Technical Report

TR-2016-001

Regularized HSS Iteration Method for Saddle-Point Linear Systems

by

Zhong-Zhi Bai, Michele Benzi

Mathematics and Computer Science

EMORY UNIVERSITY

Regularized HSS Iteration Methods for Saddle-Point Linear Systems*

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
Email: bzz@lsec.cc.ac.cn

Michele Benzi

Department of Mathematics and Computer Science
Emory University, Atlanta GA 30322, USA
Email: benzi@mathcs.emory.edu

January 28, 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, showing that the eigenvalues are clustered near 0 and 2 when the iteration parameter is close to 0. 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 optimal convergence behavior 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.

AMS(MOS) Subject Classifications: 65F08, 65F10, 65F15, 65F22; CR: G1.3.

1 Introduction

Consider the following linear system in saddle-point form:

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

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

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 [9]. 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., [15, 11, 13, 9, 2, 22].

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 [8] Benzi and Golub proposed to apply the *Hermitian and skew-Hermitian splitting* (**HSS**) iteration method in [6] 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 [20] Simoncini and Benzi analyzed the spectral properties of the preconditioned matrix corresponding to the HSS preconditioner; see also [10]. To improve the convergence rate of the HSS iteration method, Bai, Golub and Pan presented in [7] 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 [19, 18].

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 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 analyze clustering properties of the eigenvalues of the RHSS-preconditioned matrix. Numerical results are given in Section 4. Finally, in Section 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. 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 $y^{(k+1/2)} \in \mathbb{C}^p$ and $u^{(k+1/2)} \in \mathbb{C}^q$ from the linear sub-systems

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

and compute $z^{(k+1/2)} \in \mathbb{C}^q$ by

$$z^{(k+1/2)} = z^{(k)} + u^{(k+1/2)};$$

(ii) compute

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

(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 [8]. Alternatively, the RHSS iteration method can be regarded as a special case of the PHSS iteration method developed in [7] with the particular preconditioning matrix

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

see also [5].

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 three linear sub-systems with respect to the Hermitian positive definite matrices

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

In general, thanks to the presence of the Hermitian positive semidefinite matrix Q, the last 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 actual computations, the afore-mentioned three 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., [21, 1, 16, 6, 18]. Note that the inexact iteration method may fail to converge unless the 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

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

and

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

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 need to use a flexible Krylov subspace method, such as FGMRES [18].

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 [21, 1, 16]. 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

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

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 [8] 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 [12, 18, 2] 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., [18, 3]. 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 [20, 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 (3.2) and (3.1), 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

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

With the change of variable

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

the equality (3.5) can then be rewritten as

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

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

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

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

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

we obtain

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

The two roots of this quadratic equation are

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

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

$$\lambda_\pm \approx 1 \pm \sqrt{1 - \frac{4\alpha}{\alpha + \nu}}.$$

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 [20]) that for α small the cluster near 2 contains p eigenvalues and the cluster near 0 the remaining q eigenvalues. As noted in [20], 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, it can be observed from the proof of Theorem 3.2 that the RHSS preconditioner does not affect the spectral properties of the (1,1)-block matrix B, instead it is preconditioning

the (1,2) block E and, symmetrically, the (2,1) block E^* as well. In other words, the RHSS preconditioning matrix is only preconditioning the Schur complement of the saddle-point matrix $A \in \mathbb{C}^{n\times n}$. As a result, we deduce that the RHSS iteration should be expected to work well 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.

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 solver in order to show the advantages of the RHSS approach relative to the older method.

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.

All experiments are started from the initial vector $x^{(0)} = 0$, terminated once the relative residual errors at the current iterates $x^{(k)}$ satisfy $\|b - Ax^{(k)}\| \leq 10^{-6}\|b\|$. 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 (ubantu 15.04). In our codes we utilize the IC (MATLAB function: `ichol(·)`) and the *modified IC* (**MIC**) (MATLAB function: `ichol(·,struct('droptol',1e-3,'michol','on'))`) factorizations to produce preconditioning matrices for certain sub-matrices precisely specified in the sequel.

Example 4.1. *[7] Consider the Stokes problem: Find u and p such that*

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

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 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 finite differences, we obtain the saddle-point linear system (1.1), in which

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

with

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

being tridiagonal matrices, 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. In both HSS and RHSS used either as linear solvers or as "exact" preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$, $\alpha I + Q$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved directly by sparse Cholesky factorization.

In our implementations of the exact and the inexact HSS and RHSS iteration methods, 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. 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 method, 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 Table 1 (the first two rows) we report the stopping tolerances used for the inner PCG iterations.

