Numer Algor (2017) 74:781–795
DOI 10.1007/s11075-016-0171-2

ORIGINAL PAPER

A new relaxed HSS preconditioner for saddle point problems

Davod Khojasteh Salkuyeh[1] · Mohsen Masoudi[1]

Received: 7 March 2016 / Accepted: 22 June 2016 / Published online: 7 July 2016

Abstract We present a preconditioner for saddle point problems. The proposed preconditioner is extracted from a stationary iterative method which is convergent under a mild condition. Some properties of the preconditioner as well as the eigenvalues distribution of the preconditioned matrix are presented. The preconditioned system is solved by a Krylov subspace method like restarted GMRES. Finally, some numerical experiments on test problems arisen from finite element discretization of the Stokes problem are given to show the effectiveness of the preconditioner.

Keywords Saddle point problems · HSS preconditioner · Preconditioning · Krylov subspace method · GMRES

Mathematics Subject Classification (2010) 65F08 · 65F10

1 Introduction

We study the solution of the system of linear equations with the following block 2×2 structure

$$\mathcal{A}u = \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \equiv b, \tag{1.1}$$

✉ Davod Khojasteh Salkuyeh
khojasteh@guilan.ac.ir

Mohsen Masoudi
masoudi_mohsen@phd.guilan.ac.ir

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

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m \times n}$ with $rank(B) = m < n$. In addition, $x, f \in \mathbb{R}^n$, and $y, g \in \mathbb{R}^m$. We also assume that the matrices A and B are large and sparse. According to Lemma 1.1 in [13], the matrix $\mathcal{A}$ is nonsingular. Such systems are called saddle point problems and appear in a variety of scientific and engineering problems; e.g., computational fluid dynamics, constrained optimization, etc. The readers are referred to [3, 14] for more discussion on this subject.

Several efficient iterative methods have been proposed during the recent decades to solve the saddle point problems (1.1), such as SOR-like method [24], modified block SSOR iteration [1, 2], generalized SOR method [10], Uzawa method [28], parametrized inexact Uzawa methods [11], Hermitian and skew-Hermitian splitting (HSS) iteration methods [4, 7, 8], and so on. However, in some situations, these iterative methods may be less efficient than the Krylov subspace methods [28]. On the other hand, when Krylov subspace methods are applied to the saddle point problem (1.1), tend to converge slowly. But these methods can produce suitable preconditioners for accelerating the rate of convergence of the Krylov subspace methods. In general, favorable rates of convergence of Krylov subspace methods are often associated with a clustering of most of the eigenvalues of preconditioned matrices around 1 and away from zero [12]. In view of this, many preconditioners have been presented in literature, e.g., block diagonal preconditioners [26, 30], constraint preconditioners [9, 25], block triangular preconditioners [6, 18, 29, 31], parametrized block triangular preconditioners [23], and HSS preconditioners [8, 13, 27].

In [7], Bai et al. proposed the HSS iteration method to solve non-Hermitian positive definite linear systems $Ax = b$ which converges unconditionally to the unique solution of the system. For a given initial guess x^0, the HSS iteration can be written as

$$\begin{cases} (\alpha I + H)\, x^{k+\frac{1}{2}} = (\alpha I - S)\, x^k + b, \\ (\alpha I + S)\, x^{k+1} = (\alpha I - H)\, x^{k+\frac{1}{2}} + b, \end{cases} \quad k = 0, 1, 2, \ldots, \tag{1.2}$$

where $\alpha > 0$ and $A = H + S$, in which $H = (A + A^*)/2$ and $S = (A - A^*)/2$, where A^* denotes the conjugate transpose of A.

Benzi and Golub in [13] have applied the HSS iteration method to the generalized saddle point problem (saddle point problems with nonzero (2, 2)-block). As they mentioned, the convergence of the method to solve the saddle point problem is typically too slow for the method to be competitive. For this reason, they proposed using a nonsymmetric Krylov subspace method like the GMRES algorithm or its restarted version to accelerate the convergence of the iteration. Since the method has promising performance and elegant mathematical properties, it has attracted many researchers attention and many algorithmic variants and theoretical analysis of the HSS iteration for saddle point problems have been presented. In [5], Bai et al. investigated the convergence properties of the HSS iteration for the saddle point problem (1.1) with A being non-Hermitian and positive semidefinite. In [15], Benzi and Guo proposed a dimensional split (DS) preconditioner for the Stokes and the linearized Navier-Stokes equations. The DS preconditioner is extracted from an HSS iteration method based on the dimensional splitting of $\mathcal{A}$. A modification of the DS preconditioner has been presented by Cao et al. in [17]. Benzi et al. have presented a relaxed version of DS in [16]. Some variants of the HSS preconditioner including their relaxed versions have also been presented in the literature (see, e.g., [19, 22, 32]). In this paper, we present a new preconditioner which can be considered as a relaxed version of the HSS preconditioner for the saddle point problem.

