$, if α is close to 0, where $M(\alpha,\omega)$ is the RHSS preconditioning matrix defined in (2.5) and φ is an eigenvalue of the matrix

$$\Phi(\omega) = \left(V_o^*\big[Q + (\omega+1)C\big]^{-1} V_o\right)^{-1} V_o^*(Q+\omega C)^{-1}V_o,$$

with $V_o \in \mathbb{C}^{q\times(q-r_E)}$ being a matrix whose columns form a basis of the null space of the matrix E.

Proof. With the notation in (3.1) and (3.2), we can rewrite the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$ as

$$A = \tilde{D}\tilde{A}\tilde{D}, \quad \text{with} \quad \tilde{A} = \begin{pmatrix} B & \tilde{E} \\ -\tilde{E}^* & \tilde{C} \end{pmatrix}.$$

Also, it follows from straightforward computations that

$$\begin{pmatrix} \alpha I & \tilde{E} \\ -\tilde{E}^* & \alpha I + \tilde{C} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\alpha} I - \frac{1}{\alpha^2}\tilde{E}\tilde{S}^{-1}\tilde{E}^* & -\frac{1}{\alpha}\tilde{E}\tilde{S}^{-1} \\ \frac{1}{\alpha}\tilde{S}^{-1}\tilde{E}^* & \tilde{S}^{-1} \end{pmatrix},$$

where

$$\tilde{S} = \alpha I + \tilde{C} + \frac{1}{\alpha}\tilde{E}^*\tilde{E}$$

is the Schur complement of the matrix $\alpha I + \tilde{G}$ with respect to its $(2,2)$-block. Hence, by making use of the expression of the matrix $M(\alpha,\omega)$ in (3.3) we obtain

$$\begin{aligned} \mathbf{A}(\alpha,\omega) &= M(\alpha,\omega)^{-1}A \\ &= 2\tilde{D}^{-1}\begin{pmatrix} \alpha I & \tilde{E} \\ -\tilde{E}^* & \alpha I + \tilde{C} \end{pmatrix}^{-1} \begin{pmatrix} \alpha(\alpha I + B)^{-1} & O \\ O & I \end{pmatrix} \begin{pmatrix} B & \tilde{E} \\ -\tilde{E}^* & \tilde{C} \end{pmatrix} \tilde{D}, \end{aligned}$$

which is similar to the matrix

$$\begin{aligned} \tilde{\mathbf{A}}(\alpha,\omega) &= 2\begin{pmatrix} \alpha(\alpha I + B)^{-1} & O \\ O & I \end{pmatrix} \begin{pmatrix} B & \tilde{E} \\ -\tilde{E}^* & \tilde{C} \end{pmatrix} \begin{pmatrix} \alpha I & \tilde{E} \\ -\tilde{E}^* & \alpha I + \tilde{C} \end{pmatrix}^{-1} \\ &= 2\begin{pmatrix} \alpha(\alpha I + B)^{-1} & O \\ O & I \end{pmatrix} \begin{pmatrix} B & \tilde{E} \\ -\tilde{E}^* & \tilde{C} \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha} I - \frac{1}{\alpha^2}\tilde{E}\tilde{S}^{-1}\tilde{E}^* & -\frac{1}{\alpha}\tilde{E}\tilde{S}^{-1} \\ \frac{1}{\alpha}\tilde{S}^{-1}\tilde{E}^* & \tilde{S}^{-1} \end{pmatrix} \\ &= 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix} \begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)\tilde{E}\tilde{S}^{-1}\tilde{E}^* & (\alpha I - B)\tilde{E}\tilde{S}^{-1} \\ -\tilde{S}^{-1}\tilde{E}^* & I - \alpha\tilde{S}^{-1} \end{pmatrix}. \end{aligned} \tag{3.6}$$

Again, with the notation in (3.1), we see that

$$\tilde{S} = \hat{C}^{-1/2}\check{S}\hat{C}^{-1/2} \quad \text{and} \quad \tilde{E}\tilde{S}^{-1} = E\check{S}^{-1}\hat{C}^{1/2},$$

where

$$\check{S} = \alpha I + Q + (\omega + 1)C + \frac{1}{\alpha}E^*E = \alpha I + \bar{C} + \frac{1}{\alpha}E^*E,$$

with

$$\bar{C} = Q + (\omega + 1)C.$$

Hence, with the substitution of these two expressions into (3.6) we have

$$\tilde{\mathbf{A}}(\alpha,\omega) = 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix}\begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)E\check{S}^{-1}E^* & (\alpha I - B)E\check{S}^{-1}\hat{C}^{1/2} \\ -\hat{C}^{1/2}\check{S}^{-1}E^* & I - \alpha\hat{C}^{1/2}\check{S}^{-1}\hat{C}^{1/2} \end{pmatrix},$$

which is further similar to the matrix

$$\begin{aligned}
\check{\mathbf{A}}(\alpha,\omega) &= 2\begin{pmatrix} I & O \\ O & \hat{C}^{-1/2} \end{pmatrix}\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix} \\
&\quad \cdot \begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)E\check{S}^{-1}E^* & (\alpha I - B)E\check{S}^{-1}\hat{C}^{1/2} \\ -\hat{C}^{1/2}\check{S}^{-1}E^* & I - \alpha\hat{C}^{1/2}\check{S}^{-1}\hat{C}^{1/2} \end{pmatrix}\begin{pmatrix} I & O \\ O & \hat{C}^{1/2} \end{pmatrix} \\
&= 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix}\begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)E\check{S}^{-1}E^* & (\alpha I - B)E\check{S}^{-1}\hat{C} \\ -\check{S}^{-1}E^* & I - \alpha\check{S}^{-1}\hat{C} \end{pmatrix} \\
&= 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix}\begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)E\check{S}^{-1}E^* & \frac{1}{\alpha}(\alpha I - B)E\check{S}^{-1}\check{C} \\ -\check{S}^{-1}E^* & I - \check{S}^{-1}\check{C} \end{pmatrix},
\end{aligned}$$

where

$$\check{C} = \alpha\hat{C} = \alpha I + Q + \omega C.$$

Let $E = U\Sigma V^*$ be the singular value decomposition of the matrix $E \in \mathbb{C}^{p\times q}$, where

$$U = \left[u_1, u_2, \ldots, u_p\right] \in \mathbb{C}^{p\times p} \quad \text{and} \quad V = \left[v_1, v_2, \ldots, v_q\right] \in \mathbb{C}^{q\times q}$$

are unitary matrices, and

$$\Sigma = \begin{pmatrix} \Sigma_{r_E} & O \\ O & O \end{pmatrix} \in \mathbb{R}^{p\times q}$$

is a diagonal matrix, where $\Sigma_{r_E} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{r_E})$, r_E is a positive integer satisfying $r_E \leq q$ and the singular values $\sigma_1, \sigma_2, \ldots, \sigma_{r_E}$ are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r_E} > 0$; see Golub & Van Loan (1996). Write

$$U_1 = \left[u_1, u_2, \ldots, u_{r_E}\right], \quad U_2 = \left[u_{r_E+1}, u_{r_E+2}, \ldots, u_p\right]$$

and

$$V_1 = \left[v_1, v_2, \ldots, v_{r_E}\right], \quad V_2 = \left[v_{r_E+1}, v_{r_E+2}, \ldots, v_q\right].$$

Then it holds that

$$EV_1 = U_1\Sigma_{r_E}, \quad EV_2 = O \quad \text{and} \quad E^*U_1 = V_1\Sigma_{r_E}, \quad E^*U_2 = O.$$

Since

$$\check{S} = V\left(\alpha I + V^*\bar{C}V + \frac{1}{\alpha}\Sigma^*\Sigma\right)V^* = V\bar{S}V^*$$

with

$$\bar{S} = \alpha I + V^*\bar{C}V + \frac{1}{\alpha}\Sigma^*\Sigma = \begin{pmatrix} \alpha I + V_1^*\bar{C}V_1 + \frac{1}{\alpha}\Sigma_{r_E}^2 & V_1^*\bar{C}V_2 \\ V_2^*\bar{C}V_1 & \alpha I + V_2^*\bar{C}V_2 \end{pmatrix} := \begin{pmatrix} \bar{S}_{11} & \bar{S}_{12} \\ \bar{S}_{21} & \bar{S}_{22} \end{pmatrix},$$

the block elements of the matrix

$$\bar{R} := \begin{pmatrix} \bar{R}_{11} & \bar{R}_{12} \\ \bar{R}_{21} & \bar{R}_{22} \end{pmatrix} = \bar{S}^{-1}$$

are given by

$$\begin{aligned} \bar{R}_{11} &= \bar{S}_{11}^{-1} + \bar{S}_{11}^{-1}V_1^*\bar{C}V_2\left(\alpha I + V_2^*\bar{C}V_2 - V_2^*\bar{C}V_1\bar{S}_{11}^{-1}V_1^*\bar{C}V_2\right)^{-1}V_2^*\bar{C}V_1\bar{S}_{11}^{-1}, \\ \bar{R}_{12} &= -\bar{S}_{11}^{-1}V_1^*\bar{C}V_2\left(\alpha I + V_2^*\bar{C}V_2 - V_2^*\bar{C}V_1\bar{S}_{11}^{-1}V_1^*\bar{C}V_2\right)^{-1}, \\ \bar{R}_{21} &= -\left(\alpha I + V_2^*\bar{C}V_2 - V_2^*\bar{C}V_1\bar{S}_{11}^{-1}V_1^*\bar{C}V_2\right)^{-1}V_2^*\bar{C}V_1\bar{S}_{11}^{-1}, \\ \bar{R}_{22} &= \left(\alpha I + V_2^*\bar{C}V_2 - V_2^*\bar{C}V_1\bar{S}_{11}^{-1}V_1^*\bar{C}V_2\right)^{-1}. \end{aligned}$$

Note that $\bar{R}_{21} = \bar{R}_{12}^*$. If we denote

$$\bar{R}_1 := \begin{pmatrix} \bar{R}_{11} \\ \bar{R}_{21} \end{pmatrix}$$

then it holds that $\bar{R}_1^* = (\bar{R}_{11}, \bar{R}_{12})$. As a result the matrix $\check{\mathbf{A}}(\alpha,\omega)$ can be written as

$$\check{\mathbf{A}}(\alpha,\omega) = 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix}\begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)U_1\Sigma_{r_E}\bar{R}_{11}\Sigma_{r_E}U_1^* & \frac{1}{\alpha}(\alpha I - B)U_1\Sigma_{r_E}\bar{R}_1^*V^*\check{C} \\ -V\bar{R}_1\Sigma_{r_E}U_1^* & I - V\bar{R}V^*\check{C} \end{pmatrix},$$

which is similar to the matrix

$$\bar{\mathbf{A}}(\alpha,\omega) = 2\begin{pmatrix} (\alpha I + B)^{-1} & O \\ O & I \end{pmatrix}\begin{pmatrix} B + \frac{1}{\alpha}(\alpha I - B)U_1\Sigma_{r_E}\bar{R}_{11}\Sigma_{r_E}U_1^* & \frac{1}{\alpha}(\alpha I - B)U_1\Sigma_{r_E}\bar{R}_1^*V^*\check{C}V \\ -\bar{R}_1\Sigma_{r_E}U_1^* & I - \bar{R}V^*\check{C}V \end{pmatrix}.$$

We can assert that the matrix $V_2^*\bar{C}V_2$ is invertible. In fact, for any $\bar{z} \in \mathbb{C}^{q-r_E}$, if $V_2^*\bar{C}V_2\bar{z} = 0$ then from $\bar{z}^*V_2^*\bar{C}V_2\bar{z} = 0$ we know that $V_2\bar{z} \in \text{null}(\bar{C}) \subset \text{null}(C)$. On the other hand, as $V_2\bar{z} \in \text{null}(E)$, it straightforwardly follows from the condition null$(C) \cap$ null$(E) = \{0\}$ that $V_2\bar{z} = 0$, which readily implies $\bar{z} = 0$.

