BIT Numer Math (2017) 57:287–311
DOI 10.1007/s10543-016-0636-7

BIT

Regularized HSS iteration methods for saddle-point linear systems

Zhong-Zhi Bai[1] · **Michele Benzi**[2]

Received: 29 January 2016 / Accepted: 20 October 2016 / Published online: 11 November 2016

Abstract We propose a class of regularized Hermitian and skew-Hermitian splitting methods for the solution of large, sparse linear systems in saddle-point form. These methods can be used as stationary iterative solvers or as preconditioners for Krylov subspace methods. We establish unconditional convergence of the stationary iterations and we examine the spectral properties of the corresponding preconditioned matrix. Inexact variants are also considered. Numerical results on saddle-point linear systems arising from the discretization of a Stokes problem and of a distributed control problem show that good performance can be achieved when using inexact variants of the proposed preconditioners.

Keywords Saddle-point linear system · Hermitian and skew-Hermitian splitting · Iterative methods · Inexact implementation · Preconditioning · Convergence

Communicated by Lars Eldén.

Supported by The National Natural Science Foundation for Creative Research Groups (No. 11321061) and The National Natural Science Foundation (No. 11671393), P.R. China, and by DOE (Office of Science) Grant ERKJ247.

✉ Zhong-Zhi Bai
bzz@lsec.cc.ac.cn

Michele Benzi
benzi@mathcs.emory.edu

1 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, People's Republic of China

2 Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA

Mathematics Subject Classification 65F08 · 65F10 · 65F15 · 65F22 · CR: G1.3

1 Introduction

Consider the following linear system in saddle-point form:

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

where $B \in \mathbb{C}^{p\times p}$ is a Hermitian positive definite matrix, $E \in \mathbb{C}^{p\times q}$ is a rectangular matrix of full column rank, $E^* \in \mathbb{C}^{q\times p}$ is the conjugate transpose of E, $O \in \mathbb{C}^{q\times q}$ is the zero matrix, and $f \in \mathbb{C}^p$, $g \in \mathbb{C}^q$, with p and q being two given positive integers such that $p \geq q$. These assumptions guarantee the existence and uniqueness of the solution of the saddle-point linear system (1.1); see [10]. 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 $n = p + q$. For background information on saddle-point linear systems in scientific and engineering applications see, e.g., [2,10,12,14,16,24].

Based on the *Hermitian and skew-Hermitian* (HS) splitting of the matrix $A \in \mathbb{C}^{n\times n}$,

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

in [9] Benzi and Golub proposed to apply the *Hermitian and skew-Hermitian splitting* (HSS) iteration method in [7] to solve the saddle-point linear system (1.1). They proved that the HSS iteration method converges unconditionally to the unique solution of the saddle-point linear system (1.1), thus extending the HSS convergence theory for (non-Hermitian) positive definite matrices to a large class of positive semidefinite matrices. Moreover, in [22] Simoncini and Benzi analyzed the spectral properties of the preconditioned matrix corresponding to the HSS preconditioner; see also [11]. To improve the convergence rate of the HSS iteration method, Bai, Golub and Pan presented in [8] the *preconditioned HSS* (PHSS) iteration method by first preconditioning the saddle-point matrix $A \in \mathbb{C}^{n\times n}$ and then iterating with the HSS iteration scheme, and Bai and Golub established in [4] the *accelerated HSS* (AHSS) iteration method by extrapolating the HSS iteration sequence with two different parameters. The PHSS and the AHSS iteration methods also induce the PHSS and the AHSS preconditioners, respectively, for the saddle-point matrix $A \in \mathbb{C}^{n\times n}$, which were shown to be quite effective in accelerating the Krylov subspace iteration methods such as GMRES [20,21].

In order to further improve the convergence behavior of the HSS iteration method, in this paper we propose a *regularized HSS* (RHSS) iteration method by introducing a Hermitian positive semidefinite matrix, called the *regularization matrix*, in the HS splitting in (1.2). This regularization strategy may considerably improve the conditioning of the inner linear sub-systems involved in the two-half steps of the HSS iteration method, so that the corresponding *inexact RHSS* (IRHSS) preconditioner can be expected to be more effective and robust when applied to the solution of the

saddle-point linear system (1.1). On the theoretical side, we prove that the RHSS iteration method converges unconditionally to the unique solution of the saddle-point linear system (1.1), 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. From the computational viewpoint, we show experimentally that the stationary RHSS iteration method and the RHSS-preconditioned GMRES method outperform the stationary HSS iteration method and the HSS-preconditioned GMRES method in terms of both iteration counts and computing time, and that the IRHSS iteration method and the IRHSS-preconditioned (flexible) GMRES method have higher computing efficiency than their exact counterparts in terms of computing time, respectively. Hence, our experiments show that the RHSS and the IRHSS methods can be efficient and robust for solving the saddle-point linear system (1.1) when they are used as preconditioners for certain types of saddle-point problems.

The organization of the paper is as follows. In Sect. 2 we present the algorithmic description of the RHSS iteration method. In Sect. 3, we prove the unconditional convergence of the RHSS iteration method and analyze clustering properties of the eigenvalues of the RHSS-preconditioned matrix. Numerical results are given in Sect. 4. Finally, in Sect. 5 we briefly summarize our main conclusions.

2 The RHSS iteration method

In this section, we derive the RHSS iteration method for the saddle-point linear system (1.1) and the RHSS preconditioning matrix for the saddle-point matrix $A \in \mathbb{C}^{n\times n}$.

To this end, for a given Hermitian positive semidefinite matrix $Q \in \mathbb{C}^{q\times q}$ we split the saddle-point matrix $A \in \mathbb{C}^{n\times n}$ in (1.1) into

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

We call the collection of these two splittings a *regularized Hermitian and skew-Hermitian* (RHS) splitting, as the matrix Q plays a regularizing role in the HS splitting in (1.2). Also, we call Q the regularization matrix. Note that when $Q = O$, the RHS splitting in (2.1) automatically reduces to the HS splitting in (1.2). The RHS splitting in (2.1) of the matrix A naturally leads to equivalent reformulations of the saddle-point linear system (1.1) into two systems of fixed-point equations as follows:

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

where $\alpha > 0$ is a prescribed iteration parameter and I is the identity matrix. By iterating alternatively between these two fixed-point systems as

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

and

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

or in their blockwise forms

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

and

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

we obtain the regularized HSS (or in short, RHSS) iteration method for solving the saddle-point linear system (1.1) as follows.

The RHSS iteration method Let α be a positive constant. 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 $y^{(k+1/2)} \in \mathbb{C}^p$ from the linear sub-system

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

(ii) compute

$$f^{(k+1/2)} = (\alpha I - B)y^{(k+1/2)} + f \quad \text{and} \quad g^{(k+1/2)} = (\alpha I + Q)z^{(k)} + E^* y^{(k)} + 2g;$$

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

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

(iv) compute

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

We remark that when $Q = O$, the RHSS iteration method automatically reduces to the HSS iteration method in [9]. Alternatively, the RHSS iteration method can be regarded as a special case of the PHSS iteration method developed in [6] with the particular preconditioning matrix

$$P = \begin{pmatrix} I & O \\ O & I + \frac{1}{\alpha} Q \end{pmatrix};$$

see also [8]. Note that in [8] the preconditioning matrix P is differently taken to be

$$P = \begin{pmatrix} B & O \\ O & C \end{pmatrix},$$

with $C \in \mathbb{C}^{q\times q}$ being a Hermitian positive definite matrix approximating the Schur complement $S = E^*B^{-1}E$ of the saddle-point matrix A in (1.1).