Throughout the paper, for a matrix X, $\rho(X)$ and X^* stand for the spectral radius and conjugate transpose of X, respectively. For a vector $x \in \mathbb{C}^n$, $\|x\|_2$ denotes the Euclidian norm of x. For a given matrix $A \in \mathbb{R}^{n\times n}$ and a vector $r \in \mathbb{R}^n$, the Krylov subspace $\mathcal{K}_m(A, r)$ is defined as $\mathcal{K}_m(A, r) = span\{r, Ar, \ldots, A^{m-1}r\}$.

This paper is organized as follows. In Section 2, we present our preconditioner. Some properties of the preconditioner are presented in Section 3. Implementation of the proposed preconditioner is presented in Section 4. Numerical experiments are given in Section 5. The paper is ended by some concluding remarks in Section 6.

2 A review of the HSS preconditioner and its relaxed version

In this section, we first briefly review the HSS iteration method and the induced HSS preconditioner for the saddle point problem. Then, a relaxed version of the HSS (RHSS) preconditioner, proposed by Cao et al. in [19], is presented. Next, we give a new relaxed HSS (REHSS) preconditioner and investigate some of its properties.

2.1 The HSS preconditioner for the saddle point problem

According to the HSS iteration, the matrix $\mathcal{A}$ is split as

$$\mathcal{A} = \mathcal{H} + \mathcal{S},$$

where

$$\mathcal{H} = \frac{1}{2}\left(\mathcal{A} + \mathcal{A}^T\right) = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{S} = \frac{1}{2}\left(\mathcal{A} - \mathcal{A}^T\right) = \begin{bmatrix} 0 & B^T \\ -B & 0 \end{bmatrix}.$$

Obviously, both of the matrices $\alpha\mathcal{I} + \mathcal{H}$ and $\alpha\mathcal{I} + \mathcal{S}$ are nonsingular. In this case, the HSS iteration for the saddle point problem (1.1) is written as

$$\begin{cases} (\alpha\mathcal{I} + \mathcal{H})\, x^{k+\frac{1}{2}} = (\alpha\mathcal{I} - \mathcal{S})\, x^k + b, \\ (\alpha\mathcal{I} + \mathcal{S})\, x^{k+1} = (\alpha\mathcal{I} - \mathcal{H})\, x^{k+\frac{1}{2}} + b. \end{cases} \tag{2.1}$$

Computing $x^{k+\frac{1}{2}}$ from the first equation and substituting it in the second equation yields the iteration

$$x^{k+1} = \Gamma_{HSS} x^k + c,$$

where

$$\Gamma_{HSS} = (\alpha\mathcal{I} + \mathcal{S})^{-1} (\alpha\mathcal{I} - \mathcal{H}) (\alpha\mathcal{I} + \mathcal{H})^{-1} (\alpha\mathcal{I} - \mathcal{S}),$$

and

$$c = 2\alpha\, (\alpha\mathcal{I} + \mathcal{S})^{-1} (\alpha\mathcal{I} + \mathcal{H})^{-1}\, b.$$

It is known that there is a unique splitting $\mathcal{A} = \mathcal{M}_\alpha - \mathcal{N}_\alpha$, with $\mathcal{M}_\alpha$ being nonsingular, which induces the iteration matrix Γ_{HSS}, i.e.,

$$\Gamma_{HSS} = \mathcal{M}_\alpha^{-1}\mathcal{N}_\alpha = \mathcal{I} - \mathcal{M}_\alpha^{-1}\mathcal{A},$$

where

$$\mathcal{M}_\alpha = \frac{1}{2\alpha}\left(\alpha\mathcal{I}+\mathcal{H}\right)\left(\alpha\mathcal{I}+\mathcal{S}\right), \quad \mathcal{N}_\alpha = \frac{1}{2\alpha}\left(\alpha\mathcal{I}-\mathcal{H}\right)\left(\alpha\mathcal{I}-\mathcal{S}\right). \tag{2.2}$$