Now, letting $\alpha \to 0$ we obtain

$$\frac{1}{\alpha}\bar{S}_{11}^{-1} = \left(\alpha^2 I + \alpha V_1^* \bar{C} V_1 + \Sigma_{r_E}^2\right)^{-1} \to \Sigma_{r_E}^{-2},$$

so that $\bar{S}_{11}^{-1} \to O$. It follows from this fact that

$$\bar{R}_{11} \to O, \quad \bar{R}_{12} \to O, \quad \bar{R}_{21} \to O, \quad \bar{R}_{22} \to \left(V_2^* \bar{C} V_2\right)^{-1}$$

and

$$\frac{1}{\alpha}\bar{R}_{11} \to \Sigma_{r_E}^{-2}, \quad \frac{1}{\alpha}\bar{R}_{12} \to -\Sigma_{r_E}^{-2} V_1^* \bar{C} V_2 \left(V_2^* \bar{C} V_2\right)^{-1}.$$

Therefore, when $\alpha \to 0$, it holds that

$$\bar{R} \to \begin{pmatrix} O & O \\ O & \left(V_2^* \bar{C} V_2\right)^{-1} \end{pmatrix}, \quad V^* \check{C} V \to V^*(Q + \omega C)V$$

and

$$\bar{R}_1 \Sigma_{r_E} U_1^* \to O, \quad \frac{1}{\alpha} U_1 \Sigma_{r_E} \bar{R}_{11} \Sigma_{r_E} U_1^* \to U_1 U_1^*,$$

as well as

$$\frac{1}{\alpha} U_1 \Sigma_{r_E} \bar{R}_1^* \to U_1 \Sigma_{r_E}^{-1} \left(I, -V_1^* \bar{C} V_2 \left(V_2^* \bar{C} V_2\right)^{-1}\right),$$

which readily implies

$$\bar{\mathbf{A}}(\alpha, \omega) \to 2 \begin{pmatrix} I - U_1 U_1^* & U_1 \bar{\bar{A}}_{12} & U_1 \bar{\bar{A}}_{13} \\ O & I & O \\ O & \bar{\bar{A}}_{32} & \bar{\bar{A}}_{33} \end{pmatrix} := \bar{\bar{\mathbf{A}}}(\omega),$$

where

$$\bar{\bar{A}}_{12} = \Sigma_{r_E}^{-1} V_1^* \left[\bar{C} V_2 (V_2^* \bar{C} V_2)^{-1} V_2^* - I\right] (Q + \omega C) V_1,$$

$$\bar{\bar{A}}_{13} = \Sigma_{r_E}^{-1} V_1^* \left[\bar{C} V_2 (V_2^* \bar{C} V_2)^{-1} V_2^* - I\right] (Q + \omega C) V_2,$$

$$\bar{\bar{A}}_{32} = -\left(V_2^* \bar{C} V_2\right)^{-1} V_2^* (Q + \omega C) V_1,$$

$$\bar{\bar{A}}_{33} = I - \left(V_2^* \bar{C} V_2\right)^{-1} V_2^* (Q + \omega C) V_2.$$

By noticing the identity

$$U^*\big(I - U_1U_1^*\big)U = \begin{pmatrix} O & O \\ O & I \end{pmatrix}$$

we see that the matrix $\bar{\bar{\mathbf{A}}}(\omega)$ is similar to the matrix

$$2\begin{pmatrix} O & O & \bar{\bar{A}}_{12} & \bar{\bar{A}}_{13} \\ O & I & O & O \\ O & O & I & O \\ O & O & \bar{\bar{A}}_{32} & \bar{\bar{A}}_{33} \end{pmatrix},$$

which, after simultaneous exchanges of the last two rows and columns, is further similar to the matrix

$$\hat{\mathbf{A}}(\omega) = 2\begin{pmatrix} O & O & \bar{\bar{A}}_{13} & \bar{\bar{A}}_{12} \\ O & I & O & O \\ O & O & \bar{\bar{A}}_{33} & \bar{\bar{A}}_{32} \\ O & O & O & I \end{pmatrix}.$$

Recalling that the sizes of the diagonal blocks of the matrix $\hat{\mathbf{A}}(\omega)$ are $r_E \times r_E$, $(p - r_E) \times (p - r_E)$, $(q - r_E) \times (q - r_E)$ and $r_E \times r_E$ in order, we then immediately achieve the conclusion that we were proving. □

For Theorem 3.2 we remark that $\varphi \in \big[\frac{\omega}{\omega+1}, 1\big]$ holds for all $\omega \geq 0$. Hence, the eigenvalues of the matrix $\Phi(\omega)$ are located in the interval $\big[0, \frac{2}{\omega+1}\big]$ for all $\omega \geq 0$; but they could be not well clustered if there is no specific restriction on the regularization matrix $Q \in \mathbb{C}^{q\times q}$ unless the $(2,2)$-block matrix $C \in \mathbb{C}^{q\times q}$ is very small in norm. Moreover, when α is close to 0, we observe from the proof of Theorem 3.2 that the eigenvalues of the RHSS-preconditioned matrix $\mathbf{A}(\alpha,\omega) = M(\alpha,\omega)^{-1}A$ are clustered only if the eigenvalues of the matrix $\bar{\bar{A}}_{33}$ are clustered or, roughly speaking, those of the matrix $\check{S}^{-1}\check{C}$ are clustered. This further implies that we should impose that $\check{S}^{-1}\check{C}$ approximates a scalar matrix so that the RHSS-preconditioned Krylov subspace methods, like GMRES, can be expected to converge quickly; see, e.g., Eisenstat *et al.* (1983), Saad (2003), Benzi & Simoncini (2006) and Bai (2015). From this observation we know that in actual computations the regularization matrix $Q \in \mathbb{C}^{q\times q}$ should be chosen such that

$$Q \approx (\alpha\gamma - \omega)C + \gamma E^*E - \alpha I, \tag{3.7}$$

or

$$Q \approx (\alpha\gamma - \omega)C + \gamma E^*E \tag{3.8}$$

if α is small enough (recall that Q must be positive semidefinite), where γ is a positive constant.

In practice, α should be chosen small enough so as to have most of the eigenvalues of the RHSS-preconditioned matrix falling into well-separated clusters but not so small that the RHSS-preconditioned matrix becomes too close to being singular. In contrast, when the RHSS iteration method is used as a stationary method the asymptotic convergence rate is maximized when the spectral radius of the iteration matrix is the smallest, and this means that the optimal α should *not* be taken small; indeed, the optimal α can be reasonably large. We refer to Bai & Benzi (2017) for more details.

In general, the RHSS iteration method involves two iteration parameters α and ω that need to be predetermined in actual applications. Note that the parameter ω is introduced only in the splitting of the (2,2)-block matrix $C \in \mathbb{C}^{q\times q}$, while the parameter α is introduced in the whole RHS splitting. Hence, the parameter ω may mainly have a local effect and the parameter α should principally have a global effect on the convergence property of the RHS splitting of the stabilized saddle-point matrix $A \in \mathbb{C}^{n\times n}$. As a result, the convergence behavior of the RHSS iteration method should be not so sensitive with respect to the parameter ω but should be relatively sensitive with respect to the parameter α. Roughly speaking, choosing the best pair (α, ω) of the parameters such that the RHSS iteration method attains its fastest convergence rate is very difficult and also problem dependent in both theory and applications. However, in accordance with the aforementioned observation, in actual implementations we may find the pair (α, ω) of the optimal parameters by the following two strategies:

I. Find the pair (α, ω) of optimal parameters experimentally for a problem of smaller size, fix the parameter ω at this optimal value for all problem sizes and then experimentally find the corresponding optimal parameter α for all other sizes of the problem by trial runs;

II. Find the pair (α, ω) of optimal parameters experimentally for a problem of smaller size, then fix and reuse this pair (α, ω) of optimal parameters for all other larger sizes of the problem.

Compared with strategy I, strategy II is simpler in algorithmic execution and cheaper in computational workload, so that it can produce the parameters α and ω more effectively for those problems that may lead to an almost parametrically insensitive convergence property of the RHSS iteration method. In this way we provide two practical and inexpensive strategies for finding an almost optimal pair (α, ω) of the parameters for the RHSS iteration method, in order that it can achieve high computational efficiency. This approach can be equally applied to determining a nearly optimal pair (α, ω) of the parameters for the RHSS-preconditioned Krylov subspace iteration methods.

We remark that for the special choices of the regularization matrix $Q \in \mathbb{C}^{q\times q}$ as described in (3.7) and (3.8), the RHSS method, meaning either the RHSS iteration or the RHSS preconditioning method, involves only two arbitrary parameters, one is α and another is γ; see Remark 3.3 for more details. As stated above, the convergence behavior of the RHSS iteration method should not be so sensitive with respect to the parameter γ, although it should be relatively sensitive with respect to the parameter α. The optimal values of these parameters α and γ can be determined by experience and trial runs such that either the iteration count or the computing time of the RHSS iteration method or the RHSS-preconditioned Krylov subspace iteration method is minimized in an analogous fashion to the determination of the parameters α and ω in the general situation of the RHSS method stated above.

Remark 3.3 When the regularization matrix $Q \in \mathbb{C}^{q\times q}$ is taken to be

(a) $Q = (\alpha\gamma - \omega)C + \gamma E^*E - \alpha I$,

(b) $Q = (\alpha\gamma - \omega)C + \gamma E^*E$ or

(c) $Q = \gamma C$,

the RHSS iteration method can be written as

$$
\begin{aligned}
&(\alpha I + B)y^{(k+1/2)} = \alpha y^{(k)} - Ez^{(k)} + f,\\
&f^{(k+1/2)} = (\alpha I - B)y^{(k+1/2)} + f,\\
&\begin{cases}
g^{(k+1/2)} = E^* y^{(k)} + \big[(\alpha\gamma - 1)C + \gamma E^* E\big] z^{(k)} + 2g, & \text{for case (a)},\\
g^{(k+1/2)} = E^* y^{(k)} + \big[\alpha I + (\alpha\gamma - 1)C + \gamma E^* E\big] z^{(k)} + 2g, & \text{for case (b)},\\
g^{(k+1/2)} = E^* y^{(k)} + \big[\alpha I + (\gamma - 1)C\big] z^{(k)} + 2g, \text{ with } \gamma := \omega + \gamma, & \text{for case (c)},
\end{cases}\\
&\begin{cases}
\left(C + \frac{1}{\alpha}E^* E\right) z^{(k+1)} = \frac{1}{\alpha\gamma+1}\left(\frac{1}{\alpha}E^* f^{(k+1/2)} + g^{(k+1/2)}\right), & \text{for case (a)},\\
\left(\frac{\alpha}{\alpha\gamma+1}I + C + \frac{1}{\alpha}E^* E\right) z^{(k+1)} = \frac{1}{\alpha\gamma+1}\left(\frac{1}{\alpha}E^* f^{(k+1/2)} + g^{(k+1/2)}\right), & \text{for case (b)},\\
\left(\alpha I + (\gamma + 1)C + \frac{1}{\alpha}E^* E\right) z^{(k+1)} = \frac{1}{\alpha}E^* f^{(k+1/2)} + g^{(k+1/2)}, \text{ with } \gamma := \omega + \gamma, & \text{for case (c)},
\end{cases}\\
&y^{(k+1)} = \frac{1}{\alpha}\left(-Ez^{(k+1)} + f^{(k+1/2)}\right),
\end{aligned}
$$

and the generalized residual equation of the form (2.7) can be solved according to the procedure

$$
\begin{aligned}
&(\alpha I + B)u_a = 2\alpha r_a,\\
&\begin{cases}
\left(C + \frac{1}{\alpha}E^* E\right) w_b = \frac{1}{\alpha\gamma+1}\left(\frac{1}{\alpha}E^* u_a + 2r_b\right), & \text{for case (a)},\\
\left(\frac{\alpha}{\alpha\gamma+1}I + C + \frac{1}{\alpha}E^* E\right) w_b = \frac{1}{\alpha\gamma+1}\left(\frac{1}{\alpha}E^* u_a + 2r_b\right), & \text{for case (b)},\\
\left(\alpha I + (\gamma + 1)C + \frac{1}{\alpha}E^* E\right) w_b = \frac{1}{\alpha}E^* u_a + 2r_b, \text{ with } \gamma := \omega + \gamma, & \text{for case (c)},
\end{cases}\\
&w_a = \frac{1}{\alpha}(u_a - Ew_b).
\end{aligned}
$$

We observe that for the special choices (a)–(c) of the regularization matrix $Q \in \mathbb{C}^{q\times q}$, the RHSS method involves only two arbitrary parameters rather than three: one is α and another is γ, with γ being, in particular, set to be $\gamma := \omega + \gamma$ in case (c), which means that the parameter ω is absorbed into γ.