In general, with a suitable choice of the regularization matrix Q the convergence rate of the RHSS iteration method can be accelerated so as to be substantially faster than the HSS iteration method, when both methods are used to solve the saddle-point linear system (1.1). In addition, the main costs at each step of the RHSS iteration method are solving two linear sub-systems with respect to the Hermitian positive definite matrices

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

In general, thanks to the presence of the Hermitian positive semidefinite matrix Q, the latter matrix can be expected to be better conditioned than its counterpart

$$\alpha I + \frac{1}{\alpha}E^*E$$

arising in the HSS iteration method. In fact, the smallest eigenvalue of the matrix $\alpha I + Q + \frac{1}{\alpha}E^*E$ will be significantly larger than that of the matrix $\alpha I + \frac{1}{\alpha}E^*E$, especially if the null space of the matrix Q does not contain any singular vector corresponding to the smallest singular value of the matrix E. As a result, for this situation the condition number of the matrix $\alpha I + Q + \frac{1}{\alpha}E^*E$ will be much smaller than that of the matrix $\alpha I + \frac{1}{\alpha}E^*E$ when the matrix Q is also bounded uniformly with respect to the matrix size q.

In actual computations, the afore-mentioned two Hermitian positive definite linear sub-systems 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. With this approach the linear sub-systems are solved inexactly, leading to the IRHSS iteration method. The choice of preconditioner to be used in the PCG method will be in general problem-dependent; standard options are the *incomplete Cholesky* (IC) factorization, *symmetric successive overrelaxation* (SSOR) iteration, or *algebraic multigrid* (AMG); see, e.g., [1,7,17,20, 23]. Note that the inexact iteration method may fail to converge unless the linear sub-systems are solved with sufficient accuracy; this is not an issue, however, if IRHSS is used as a preconditioner (as we discuss below).

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

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

where

$$M(\alpha) = \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}I & O \\ O & (\alpha I + Q)^{-1} \end{pmatrix}(\alpha I + H_+)(\alpha I + S_+)$$
$$= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}(\alpha I + B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & E \\ -E^* & \alpha I + Q \end{pmatrix} \tag{2.4}$$

and

$$N(\alpha) = \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}I & O \\ O & (\alpha I + Q)^{-1} \end{pmatrix}(\alpha I - H_-)(\alpha I - S_-)$$
$$= \frac{1}{2}\begin{pmatrix} \frac{1}{\alpha}(\alpha I - B) & O \\ O & I \end{pmatrix}\begin{pmatrix} \alpha I & -E \\ E^* & \alpha I + Q \end{pmatrix}. \tag{2.5}$$

Note that $L(\alpha) = M(\alpha)^{-1}N(\alpha)$ is the iteration matrix of the RHSS iteration method. From this equivalent reformulation we see that the RHSS iteration method can also be induced by the matrix splitting $A = M(\alpha) - N(\alpha)$. The splitting matrix $M(\alpha)$ can be employed to precondition the saddle-point matrix A, 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

$$M(\alpha)\, w = r,$$

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 implementation, this generalized residual equation can be solved according to the following procedure:

(i) solve $u_a \in \mathbb{C}^p$ from the linear sub-system

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

(ii) solve $w_b \in \mathbb{C}^q$ from the linear sub-system

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

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

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

Hence, analogously to the computational implementation of the RHSS iteration method, the action of the RHSS preconditioning matrix $M(\alpha)$ also requires to solve two linear sub-systems with the Hermitian positive definite coefficient matrices

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

Again, these linear sub-systems can be solved either exactly by Cholesky factorization or inexactly by PCG method, depending on the sizes of these matrices. Of course, in the case of IRHSS we may need to use a flexible Krylov subspace method, such as FGMRES [20].

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 $M(\alpha)^{-1}A$ 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 $L(\alpha) = M(\alpha)^{-1}N(\alpha)$ is less than one, i.e., $\rho(L(\alpha)) < 1$, where $M(\alpha)$ and $N(\alpha)$ are defined in (2.4) and (2.5), respectively; see [1,17,23]. The following theorem precisely describes the asymptotic convergence property of the RHSS iteration method.

Theorem 3.1 *Let $B \in \mathbb{C}^{p\times p}$ be Hermitian positive definite, $E \in \mathbb{C}^{p\times q}$ be of full column rank, and $\alpha > 0$ be a given constant. Assume that $Q \in \mathbb{C}^{q\times q}$ is a Hermitian positive semidefinite matrix. Then it holds that $\rho(L(\alpha)) < 1$, i.e., the RHSS iteration method converges unconditionally to the exact solution of the saddle-point linear system* (1.1).

Proof Denote by

$$\widetilde{W}(\alpha) = \begin{pmatrix} (\alpha I + B)^{-1}(\alpha I - B) & O \\ O & I \end{pmatrix}$$

and

$$\widetilde{U}_-(\alpha) = \begin{pmatrix} \alpha I & -E \\ E^* & \alpha I + Q \end{pmatrix}, \quad \widetilde{U}_+(\alpha) = \begin{pmatrix} \alpha I & E \\ -E^* & \alpha I + Q \end{pmatrix}.$$

Then the iteration matrix $L(\alpha)$ of the RHSS iteration method is similar to the matrix

$$\widetilde{L}(\alpha) = \widetilde{W}(\alpha)\,\widetilde{U}_-(\alpha)\,\widetilde{U}_+(\alpha)^{-1}.$$

With the block-diagonal matrix

$$\widetilde{D}(\alpha) = \begin{pmatrix} I & O \\ O & \sqrt{\alpha}(\alpha I + Q)^{-1/2} \end{pmatrix}$$

and the matrix

$$\widetilde{E}(\alpha) = \sqrt{\alpha} E(\alpha I + Q)^{-1/2},$$

we know that $\widetilde{L}(\alpha)$ is further similar to the matrix

$$\widehat{L}(\alpha) = \widehat{W}(\alpha)\widehat{U}_-(\alpha)\widehat{U}_+(\alpha)^{-1},$$

where

$$\widehat{W}(\alpha) = \widetilde{D}(\alpha)\,\widetilde{W}(\alpha)\,\widetilde{D}(\alpha)^{-1} = \widetilde{W}(\alpha),$$

$$\widehat{U}_-(\alpha) = \widetilde{D}(\alpha)\,\widetilde{U}_-(\alpha)\,\widetilde{D}(\alpha) = \begin{pmatrix} \alpha I & -\widetilde{E}(\alpha) \\ \widetilde{E}(\alpha)^* & \alpha I \end{pmatrix}$$

and

$$\widehat{U}_+(\alpha) = \widetilde{D}(\alpha)\,\widetilde{U}_+(\alpha)\,\widetilde{D}(\alpha) = \begin{pmatrix} \alpha I & \widetilde{E}(\alpha) \\ -\widetilde{E}(\alpha)^* & \alpha I \end{pmatrix}.$$

That is to say,

$$\begin{aligned}\widehat{L}(\alpha) = &\begin{pmatrix} \alpha I + B & O \\ O & \alpha I \end{pmatrix}^{-1} \begin{pmatrix} \alpha I - B & O \\ O & \alpha I \end{pmatrix} \\ &\cdot \begin{pmatrix} \alpha I & -\widetilde{E}(\alpha) \\ \widetilde{E}(\alpha)^* & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & \widetilde{E}(\alpha) \\ -\widetilde{E}(\alpha)^* & \alpha I \end{pmatrix}^{-1},\end{aligned}$$

which is also similar to the matrix

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

where

$$\begin{aligned}\check{M}(\alpha) &= \frac{1}{2\alpha}\begin{pmatrix} \alpha I + B & O \\ O & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & \widetilde{E}(\alpha) \\ -\widetilde{E}(\alpha)^* & \alpha I \end{pmatrix} \\ &= \frac{1}{2\alpha}(\alpha I + \check{H}(\alpha))(\alpha I + \check{S}(\alpha)) \qquad (3.1)\end{aligned}$$

and

$$\begin{aligned}\check{N}(\alpha) &= \frac{1}{2\alpha}\begin{pmatrix} \alpha I - B & O \\ O & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & -\widetilde{E}(\alpha) \\ \widetilde{E}(\alpha)^* & \alpha I \end{pmatrix} \\ &= \frac{1}{2\alpha}(\alpha I - \check{H}(\alpha))(\alpha I - \check{S}(\alpha)),\end{aligned}$$

with

$$\check{H}(\alpha) = \begin{pmatrix} B & O \\ O & O \end{pmatrix} \quad \text{and} \quad \check{S}(\alpha) = \begin{pmatrix} O & \widetilde{E}(\alpha) \\ -\widetilde{E}(\alpha)^* & O \end{pmatrix}.$$