Method	Inner-Termination Tolerance				
	64	96	128	192	256
IHSS	1E-02	1E-02	1E-02	1E-05	1E-05
IRHSS	5E-02	5E-02	5E-02	1E-02	1E-02
IHSS-FGMRES	1E-02	1E-01	1E-02	1E-01	1E-02
IRHSS-FGMRES	1E-04	1E-03	1E-03	1E-03	1E-02

Table 1: Inner PCG stopping tolerances for IHSS and IRHSS methods for Example 4.1.

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 the HSS or the RHSS iteration method. We note that for RHSS, the optimal α slowly decreases as h decreases, while γ increases at a similar rate.

Method	Index	m				
		64	96	128	192	256
HSS	α	0.23	0.21	0.17	0.13	0.11
RHSS	α	0.08	0.06	0.05	0.04	0.03
	γ	8.0	10.0	12.0	14.0	18.0

Table 2: The values of iteration parameter α and regularization parameter γ in HSS and RHSS iteration methods for Example 4.1.

In Table 3 we report iteration counts and computing times for the exact and the inexact HSS and RHSS iteration methods. In the IHSS and the IRHSS iteration methods, we adopt the same parameters 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 IHSS 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. 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 [6] for IHSS can be adapted to account for the observed convergence of IRHSS. As the results show, all the four iteration methods tested here converge slowly on the Stokes problem, and the number of iterations is seen to increase as the mesh is refined.

Method	Index	m				
		64	96	128	192	256
HSS	IT	335	455	632	1055	1533
	CPU	6.72	36.18	101.92	481.25	1639.61
RHSS	IT	179	225	266	337	417
	CPU	4.72	21.05	56.74	206.39	598.20
IHSS	IT	369	455	963	1083	1577
	CPU	3.23	7.77	25.03	153.07	431.87
IRHSS	IT	208	311	438	405	598
	CPU	2.47	5.75	13.36	45.53	117.51

Table 3: Numerical results for HSS, RHSS and IHSS, IRHSS iteration methods for Example 4.1.

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 $Q = \gamma\,\mathrm{diag}(E^TE)$, where γ is a regularization parameter and $\mathrm{diag}(\cdot)$ represents the diagonal matrix constituted by the diagonal components of the corresponding matrix. In both IHSS and IRHSS iteration preconditioners, the linear sub-systems with the coefficient matrices $\alpha I + B$, $\alpha I + \frac{1}{\alpha}E^TE$ and $\alpha I + Q + \frac{1}{\alpha}E^TE$ are solved iteratively by the PCG method, for which all these three matrices are preconditioned by their MIC factorizations. The inner iteration stopping tolerances can be found in Table 1 (the last two 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 either the HSS-GMRES or the RHSS-GMRES method. 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 and RHSS-(F)GMRES methods. The GMRES method is used without restarts. In the IHSS-FGMRES and the IRHSS-FGMRES methods, we adopt the same parameters α and (or) γ as in the HSS-GMRES and the 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 preconditioners.

Method	Index	m				
		64	96	128	192	256
HSS-GMRES	α	110.0	160.0	185.0	205.0	220.0
RHSS-GMRES	α	0.004	0.006	0.010	0.060	0.200
	γ	200.0	150.0	100.0	30.0	10.0

Table 4: The values of iteration parameter α and regularization parameter γ in HSS-GMRES and RHSS-GMRES methods for Example 4.1.

Method	Index	m				
		64	96	128	192	256
HSS-GMRES	IT	72	87	100	125	150
	CPU	4.53	15.18	37.97	133.38	387.61
RHSS-GMRES	IT	40	44	48	56	63
	CPU	2.03	7.32	16.60	58.09	156.61
IHSS-FGMRES	IT	74	98	106	142	157
	CPU	2.23	5.77	11.09	34.75	104.52
IRHSS-FGMRES	IT	44	56	60	63	81
	CPU	1.70	3.70	8.14	23.74	52.75

Table 5: Numerical results for HSS-, RHSS-GMRES and IHSS-, IRHSS-FGMRES methods for Example 4.1.

We also note, however, that the convergence rate 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 one, it cannot be recommended as a solver for the discrete steady Stokes problem, for which several optimal, h-independent solvers are known [14]. The next example, however, shows that there are saddle-point linear systems for which the new techniques provide (nearly) optimal complexity.