4. Numerical results

In this section we implement HSS, IHSS and RHSS, IRHSS methods by using them either as linear iteration solvers or as matrix splitting preconditioners for GMRES or FGMRES. The corresponding methods in the preconditioning case are abbreviated as HSS-GMRES, IHSS-FGMRES and RHSS-GMRES, IRHSS-FGMRES. We also implement the preconditioned GMRES method incorporated with a block-triangular preconditioner, called BT-GMRES, in, e.g., Murphy *et al.* (2000), Elman *et al.* (2014) and Beik *et al.* (2017) and the preconditioned MINRES method incorporated with a block-diagonal preconditioner, called BD-MINRES, in, e.g., Silvester & Wathen (1994), Perugia & Simoncini (2000), Pearson *et al.* (2014) and Herzog & Soodhalter (2017). For MINRES the stabilized saddle-point problem

(1.1) is reformulated into its Hermitian variant by multiplying -1 on both sides of the second block equation. The BD and the BT preconditioners are taken to be

$$P_{\mathrm{BD}} = \begin{pmatrix} \widehat{\widehat{B}} & O \\ O & \widehat{\widehat{S}} \end{pmatrix} \quad \text{and} \quad P_{\mathrm{BT}} = \begin{pmatrix} \widehat{\widehat{B}} & E \\ O & \widehat{\widehat{S}} \end{pmatrix}, \tag{4.1}$$

with $\widehat{\widehat{B}}$ and $\widehat{\widehat{S}}$ being certain approximations to the (1,1)-block matrix B and the Schur complement $S = C + E^*B^{-1}E$, respectively, of the stabilized saddle-point matrix A in (1.1).

The numerical behaviors of these methods are tested and evaluated in terms of the number of iteration steps (denoted 'IT') and the computing time in seconds (denoted 'CPU'). We are going to solve three types of stabilized saddle-point problems from optimal boundary control (Herzog & Soodhalter, 2017), binary image inpainting (Bosch *et al.*, 2014) and nonlinear image restoration (Benzi & Ng, 2006). With these experiments we will show the advantages of the RHSS approach relative to the older methods in the classes of HSS, GMRES and MINRES and will also identify suitable choices of the regularization matrix Q, the normalization parameter ω and the iteration parameter α.

In the sequel we use $(\cdot)^{\mathrm{T}}$ to denote the transpose of either a vector or a matrix. In our implementations, in the exact HSS and RHSS, the linear subsystems with the coefficient matrices $\alpha I + B$, $\alpha I + C$, $\alpha I + \frac{1}{\alpha}E^{\mathrm{T}}E$ and $\alpha I + Q + (1+\omega)C + \frac{1}{\alpha}E^{\mathrm{T}}E$ are solved directly by sparse Cholesky factorizations. All experiments are started from the initial vector $x^{(0)} = 0$, and terminated once the relative residual errors at the current iterates $x^{(k)}$ satisfy $\|b - Ax^{(k)}\| \leq 10^{-6} \times \|b\|$. In addition, all experiments are carried out using MATLAB (version R2015a) on a personal computer with 2.83 GHz central processing unit (Intel(R) Core(TM)2 Quad CPU Q9550), 8.00 GB memory and the Linux operating system (Ubuntu 15.04). In our codes, in order to construct approximate solvers for certain symmetric positive definite linear subsystems precisely specified in the sequel, we utilize preconditioners based on the *modified incomplete Cholesky* (MIC) factorization implemented in MATLAB by the function `ichol(·, struct('droptol', 1e-3, 'michol', 'on'))`. Moreover, we build up their AMG approximations or preconditioners by utilizing the package HSL_MI20 (hsl_mi20_precondition) with default parameters.

EXAMPLE 4.1 (Herzog & Soodhalter, 2017). Consider the optimal boundary control problem for the stationary heat equation

$$\min_{u,\tilde{f}} \frac{1}{2}\|u - u_*\|^2_{\mathcal{L}_2(\Omega)} + \frac{\beta}{2}\|\tilde{f}\|^2_{\mathcal{L}_2(\partial\Omega)} \tag{4.2}$$

$$\text{s.t.} \quad \begin{cases} -\nu\,\Delta u + \eta u = 0 & \text{in} \quad \Omega, \\ \nu\frac{\partial u}{\partial \vec{n}} = \tilde{f} & \text{on} \quad \partial\Omega, \end{cases}$$

where u is a vector-valued function representing the temperature, u_* is a given function that represents the desired state, $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$ is the unit square domain with its boundary being indicated by $\partial\Omega$, $\|\cdot\|_{\mathcal{L}_2(\Omega)}$ denotes the $\mathcal{L}_2$-norm defined on Ω, $\beta > 0$ is the regularization parameter, Δ denotes the componentwise Laplacian operator, ν is the thermal diffusivity, $\eta > 0$ is a prescribed positive constant and $\vec{n}$ is the unit outward normal vector at $\partial\Omega$. After eliminating the control function $\tilde{f}$ from problem (4.2) we obtain a stabilized saddle-point problem in the continuous form with respect to the state u and the adjoint state $\tilde{p}$. Then a further discretization of this problem by piecewise linear and continuous elements for u and $\tilde{p}$, respectively, results in the stabilized saddle-point problem (1.1) of the coefficient matrix

$$\begin{pmatrix} M & \eta M + \nu K \\ -(\eta M + \nu K) & \frac{1}{\beta}\breve{M} \end{pmatrix},$$

where M and K are the mass and the stiffness matrices on the domain Ω, respectively, and $\breve{M}$ is the mass matrix on the boundary $\partial\Omega$. In actual computations we set $u_* = x$, $\beta = 10^{-4}$, $\nu = 1$ and $\eta = 1$. As for the right-hand side subvectors f and g, the former is obtained from evaluating the linear form $f_u(v) = \int_\Omega u_* v \,\mathrm{d}x = \int_\Omega xv \,\mathrm{d}x$ on the corresponding basis functions and the latter is set to be $g = 0$.

For this example we have

$$B = M, \quad E = \eta M + \nu K, \quad C = \frac{1}{\beta}\breve{M}$$

and $p = m^2$, $q = m^2$, where m is the number of grids that corresponds to the step size $h = \frac{1}{m-1}$ of the discretization mesh. Note that the dimension of the stabilized saddle-point matrix $A \in \mathbb{R}^{n\times n}$ is $n = 2m^2$.

As suggested in Section 3 we take the regularization matrix to be

$$Q = (\alpha\gamma - \omega)C + \gamma E^T E \quad \text{(i.e., case (b) in Remark 3.3)}$$

in the RHSS iteration method and

$$Q = (\alpha\gamma - \omega)C + \gamma E^T E - \alpha I \quad \text{(i.e., case (a) in Remark 3.3)}$$

or

$$Q = (\alpha\gamma - \omega)C + \gamma E^T E \quad \text{(i.e., case (b) in Remark 3.3)}$$

in both the RHSS-GMRES and IRHSS-FGMRES methods, where γ is a positive constant to be determined according to the problem and the method. For these two cases of the regularization matrix Q the corresponding methods are indicated by RHSS(#) and RHSS-GMRES(#), with #=a, b. In this fashion the meanings of the notation IRHSS(#) and IRHSS-FGMRES(#) are obvious. We refer to Remark 3.3 for a precise description of RHSS iteration methods and RHSS preconditioners. Note that RHSS and, correspondingly, IRHSS, now possess only two arbitrary parameters: one is the iteration parameter α and the other is the regularization parameter γ. The normalization parameter ω vanishes for these special choices of the regularization matrix Q.

In the implementations of both IHSS and IRHSS used either as a linear solver or as a right preconditioner, the involved linear subsystems in the residual-updating form (Bai & Rozložník, 2015) are solved iteratively by the PCG method starting from the initial guess 0, with the stopping tolerance 0.1, except for the mesh size $m = 256$ which adopts the stopping tolerance 0.05, i.e., the Euclidean norm of the residual corresponding to the current inner iterate achieves a reduction 0.1 or 0.05 relative to that corresponding to the initial inner iterate. Specifically, the coefficient matrices $\alpha I + B$ and $\alpha I + C$ are preconditioned by their MIC factorizations. In IHSS the matrix $\alpha I + \frac{1}{\alpha}E^{\mathrm{T}}E$ is preconditioned by $\widehat{\widehat{E}}_\alpha^{\mathrm{T}}\widehat{\widehat{E}}_\alpha$ with $\widehat{\widehat{E}}_\alpha$ being the AMG approximation to $E_\alpha = \sqrt{\alpha}I + \frac{1}{\sqrt{\alpha}}E$. As for IRHSS, the preconditioners adopted for solving the involved inner linear subsystems are dependent: in the IRHSS solver the matrix

TABLE 1 *Numerical results for RHSS(b), HSS-GMRES and RHSS-GMRES(#) for Example 4.1*

		m					
Method	Index	64	96	128	192	256	384
RHSS(b)	IT	30	31	39	57	95	205
	CPU	1.61	6.01	21.09	140.53	484.35	25697.14
HSS-GMRES	IT	95	115	142	172	203	288
	CPU	10.82	44.13	126.49	461.68	1471.65	96716.97
RHSS-GMRES(a)	IT	8	7	7	7	7	7
	CPU	1.05	2.81	7.95	59.47	222.79	8385.43
RHSS-GMRES(b)	IT	8	8	7	8	7	8
	CPU	0.90	3.21	7.45	61.53	223.08	12097.14

$\frac{\alpha}{\alpha\gamma+1}I + C + \frac{1}{\alpha}E^{\mathrm{T}}E$ is preconditioned by its AMG approximation and in the IRHSS preconditioner the preconditioning matrix for $C + \frac{1}{\alpha}E^{\mathrm{T}}E$ in case (a) is taken to be $\widehat{\widehat{F}}_{\alpha}^{\mathrm{T}}\widehat{\widehat{F}}_{\alpha}$, with $\widehat{\widehat{F}}_{\alpha}$ being the AMG approximation of $\frac{1}{\sqrt{\alpha}}E$, while the preconditioning matrix for $\frac{\alpha}{\alpha\gamma+1}I + C + \frac{1}{\alpha}E^{\mathrm{T}}E$ in case (b) is taken to be $\widehat{\widehat{F}}_{\alpha,\gamma}^{\mathrm{T}}\widehat{\widehat{F}}_{\alpha,\gamma}$, with $\widehat{\widehat{F}}_{\alpha,\gamma}$ being the AMG approximation of $\sqrt{\frac{\alpha}{\alpha\gamma+1}I + \frac{1}{\sqrt{\alpha}}E}$.