The above analysis readily shows that $\check{L}(\alpha)$ is the HSS iteration matrix of the saddle-point linear system with the coefficient matrix

$$\check{A}(\alpha) = \check{H}(\alpha) + \check{S}(\alpha) = \begin{pmatrix} B & \widetilde{E}(\alpha) \\ -\widetilde{E}(\alpha)^* & O \end{pmatrix}. \tag{3.2}$$

As the matrix $\widetilde{E}(\alpha)$ is of full column rank, from [9] we straightforwardly know that $\rho(\check{L}(\alpha)) < 1$. As a result, we immediately have $\rho(L(\alpha)) < 1$ due to the similarity of the matrices $\check{L}(\alpha)$ and $L(\alpha)$. □

We observe that the convergence of the RHSS iteration method can also be obtained as a consequence of the general convergence theory for the PHSS iteration method since, as already mentioned, RHSS is a special case of PHSS for a particular choice of preconditioner. However, the proof just given yields additional insight into the properties of the method, which cannot be obtained from the general theory of PHSS. Some of this additional information is explicitly used in the proof of Theorem 3.2 below. We also observe that while in the original PHSS iteration method the main focus was on preconditioning the (1,1) block B, the RHSS iteration method acts as a preconditioner for the off-diagonal blocks E and E^* of the saddle-point matrix.

It is known from [2,13,20] that if the coefficient matrix of a linear system has only a few distinct eigenvalues or if these eigenvalues are sufficiently clustered away from the origin, then there are polynomials of low degree which will be small at the eigenvalues. Provided that the preconditioned matrix is diagonalizable and the matrix formed by the eigenvectors is not too ill-conditioned, it is well known that optimal Krylov subspace methods, like GMRES, can be expected to converge quickly; see, e.g., [3,20]. Hence, to estimate the convergence rate of the preconditioned Krylov subspace iteration methods such as GMRES with respect to the RHSS preconditioner, in the next theorem we describe the clustering property of the eigenvalues of the preconditioned matrix $M(\alpha)^{-1}A$. We note that a similar result was obtained for the HSS preconditioner in [22, Prop. 3.3]. Since RHSS is a special case of PHSS, which in turn is a special case of HSS, the next result is not surprising; however, the proof given below yields additional insights into the matter which are not readily available from the general theory.

Theorem 3.2 *Let $B \in \mathbb{C}^{p\times p}$ be Hermitian positive definite, $E \in \mathbb{C}^{p\times q}$ be of full column rank, $p \geq q$, and α be a given positive constant. Assume that $Q \in \mathbb{C}^{q\times q}$ is a Hermitian positive semidefinite matrix. Then the eigenvalues of the preconditioned matrix $\mathbf{A}(\alpha) = M(\alpha)^{-1}A$ are clustered at 0_+ and 2_- if α is close to 0, where $M(\alpha)$ is the RHSS preconditioning matrix defined in (2.4).*

Proof Let λ be an eigenvalue of the matrix $\mathbf{A}(\alpha) = M(\alpha)^{-1}A$. As A is nonsingular, we know that $\lambda \neq 0$. Based on the relationships

$$M(\alpha)^{-1}A = I - L(\alpha) \quad \text{and} \quad \check{M}(\alpha)^{-1}\check{A}(\alpha) = I - \check{L}(\alpha),$$

from the similarity of the matrices $L(\alpha)$ and $\check{L}(\alpha)$ demonstrated in the proof of Theorem 3.1, we see that λ is also an eigenvalue of the matrix $\check{M}(\alpha)^{-1}\check{A}(\alpha)$, where $\check{M}(\alpha)$

and $\check{A}(\alpha)$ are defined as in (3.1) and (3.2), respectively. That is to say, there exists a nonzero vector $\check{x} \in \mathbb{C}^n$ such that

$$\check{A}(\alpha)\,\check{x} = \lambda\,\check{M}(\alpha)\,\check{x}. \tag{3.3}$$

Let now

$$\check{x} = \begin{pmatrix} \check{y} \\ \check{z} \end{pmatrix}, \quad \text{with} \quad \check{y} \in \mathbb{C}^p \quad \text{and} \quad \check{z} \in \mathbb{C}^q.$$

Then the equation (3.3) can be equivalently written as

$$\begin{cases} 2\alpha(B\check{y} + \tilde{E}(\alpha)\,\check{z}) = \lambda(\alpha I + B)(\alpha\check{y} + \tilde{E}(\alpha)\,\check{z}), \\ -2\alpha\tilde{E}(\alpha)^*\,\check{y} = \alpha\lambda(-\tilde{E}(\alpha)^*\,\check{y} + \alpha\check{z}). \end{cases} \tag{3.4}$$

We claim that $\check{y} \neq 0$. Otherwise, the second equality in (3.4) implies that $\check{z} = 0$, which contradicts the assumption that $\check{x}$ is an eigenvector of the matrix $\check{M}(\alpha)^{-1}\check{A}(\alpha)$.

The second equality in (3.4) gives

$$\check{z} = \frac{\lambda - 2}{\alpha\lambda}\tilde{E}(\alpha)^*\,\check{y}.$$

Substituting it into the first equality in (3.4), after suitable manipulations we have

$$\begin{aligned} &\lambda^2(\alpha I + B)(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)\,\check{y} \\ &\quad - 2\lambda\left[(\alpha I + B)\left(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*\right) - \alpha\left(\alpha^2 I - \tilde{E}(\alpha)\tilde{E}(\alpha)^*\right)\right]\check{y} \\ &\quad + 4\alpha\tilde{E}(\alpha)\tilde{E}(\alpha)^*\,\check{y} = 0. \end{aligned} \tag{3.5}$$

With the change of variable

$$\tilde{y} = (\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)\,\check{y},$$

the equality (3.5) can then be rewritten as

$$\begin{aligned} &\lambda^2(\alpha I + B)\tilde{y} - 2\lambda\left[(\alpha I + B) - \alpha(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)^{-1}(\alpha^2 I - \tilde{E}(\alpha)\tilde{E}(\alpha)^*)\right]\tilde{y} \\ &\quad + 4\alpha(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)^{-1}\tilde{E}(\alpha)\tilde{E}(\alpha)^*\tilde{y} = 0. \end{aligned} \tag{3.6}$$

As $\check{y} \neq 0$, we know that $\tilde{y} \neq 0$. Hence, without loss of generality, in the subsequent discussion we can assume that $\|\tilde{y}\| = 1$. Here and in the sequel, $\|\cdot\|$ represents the Euclidean norm of either a vector or a matrix. Also, for simplicity we adopt the notation

$$\nu = \tilde{y}^* B\tilde{y} \quad \text{and} \quad \delta = \tilde{y}^*(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)^{-1}(\alpha^2 I - \tilde{E}(\alpha)\tilde{E}(\alpha)^*)\tilde{y}.$$

By multiplying both sides of (3.6) from left with y^*, and using the identity

$$\begin{aligned} I - (\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)^{-1}(\alpha^2 I - \tilde{E}(\alpha)\tilde{E}(\alpha)^*) \\ = 2(\alpha^2 I + \tilde{E}(\alpha)\tilde{E}(\alpha)^*)^{-1}\tilde{E}(\alpha)\tilde{E}(\alpha)^*, \end{aligned}$$

we obtain

$$(\alpha + \nu)\lambda^2 - 2[\alpha(1-\delta) + \nu]\lambda + 2\alpha(1-\delta) = 0.$$