Benzi et al. in [13] have shown that for all $\alpha > 0$, the HSS iteration is convergent unconditionally to the unique solution of the saddle point problem (1.1). As we know, the HSS iteration serves the preconditioner $\mathcal{M}_\alpha$ for the system (1.1) which is called the HSS preconditioner. Since the pre-factor $\frac{1}{2\alpha}$ in the HSS preconditioner $\mathcal{M}_\alpha$ has no effect on the preconditioned system, the HSS preconditioner can be written in the form

$$\begin{aligned}\mathcal{P}_{HSS} &= \frac{1}{\alpha}\left(\alpha\mathcal{I}+\mathcal{H}\right)\left(\alpha\mathcal{I}+\mathcal{S}\right) = \frac{1}{\alpha}\begin{bmatrix} A+\alpha I & 0 \\ 0 & \alpha I\end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & \alpha I\end{bmatrix} \\ &= \begin{bmatrix} A+\alpha I & B^T+\frac{1}{\alpha}AB^T \\ -B & \alpha I\end{bmatrix}.\end{aligned} \tag{2.3}$$

The difference between the HSS preconditioner $\mathcal{P}_{HSS}$ and the coefficient matrix $\mathcal{A}$ is

$$\mathcal{R}_{HSS} = \mathcal{P}_{HSS} - \mathcal{A} = \begin{bmatrix} \alpha I & \frac{1}{\alpha}AB^T \\ 0 & \alpha I\end{bmatrix}. \tag{2.4}$$

2.2 The RHSS preconditioner

From (2.4), we see that as α tends to zero, the diagonal blocks tend to zero while the nonzero off-diagonal block becomes unbounded. Hence, it is sought an appropriate α to balance the weight of both parts. To do so, Cao et al. in [19] consider the following relaxed HSS (RHSS) preconditioner for the saddle point problem (1.1)

$$\mathcal{P}_{RHSS} = \frac{1}{\alpha}\begin{bmatrix} A & 0 \\ 0 & \alpha I\end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & 0\end{bmatrix} = \begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & 0\end{bmatrix}. \tag{2.5}$$

In this case, the difference between the RHSS preconditioner and the matrix $\mathcal{A}$ is given by

$$\mathcal{R}_{RHSS} = \mathcal{P}_{RHSS} - \mathcal{A} = \begin{bmatrix} 0 & \left(\frac{1}{\alpha}A - I\right)B^T \\ 0 & 0\end{bmatrix}. \tag{2.6}$$

Here, we see that as the parameter α tends to zero, the $(1,2)$-block of $\mathcal{R}_{RHSS}$ becomes unbounded.

From (2.6), we have $\mathcal{A} = \mathcal{P}_{RHSS} - \mathcal{R}_{RHSS}$ which produces the RHSS iteration

$$\mathcal{P}_{RHSS}x^{k+1} = \mathcal{R}_{RHSS}x^k + b, \quad k = 0, 1, \ldots,$$

where x^0 is an initial guess. Hence, the iteration matrix of the RHSS iteration is given by $\Gamma_{RHSS} = \mathcal{P}_{RHSS}^{-1}\mathcal{R}_{RHSS}$. In [19], it was shown that $\rho\left(\Gamma_{RHSS}\right) < 1$ for all $0 < \alpha < \frac{2}{\mu_1}$ and the optimal value of α is $\alpha_{opt} = 2/(\mu_1+\mu_m)$, where μ_1 and μ_m are, respectively, the largest and smallest eigenvalues of the matrix $\left(BB^T\right)^{-1}\left(BA^{-1}B^T\right)$.

3 The REHSS preconditioner

As we mentioned when α tends to zero, the $(1,2)$-block in both of the matrices $\mathcal{R}_{HSS}$ and $\mathcal{R}_{RHSS}$ become unbounded. To overcome this problem, we consider the following splitting for the matrix $\mathcal{A}$ as

$$\mathcal{A} = \mathcal{P}_{REHSS} - \mathcal{R}_{REHSS} = \begin{bmatrix} A & AB^T \\ -B & \alpha I \end{bmatrix} - \begin{bmatrix} 0 & (A-I)B^T \\ 0 & \alpha I \end{bmatrix}, \tag{3.1}$$

where $\alpha > 0$. As α tends to zero the $(2,2)$-block of $\mathcal{R}_{REHSS}$ tends to zero and in contrast with the HSS and the RHSS preconditioners the $(1,2)$-block remains bounded. This means that, for small values of α the REHSS preconditioner should be closer to the coefficient matrix $\mathcal{A}$ than the HSS and the RHSS preconditioners.