In addition, the MINRES method is preconditioned by the optimal block-diagonal preconditioner proposed in Herzog & Soodhalter (2017), i.e., Diag$(\widehat{\widehat{B}}, \widehat{\widehat{S}})$, with $\widehat{\widehat{B}}$ and $\widehat{\widehat{S}}$ being the AMG approximations of the matrices $M + K$ and $\frac{1}{\beta}(M + K)$, respectively. In addition, in the block-triangular preconditioner of the form in (4.1) the (1,1)-block matrix $\widehat{\widehat{B}}$ is the 25-step Chebyshev semi-iteration approximation of the matrix B and the (2,2)-block matrix $\widehat{\widehat{S}}$ is the AMG approximation of the approximated Schur complement $\widehat{S} = C + E^{\mathrm{T}}\,\mathrm{diag}(B)^{-1}\,E$, where diag$(B)$ is the diagonal part of the matrix B.

First of all we remark that for this example both HSS and IHSS iteration methods fail to converge within 10000 steps for all tested values of m, regardless of the values of the iteration parameter α used.

In Table 1 we report iteration counts and CPU times for RHSS(b), HSS-GMRES and RHSS-GMRES(#) with #=a, b, for which the parameters α and (or) γ are taken to be the experimentally computed optimal ones that minimize the total number of iteration steps of the corresponding method; see Table 2. As the results in Table 1 show, for each mesh size m the RHSS(b) iteration method succeeds in solving the stabilized saddle-point problem and even requires much smaller iteration step and costs much less CPU time than the HSS-GMRES method. Although the numbers of iteration steps of RHSS(b) and HSS-GMRES are increasing when m is growing, those of HSS-GMRES are increasing significantly more quickly. Roughly speaking, for a smaller value of m such as $m \leq 128$, the iteration steps and the CPU times of HSS-GMRES are at least three and six times of those of RHSS(b), while for a larger value of m such as $m \geq 192$ the iteration steps and the CPU times of HSS-GMRES are at least two and three times of those of RHSS(b), respectively. Hence, compared with HSS-GMRES, RHSS(b) is the winner even if it is purely a linear iteration solver.

Both RHSS-GMRES(a) and RHSS-GMRES(b) significantly outperform RHSS(b) and HSS-GMRES in terms of iteration steps and CPU times. In addition, for all tested values of the mesh size m, RHSS-GMRES(a) and RHSS-GMRES(b) are convergent in almost the same iteration steps and CPU times, except for the largest case $m = 384$ which makes a difference of 3711.71 seconds only

Table 2 *Experimentally computed optimal values of the iteration parameter α and (or) the regularization parameter γ in RHSS(b), HSS-GMRES and RHSS-GMRES(#) for Example 4.1*

		m					
Method	Index	64	96	128	192	256	384
	γ	805	1460	1850	2770	2650	2600
RHSS(b)	α	7.0E−4	4.0E−4	3.0E−4	2.0E−4	2.0E−4	2.0E−4
HSS-GMRES	α	0.0073	0.0064	0.0057	0.0046	0.0041	0.0041
	γ	0.001	0.0001	0.001	0.0001	0.001	0.001
RHSS-GMRES(a)	α	2.0E−4	1.1E−4	6.0E−5	3.0E−5	1.4E−5	8.0E−6
	γ	0.001	0.001	0.001	0.001	0.001	0.001
RHSS-GMRES(b)	α	2.0E−4	1.0E−4	6.0E−5	3.0E−5	1.4E−5	8.0E−6

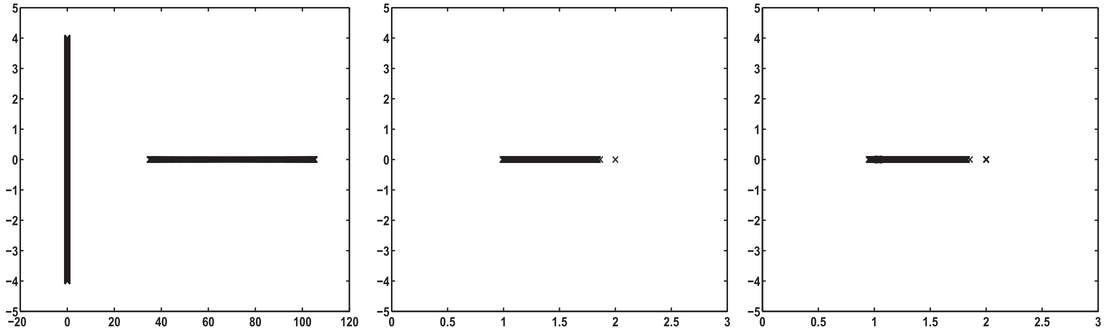

Fig. 1. Eigenvalue distributions of the original coefficient matrix (left), the RHSS(a)-preconditioned matrix (middle) and the RHSS(b)-preconditioned matrix (right), with their numerically computed optimal parameters α and γ, respectively, when $m = 96$ for Example 4.1.

in time. This also implies that both RHSS-GMRES(a) and RHSS-GMRES(b) possess h-independent convergence behavior, thanks to the tightly clustered eigenvalues of the RHSS-preconditioned matrices, especially when the parameter α is small enough; see Figs 1 and 2 for an intuitive visualization. We refer to Theorem 3.2 for a theoretical illustration.

In Table 3 we report iteration counts and CPU times for IRHSS(b), IHSS-FGMRES, IRHSS-FGMRES(#), BT-GMRES and BD-MINRES. The parameters α and (or) γ are taken to be the same as their exact counterparts; see Table 2. BT-GMRES outperforms either IRHSS(b) or IHSS-FGMRES in terms of both iteration counts and CPU times, with its convergence behavior being, however, uniformly h-dependent, and BD-MINRES performs much better in CPU time than IRHSS(b), IHSS-FGMRES and BT-GMRES, with its convergence behavior being, roughly speaking, h-independent, though it requires more iteration steps than the other methods except for IRHSS(b). Moreover, both IRHSS-FGMRES(a) and IRHSS-FGMRES(b) converge to the solution of the stabilized saddle-point problem in about the same iteration step and CPU time for all tested values of m and both of them significantly outperform the other methods such as BT-GMRES and BD-MINRES in iteration steps and CPU times. Remarkably, the numbers of iteration steps of both IRHSS-FGMRES(a) and IRHSS-FGMRES(b) remain about

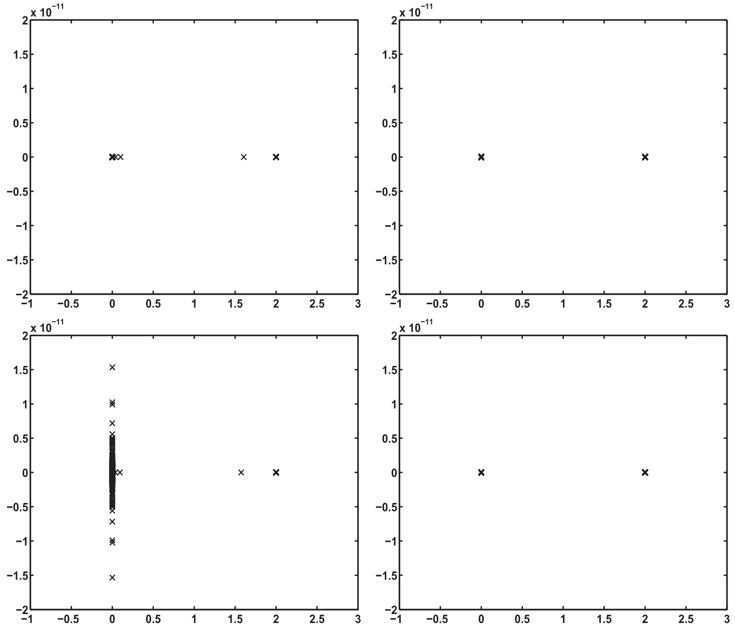

FIG. 2. Eigenvalue distributions of the RHSS(a)-preconditioned matrix with $\alpha = 1.1\mathrm{E}-8$ (upper left), $\alpha = 1.1\mathrm{E}-12$ (upper right) and $\gamma = 0.0001$, and eigenvalue distributions of the RHSS(b)-preconditioned matrix with $\alpha = 1.0\mathrm{E}-8$ (bottom left), $\alpha = 1.0\mathrm{E}-12$ (bottom right) and $\gamma = 0.001$, when $m = 96$ for Example 4.1.

TABLE 3 *Numerical results for IRHSS(b), IHSS-FGMRES, IRHSS-FGMRES(#), BT-GMRES and BD-MINRES for Example* 4.1

		m					
Method	Index	64	96	128	192	256	384
IRHSS(b)	IT	359	543	826	1189	2121	5520
	CPU	3.54	11.52	33.43	128.25	524.48	3511.16
IHSS-FGMRES	IT	112	135	161	195	239	343
	CPU	8.72	21.88	45.53	127.82	298.09	1238.62
IRHSS-FGMRES(a)	IT	19	16	17	18	18	19
	CPU	1.16	2.36	4.47	11.56	19.58	50.39
IRHSS-FGMRES(b)	IT	19	18	18	18	18	19
	CPU	1.17	2.50	5.19	10.85	19.43	48.00
BT-GMRES	IT	70	88	100	128	149	206
	CPU	2.45	6.99	14.08	46.61	101.60	426.01
BD-MINRES	IT	268	278	276	282	280	295
	CPU	1.68	3.91	7.00	16.17	29.78	71.58

TABLE 4 *Numerical results for RHSS(b), IRHSS(b), RHSS-GMRES(#) and IRHSS-FGMRES(#) for Example 4.1, which are produced by adopting the values of the parameters α and γ in Table 2 corresponding to the case $m = 64$*

		m					
Method	Index	64	96	128	192	256	384
RHSS(b)	IT	30	50	84	181	319	715
	CPU	1.61	9.22	39.62	288.10	1231.63	73606.19
RHSS-GMRES(a)	IT	8	9	12	16	19	26
	CPU	1.05	3.42	13.85	104.43	264.88	10393.02
RHSS-GMRES(b)	IT	8	10	12	16	21	31
	CPU	0.90	5.33	14.00	104.51	280.80	11453.53
IRHSS(b)	IT	359	751	1323	3548	8872	33821
	CPU	3.54	16.21	59.49	643.14	2458.80	25863.51
IRHSS-FGMRES(a)	IT	19	21	24	28	36	47
	CPU	1.16	3.85	9.15	36.91	99.81	440.42
IRHSS-FGMRES(b)	IT	19	20	23	34	42	53
	CPU	1.17	3.39	8.59	36.28	96.37	437.89

the same constant for different mesh sizes, indicating that these two methods possess h-independent convergence behavior, too. By comparing Table 3 with Table 1 we see that the inexact methods IRHSS-FGMRES(a) and IRHSS-FGMRES(b) can achieve much higher computational efficiency than their exact counterparts and they are the best among all these methods for solving this stabilized saddle-point problem.