The two roots of this quadratic equation are

$$\begin{aligned} \lambda_\pm &= \frac{\alpha(1-\delta) + \nu \pm \sqrt{[\alpha(1-\delta) + \nu]^2 - 2\alpha(\alpha+\nu)(1-\delta)}}{\alpha + \nu} \\ &= \frac{\alpha(1-\delta) + \nu \pm \sqrt{\nu^2 - \alpha^2(1-\delta^2)}}{\alpha + \nu}. \end{aligned} \tag{3.7}$$

Because ν is bounded from below and above by the smallest and the largest eigenvalues of the Hermitian positive definite matrix B, and $|\delta| \leq 1$ holds uniformly for all $\alpha > 0$, by taking limits in the formula for the eigenvalue λ in (3.7), we see that

$$\lim_{\alpha \to 0} \lambda_- = 0 \quad \text{and} \quad \lim_{\alpha \to 0} \lambda_+ = 2.$$

Recalling that Theorem 3.1 guarantees that all eigenvalues of the matrix $\mathbf{A}(\alpha)$ are included in a disk centered at 1 with the radius $\rho(L(\alpha)) < 1$, we know that all eigenvalues of the matrix $\mathbf{A}(\alpha)$ are clustered at 0_+ and 2_- when α is close to 0. □

From (3.7) we see that if $\delta \approx 0$ then

$$\lambda_\pm \approx 1 \pm \sqrt{\frac{\nu - \alpha}{\nu + \alpha}}.$$

Hence, when α is close to 0, the eigenvalues of the preconditioned matrix $\mathbf{A}(\alpha)$ are clustered at 0_+ and 2_-. Note that $\delta \approx 0$ if

$$\alpha^2 I \approx \tilde{E}(\alpha)\tilde{E}(\alpha)^* = \alpha E(\alpha I + Q)^{-1}E^*,$$

or in other words,

$$\alpha I \approx E(\alpha I + Q)^{-1}E^*,$$

which implies that

$$\alpha I + Q \approx \frac{1}{\alpha}E^*E.$$

From this observation we know that in actual computations the regularization matrix $Q \in \mathbb{C}^{q\times q}$ should be chosen such that

$$Q \approx \frac{1}{\alpha} E^* E - \alpha I,$$

or

$$Q \approx \frac{1}{\alpha} E^* E$$

if α is small enough (recall that Q must be positive semidefinite).

It can be shown (see [22]) that for α small the cluster near 2 contains p eigenvalues and the cluster near 0 the remaining q eigenvalues. As noted in [22], clustering near 0 and 2 can lead to fast convergence of the preconditioned Krylov subspace iteration method provided that the leftmost cluster is not *too* close to zero. In practice, α should be chosen small enough so as to have most of the eigenvalues falling into two well-separated clusters, but not so small that the preconditioned matrix becomes too close to being singular. In contrast, when the RHSS iteration method is used as a stationary method the asymptotic rate of convergence 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 quite large.

Also, as already observed the RHSS preconditioner does not affect the spectral properties of the (1,1)-block matrix B, instead it has the effect of preconditioning the (1,2) block E and, symmetrically, the (2,1) block E^* as well. As a consequence, the RHSS preconditioner can be thought of as preconditioning the Schur complement of the saddle-point matrix $A \in \mathbb{C}^{n\times n}$. Therefore, we deduce that the RHSS iteration should be expected to be especially useful when it is used to solve or precondition saddle-point linear systems (1.1) having a well-conditioned (1,1)-block matrix and an ill-conditioned Schur complement, provided that a suitable choice of Q is available. On the other hand, the RHSS preconditioner is likely to be less effective when applied to saddle-point problems with a well-conditioned Schur complement and an ill-conditioned (1,1) block. The numerical results presented in the following section confirm this expectation.

4 Numerical results

In this section we report on numerical experiments using the RHSS iteration method to solve two types of saddle-point problems, one from fluid mechanics and the other from optimal control. We are interested in the performance of RHSS as a solver and as a preconditioner for Krylov subspace iteration methods. Both the exact and the inexact versions are tested. The experiments also aim at identifying suitable choices of the matrix Q and of the parameter α. Furthermore, we compare the performance of the proposed method with the corresponding variants of the standard HSS and PHSS solvers, as well as the *preconditioned MINRES* (PMINRES) solver incorporated with the optimal block-diagonal preconditioners in [15,19], in order to show the advantages of the RHSS approach relative to the older methods.

In the tests we report both the number of iterations (denoted as "IT") and the computing time in seconds (denoted as "CPU"). In the sequel, we use $(\cdot)^T$ to denote the transpose of either a vector or a matrix, and "—" to indicate that the corresponding iteration method either does not satisfy the prescribed stopping criterion until 5, 000 iteration steps, or cannot be executed due to insufficient computer memory.

All experiments are started from the initial vector $x^{(0)} = 0$, terminated once the relative residual errors at the current iterates $x^{(k)}$ satisfy $\|b - Ax^{(k)}\| \leq 10^{-5} \times \|b\|$. 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 Linux operating system (Ubuntu 15.04). In our codes, in order to construct approximate solvers for certain linear sub-systems precisely specified in the sequel, we utilize preconditioners based on the incomplete Cholesky and the *modified incomplete Cholesky* (MIC) factorizations implemented in MATLAB by the functions `ichol(·)` and `ichol(·,struct('droptol',1e-3,'michol','on'))`. Moreover, we build up their algebraic multigrid approximations or preconditioners by utilizing the package HSL_MI20 (hsl_mi20_precondition) with default parameters.

Example 4.1 [8] Consider the Stokes problem: Find u and p such that

$$\begin{cases} -\Delta u + \nabla p = \tilde{f}, \\ \nabla \cdot u = 0, \end{cases} \quad \text{in} \quad \Omega, \tag{4.1}$$

under the boundary and the normalization conditions $u = 0$ on $\partial\Omega$ and $\int_\Omega p = 0$, where $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$, $\partial\Omega$ is the boundary of Ω, Δ is the componentwise Laplacian operator, ∇ and $\nabla\cdot$ denote the gradient and the divergence operators, u is a vector-valued function representing the velocity, and p is a scalar function representing the pressure. By discretizing this problem with the upwind finite-difference scheme, we obtain the saddle-point linear system (1.1), in which

$$B = \begin{pmatrix} I \otimes T + T \otimes I & O \\ O & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times 2m^2}$$
$$\text{and} \quad E = \begin{pmatrix} I \otimes \Upsilon \\ \Upsilon \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times m^2},$$

with

$$T = \frac{1}{h^2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m} \quad \text{and} \quad \Upsilon = \frac{1}{h} \cdot \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{m \times m},$$

and $f = (1, 1, \ldots, 1)^T \in \mathbb{R}^{2m^2}$ and $g = (0, 0, \ldots, 0)^T \in \mathbb{R}^{m^2}$ being constant vectors. Here, $h = \frac{1}{m+1}$ represents the discretization stepsize and $\otimes$ denotes the Kronecker product symbol.

In our implementations, we first symmetrically scale the saddle-point matrix $A \in \mathbb{R}^{3m^2 \times 3m^2}$ such that its nonzero diagonal entries are all equal to 1. We found this scaling to be important in order to have good performance; see [18]. In both HSS and RHSS

Table 1 Inner PCG and inner PMINRES stopping tolerances for IHSS, IPHSS, IRHSS and IHSS-, IPHSS-, IRHSS-FGMRES methods for Example 4.1

Method	m					
	64	96	128	192	256	384
IHSS	1E−02	1E−02	1E−02	1E−05	1E−05	1E−05
IPHSS	1E−01	1E−01	1E−01	1E−01	1E−01	1E−02
IRHSS	1E−01	1E−01	1E−01	1E−01	1E−01	1E−01
	4E−02	1E−02	1E−02	1E−02	5E−03	5E−03
IHSS-FGMRES	1E−01	1E−01	1E−01	1E−01	1E−01	1E−01
IPHSS-FGMRES	1E−01	1E−01	1E−01	1E−01	1E−01	1E−01
IRHSS-FGMRES	1E−02	1E−02	1E−02	1E−02	1E−02	1E−02

used either as linear solvers or as "exact" preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved directly by sparse Cholesky factorizations. In addition, the linear sub-systems occurring in the two half-steps of the PHSS solver or the PHSS preconditioner are also solved directly by sparse Cholesky factorizations.