From the REHSS splitting (3.1), we state the REHSS iteration as

$$\begin{bmatrix} A & AB^T \\ -B & \alpha I \end{bmatrix} u^{k+1} = \begin{bmatrix} 0 & (A-I)B^T \\ 0 & \alpha I \end{bmatrix} u^k + \begin{bmatrix} f \\ g \end{bmatrix},$$

to solve the saddle point problem (1.1). In this case, the iteration matrix of the REHSS iteration is given by

$$\Gamma_{REHSS} = \mathcal{P}_{REHSS}^{-1}\mathcal{R}_{REHSS} = \begin{bmatrix} A & AB^T \\ -B & \alpha I \end{bmatrix}^{-1} \begin{bmatrix} 0 & (A-I)B^T \\ 0 & \alpha I \end{bmatrix}. \tag{3.2}$$

The next theorem provides a sufficient condition for the convergence of the REHSS iteration.

Theorem 1 *Let* $Q = B\left(\frac{1}{2}A^{-1} - I\right)B^T$. *If* $\delta = \lambda_{\max}(Q)$, *then for every* $\alpha > \max\{\delta, 0\}$, *it holds that* $\rho(\Gamma_{REHSS}) < 1$.

Proof We have

$$\mathcal{P}_{REHSS} = \mathcal{M}_1\mathcal{M}_2 = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & B^T \\ -B & \alpha I \end{bmatrix},$$

where

$$\mathcal{M}_2 = \begin{bmatrix} I & B^T \\ -B & \alpha I \end{bmatrix} = \begin{bmatrix} I & 0 \\ -B & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \alpha I + BB^T \end{bmatrix}\begin{bmatrix} I & B^T \\ 0 & I \end{bmatrix}.$$

Therefore,

$$\begin{aligned} \mathcal{P}_{REHSS}^{-1} &= \mathcal{M}_2^{-1}\mathcal{M}_1^{-1} \\ &= \begin{bmatrix} I & -B^T \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & (\alpha I + BB^T)^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ B & I \end{bmatrix}\begin{bmatrix} A^{-1} & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} A^{-1} - B^TS^{-1}BA^{-1} & -B^TS^{-1} \\ S^{-1}BA^{-1} & S^{-1} \end{bmatrix}, \end{aligned} \tag{3.3}$$

where $S = \alpha I + BB^T$. Hence, we get

$$\mathcal{P}_{REHSS}^{-1}\mathcal{A} = \begin{bmatrix} I & \tilde{A} \\ 0 & \hat{A} \end{bmatrix}, \tag{3.4}$$

where $\tilde{A} = A^{-1}B^T - B^T S^{-1} B A^{-1} B^T$ and $\hat{A} = S^{-1} B A^{-1} B^T$. As a result, we obtain

$$\begin{aligned}\Gamma_{REHSS} &= \mathcal{P}_{REHSS}^{-1}\mathcal{R}_{REHSS} = \mathcal{P}_{REHSS}^{-1}(\mathcal{P}_{REHSS} - \mathcal{A}) \\ &= I - \mathcal{P}_{REHSS}^{-1}\mathcal{A} = \begin{bmatrix} 0 & -\tilde{A} \\ 0 & I - \hat{A} \end{bmatrix}.\end{aligned}$$

Hence, if λ is an eigenvalue of the matrix Γ_{REHSS}, then $\lambda = 0$ or $\lambda = 1 - \mu$, where μ is an eigenvalue of the matrix $\hat{A}$. Therefore, there exists a vector $x \neq 0$ such that

$$\hat{A}x = \left(\alpha I + BB^T\right)^{-1} BA^{-1}B^T x = \mu x.$$

Without loss of generality, we assume that $\|x\|_2 = 1$. Since $B^T x \neq 0$, we have

$$\mu = \frac{x^* BA^{-1}B^T x}{\alpha + x^* BB^T x} > 0.$$

Hence, $|\lambda| < 1$ if and only if

$$\frac{x^* BA^{-1}B^T x}{\alpha + x^* BB^T x} < 2.$$

which is equivalent to

$$\alpha > x^* B\left(\frac{1}{2}A^{-1} - I\right)B^T x = x^* Q x. \tag{3.5}$$

Therefore, a sufficient condition to have $|\lambda| < 1$ is

$$\alpha > \max_{\|x\|_2=1} x^* Q x = \lambda_{\max}(Q) = \delta.$$

It is necessary to mention that the matrix Q is symmetric and hence all of its eigenvalues are real. □

Corollary 1 *Assume that*

$$\lambda_{\min}(A) > \frac{1}{2}\kappa(B)^2, \tag{3.6}$$

where $\kappa(B)$ and $\lambda_{\min}(A)$ stand for the spectral condition number and smallest eigenvalue of A. Then, for every $\alpha > 0$, it holds that $\rho(\Gamma_{REHSS}) < 1$.