Going back to Table 2 we see that for RHSS(b) the optimal values of the parameter α are quite small but all are about the same order $\mathcal{O}(10^{-4})$ in magnitude, for different mesh size m; in contrast, the optimal values of the parameter γ are very large and changing very quickly, with the orders being approximately reciprocal to those of α. In fact, it holds that $\alpha\gamma \in [0.52, 0.584]$. For RHSS-GMRES(a) and RHSS-GMRES(b), as predicted in Theorem 3.2, the optimal values of the parameter γ are small and those of the parameter α are even smaller for all tested mesh sizes; however, the former remains almost the same constant 0.001 except for the two cases $m = 96$ and 192 for which $\gamma = 0.0001$, while the latter decreases gradually when the mesh size m is increasing. This is, in nature, equivalent to strategy I recommended in Section 3 for determining the pair (α, γ) of the optimal parameters, with ω there being replaced by γ here. We should point out that for this example strategy II in Section 3 works but it does not work very well, especially when the mesh sizes are far away from the referenced smallest one, i.e., $m = 64$. This phenomenon is evident from the data in Table 4, which, for all mesh sizes m, are produced by adopting the values of the parameters α and γ in Table 2 corresponding to the smallest mesh size $m = 64$. Indeed, the data in Table 4 do not match aptly with the data in Tables 1 and 3, especially when m is much larger than 64. The reason may be that the RHSS method is quite sensitive to the parameter α or γ, which is, in turn, strongly dependent on the mesh size m, for this problem; see Fig. 3. However, from Table 4 we see that though both IRHSS-FGMRES(a) and IRHSS-FGMRES(b) are much less effective than BD-MINRES in terms of CPU time, especially when m is larger than 128, they apparently outperform BT-GMRES in iteration counts and CPU times and their

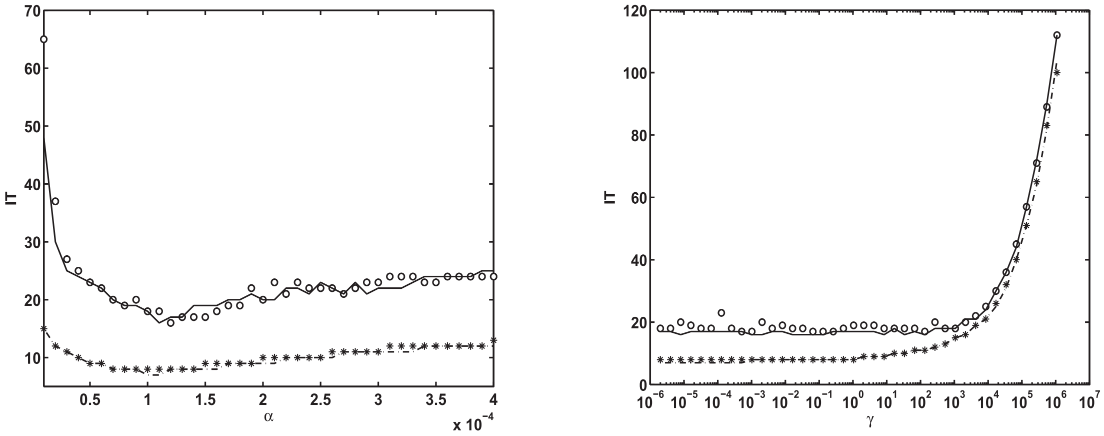

FIG. 3. Pictures of IT vs. α (left) and γ (right) for (I)RHSS-(F)GMRES(a) and (I)RHSS-(F)GMRES(b) methods with their numerically computed optimal parameter γ or α when $m = 96$ for Example 4.1. RHSS-GMRES(a): $-\cdot-\cdot-\cdot$, RHSS-GMRES(b): $***$, IRHSS-FGMRES(a): — and IRHSS-FGMRES(b): $\circ\circ\circ$.

iteration steps are growing gently with respect to the increase of m. In addition, the iteration counts of BD-MINRES are very stable with respect to the mesh size, so that it could outperform the other methods if the mesh is refined further. Because the parameter-free method BD-MINRES vastly outperforms all (I)RHSS-preconditioned (F)GMRES methods that adopt suboptimal choices of the parameters α and γ, practically it should be a good option for solving this stabilized saddle-point problem.

EXAMPLE 4.2 (Bosch *et al.*, 2014). Consider the modified smooth Cahn–Hilliard inpainting problem

$$\begin{cases} \partial_t u = -\Delta\big(\eta\varepsilon\,\Delta u - \frac{1}{\varepsilon}\,\psi_0'(u)\big) + \beta\,(\tilde{f} - u) & \text{in} \quad \Omega, \\ \frac{\partial u}{\partial \vec{n}} = \frac{\partial\,\Delta u}{\partial \vec{n}} = 0 & \text{on} \quad \partial\Omega, \end{cases} \tag{4.3}$$

where $\tilde{f}(x)$ is a given binary image in a domain Ω with its boundary being indicated by $\partial\Omega$, $\varepsilon > 0$ is a constant proportional to the thickness of the interfacial region, $\eta > 0$ is a constant related to the interfacial energy density, $\vec{n}$ is the unit outward normal vector at $\partial\Omega$, $\psi_0(u)$ is a smooth double-well potential and

$$\beta(x) = \begin{cases} 0 & \text{if} \quad x \in \mathbb{D}, \\ \beta_0 & \text{if} \quad x \in \Omega \setminus \mathbb{D}, \end{cases}$$

with β_0 being a fitting constant and $\mathbb{D} \subset \Omega$ being the inpainting domain (damaged or missing parts). Then the variable $u(x,t)$ evolves in time to become a fully inpainted version of $\tilde{f}(x)$ under the system (4.3). The corresponding energy functionals

$$\mathcal{E}_1(u) = \int_\Omega \left(\frac{\eta\varepsilon}{2}\,|\nabla u|^2 + \frac{1}{\varepsilon}\,\psi_0(u)\right)\mathrm{d}x \quad \text{and} \quad \mathcal{E}_2(u) = \frac{1}{2}\int_\Omega \beta\,(\tilde{f} - u)^2\,\mathrm{d}x$$

are divided into the convex parts $\mathcal{E}_{1a}(u)$, $\mathcal{E}_{2a}(u)$ and the concave parts $\mathcal{E}_{1b}(u)$, $\mathcal{E}_{2b}(u)$ such that

$$\mathcal{E}_1(u) = \mathcal{E}_{1a}(u) - \mathcal{E}_{1b}(u) \quad \text{and} \quad \mathcal{E}_2(u) = \mathcal{E}_{2a}(u) - \mathcal{E}_{2b}(u),$$

where

$$\mathcal{E}_{1a}(u) = \int_\Omega \left(\frac{\eta\varepsilon}{2}\,|\nabla u|^2 + \frac{c_1}{2}\,|u|^2\right) \mathrm{d}x, \quad \mathcal{E}_{1b}(u) = \int_\Omega \left(-\frac{1}{\varepsilon}\,\psi_0(u) + \frac{c_1}{2}\,|u|^2\right) \mathrm{d}x$$

and

$$\mathcal{E}_{2a}(u) = \int_\Omega \frac{c_2}{2}\,|u|^2\,\mathrm{d}x, \quad \mathcal{E}_{2b}(u) = \int_\Omega \left(-\frac{\beta}{2}\,(\tilde{f} - u)^2 + \frac{c_2}{2}\,|u|^2\right) \mathrm{d}x.$$

The constants c_1 and c_2 are positive, with c_2 satisfying $c_2 > \beta_0$. These splittings, together with the backward Euler discretization for the time derivative $\partial_t u$ and the Q_1 finite element discretization of rectangular elements for the smooth time-discrete Cahn–Hilliard problem, result in the stabilized saddle-point problem (1.1) of the coefficient matrix

$$\begin{pmatrix} M & \eta\varepsilon K \\ -\eta\varepsilon K & \eta\varepsilon\big[\big(\frac{1}{\tau} + c_2\big)M + c_1 K\big] \end{pmatrix},$$

where M and K are the mass and the stiffness matrices, respectively, and τ is the time-step size. In actual computations we set $\eta = 1$, $\varepsilon = \tau = 0.01$, $c_1 = 300$ and $c_2 = 300000$.

For this example, we have

$$B = M, \quad E = \eta\varepsilon K \quad \text{and} \quad C = \eta\varepsilon\left[\left(\frac{1}{\tau} + c_2\right)M + c_1 K\right].$$

In addition, we set the subvectors to be $f = \text{ones}(1,p)^{\mathrm{T}}$ and $g = \text{ones}(1,q)^{\mathrm{T}}$, so that the right-hand side of the stabilized saddle-point problem (1.1) is $b = \text{ones}(1, p+q)^{\mathrm{T}}$. Here $p = q = m^2$, with m being the number of inner grids, which corresponds to the step size $h = \frac{1}{m+1}$ of the discretization mesh, so that the dimension of the stabilized saddle-point matrix $A \in \mathbb{R}^{n\times n}$ is $n = 2m^2$. Note that the matrix E is highly ill-conditioned, especially when the step size h becomes sufficiently small.

Also, as suggested in Remark 3.3, we take the regularization matrix to be

$$Q = (\alpha\gamma - \omega)C + \gamma E^{\mathrm{T}}E$$

or $Q = \gamma C$ in the (I)RHSS and (I)RHSS-(F)GMRES methods, where γ is a positive constant to be determined according to the problem and the method. These two cases of the regularization matrix Q are denoted case (b) and case (c), respectively, and the corresponding methods are indicated by (I)RHSS(#) and (I)RHSS-(F)GMRES(#), with #=b, c. We refer to Remark 3.3 for a precise description of the RHSS iteration methods and the RHSS preconditioners. Again, note that RHSS and, correspondingly, IRHSS now possess only two arbitrary parameters: one is the iteration parameter α and the other is the regularization parameter γ. The normalization parameter ω is merged with the parameter γ for the special choice, case (c), of the regularization matrix Q.

TABLE 5 *Numerical results for RHSS(#), HSS-GMRES and RHSS-GMRES(#) for Example 4.2*

		m					
Method	Index	64	96	128	192	256	384
RHSS(b)	IT	8	10	11	13	17	30
	CPU	0.32	1.02	2.48	10.60	39.97	216.05
RHSS(c)	IT	7	8	10	14	19	32
	CPU	0.24	0.79	2.16	10.55	41.20	218.61
HSS-GMRES	IT	24	19	22	29	35	47
	CPU	1.41	4.37	11.55	45.57	119.93	568.83
RHSS-GMRES(b)	IT	6	5	6	6	6	6
	CPU	0.32	0.94	2.47	8.24	26.41	149.94
RHSS-GMRES(c)	IT	6	5	6	6	6	6
	CPU	0.32	0.94	2.49	8.21	24.70	150.12

In the implementations of both IHSS and IRHSS used either as a linear solver or as a right preconditioner, the involved linear subsystems in the direct-splitting form (Bai & Rozložník, 2015) are solved iteratively by the PCG method starting from the initial guess 0, with the stopping tolerance 10^{-10} with respect to IHSS and 0.1 with respect to all other methods. Here we should address that in order to guarantee the convergence of the inexact iteration methods the linear subsystems involved in IHSS need to be solved much more accurately than those involved in other methods such as IRHSS(#), IHSS-FGMRES and IRHSS-FGMRES(#). According to preconditioners adopted for the PCG method, except for the matrix $\alpha I + B$ which is preconditioned by its MIC factorization, all other matrices are preconditioned by their AMG approximations.

In addition, the MINRES method is preconditioned by the optimal block-diagonal preconditioner proposed in Bosch *et al.* (2014), i.e., Diag$(\widehat{\widehat{M}}, \widehat{\widehat{K}}M^{-1}\widehat{\widehat{K}})$, and the GMRES method is preconditioned by the optimal block-triangular preconditioner of the form in (4.1), with the (1,1)-block matrix $\widehat{\widehat{B}} = \widehat{\widehat{M}}$ and the (2,2)-block matrix $\widehat{\widehat{\widehat{S}}} = \widehat{\widehat{K}}M^{-1}\widehat{\widehat{K}}$. Here $\widehat{\widehat{M}}$ is the 25-step Chebyshev semi-iteration approximation of M and $\widehat{\widehat{K}}$ is the AMG approximation of $\eta\varepsilon K + \sqrt{\eta\varepsilon\left(\frac{1}{\tau} + c_2\right)}\, M$.