In our implementations of the exact and the inexact RHSS iteration methods for solving the discrete Stokes problem (4.1) we choose the regularization matrix Q to be $Q = \gamma\, E^T E$, where γ is a regularization parameter. Note that this is a discrete scaled Neumann Laplacian and is fairly sparse. In both IHSS and IRHSS iteration methods, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$, $\alpha I + Q$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved iteratively by the PCG methods, for which in IHSS the matrices $\alpha I + B$ and $\alpha I + \frac{1}{\alpha} E^T E$ are preconditioned by their IC factorizations, while in IRHSS the matrix $\alpha I + B$ is preconditioned by its IC factorization, and the matrices $\alpha I + Q$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are preconditioned by their MIC factorizations. In addition, it is required to solve a linear sub-system with the coefficient matrix

$$\begin{pmatrix} \alpha B & E \\ -E^T & \alpha C \end{pmatrix}, \quad C := E^T E, \tag{4.2}$$

in each step of the inexact PHSS (IPHSS) iteration method. We first equivalently reformulate this linear sub-system into its symmetric form with the coefficient matrix

$$\begin{pmatrix} \alpha B & E \\ E^T & -\alpha C \end{pmatrix} \tag{4.3}$$

and then adopt the PMINRES method incorporated with the optimal block-diagonal preconditioner $\mathrm{Diag}(\widehat{B}, I)$ to solve it, with $\widehat{B}$ being the algebraic multigrid approximation of the sub-matrix B; see [15]. In Table 1 we report the stopping tolerances used for the inner PCG iterations in the first, the third, the fourth and the last rows, and those used for the inner PMINRES iterations in the second and the fifth rows.

Table 2 The values of iteration parameter α and (or) regularization parameter γ in HSS, PHSS, IPHSS and RHSS iteration methods for Example 4.1

Method	Index	m					
		64	96	128	192	256	384
HSS	α	0.23	0.21	0.17	0.13	0.11	0.07
PHSS	α	3.60	4.38	—	—	—	—
IPHSS	α	3.60	4.38	5.00	5.63	6.30	7.10
RHSS	α	0.07	0.05	0.04	0.03	0.02	0.02
	γ	3.5	5.0	7.0	10.0	17.0	20.0

In Table 2 we list the iteration parameter α and the regularization parameter γ used in our implementations, which are the experimentally computed optimal ones that minimize the total number of iterations for either HSS, or IPHSS, or RHSS iteration method. For PHSS, the optimal iteration parameter α can be computed by the analytic formula derived in [8]; see also [5]. However, for $m \geq 128$ the method becomes prohibitively expensive (see below), and therefore we do not report values of the optimal α. We note that for RHSS, the optimal α slowly decreases as h decreases, while γ increases at a similar rate. In Fig. 1 we show the strong clustering of the eigenvalues near 0 and 2, as predicted by Theorem 3.2 for small values of α.

In Table 3 we report iteration counts and computing times for the exact and the inexact HSS, PHSS and RHSS iteration methods. In IHSS and IRHSS iteration methods, we adopt the same parameters as in HSS and RHSS iteration methods reported in Table 2 for α and (or) γ. From Table 3, we observe that when used as a fixed-point iteration, RHSS and its inexact variant IRHSS significantly outperform HSS and IHSS, both in terms of iteration counts and CPU time; the advantage of the new techniques become more pronounced as the system size increases. By comparing with PHSS and IPHSS, we find that for smaller system sizes such as $m = 64$ and 96, both RHSS and IRHSS cost much less CPU times, although they require more iteration steps to achieve the convergence. For larger system sizes such as $m \geq 128$, the PHSS iteration method fails, as it demands unavailable huge computer memory in solving a linear sub-system with the coefficient matrix (4.2); the IPHSS iteration method is, however, successfully convergent until the prescribed stopping criterion is satisfied. Moreover, both RHSS and IRHSS successfully converge to a satisfactory approximate solution of the saddle-point linear system (1.1) arising from Example 4.1 in much less CPU times than IPHSS. We remark that the IPHSS iteration method utilizes its own experimentally computed optimal values of the iteration parameter α, while both IHSS and IRHSS iteration methods adopt only those of the HSS and the RHSS iteration methods, respectively. These experiments also show that the slightly higher cost of RHSS per iteration is more than offset by its faster convergence. Moreover, we can see that the inexact variant IRHSS is much more efficient than the exact method RHSS, as expected. We emphasize that the fact that IRHSS iteration method is convergent is not obvious, as the convergence theory presented in this paper only covers the exact iteration method, RHSS; however, the theory developed in [7] for IHSS can be adapted to account for the observed convergence of

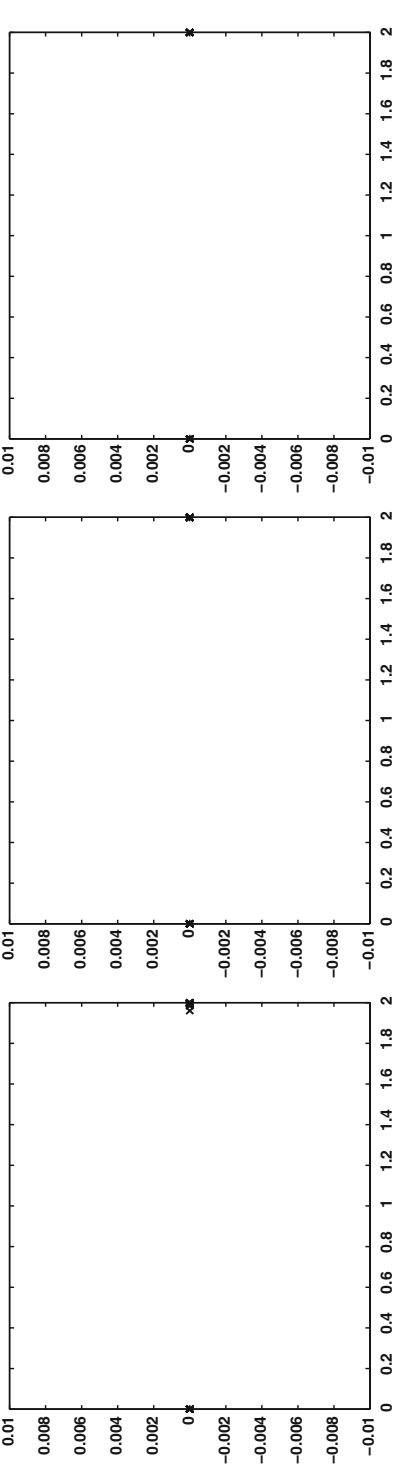

Fig. 1 The eigenvalue distributions of the RHSS-preconditioned matrices for Example 4.1 when $m = 64$ and $Q = \gamma\,\mathrm{diag}(E^T E)$, with $\gamma = 200.0$ and $\alpha = 0.4$ (*left*), $\alpha = 0.04$ (*middle*) and $\alpha = 0.004$ (*right*)

Table 3 Numerical results for HSS, PHSS, RHSS and IHSS, IPHSS, IRHSS iteration methods for Example 4.1

Method	Index	m					
		64	96	128	192	256	384
HSS	IT	268	368	478	772	1114	1693
	CPU	5.31	25.24	73.51	351.73	1190.73	6177.25
PHSS	IT	49	61	—	—	—	—
	CPU	159.16	1723.12	—	—	—	—
RHSS	IT	88	107	128	186	246	434
	CPU	1.77	7.25	21.27	90.05	270.63	1601.15
IHSS	IT	269	391	691	773	1115	1843
	CPU	2.36	6.59	18.46	113.44	318.90	1482.51
IPHSS	IT	54	71	86	101	120	136
	CPU	7.12	23.09	47.99	151.15	407.41	1340.37
IRHSS	IT	276	235	281	336	537	669
	CPU	2.82	5.13	10.75	30.54	100.69	325.47

IRHSS. As the results show, all these iteration methods tested here converge slowly on the Stokes problem, and the number of iterations is seen to increase as the mesh is refined.