Proof From [28, Theorem 1.22], we have

$$\begin{aligned} x^* BA^{-1}B^T x &\le \lambda_{\max}(A^{-1}) x^* BB^T x \le \frac{1}{\lambda_{\min}(A)}\lambda_{\max}(BB^T) x^* x = \frac{\sigma_{\max}(B)^2}{\lambda_{\min}(A)}, \\ x^* BB^T x &\ge \lambda_{\min}(BB^T) x^* x = \sigma_{\min}(B)^2, \end{aligned}$$

where $\sigma_{\min}(B)$ and $\sigma_{\max}(B)$ stand for the smallest and largest singular values of B. From these inequalities and (3.5), we deduce

$$x^* Qx = \frac{1}{2} x^* BA^{-1}B^T x - x^* BB^T x \leq \frac{1}{2}\frac{\sigma_{\max}(B)^2}{\lambda_{\min}(A)} - \sigma_{\min}(B)^2 = \theta.$$

It follows from this equation that $\delta \leq \theta$, where δ was defined in Theorem 1. Hence, if $\alpha > \max\{0, \theta\}$, then the convergence of the REHSS iteration is achieved. Now, if $\theta < 0$ then for every $\alpha > 0$, we get $\rho(\Gamma_{REHSS}) < 1$. Obviously, $\theta < 0$ is equivalent to the condition (3.6). □

The next theorem analyses the behavior of $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$.

Theorem 2 (a) *For $\alpha > 0$, the preconditioned matrix $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$ has eigenvalue 1 of algebraic multiplicity at least n. The remaining eigenvalues are μ_i, where μ_i are the eigenvalues of the $m \times m$ matrix $\hat{A} = \left(\alpha I + BB^T\right)^{-1} BA^{-1}B^T$.*

(b) *Let $(\mu, [x; y])$ be an eigenpair of $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$. Then, $x \neq 0$ and μ is either equal to 1 or can be written as $\mu = (\alpha\hat{b} + \hat{c})/\hat{a}$, where*

$$\hat{a} = x^* \left(\alpha I + B^T B\right) A \left(\alpha I + B^T B\right) x,\ \hat{b} = x^* \left(B^T B\right) x,\ \hat{c} = x^* \left(B^T B\right)^2 x.$$

Moreover, when $\alpha \to 0$, then μ is either equal to 1 or

$$\frac{1}{\mu_{\max}(A)} \leqslant \mu \leqslant \frac{1}{\mu_{\min}(A)},$$

where $\mu_{\min}(A)$ and $\mu_{\min}(A)$ are the smallest and largest eigenvalues of A, respectively.

Proof Part (a) follows immediately from (3.4). To prove (b), let $(\mu, [x; y])$ be an eigenpair of $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$. Therefore,

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} = \mu \mathcal{P}_{REHSS}\begin{bmatrix} x \\ y \end{bmatrix},$$

which is equivalent to

$$\begin{aligned} Ax + B^T y &= \mu Ax + \mu AB^T y, \\ -Bx &= -\mu Bx + \mu\alpha y. \end{aligned} \tag{3.7}$$

Hence

$$(\mu - 1)\, Ax + (\mu A - I)\, B^T y = 0, \tag{3.8}$$

$$\mu\alpha y = (\mu - 1)\, Bx. \tag{3.9}$$

Premultiplying both sides of (3.9) by B^T yields

$$\mu\alpha B^T y = (\mu - 1)\, B^T Bx. \tag{3.10}$$

Multiplying both sides of (3.8) by $\mu\alpha$ and substituting (3.10) in it, gives

$$\mu\alpha\left(\mu-1\right)Ax+\left(\mu A-I\right)\left(\mu-1\right)B^TBx=0.$$

We show that $x\neq 0$. Otherwise, from (3.7), we get $\mu=0$ or $y=0$. In fact, neither of them can be zero. So $x\neq 0$. Without loss of generality, let $\|x\|_2=1$. Hence,

$$\mu^2A\left(\alpha I+B^TB\right)x-\mu\left(A\left(\alpha I+B^TB\right)+B^TB\right)x+B^TBx=0. \qquad (3.11)$$

Multiplying $x^*\left(\alpha I+B^TB\right)$ to both sides of (3.11), yields

$$\hat{a}\mu^2-\left(\hat{a}+\alpha\hat{b}+\hat{c}\right)\mu+\left(\alpha\hat{b}+\hat{c}\right)=0.$$

The roots of this quadratic equation are $\mu=1$ and

$$\mu=\frac{\alpha\hat{b}+\hat{c}}{\hat{a}}=\frac{\alpha\hat{b}+\hat{c}}{\alpha^2x^*Ax+\alpha\left(x^*B^TBAx+x^*AB^TBx\right)+x^*B^TBAB^TBx}.$$