Example 4.2. *[17] Consider the optimal control problem*

$$\begin{cases} \min\limits_{u,v} J(u,v) & := & \frac{1}{2}\|u-u_d\|^2_{\mathcal{L}^2(\Omega)} + \frac{\beta}{2}\|v\|^2_{\mathcal{L}^2(\Omega)}, & in \quad \Omega, \\ -\Delta u & = & v, \end{cases} \tag{4.2}$$

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, $\mathcal{L}^2(\Omega)$ is the $\mathcal{L}^2$-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

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

and

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

Here, the matrix $G_{\mathcal{I}}$ is a projection onto the active set $\mathcal{I} = \mathcal{I}_+ \cup \mathcal{I}_-$ with $\mathcal{I}_+ = \{i \mid u_i > \bar{u}_i\}$ and $\mathcal{I}_- = \{i \mid u_i < \underline{u}_i\}$, $M \in \mathbb{R}^{m^2 \times m^2}$ and $K \in \mathbb{R}^{m^2 \times m^2}$ are the mass and the stiffness matrices, respectively, λ is the Lagrangian multiplier, $c_{\mathcal{I}} = Mu_d + \epsilon^{-1}(G_{\mathcal{I}_+} M G_{\mathcal{I}_+} \bar{u} + G_{\mathcal{I}_-} M G_{\mathcal{I}_-} \underline{u})$, 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 $\mathcal{I}$ 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 $\alpha I + B$, $\alpha I + \frac{1}{\alpha} E^T E$, $\alpha I + Q$ and $\alpha I + Q + \frac{1}{\alpha} E^T E$ are solved directly by sparse Cholesky factorization.

In our implementations of the exact and the inexact HSS and RHSS iteration methods, again we choose the regularization matrix Q to be $Q = \gamma\, E^T E$, where γ is a regularization parameter. 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 method, 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 $\hat{K}(\alpha)^T \hat{K}(\alpha)$ with $\hat{K}(\alpha)$ being the algebraic multigrid preconditioner for $\alpha I + K$ obtained by utilizing the package HSL_MI20 (hsl_mi20_precondition) with default parameters. For the inexact variants, the inner PCG 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 the RHSS iteration method. We note that the general trend as $h \to 0$ is qualitatively similar to that in the case of Example 4.1.

For this problem, the number of iterations for both HSS and IHSS iteration methods is prohibitively high; both methods fail to converge within 10,000 iterations for all values of m, regardless of the value of α used. For this reason we do not report the values of α for HSS and IHSS in Table 6. Iteration counts and CPU times for the RHSS and the IRHSS iteration methods 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. On the other hand, we note that the number of iterations, while high, appears to grow slowly for both RHSS and IRHSS as the mesh is refined.

Method	Index	m				
		64	96	128	192	256
RHSS	α	6E-04	3E-04	2E-04	1E-04	6E-05
	γ	540	850	910	1440	2000

Table 6: The values of iteration parameter α and regularization parameter γ in RHSS iteration method for Example 4.2.

Method	Index	m				
		64	96	128	192	256
RHSS	IT	1093	1207	1367	1543	1699
	CPU	43.07	157.47	433.59	1406.88	3707.51
IRHSS	IT	1365	1471	1742	2078	2320
	CPU	22.30	46.08	96.12	275.21	553.20

Table 7: Numerical results for RHSS and IRHSS iteration methods for Example 4.2.

Next, we turn to the use of HSS and RHSS (IHSS 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 subsystems with the coefficient matrices $\alpha I+B$, $\alpha I+\frac{1}{\alpha}E^TE$ and $\alpha I+Q+\frac{1}{\alpha}E^TE$ are solved iteratively by the PCG method, for which the matrix $\alpha I + B$ is preconditioned by its MIC factorization. Moreover, in IHSS the matrix $\alpha I + \frac{1}{\alpha}E^TE$ is preconditioned by $\hat{K}(\alpha)^T\hat{K}(\alpha)$ with $\hat{K}(\alpha)$ being the algebraic multigrid preconditioning for $\alpha I + K$, and in IRHSS the matrix $\alpha I + Q + \frac{1}{\alpha}E^TE$ is preconditioned by $\hat{K}^T\hat{K}$ with $\hat{K}$ being the algebraic multigrid preconditioning of K for both choices (a) and (b) of Q. As before, the algebraic multigrid preconditioning operators are built by utilizing the package HSL_MI20 (hsl_mi20_precondition) with default parameters.

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 either the HSS-GMRES or the RHSS-GMRES method. Also, in Table 9 we report the stopping tolerances used for the inner PCG 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 and IRHSS preconditioners.

Method	Q		Index	m				
				64	96	128	196	256
HSS-GMRES	0		α	0.023	0.010	0.006	0.002	0.0008
RHSS-GMRES	(a)	$\gamma E^TE - \alpha I$	α	9.50	6.00	3.00	1.50	0.80
			γ	1E-08	1E-08	1E-08	1E-07	1E-07
	(b)	$\gamma KK^T - \alpha I$	α	10.00	6.00	4.00	2.00	1.00
			γ	1E-08	1E-08	1E-07	1E-07	1E-07

Table 8: The values of regularization parameter Q, iteration parameter α and regularization parameter γ in HSS-GMRES and RHSS-GMRES methods for Example 4.2.