Next, we consider the use of these methods as preconditioners for (F)GMRES. In the next set of experiments, we choose the regularization matrix Q to be $Q = \gamma\,\mathrm{diag}(E^T E)$, where γ is a regularization parameter. In both IHSS and IRHSS iteration preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved iteratively by the PCG methods, for which all these three matrices are preconditioned by their algebraic multigrid approximations. In addition, the linear sub-system with the coefficient matrix (4.2) involved in the IPHSS preconditioner is first equivalently reformulated into its symmetric form with the coefficient matrix (4.3) and then solved by PMINRES incorporated with the near-optimal block-diagonal preconditioner $\mathrm{Diag}(\widehat{B}, I)$ [15], with $\widehat{B}$ being the algebraic multigrid approximation of the sub-matrix B. This same preconditioner $\mathrm{Diag}(\widehat{B}, I)$ is also adopted in the standard PMINRES method. The inner iteration stopping tolerances can be found in Table 1 (the last three rows).

In Table 4 we list the iteration parameter α and the regularization parameter γ used in our implementations, which are the experimentally computed optimal ones that minimize the total number of iteration steps of HSS-, PHSS-, IPHSS- and RHSS-(F)GMRES methods. It is worth noting that the values and behavior of the optimal parameters as $h \to 0$ are very different from the case where the methods are used without Krylov subspace acceleration. However, in both cases the optimal γ behaves reciprocally to the optimal α.

In Table 5 we report iteration counts and CPU times for the exact and the inexact HSS-(F)GMRES, PHSS-(F)GMRES and RHSS-(F)GMRES methods, as well as the PMINRES method. The GMRES method is used without restarts. In IHSS-FGMRES

Table 4 The values of iteration parameter α and (or) regularization parameter γ in HSS-GMRES, PHSS-GMRES, IPHSS-FGMRES and RHSS-GMRES methods for Example 4.1

Method	Index	m					
		64	96	128	192	256	384
HSS-GMRES	α	110.0	160.0	185.0	205.0	220.0	230.0
PHSS-GMRES	α	2E−04	2E−04	—	—	—	—
IPHSS-FGMRES	α	2E−04	2E−04	1E−03	1E−03	1E−03	1E−03
RHSS-GMRES	α	0.004	0.006	0.010	0.060	0.200	0.200
	γ	200.0	150.0	100.0	30.0	10.0	3.0

Table 5 Numerical results for HSS-, PHSS-, RHSS-GMRES and IHSS-, IPHSS-, IRHSS-FGMRES methods, as well as for PMINRES method, for Example 4.1

Method	Index	m					
		64	96	128	192	256	384
HSS-GMRES	IT	63	79	91	112	135	177
	CPU	3.62	13.70	35.52	124.21	349.49	1612.40
PHSS-GMRES	IT	3	3	—	—	—	—
	CPU	151.47	1664.80	—	—	—	—
RHSS-GMRES	IT	37	41	43	50	57	62
	CPU	1.82	6.32	15.65	52.49	141.77	631.16
IHSS-FGMRES	IT	72	87	101	128	153	201
	CPU	2.17	5.32	10.93	34.45	85.87	373.10
IPHSS-FGMRES	IT	30	30	30	30	30	31
	CPU	7.29	17.50	28.81	70.08	132.97	343.17
IRHSS-FGMRES	IT	48	53	55	59	63	74
	CPU	1.58	3.45	6.04	14.65	30.48	95.30
PMINRES	IT	31	35	36	38	40	43
	CPU	0.29	0.60	1.08	2.65	5.08	12.77

and IRHSS-FGMRES methods, we adopt the same parameters α and (or) γ as in HSS-GMRES and RHSS-GMRES methods, respectively; see Table 4.

First, we note that (F)GMRES acceleration significantly improves convergence for all methods, as expected. Second, these results show that RHSS and IRHSS, when used as preconditioners for GMRES and FGMRES (resp.), are much better than the HSS and IHSS, as well as the PHSS and IPHSS preconditioners, correspondingly. We also note, however, that the convergence rate for both RHSS and IRHSS preconditioning is not h-independent; in particular, the convergence deteriorates when the inexact version of RHSS preconditioning is used. Hence, while the RHSS approach appears to be superior to the HSS and the PHSS one, it cannot be recommended as a solver for the discrete steady Stokes problem, for which PMINRES with inexact block diagonal preconditioning is clearly the method of choice [15]. We note that for the discrete

Stokes problem considered here, the Schur complement is well-conditioned (uniformly in h), while the conditioning of the (1, 1) block deteriorates as $h \to 0$. Hence, it is not surprising that for this problem the (I)RHSS preconditioner does not achieve h-independent convergence behavior. The next example, however, shows that there are saddle-point linear systems for which the new techniques provide (nearly) optimal complexity and can thus be competitive.

Example 4.2 [19] Consider the optimal control problem

$$\begin{cases} \min\limits_{u,v} J(u,v) := \frac{1}{2}\|u - u_d\|^2_{\mathscr{L}^2(\Omega)} + \frac{\beta}{2}\|v\|^2_{\mathscr{L}^2(\Omega)}, \\ -\Delta u = v, \end{cases} \quad \text{in} \quad \Omega,$$

under the boundary condition and constraint $u_{|\partial\Omega} = 0$ and $\underline{u} \le u \le \bar{u}$, where $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$, $\partial\Omega$ is the boundary of Ω, $\|\cdot\|_{\mathscr{L}^2(\Omega)}$ is the $\mathscr{L}^2$-norm on Ω, u_d is a given function that represents the desired state, Δ is the Laplacian operator, $\beta > 0$ is a regularization parameter, and $\underline{u}$ and $\bar{u}$ are prescribed constants. By making use of the Moreau–Yosida penalty function method and the finite element discretization on a rectangular grid with bilinear basis functions, at each iteration step of the semismooth Newton method we need to solve the saddle-point linear system (1.1), in which

$$B = \begin{pmatrix} M + \varepsilon^{-1} G_{\mathscr{I}} M G_{\mathscr{I}} & O \\ O & \beta M \end{pmatrix}, \quad E = \begin{pmatrix} -K^T \\ M \end{pmatrix}$$

and

$$y = \begin{pmatrix} u \\ v \end{pmatrix}, \quad f = \begin{pmatrix} c_{\mathscr{I}} \\ 0 \end{pmatrix} \quad \text{and} \quad z = \lambda, \quad g = 0.$$

Here, the matrix $G_{\mathscr{I}}$ is a projection onto the active set $\mathscr{I} = \mathscr{I}_+ \cup \mathscr{I}_-$ with $\mathscr{I}_+ = \{i \mid u_i > \bar{u}_i\}$ and $\mathscr{I}_- = \{i \mid u_i < \underline{u}_i\}$, $M \in \mathbb{R}^{m^2 \times m^2}$ and $K \in \mathbb{R}^{m^2 \times m^2}$ are the mass and the stiffness matrices, respectively, λ is the Lagrangian multiplier, $c_{\mathscr{I}} = M u_d + \epsilon^{-1}(G_{\mathscr{I}_+} M G_{\mathscr{I}_+} \bar{u} + G_{\mathscr{I}_-} M G_{\mathscr{I}_-} \underline{u})$, and $h = \frac{1}{m+1}$ represents the discretization stepsize. Finally, $\epsilon > 0$ is a user-defined parameter.

In our experiments we set $\epsilon = \beta = 0.01$, $\underline{u} = -\infty$, $\bar{u} = 0.1$, and $u_d = \sin(2\pi x_1 x_2)$ with $(x_1, x_2) \in \Omega$. In addition, the index set $\mathscr{I}$ is determined by taking $u = u_d$. In both HSS and RHSS used as either linear solvers or as "exact" preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved directly by sparse Cholesky factorizations. In addition, the linear sub-systems involved in the PHSS solver or the PHSS preconditioner are also solved directly by sparse Cholesky factorizations.