In Table 5 we report iteration counts and CPU times for RHSS(#), HSS-GMRES and RHSS-GMRES(#), for which the parameters α and (or) γ are taken to be the experimentally computed optimal ones that minimize the total number of iteration steps of the corresponding method; see Table 6. We remark that the optimal HSS iteration method fails to converge within 1900 iteration steps so that the corresponding computing results are not reported here. These numerical results show that RHSS(#) and RHSS-GMRES(#) significantly outperform HSS and HSS-GMRES in both iteration counts and CPU times. In addition, both RHSS-GMRES(a) and RHSS-GMRES(b) take smaller iteration step and less CPU time than all other methods and they also show convergence behavior independent of the mesh size. This phenomenon, just as demonstrated in Theorem 3.2, could be caused by the tightly clustered eigenvalues of the RHSS-preconditioned matrices, especially when the parameter α is small enough; see Figs 4 and 5 for an intuitive visualization. Admittedly, we should point out that RHSS-GMRES(b) and RHSS-GMRES(c), while requiring a constant number of iteration steps to converge regardless of the mesh size, scale poorly in terms of CPU time.

TABLE 6 *Experimentally computed optimal values of the iteration parameter α and (or) the regularization parameter γ in HSS, RHSS(#), HSS-GMRES and RHSS-GMRES(#) for Example 4.2*

		m					
Method	Index	64	96	128	192	256	384
HSS	α	0.028	0.028	0.023	0.014	0.010	0.007
	γ	3800	6000	6000	10000	13000	16000
RHSS(b)	α	2.0E−4	1.1E−4	1.0E−4	5.0E−5	4.0E−5	3.0E−5
	γ	1.0	1.1	1.3	1.8	2.1	2.7
RHSS(c)	α	2.0E−4	1.3E−4	9.0E−5	5.0E−5	4.0E−5	3.0E−5
HSS-GMRES	α	0.065	0.057	0.031	0.020	0.014	0.009
	γ	0.001	0.001	0.001	0.001	0.001	0.001
RHSS-GMRES(b)	α	2.0E−4	1.0E−4	5.0E−5	2.0E−5	1.1E−5	6.0E−6
	γ	0.002	0.002	0.002	0.002	0.002	0.002
RHSS-GMRES(c)	α	2.0E−4	1.0E−4	5.0E−5	2.0E−5	1.1E−5	6.0E−6

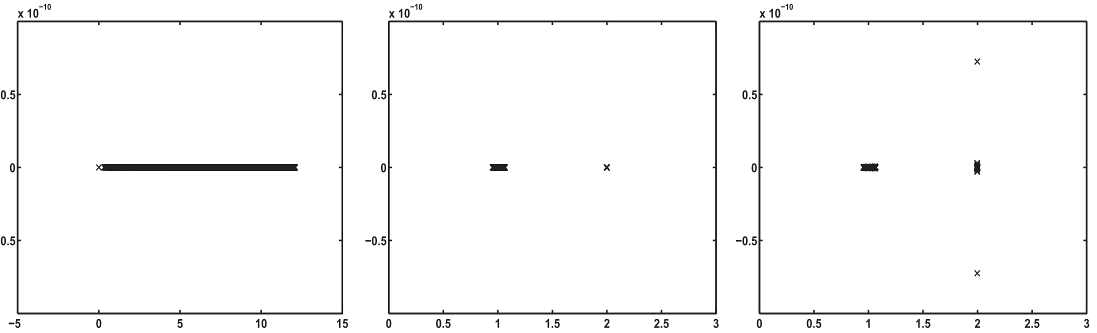

FIG. 4. Eigenvalue distributions of the original coefficient matrix (left), the RHSS(b)-preconditioned matrix (middle) and the RHSS(c)-preconditioned matrix (right), with their numerically computed optimal parameters α and γ, respectively, when $m = 96$ for Example 4.2.

In Table 7 we list the results for the inexact implementations of these methods, as well as for BT-GMRES and BD-MINRES. From this table we observe that IRHSS(#) and IRHSS-FGMRES(#) take many fewer iteration steps and much less CPU times than IHSS and IHSS-FGMRES. Both IRHSS-FGMRES(b) and IRHSS-FGMRES(c) significantly outperform IRHSS(b), IRHSS(c) and BD-MINRES and they show convergence behavior almost independent of the mesh size. Although BT-GMRES also shows h-independent convergence behavior, its iteration step and CPU time are, roughly speaking, slightly more than either IRHSS-FGMRES(b) or IRHSS-FGMRES(c). Despite this, in view of the moderate computing times and the parameter-free property, BT-GMRES is still an effective method for solving the stabilized saddle-point problem for this example. In addition, IRHSS-FGMRES(c) requires negligibly smaller iteration step and costs slightly less CPU time than IRHSS-FGMRES(b). By comparing Table 7 with Table 5 we see again that all inexact methods can achieve much higher computational efficiency than their exact counterparts because the inexact iteration methods for solving the linear subsystems can save lots of computing time compared with the direct decomposition methods.

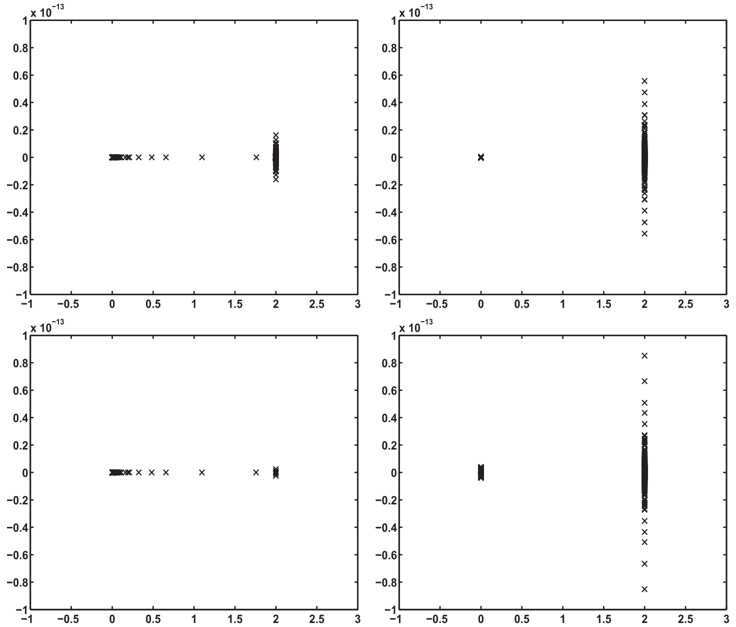

FIG. 5. Eigenvalue distributions of the RHSS(b)-preconditioned matrix with $\alpha = 1.0\text{E}{-8}$ (upper left), $\alpha = 1.0\text{E}{-12}$ (upper right) and $\gamma = 0.001$, and eigenvalue distributions of the RHSS(c)-preconditioned matrix with $\alpha = 1.0\text{E}{-8}$ (bottom left), $\alpha = 1.0\text{E}{-12}$ (bottom right) and $\gamma = 0.002$, when $m = 96$ for Example 4.2.

And among all these methods IRHSS-FGMRES(b) and IRHSS-FGMRES(c) are the best for solving this stabilized saddle-point problem. In addition, we remark that both IRHSS-FGMRES(b) and IRHSS-FGMRES(c) scale nicely in terms of CPU time.

Going back to Table 6, we see that for RHSS(b) and RHSS(c) the optimal values of the parameter α are quite small and monotonically decreasing with respect to the mesh size m, with orders of magnitude ranging from $\mathcal{O}(10^{-4})$ to $\mathcal{O}(10^{-5})$. In contrast, the optimal values of the parameter γ are relatively large and monotonically increasing when the mesh size m is growing; in particular, γ changes very quickly for RHSS(b). For different values of the mesh size m, the orders of γ are approximately reciprocal to those of α. In fact, it holds that $\alpha\gamma \in [0.48, 0.76]$ for RHSS(b) and $\alpha\gamma \in [0.84, 1.43] \times 10^{-4}$ for RHSS(c). For RHSS-GMRES(b) and RHSS-GMRES(c), as predicted in Theorem 3.2, the optimal values of the parameter γ are small and those of the parameter α are even smaller for all tested mesh sizes; however, the former remains exactly the same constant, 0.001 for RHSS-GMRES(b) and 0.002 for RHSS-GMRES(c), while the latter decreases gradually when the mesh size m is increasing. This is, in nature, equivalent to strategy I recommended in Section 3 for determining the pair (α, γ) of optimal parameters, with ω there being replaced by γ here. We should point out again that for this example, strategy II in Section 3 works, but it does not work very well, especially when the mesh sizes are far away from the referenced smallest one, i.e., $m = 64$. This phenomenon is evident from the data in Table 8, which, for all mesh sizes m, are produced by adopting the values of the parameters α and γ in Table 6 corresponding to the smallest mesh size $m = 64$. Indeed, the data in Table 8 do not match aptly with the data in Tables 5 and 7, especially when m is much larger than 64. The reason may be that the RHSS method is quite sensitive to the parameter α or γ, which, in turn, heavily relies on the mesh size

Table 7 *Numerical results for IHSS, IRHSS(#), IHSS-FGMRES, IRHSS-FGMRES(#), BT-GMRES and BD-MINRES for Example* 4.2

		m					
Method	Index	64	96	128	192	256	384
IHSS	IT	1825	2314	2762	3647	4678	6969
	CPU	54.41	152.32	352.29	1173.83	2974.32	10571.63
IRHSS(b)	IT	13	20	23	35	46	63
	CPU	0.16	0.41	0.85	2.95	7.03	22.14
IRHSS(c)	IT	13	17	24	35	46	67
	CPU	0.15	0.36	0.88	2.85	6.58	22.74
IHSS-FGMRES	IT	27	20	22	31	36	51
	CPU	0.95	1.17	2.28	5.56	12.46	43.70
IRHSS-FGMRES(b)	IT	8	8	9	13	10	14
	CPU	0.18	0.26	0.52	2.18	3.30	11.30
IRHSS-FGMRES(c)	IT	8	8	9	9	10	13
	CPU	0.14	0.25	0.51	1.66	3.35	10.57
BT-GMRES	IT	10	10	11	11	11	12
	CPU	0.30	0.61	1.17	3.07	5.57	14.26
BD-MINRES	IT	27	31	33	35	37	38
	CPU	0.37	0.90	1.62	4.58	8.79	20.50

m, for this problem; see Fig. 6. However, from Table 8 we see that IRHSS-FGMRES(b) and IRHSS-FGMRES(c) outperform (for $m \leq 256$) or about comparable with (for $m = 384$) both BT-GMRES and BD-MINRES in CPU time, and their iteration steps are growing mildly with respect to the increase of m.