To prove the last part of theorem, we show that if $Bx=0$, then $\mu=1$. If $Bx=0$, then it follows from (3.9) that $y=0$. Substituting this in (3.8), yields $(\mu-1)Ax=0$. Now, since $Ax\neq 0$, we conclude that $\mu=1$. Therefore, if $\alpha\to 0$, then $\mu=1$ or

$$\mu=\frac{x^*\left(B^TB\right)^2x}{x^*B^TBAB^TBx}=\frac{(B^TBx)^*(B^TBx)}{(B^TBx)^*A(B^TBx)}=\frac{z^*z}{z^*Az},$$

where $z=B^TBx$. Since, A is symmetric positive definite we have

$$\frac{1}{\mu_{\max}(A)}\leqslant\frac{z^*z}{z^*Az}\leqslant\frac{1}{\mu_{\min}(A)},$$

which completes the proof. □

Theorem 3 *The degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$ is at most $m+1$. Thus, the dimension of the Krylov subspace $\mathcal{K}_n(\mathcal{P}_{REHSS}^{-1}\mathcal{A},b)$ is at most $m+1$.*

Proof Let χ be the characteristic polynomial of the preconditioned matrix $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$. By using (3.4), we have

$$\chi(x)=(x-1)^n\prod_{i=1}^{m}\left(x-\mu_i\right),$$

where μ_i, for $i=1,\ldots,m$, are the eigenvalues of the matrix $\hat{A}$. Let

$$p(x)=(x-1)\prod_{i=1}^{m}\left(x-\mu_i\right).$$

Therefore,

$$\begin{aligned} p(\mathcal{P}_{REHSS}^{-1}\mathcal{A}) &= \left(\mathcal{P}_{REHSS}^{-1}\mathcal{A}-\mathcal{I}\right)\prod_{i=1}^{m}\left(\mathcal{P}_{REHSS}^{-1}\mathcal{A}-\mu_i\mathcal{I}\right) \\ &= \begin{bmatrix} 0 & \tilde{A} \\ 0 & \hat{A}-I_m \end{bmatrix}\prod_{i=1}^{m}\begin{bmatrix} (1-\mu_i)I_n & \tilde{A} \\ 0 & \hat{A}-\mu_i I_m \end{bmatrix} \\ &= \begin{bmatrix} 0 & \tilde{A}\prod_{i=1}^{m}(\hat{A}-\mu_i I_m) \\ 0 & (\hat{A}-I_m)\prod_{i=1}^{m}(\hat{A}-\mu_i I_m) \end{bmatrix}. \end{aligned}$$

Since for $i=1,\ldots,m$, μ_i is an eigenvalue of the matrix $\hat{A}$, we have

$$\prod_{i=1}^{m}(\hat{A}-\mu_i I_m)=0,$$

and so $p(\mathcal{P}_{REHSS}^{-1}\mathcal{A}) = 0$. Therefore, the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$ is at most $m+1$. Hence, by [28, Proposition 6.1], the dimension of the Krylov subspace $\mathcal{K}_n(\mathcal{P}_{REHSS}^{-1}\mathcal{A}, b)$ is also at most $m+1$. □

4 Implementation of P_{REHSS}

We use the restarted version of the GMRES (denoted by GMRES(m)) in conjunction with the preconditioner P_{REHSS} to solve the saddle point problem (1.1). At each step of applying the preconditioner P_{REHSS} within the GMRES(m) algorithm, we need to compute a vector of the form $z = P_{REHSS}^{-1}r$ for a given vector $r=[r_1; r_2]$ where $r_1 \in \mathbb{R}^n$ and $r_2 \in \mathbb{R}^m$. Let $z=[z_1; z_2]$, where $z_1 \in \mathbb{R}^n$ and $z_2 \in \mathbb{R}^m$. Now, form (3.4), we can compute the vector z via

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} I & -B^T \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \left(\alpha I + BB^T\right)^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ B & I \end{bmatrix}\begin{bmatrix} A^{-1} & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}.$$

We can use Algorithm 1 to compute the vector z.

Algorithm 1 Computation of $z = P_{REHSS}^{-1}r$

1. Solve $Aw_1 = r_1$ for w_1.
2. Solve $(\alpha I + BB^T)w_2 = Bw_1 + r_2$ for w_2.
3. $z_2 := w_2$.
4. $z_1 := w_1 - B^T w_2$.