In our implementations of the exact and the inexact RHSS iteration methods, again we choose the regularization matrix Q to be

$$Q = \gamma\, E^T E = \gamma(K K^T + M^T M) = \gamma(K^2 + M^2),$$

where γ is a regularization parameter. Here we have used the fact that both M and K are symmetric positive definite matrices. In both IHSS and IRHSS iteration methods, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$, $\alpha I + Q$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved iteratively by the PCG methods, for which in IHSS the matrices $\alpha I + B$ and $\alpha I + \frac{1}{\alpha} E^T E$ are preconditioned by their IC factorizations, while in IRHSS the matrix $\alpha I + B$ is preconditioned by its MIC factorization, the matrix $\alpha I + Q$ is preconditioned by its IC factorization, and the matrix $\alpha I + Q + \frac{1}{\alpha} E^T E$ is preconditioned by $\widehat{K}(\alpha)^T \widehat{K}(\alpha)$ with $\widehat{K}(\alpha)$ being the algebraic multigrid preconditioner for $\sqrt{\alpha}\, I + \sqrt{\gamma + \frac{1}{\alpha}}\, (K + M)$. In addition, a linear sub-system with the coefficient matrix (4.2) with $C = K K^T + \frac{1}{\beta} M$ must be solved in each step of the IPHSS iteration method. We first equivalently reformulate this linear sub-system into its symmetric form with the coefficient matrix (4.3) and then adopt the PMINRES method incorporated with the optimal block-diagonal preconditioner $\mathrm{Diag}(\widehat{B}, \widehat{S})$ to solve it, where $\widehat{B}$ is the 20-step Chebyshev semi-iteration approximation of B, $\widehat{S} = \widehat{K}\, M_G\, \widehat{K}^T$ with $M_G = (I + \frac{1}{\sqrt{\varepsilon}}\, G_{\mathscr{J}}) M (I + \frac{1}{\sqrt{\varepsilon}}\, G_{\mathscr{J}})$, and $\widehat{K}$ is the algebraic multigrid approximation of $K + \frac{1}{\sqrt{\beta}}\, M\, (I + \frac{1}{\sqrt{\varepsilon}}\, G_{\mathscr{J}})$; see [19]. For the inexact variants, the inner tolerances are reported in Table 9.

In Table 6 we list the iteration parameter α and the regularization parameter γ used in our implementations, which are the experimentally computed optimal ones that minimize the total number of iteration steps of HSS, IPHSS and RHSS iteration methods. We note that the general trend as $h \to 0$ is qualitatively similar to that in the case of Example 4.1. Here we should point out again that the optimal iteration parameter α for the PHSS iteration method is computed by the analytic formula derived in [5,8], which is, however, not computable when m is larger than or equal to 128 due to insufficient computer memory in computing the extremal singular values of the matrix $B^{-1/2} E C^{-1/2}$.

For this problem, the number of iterations for both HSS and IHSS iteration methods is prohibitively high; both methods fail to converge within 5000 iterations for all values of m, regardless of the value of α used. Also, the PHSS iteration method fails to compute an approximate solution to the saddle-point linear system (1.1) arising from Example 4.2 when m is greater than or equal to 128 due to insufficient computer memory. For this reason we do not report the values of α for HSS and PHSS in Table 6 when m is greater than or equal to 192 and 128, respectively. Iteration counts and CPU times for HSS, PHSS and RHSS iteration methods as well as their inexact variants IHSS, IPHSS and IRHSS are reported in Table 7. As these results show, RHSS and IRHSS succeed in solving the problem for all h; however, the convergence is slow.

Next, we turn to the use of HSS, PHSS and RHSS (IHSS, IPHSS and IRHSS) as preconditioners for GMRES (respectively, FGMRES). We consider two choices of the regularization matrix Q as in Table 8, where γ is a regularization parameter. In both IHSS and IRHSS preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved iteratively by the PCG methods, for which the matrix $\alpha I + B$ is preconditioned by its MIC factorization. Moreover, in IHSS the matrix $\alpha I + \frac{1}{\alpha} E^T E$ is preconditioned by $\widehat{K}(\alpha)^T \widehat{K}(\alpha)$ with $\widehat{K}(\alpha)$ being the algebraic multigrid preconditioning for $\alpha I + K$, and in IRHSS the

Table 6 The values of iteration parameter α and (or) regularization parameter γ in HSS, PHSS, IPHSS and RHSS iteration methods for Example 4.2

Method	Index	m					
		64	96	128	192	256	384
HSS	α	0.023	0.015	0.012	—	—	—
PHSS	α	1.15	1.15	—	—	—	—
IPHSS	α	1.15	1.15	1.15	1.15	1.16	1.16
RHSS	α	6E−04	3E−04	2E−04	1E−04	7E−05	6E−05
	γ	820	1650	2500	5000	7150	8300

Table 7 Numerical results for HSS, PHSS, RHSS and IHSS, IPHSS, IRHSS iteration methods for Example 4.2

Method	Index	m					
		64	96	128	192	256	384
HSS	IT	20,543	30,147	40,811	—	—	—
	CPU	554.81	2732.26	7613.35	—	—	—
PHSS	IT	13	14	—	—	—	—
	CPU	157.61	1732.26	—	—	—	—
RHSS	IT	726	776	862	928	1077	1842
	CPU	18.89	66.45	181.58	568.08	1527.60	8472.22
IHSS	IT	19,901	28,743	40,561	—	—	—
	CPU	357.51	1411.31	4737.73	—	—	—
IPHSS	IT	43	45	47	50	52	52
	CPU	6.42	17.39	32.93	98.34	189.42	638.19
IRHSS	IT	1137	1296	1381	1616	1959	3533
	CPU	17.65	42.06	84.90	281.75	671.34	2837.21

matrix $\alpha I + Q + \frac{1}{\alpha} E^T E$ is preconditioned by $\widehat{K}^T \widehat{K}$ with $\widehat{K}$ being the algebraic multigrid preconditioning of $K + M$ and $\sqrt{1+\alpha\gamma}\, K + M$ for both choices (a) and (b) of Q, respectively. In addition, the linear sub-system with the coefficient matrix (4.2) involved in the IPHSS preconditioner is first equivalently reformulated into its symmetric form with the coefficient matrix (4.3) and then solved by PMINRES with the block-diagonal preconditioner $\mathrm{Diag}(\widehat{B}, \widehat{S})$, where $\widehat{B}$ is the 20-step Chebyshev semi-iteration approximation of B, $\widehat{S} = \widehat{K}\, M_G\, \widehat{K}^T$ with $M_G = (I + \frac{1}{\sqrt{\varepsilon}} G_{\mathscr{J}})\, M\, (I + \frac{1}{\sqrt{\varepsilon}} G_{\mathscr{J}})$, and $\widehat{K}$ is the algebraic multigrid approximation of $K + \frac{1}{\sqrt{\beta}} M\, (I + \frac{1}{\sqrt{\varepsilon}} G_{\mathscr{J}})$; see [19] again. This block-diagonal preconditioner is adopted also in the standard PMINRES method.