Example 4.3 (Benzi & Ng, 2006). Consider the nonlinear image restoration problem

$$\min_{y} \left\| \tilde{f} - s(Ky) \right\|_2^2 + \beta \|y\|_2^2, \tag{4.4}$$

where $\tilde{f}$ and $y \in \mathbb{R}^p$ represent the observed and the original images, respectively, $s : \mathbb{R}^p \to \mathbb{R}$ denotes a nonlinear point spread function and $K = ([K]_{ij}) \in \mathbb{R}^{p \times p}$ is a blurring matrix. When this regularized nonlinear least-squares problem (4.4) is solved by the Gauss–Newton iteration method, at each step for the currently available approximant $y_c = ([y_c]_1, [y_c]_2, \ldots, [y_c]_p)$, we need to solve a linear system of the form

$$\left(\beta I + K^{\mathrm{T}} D^2 K\right) y = K^{\mathrm{T}} D \left[\tilde{f} - s(Ky_c) + DKy_c\right]$$

to obtain the next approximant, where $D = \operatorname{diag}([D]_1, [D]_2, \ldots, [D]_p) \in \mathbb{R}^{p \times p}$ is a diagonal matrix with its diagonal entries being given by

$$[D]_i = \left. \frac{\partial s}{\partial \xi} \right|_{\xi = \sum_{j=1}^{p} [K]_{ij} [y_c]_j}, \quad i = 1, 2, \ldots, p.$$

TABLE 8 *Numerical results for RHSS(#), IRHSS(#), RHSS-GMRES(#) and IRHSS-FGMRES(#) for Example 4.2, which are produced by adopting the values of the parameters α and γ in Table 6 corresponding to the case $m = 64$*

		m					
Method	Index	64	96	128	192	256	384
RHSS(b)	IT	8	12	22	50	89	200
	CPU	0.32	1.13	4.28	31.72	132.37	817.40
RHSS(c)	IT	7	13	22	50	89	200
	CPU	0.24	1.08	3.89	29.70	125.63	780.65
RHSS-GMRES(b)	IT	6	6	9	14	18	28
	CPU	0.32	1.04	3.31	19.78	64.89	277.21
RHSS-GMRES(c)	IT	6	6	9	14	18	28
	CPU	0.32	1.12	3.26	19.93	65.36	277.61
IRHSS(b)	IT	13	17	23	50	89	200
	CPU	0.16	0.37	0.88	4.08	12.92	103.18
IRHSS(c)	IT	13	15	23	51	89	200
	CPU	0.15	0.33	0.88	4.15	13.00	66.16
IRHSS-FGMRES(b)	IT	8	9	13	19	25	38
	CPU	0.18	0.29	0.74	2.46	5.88	23.19
IRHSS-FGMRES(c)	IT	8	9	13	20	26	37
	CPU	0.14	0.29	0.72	2.52	6.08	22.30

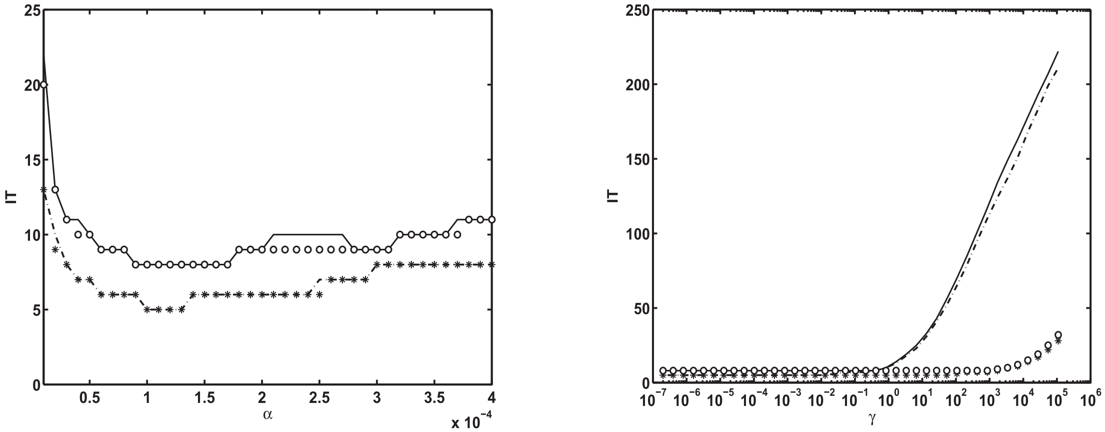

FIG. 6. Pictures of IT vs. α (left) and γ (right) for (I)RHSS-(F)GMRES(b) and (I)RHSS-(F)GMRES(c) methods with their numerically computed optimal parameter γ or α when $m = 96$ for Example 4.2. RHSS-GMRES(b): $***$, RHSS-GMRES(c): $- \cdot - \cdot -\cdot$, IRHSS-FGMRES(b): $\circ\circ\circ$ and IRHSS-FGMRES(c): —.

This linear system can be equivalently reformulated into the stabilized saddle-point problem (1.1), in which

$$B = D^{-2}, \quad E = K, \quad C = \beta I$$

and

$$f = D^{-1}\big[\tilde{f} - s(Ky_c)\big] + Ky_c, \quad g = 0.$$

In actual computations we set

$$\tilde{f} = \left[\frac{254}{p} : \frac{254}{p} : 254\right]^{\mathrm{T}}$$

and

$$y_c = \left[\left[0.5 + \frac{508}{p} : \frac{508}{p} : 254.5\right], \left[254.5 : -\frac{508}{p} : 0.5 + \frac{508}{p}\right]\right]^{\mathrm{T}}$$

and take $\beta = 10^{-3}$,

$$s(\xi) = 30\,\log(\xi), \quad [K]_{ij} = \frac{1}{\sqrt{2\pi}\,\mu}\,\exp\left(\frac{-|i-j|^2}{2\mu^2}\right), \quad i,j = 1, 2, \ldots, p,$$

with $\mu = 2$. Here, the notation ':' is a standard MATLAB symbol used in indicating a row vector of the form

$$[x_f : \delta x : x_l] \equiv [x_f, x_f + \delta x, \ldots, x_f + k\,\delta x, \ldots, x_l],$$

for which x_f and x_l are the first and the last elements, δx is a prescribed increment such that $x_l - x_f$ is its multiple and k is a non-negative integer ranging from 0 to $\frac{x_l - x_f}{\delta x}$.

For this example we have $p = q$, so that the dimension of the stabilized saddle-point matrix $A \in \mathbb{R}^{n\times n}$ is $n = 2p$. Note that the Toeplitz matrix E is highly ill-conditioned with rapidly decaying singular values so that it is almost rank-deficient, especially when the problem size p becomes sufficiently large.

Also as suggested in Remark 3.3 we take the regularization matrix to be

$$Q = (\alpha\gamma - \omega)C + \gamma E^{\mathrm{T}}E - \alpha I$$

in the (I)RHSS and (I)RHSS-(F)GMRES methods, where γ is a positive constant to be determined according to the problem and the method. This case of the regularization matrix Q is denoted case (a), and the corresponding methods are indicated by (I)RHSS(a) and (I)RHSS-(F)GMRES(a). We refer to Remark 3.3 for a precise description of the RHSS(a) iteration method and the RHSS(a) preconditioner. Note that RHSS(a) and, correspondingly, IRHSS(a) now possess only two arbitrary parameters: one is the iteration parameter α and the other is the regularization parameter γ.

In the implementations of both IHSS and IRHSS used either as a linear solver or as a right preconditioner, the involved linear subsystems in the residual-updating form (Bai & Rozložník, 2015) are solved iteratively by the PCG method starting from the initial guess 0, with the stopping tolerance 0.1 for IHSS, and 0.01 for IRHSS(a), except for the case $p = 512$ for IRHSS(a) in which we use the stopping tolerance 0.1. According to preconditioners adopted for the PCG method, in IHSS the matrix $\alpha I + \frac{1}{\alpha}E^{\mathrm{T}}E$ is preconditioned by the circulant matrix $\alpha I + \frac{1}{\alpha}T^2$ and in IRHSS the matrix $C + \frac{1}{\alpha}E^{\mathrm{T}}E$ is preconditioned

TABLE 9 *Numerical results for HSS, RHSS(a), HSS-GMRES and RHSS-GMRES(a) for Example* 4.3

		p					
Method	Index	512	1024	2048	4096	8192	16384
HSS	IT	501	490	489	488	488	488
	CPU	0.71	2.06	7.56	29.66	134.27	693.36
RHSS(a)	IT	154	144	92	51	29	24
	CPU	0.28	0.64	1.54	4.22	17.27	108.29
HSS-GMRES	IT	96	97	105	125	173	235
	CPU	0.48	1.27	8.10	22.27	116.82	705.57
RHSS-GMRES(a)	IT	40	41	33	25	18	15
	CPU	0.17	0.42	1.56	4.21	18.61	112.62

TABLE 10 *Experimentally computed optimal values of the iteration parameter α and (or) the regularization parameter γ in HSS, RHSS(a), HSS-GMRES and RHSS-GMRES(a) for Example* 4.3

		p					
Method	Index	512	1024	2048	4096	8192	16384
HSS	α	0.70	0.76	0.73	0.75	0.75	0.75
RHSS(a)	γ	0.20	0.18	0.11	0.06	0.04	0.02
	α	2.60	2.90	4.80	9.00	16.00	28.00
HSS-GMRES	α	0.56	0.90	0.90	0.80	0.66	0.60
RHSS-GMRES(a)	γ	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	α	2.00	2.10	3.50	5.80	20.00	17.00

by the circulant matrix $\beta I+\frac{1}{\alpha}T^2$, where T is the Strang circulant approximation to the matrix K; see, e.g., Strang (1986). In addition, the MINRES and the GMRES methods are preconditioned by block-diagonal and block-triangular matrices of the forms in (4.1), respectively, with the (1,1)-block matrix $\widehat{\widehat{B}} = D^{-2}$ and the (2,2)-block matrix $\widehat{\widehat{S}} = \beta I + \bar{d}^2 T^2$, where $\bar{d}$ is the arithmetic mean of all diagonal elements of the matrix D and T is the same Strang circulant approximation to the matrix K as above. Note that these circulant matrices can be effectively inverted by making use of fast Fourier transforms.

In Table 9 we report iteration counts and CPU times for HSS, RHSS(a), HSS-GMRES and RHSS-GMRES(a), for which the parameters α and (or) γ are taken to be the experimentally computed optimal ones that minimize the total number of iteration steps of the corresponding method; see Table 10. The results in this table show that RHSS(a) and RHSS-GMRES(a) significantly outperform HSS and HSS-GMRES, respectively, in both iteration counts and CPU times. Roughly speaking, RHSS(a) and RHSS-GMRES(a) take the fewest iteration steps and least CPU times, and their iteration steps are monotonically decreasing when the problem size p is growing, with RHSS-GMRES(a) requiring much smaller iteration step than, but costing about the same CPU time as, RHSS(a). This favorable numerical behavior shown by RHSS-GMRES(a) should be due to the tightly clustered eigenvalues of the RHSS-preconditioned matrices; see Fig. 7 for an intuitive visualization. Again, we refer to Theorem 3.2 for a theoretical explanation.

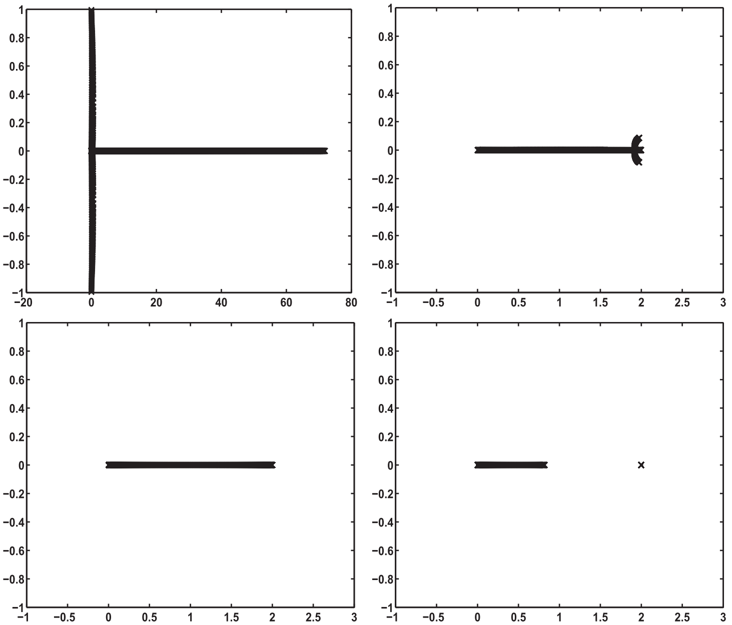

Fig. 7. Eigenvalue distributions of the original coefficient matrix (upper left), and the RHSS(a)-preconditioned matrices with $\alpha = 20$ (upper right), $\alpha = 2.0\text{E}-6$ (bottom left) and $\alpha = 2.0\text{E}-14$ (bottom right), when $\gamma = 0.0001$ and $p = 8192$, for Example 4.3.