Both of the matrices A and $\alpha I + BB^T$ are symmetric positive definite. Hence, we can solve the systems appeared in steps 1 and 2 of Algorithm 4 exactly by the Cholesky factorization or approximately by the conjugate gradient (CG) or the preconditioned conjugate gradient (PCG) iterative method.

5 Numerical experiments

In this section, we present some numerical experiments to illustrate the effectiveness of the preconditioner $\mathcal{P}_{REHSS}$ for the saddle point problem (1.1). The restarted GMRES method [28] with restarting frequency 30, i.e., GMRES (30), is applied to the left preconditioned saddle point problem (1.1) in conjunction with the preconditioner $\mathcal{P}_{REHSS}$ and the corresponding numerical results are compared with those of the preconditioners $\mathcal{P}_{HSS}$ and $\mathcal{P}_{RHSS}$ in terms of iteration counts and CPU timings. All runs are performed in MATLAB 2014 on an Intel core i7 (12G RAM) Windows 8 system.

In all the tests, the initial vector is set to be a zero vector and the right-hand side vector $b = [f; g] \in \mathbb{R}^{n+m}$ is chosen such that the exact solution of the saddle point problem (1.1) is a vector of all ones. We use the stopping criterion

$$\|\mathcal{P}r_k\|_2 \leqslant 10^{-12} \|\mathcal{P}b\|_2 ,$$

where $r_k = b - \mathcal{A}u_k$ is the residual at the kth iteration and $\mathcal{P}$ is one of the preconditioners $\mathcal{P}_{HSS}$, $\mathcal{P}_{RHSS}$, or $\mathcal{P}_{REHSS}$. The maximum number of the iterations and the maximum elapsed CPU time are set to be $k_{\max} = 500$ and $t_{\max} = 3600$ s, respectively. Throughout this section, “IT” and “CPU” stand for the numbers of the restarts in GMRES(m) and the CPU time, respectively. In all the tables, a dagger (†) shows that the method has not converged in at most $k_{\max}$ iterations. Similarly, a “‡” shows that the method has not converged after elapsing $t_{\max}$ seconds. At each step of applying the preconditioners $\mathcal{P}_{HSS}$, $\mathcal{P}_{RHSS}$, and $\mathcal{P}_{REHSS}$, we need to solve two sub-systems with symmetric positive definite coefficients matrix (see Algorithm 1 and [19, Algorithm 3.3 and Algorithm 3.4]) and all of these systems are solved by the Cholesky factorization.

Consider the Stokes problem (see [20] or [21, page 221])

$$\begin{cases} -\triangle \mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla . \mathbf{u} = 0, \end{cases} \tag{5.1}$$

in $\Omega = [-1, 1] \times [-1, 1]$, where Δ, ∇, $\mathbf{u}$, and $\mathbf{p}$ stand for the Laplace operator, the gradient operator, velocity, and pressure of the fluid, respectively, with suitable boundary condition on $\partial\Omega$. It is known that many discretization schemes for (5.1) will lead to saddle point problems of the form (1.1). We consider Q2-P1 finite element discretizations on uniform grids on the unit square of the tree standard model problems (see [20, 21])

1. The leaky lid-driven cavity problem;
2. The channel domain problem;
3. The colliding flow problem.

Table 1 The size of the matrices A and B for different of grids

	Channel domain problem				Lid driven cavity and colliding flow problem			
Grid	n	m	$nnz(A)$	$nnz(B)$	n	m	$nnz(A)$	$nnz(B)$
16×16	578	192	6698	2084	578	190	6178	1967
32×32	2178	768	29,546	10,142	2178	766	28,418	9868
64×64	8450	3072	124,550	45,062	8450	3070	122,206	44,516
128×128	33,282	12,288	511,152	192,174	33,282	12,286	506,376	191,084
256×256	132,098	49,152	2,070,764	791,738	132,098	49,150	2,061,140	789,560

We use the IFISS software package developed by Elman et al. [20] to generate the linear systems corresponding to 16×16, 32×32, 64×64, 128×128, and 256×256 meshes. The IFISS software provides the matrices `Ast` and `Bst` for the matrices A and B, respectively. For the channel domain problem, the matrix `Bst` is of full rank, but for the colliding flow and lid-driven cavity problem is rank deficient. Therefore, in these cases, we drop two first rows of `Bst` to get a full rank matrix. Generic information of the test problems, including n, m, $nnz(A)$ and $nnz(B)$, are given in Table 1 where nnz stand for the number of the nonzero entries of a matrix. We present the numerical results for different values of α ($\alpha = 10^{-4}, 10^{-2}, 1, 10^{2}$).