In Table 8 we also list the values of the iteration parameter α and of the regularization parameter γ used in our implementations. The parameter values are the experimentally computed optimal ones that minimize the total number of iteration steps of HSS-, PHSS-, IPHSS- and RHSS-(F)GMRES methods. Also, in Table 9 we report the

Table 8 The values of regularization parameter Q, iteration parameter α and (or) regularization parameter γ in HSS-GMRES, PHSS-GMRES, IPHSS-FGMRES and RHSS-GMRES methods for Example 4.2

Method	Q	Index	m					
			64	96	128	196	256	384
HSS-GMRES	0	α	0.023	0.010	0.006	0.002	0.0008	0.0004
PHSS-GMRES	—	α	1E−04	1E−03	—	—	—	—
IPHSS-FGMRES	—	α	1E−04	1E−03	1E−04	1E−02	1E−03	1E−03
RHSS-GMRES	(a) $\gamma E^T E - \alpha I$	α	9.50	6.00	3.00	1.50	0.80	0.30
		γ	1E−08	1E−08	1E−08	1E−07	1E−07	1E−07
	(b) $\gamma K K^T - \alpha I$	α	10.00	6.00	4.00	2.00	1.00	0.30
		γ	1E−08	1E−08	1E−07	1E−07	1E−07	1E−07

Table 9 Inner PCG and inner PMINRES stopping tolerances for IHSS, IPHSS, IRHSS and IHSS-, IPHSS-, IRHSS-FGMRES methods for Example 4.2

Method	m					
	64	96	128	192	256	384
IHSS	1E−02	1E−02	1E−02	—	—	—
	1E−06	1E−06	1E−07	—	—	—
IPHSS	5E−01	5E−01	5E−01	5E−01	5E−01	5E−01
IRHSS	8E−01	8E−01	8E−01	8E−01	8E−01	8E−01
IHSS-FGMRES	1E−02	1E−02	1E−02	1E−03	1E−03	1E−03
IPHSS-FGMRES	1E−01	1E−01	1E−01	1E−01	1E−01	1E−01
IRHSS-FGMRES(a)	1E−05	1E−04	1E−04	1E−05	1E−05	1E−05
IRHSS-FGMRES(b)	1E−05	1E−04	1E−04	1E−05	1E−05	1E−05

stopping tolerances used for the inner PCG and the inner PMINRES iterations when the inexact versions of the preconditioners are used. We observe that for this problem, the inner tolerances need to be tighter than in the case of Example 4.1. Indeed, we found that using larger inner stopping tolerances leads to a deterioration in the robustness and performance of the IHSS, IPHSS and IRHSS preconditioners.

In Table 10 we report iteration counts and computing times for the exact and the inexact HSS-(F)GMRES, PHSS-(F)GMRES and RHSS-(F)GMRES methods corresponding to the two choices of the regularization matrix Q given in Table 8, as well as for the PMINRES method. In IHSS-FGMRES and IRHSS-FGMRES methods, we use the same parameters α and (or) γ as in HSS-GMRES and RHSS-GMRES methods, respectively; see Table 8. From Table 10, we see that all methods achieve h-independent convergence rates on this problem, except for PMINRES and PHSS-GMRES (the latter method fails to compute an approximate solution to the saddle-point linear system (1.1) arising from Example 4.2 when m is greater than or equal to 128 due to lack of sufficient computer memory). However, HSS-GMRES and IHSS-FGMRES, PHSS-GMRES and IPHSS-FGMRES are all outperformed by RHSS-GMRES and

Table 10 Numerical results for HSS-, PHSS-, RHSS-GMRES and IHSS-, IPHSS-, IRHSS-FGMRES methods, as well as for PMINRES method, for Example 4.2

Method	Index	m					
		64	96	128	192	256	384
HSS-GMRES	IT	47	48	48	48	49	48
	CPU	3.40	10.60	24.61	67.50	167.11	573.29
PHSS-GMRES	IT	3	6	—	—	—	—
	CPU	156.34	1728.30	—	—	—	—
RHSS-GMRES(a)	IT	17	16	16	16	16	16
	CPU	1.32	4.43	9.01	29.15	72.14	192.05
RHSS-GMRES(b)	IT	17	16	16	16	16	16
	CPU	1.27	4.46	8.97	28.82	75.01	210.15
IHSS-FGMRES	IT	52	52	53	52	55	55
	CPU	2.07	4.33	8.01	21.49	42.46	104.47
IPHSS-FGMRES	IT	8	8	8	9	8	10
	CPU	3.96	9.92	18.48	62.43	117.71	434.67
IRHSS-FGMRES(a)	IT	18	20	21	18	18	19
	CPU	0.80	1.83	3.43	8.46	16.14	41.19
IRHSS-FGMRES(b)	IT	19	20	21	18	18	19
	CPU	0.82	1.82	3.50	8.23	16.20	41.81
PMINRES	IT	31	35	37	47	53	65
	CPU	0.63	1.35	2.57	8.42	17.03	47.43

IRHSS-FGMRES in terms of both iteration counts and CPU time, often by a large margin. Moreover, IRHSS-FGMRES outperforms PMINRES for sufficiently fine meshes, thanks to its h-independent convergence behavior; note the almost perfect scaling of this method in terms of CPU time. Finally, we observe that for this problem the condition number of the Schur complement deteriorates with increasing problem size, whereas the (1,1) block has bounded condition number as $h \to 0$ for fixed values of ε and β.

5 Concluding remarks

The regularized HSS iteration method is a further generalization of the HSS iteration method that was initially proposed in [7] and extended to saddle-point linear systems in [9]. It is also a special case of the preconditioned HSS iteration method that was first discussed in [8]. Theoretical analyses and numerical experiments have shown that the regularized HSS iteration method can be an effective solver for saddle-point linear systems, especially in the case of a relatively well-conditioned (1,1)-block B and an ill-conditioned Schur complement. When used to precondition Krylov subspace iteration methods, the exact and the inexact RHSS-preconditioned (F)GMRES methods outperform the corresponding solvers based on the original HSS and PHSS

methods. Moreover, in the case of the optimal control problem for Poisson's equation the inexact RHSS-preconditioner can also outperform the standard PMINRES solver with the block-diagonal preconditioner proposed and analyzed in [19], in terms of both iteration counts and computing time, leading to (almost) optimal scaling behavior with respect to mesh size. Moreover, the inexact variants of the RHSS method are consistently faster in terms of computing time than the exact ones.

Future work should focus on techniques for estimating good values of the parameters α and γ used in the (I)RHSS preconditioner.

Acknowledgements The authors are very much indebted to Kang–Ya Lu for running the numerical results. They are also thankful to the referees for their constructive comments and valuable suggestions, which greatly improved the original manuscript of this paper.

References

1. Axelsson, O.: Iterative Solution Methods. Cambridge University Press, Cambridge (1996)
2. Bai, Z.-Z.: Structured preconditioners for nonsingular matrices of block two-by-two structures. Math. Comput. **75**, 791–815 (2006)
3. Bai, Z.-Z.: Motivations and realizations of Krylov subspace methods for large sparse linear systems. J. Comput. Appl. Math. **283**, 71–78 (2015)
4. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
5. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. SIAM J. Sci. Comput. **28**, 583–603 (2006)
6. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. Math. Comput. **76**, 287–298 (2007)
7. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix. Anal. Appl. **24**, 603–626 (2003)
8. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
9. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix. Anal. Appl. **26**, 20–41 (2004)
10. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
11. Benzi, M., Simoncini, V.: On the eigenvalues of a class of saddle point matrices. Numer. Math. **103**, 173–196 (2006)
12. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Element Methods. Springer, New York, London (1991)
13. Eisenstat, S.C., Elman, H.C., Schultz, M.H.: Variational iterative methods for nonsymmetric systems of linear equations. SIAM J. Numer. Anal. **20**, 345–357 (1983)
14. Elman, H.C., Silvester, D.J., Wathen, A.J.: Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations. Numer. Math. **90**, 665–688 (2002)
15. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics, 2nd edn. Oxford University Press, Oxford (2014)
16. Fortin, M., Glowinski, R.: Augmented Lagrangian Methods. Applications to the Numerical Solution of Boundary Value Problems. North-Holland, Amsterdam (1983)
17. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
18. Greenbaum, A.: Iterative Methods for Solving Linear Systems. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1997)
19. Pearson, J.W., Stoll, M., Wathen, A.J.: Preconditioners for state-constrained optimal control problems with Moreau–Yosida penalty function. Numer. Linear Algebra Appl. **21**, 81–97 (2014)
20. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2003)

21. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
22. Simoncini, V., Benzi, M.: Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 377–389 (2004)
23. Varga, R.S.: Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, N.J. (1962)
24. Wathen, A.J.: Preconditioning. Acta Numer. **24**, 329–376 (2015)