In Table 10 we report iteration counts and computing times for the exact and the inexact HSS-(F)GMRES and RHSS-(F)GMRES methods corresponding to the two choices of the regularization matrix Q given in Table 8. In the IHSS-FGMRES and the IRHSS-FGMRES methods, we use the same parameters α and (or) γ as in the HSS-GMRES and the RHSS-GMRES methods, respectively; see Table 8. From Table 10, we see that all methods achieve h-independent

Method	Inner-Termination Tolerance				
	64	96	128	192	256
IRHSS	5E-01	7E-01	7E-01	7E-01	5E-01
IHSS-FGMRES	1E-02	1E-02	1E-02	1E-03	1E-03
IRHSS-FGMRES(a)	1E-05	1E-05	1E-04	1E-06	1E-05
IRHSS-FGMRES(b)	1E-05	1E-05	1E-04	1E-05	1E-05

Table 9: Inner PCG stopping tolerances for IHSS and IRHSS methods for Example 4.2.

convergence rates on this problem. However, the HSS-GMRES and IHSS-FGMRES methods are vastly outperformed by RHSS-GMRES and IRHSS-FGMRES in terms of both iteration counts and CPU time. Finally, we observe that IRHSS-FGMRES exhibits almost optimal scaling in terms of CPU time as the mesh is refined.

Method	Index	m				
		64	96	128	192	256
HSS-GMRES	IT	62	58	57	58	58
	CPU	4.57	12.56	26.47	80.52	194.17
RHSS-GMRES(a)	IT	22	19	19	19	19
	CPU	1.65	5.16	12.09	34.61	73.39
RHSS-GMRES(b)	IT	22	19	19	19	19
	CPU	1.61	5.41	11.96	33.88	74.25
IHSS-FGMRES	IT	66	61	62	61	65
	CPU	3.45	5.43	9.52	26.16	50.63
IRHSS-FGMRES(a)	IT	25	22	25	20	22
	CPU	1.25	2.27	4.11	9.98	19.05
IRHSS-FGMRES(b)	IT	25	22	25	23	22
	CPU	1.30	2.23	4.19	10.24	19.30

Table 10: Numerical results for HSS-, RHSS-GMRES and IHSS-, IRHSS-FGMRES methods for Example 4.2.

5 Concluding Remarks

The regularized HSS iteration method is a further generalization of the HSS iteration method that was initially proposed in [6] and extended to saddle-point linear systems in [8]. It is also a special case of the preconditioned HSS iteration method that was first discussed in [7]. 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 preconditioning Krylov subspace iteration methods, the exact and the inexact RHSS-preconditioned (F)GMRES methods outperform the corresponding solvers based on the original HSS method in terms of both iteration counts and computing time and lead to (almost) optimal scaling behavior in some

cases. Moreover, the inexact variants of the RHSS method are 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.

Acknowledgments: The authors are very much indebted to Kang-Ya Lu for running the numerical results.

References

[1] O. Axelsson, Iterative Solution Methods, *Cambridge University Press*, Cambridge, 1996.

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

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

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

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

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

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

[8] M. Benzi and G.H. Golub, A preconditioner for generalized saddle point problems, *SIAM J. Matrix. Anal. Appl.*, 26(2004), 20-41.

[9] M. Benzi, G.H. Golub and J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, 14(2005), 1-137.

[10] M. Benzi and V. Simoncini, On the eigenvalues of a class of saddle point matrices, *Numer. Math.*, 103(2006), 173-196.

[11] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods, *Springer-Verlag*, New York and London, 1991.

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

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

[14] H.C. Elman, D.J. Silvester and A.J. Wathen, Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics, Second Edition, *Oxford University Press*, Oxford, 2014.

[15] M. Fortin and R. Glowinski, Augmented Lagrangian Methods, Applications to the Numerical Solution of Boundary Value Problems, *North-Holland*, Amsterdam, 1983.

[16] G.H. Golub and C.F. Van Loan, Matrix Computations, 3rd Edition, *The Johns Hopkins University Press*, Baltimore, 1996.

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

[18] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition, *Society for Industrial and Applied Mathematics*, Philadelphia, 2003.

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

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

[21] R.S. Varga, Matrix Iterative Analysis, *Prentice-Hall*, Englewood Cliffs, N.J., 1962.

[22] A.J. Wathen, Preconditioning, *Acta Numer.*, 24(2015), 329-376.