Table 11 *Numerical results for IHSS, IRHSS(a), IHSS-FGMRES, IRHSS-FGMRES(a), BT-GMRES and BD-MINRES for Example* 4.3

		p					
Method	Index	512	1024	2048	4096	8192	16384
IHSS	IT	1213	1085	1153	1130	1219	1313
	CPU	1.40	1.81	3.21	6.07	14.05	29.55
IRHSS(a)	IT	377	813	1333	1009	640	386
	CPU	0.65	2.19	5.14	6.84	8.72	10.19
IHSS-FGMRES	IT	105	116	139	188	222	340
	CPU	0.40	0.61	3.82	10.47	15.08	52.97
IRHSS-FGMRES(a)	IT	44	46	33	30	21	17
	CPU	0.20	0.27	0.38	0.56	0.64	0.93
BT-GMRES	IT	87	116	149	191	256	352
	CPU	0.23	0.52	4.00	9.22	22.44	64.73
BD-MINRES	IT	389	666	1105	1782	2691	3776
	CPU	0.34	0.41	1.23	3.23	9.31	27.59

TABLE 12 *Numerical results for RHSS(a), IRHSS(a), RHSS-GMRES(a) and IRHSS-FGMRES(a) for Example 4.3, which are produced by adopting the values of the parameters α and γ in Table 10 corresponding to the case $p = 512$*

		p					
Method	Index	512	1024	2048	4096	8192	16384
RHSS(a)	IT	154	152	147	146	145	145
	CPU	0.28	0.76	2.25	8.93	38.33	193.25
RHSS-GMRES(a)	IT	40	42	41	41	41	41
	CPU	0.17	0.45	1.62	6.20	27.06	149.93
IRHSS(a)	IT	377	784	784	1980	2520	2819
	CPU	0.65	2.12	5.85	15.35	39.09	83.69
IRHSS-FGMRES(a)	IT	44	46	42	42	42	42
	CPU	0.20	0.29	0.55	0.92	1.63	3.22

In Table 11 we list the results for the inexact implementations of these methods, as well as for BT-GMRES and BD-MINRES. From this table we observe that IRHSS(a) and IRHSS-FGMRES(a) take many fewer iteration steps and much less CPU times than IHSS and IHSS-FGMRES, respectively. IRHSS-FGMRES(a) significantly outperforms IRHSS(a), BT-GMRES and BD-MINRES in terms of both iteration counts and CPU times. By comparing Table 11 with Table 9 we see again that all inexact methods can achieve much higher computational efficiency than their exact counterparts and among all these methods IRHSS-FGMRES(a) is the best for solving this stabilized saddle-point problem.

Going back to Table 10 we see that for RHSS(a) the optimal values of the parameter α are monotonically increasing, while those of the parameter γ are monotonically decreasing, when the problem size p is growing, with the orders of γ being approximately reciprocal to those of α. In fact, it holds that $\alpha\gamma \in [0.52, 0.64]$. For RHSS-GMRES(a) the optimal values of the parameter γ

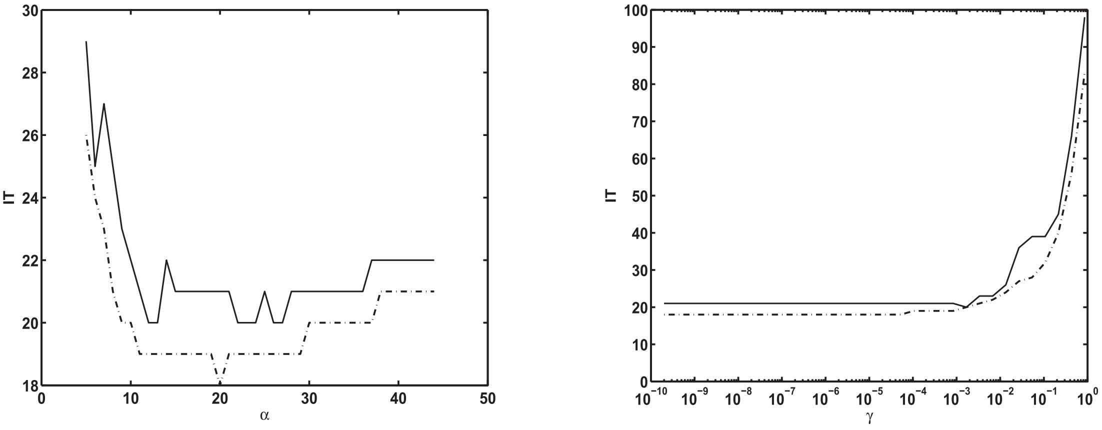

FIG. 8. Pictures of IT vs. α (left) and γ (right) for (I)RHSS-(F)GMRES(a) methods with their numerically computed optimal parameter γ or α when $p = 8192$ for Example 4.3. RHSS-GMRES(a): — · — · —·, and IRHSS-FGMRES(a): —.

are small and those of the parameter α are much larger for all tested problem sizes; however, the former remains exactly the same constant 0.0001, while the latter increases gradually when the problem size p is increasing. This is, in nature, equivalent to strategy I recommended in Section 3 for determining the pair (α, γ) of optimal parameters, with ω there being replaced by γ here. We should point out that for this example, strategy II in Section 3 works reasonably well. This phenomenon is evident from the data in Table 12, which, for all problem sizes p, are produced by adopting the values of the parameters α and γ in Table 10 corresponding to the smallest problem size $p = 512$. Indeed, the data in Table 12 match quite well with the data in Tables 9 and 11, especially when p is not far away from 512. The reason may be that RHSS(a) is not so sensitive to the parameters α and γ, which are, in turn, relatively independent of the problem size p, for this problem; see Fig. 8. Moreover, from Table 12 we see that IRHSS-FGMRES(a) even significantly outperforms both BT-GMRES and BD-MINRES in iteration counts and CPU times and its iteration step remains almost constant for different p.

5. Concluding remarks

The RHSS iteration method for solving standard saddle-point problems has been further developed to solve stabilized saddle-point problems. As a solver, this iteration method converges unconditionally to the unique solution of the stabilized saddle-point problem, and as a preconditioner, the corresponding preconditioned matrix shows clustered eigenvalue distribution, which can lead to fast convergence of the preconditioned Krylov subspace iteration methods such as GMRES. Numerical experiments have verified that the RHSS method significantly outperforms the HSS method in terms of both iteration steps and computing times, when they are employed as either linear iteration solvers or matrix splitting preconditioners. This numerical property is equally exhibited by their inexact counterparts, IHSS and IRHSS. In particular, it is remarkable that the IRHSS-preconditioned FGMRES method shows constant iteration step independent of the problem size and even has considerably higher computational efficiency than the preconditioned MINRES and the preconditioned GMRES method incorporated with an optimal block-diagonal and an optimal block-triangular preconditioner, respectively. Hence, the RHSS iteration method, especially the IRHSS-FGMRES method, can be a useful tool for solving certain types of large sparse stabilized saddle-point problems.

Acknowledgements

The author is very much indebted to Kang-Ya Lu for running the numerical results. He is also thankful to the referees for their constructive comments and valuable suggestions, which greatly improved the original manuscript of this paper.

Funding

National Natural Science Foundation, P. R. China (11671393).

References

Axelsson, O. (1996) *Iterative Solution Methods*. Cambridge: Cambridge University Press.

Bai, Z.-Z. (2006) Structured preconditioners for nonsingular matrices of block two-by-two structures. *Math. Comput.*, **75**, 791–815.

Bai, Z.-Z. (2015) Motivations and realizations of Krylov subspace methods for large sparse linear systems. *J. Comput. Appl. Math.*, **283**, 71–78.

Bai, Z.-Z. & Benzi, M. (2017) Regularized HSS iteration methods for saddle-point linear systems. *BIT Numer. Math.*, **57**, 287–311.

Bai, Z.-Z. & Golub, G. H. (2007) Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. *IMA J. Numer. Anal.*, **27**, 1–23.

Bai, Z.-Z., Golub, G. H. & Li, C.-K. (2007) Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. *Math. Comput.*, **76**, 287–298.

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

Bai, Z.-Z., Golub, G. H. & Pan, J.-Y. (2004) Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. *Numer. Math.*, **98**, 1–32.

Bai, Z.-Z. & Hadjidimos, A. (2014) Optimization of extrapolated Cayley transform with non-Hermitian positive definite matrix. *Linear Algebra Appl.*, **463**, 322–339.

Bai, Z.-Z. & Rozložník, M. (2015) On the numerical behavior of matrix splitting iteration methods for solving linear systems. *SIAM J. Numer. Anal.*, **53**, 1716–1737.

Beik, F.-P. A., Benzi, M. & Chaparpordi, S.-H. A. (2017) On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems. *J. Comput. Appl. Math.*, **326**, 15–30.

Benzi, M. & Golub, G. H. (2004) A preconditioner for generalized saddle point problems. *SIAM J. Matrix Anal. Appl.*, **26**, 20–41.

Benzi, M., Golub, G. H. & Liesen, J. (2005) Numerical solution of saddle point problems. *Acta Numer.*, **14**, 1–137.

Benzi, M. & Ng, M. K. (2006) Preconditioned iterative methods for weighted Toeplitz least squares problems. *SIAM J. Matrix Anal. Appl.*, **27**, 1106–1124.

Benzi, M. & Simoncini, V. (2006) On the eigenvalues of a class of saddle point matrices. *Numer. Math.*, **103**, 173–196.

Bosch, J., Kay, D., Stoll, M. & Wathen, A. J. (2014) Fast solvers for Cahn–Hilliard inpainting. *SIAM J. Imaging Sci.*, **7**, 67–97.

Brezzi, F. & Fortin, M. (1991) *Mixed and Hybrid Finite Element Methods*. New York and London: Springer.

Eisenstat, S. C., Elman, H. C. & Schultz, M. H. (1983) Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, **20**, 345–357.

Elman, H. C., Silvester, D. J. & Wathen, A. J. (2002) Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations. *Numer. Math.*, **90**, 665–688.

Elman, H. C., Silvester, D. J. & Wathen, A. J. (2014) *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, 2nd edn. Oxford: Oxford University Press.

Fortin, M. & Glowinski, R. (1983) *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary Value Problems*. Amsterdam: North-Holland.

Golub, G. H. & Van Loan, C. F. (1996) *Matrix Computations*, 3rd edn. Baltimore: The Johns Hopkins University Press.

Greenbaum, A. (1997) *Iterative Methods for Solving Linear Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Herzog, R. & Soodhalter, K. M. (2017) A modified implementation of MINRES to monitor residual subvector norms for block systems. *SIAM J. Sci. Comput.*, **39**, A2645–A2663.

Kellogg, R. B. (1963) Another alternating-direction-implicit method. *J. Soc. Indust. Appl. Math.*, **11**, 976–979.

Murphy, M. F., Golub, G. H. & Wathen, A. J. (2000) A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, **21**, 1969–1972.

Pearson, J. W., Stoll, M. & Wathen, A. J. (2014) Preconditioners for state-constrained optimal control problems with Moreau–Yosida penalty function. *Numer. Linear Algebra Appl.*, **21**, 81–97.

Perugia, I. & Simoncini, V. (2000) Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numer. Linear Algebra Appl.*, **7**, 585–616.

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, 2nd edn. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Saad, Y. & Schultz, M. H. (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869.

Silvester, D. J. & Kechkar, N. (1990) Stabilised bilinear-constant velocity-pressure finite elements for the conjugate gradient solution of the Stokes problems. *Comput. Methods Appl. Mech. Eng.*, **79**, 71–86.

Silvester, D. J. & Wathen, A. J. (1994) Fast iterative solution of stabilised Stokes systems II: Using general block preconditioners. *SIAM J. Numer. Anal.*, **31**, 1352–1367.

Simoncini, V. & Benzi, M. (2004) Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems. *SIAM J. Matrix Anal. Appl.*, **26**, 377–389.

Strang, G. (1986) A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.*, **74**, 171–176.

Varga, R. S. (1962) *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Wathen, A. J. (2015) Preconditioning. *Acta Numer.*, **24**, 329–376.

Wathen, A. J. & Silvester, D. J. (1993) Fast iterative solution of stabilised Stokes systems I: Using simple diagonal preconditioners. *SIAM J. Numer. Anal.*, **30**, 630–649.