Numerical results for the leaky lid-driven cavity, the channel domain, and the colliding flow problems are, respectively, presented in Tables 2, 3, and 4. In all the

Table 2 Numerical results lid driven cavity problem on $2^r \times 2^r$ grid

		$\alpha = 10^{-4}$		$\alpha = 10^{-2}$		$\alpha = 1$		$\alpha = 10^{2}$	
r	Preconditioner	IT	CPU	IT	CPU	IT	CPU	IT	CPU
	$\mathcal{P}_{HSS}$	4	0.04	5	0.06	13	0.16	106	1.46
4	$\mathcal{P}_{RHSS}$	3	0.02	3	0.02	3	0.02	4	0.04
	$\mathcal{P}_{REHSS}$	3	0.02	3	0.02	3	0.02	3	0.02
	$\mathcal{P}_{HSS}$	8	0.37	9	0.45	144	8.45	†	–
5	$\mathcal{P}_{RHSS}$	5	0.19	5	0.20	5	0.21	9	0.43
	$\mathcal{P}_{REHSS}$	5	0.21	4	0.12	3	0.07	3	0.07
	$\mathcal{P}_{HSS}$	14	3.87	47	14.21	†	–	†	–
6	$\mathcal{P}_{RHSS}$	8	1.97	8	2.03	9	2.29	27	8.10
	$\mathcal{P}_{REHSS}$	11	3.08	3	0.47	3	0.42	3	0.39
	$\mathcal{P}_{HSS}$	38	76.32	†	-	†	–	†	–
7	$\mathcal{P}_{RHSS}$	15	30.85	14	25.95	17	33.46	79	160.91
	$\mathcal{P}_{REHSS}$	9	17.30	3	3.86	3	3.69	3	3.86
	$\mathcal{P}_{HSS}$	115	2064.44	-	‡	–	‡	–	‡
8	$\mathcal{P}_{RHSS}$	37	641.50	28	471.97	38	648.74	–	‡
	$\mathcal{P}_{REHSS}$	5	67.63	3	33.11	3	31.91	3	27.43

Table 3 Numerical results for channel domain problem on $2^r \times 2^r$ grid

		$\alpha = 10^{-4}$		$\alpha = 10^{-2}$		$\alpha = 1$		$\alpha = 10^{2}$	
r	Preconditioner	IT	CPU	IT	CPU	IT	CPU	IT	CPU
	$\mathcal{P}_{HSS}$	5	0.05	6	0.06	7	0.07	17	0.21
4	$\mathcal{P}_{RHSS}$	3	0.02	3	0.03	3	0.02	4	0.03
	$\mathcal{P}_{REHSS}$	3	0.02	3	0.02	3	0.02	3	0.02
	$\mathcal{P}_{HSS}$	9	0.48	10	0.50	13	0.72	47	2.84
5	$\mathcal{P}_{RHSS}$	5	0.22	5	0.22	5	0.20	9	0.47
	$\mathcal{P}_{REHSS}$	5	0.21	3	0.09	3	0.07	3	0.06
	$\mathcal{P}_{HSS}$	21	5.94	13	3.46	28	8.39	498	154.59
6	$\mathcal{P}_{RHSS}$	8	2.08	8	2.06	8	1.96	21	6.15
	$\mathcal{P}_{REHSS}$	6	1.45	3	0.40	3	0.40	3	0.38
	$\mathcal{P}_{HSS}$	48	96.57	15	28.06	83	169.45	320	1089.81
7	$\mathcal{P}_{RHSS}$	17	32.79	16	30.53	15	28.83	86	178.10
	$\mathcal{P}_{REHSS}$	5	7.45	3	2.84	3	2.65	3	2.52
	$\mathcal{P}_{HSS}$	169	3093.67	20	336.76	–	‡	–	‡
8	$\mathcal{P}_{RHSS}$	42	758.74	37	654.38	34	603.85	–	‡
	$\mathcal{P}_{REHSS}$	4	40.52	3	24.29	3	22.11	2	16.03

Table 4 Numerical results for the colliding flow problem on $2^r \times 2^r$ grid

		$\alpha = 10^{-4}$		$\alpha = 10^{-2}$		$\alpha = 1$		$\alpha = 10^{2}$	
r	Preconditioner	IT	CPU	IT	CPU	IT	CPU	IT	CPU
	$\mathcal{P}_{HSS}$	4	0.04	5	0.05	13	0.16	90	1.17
4	$\mathcal{P}_{RHSS}$	3	0.02	3	0.02	3	0.02	4	0.03
	$\mathcal{P}_{REHSS}$	3	0.02	3	0.02	3	0.02	3	0.02
	$\mathcal{P}_{HSS}$	8	0.37	9	0.43	115	7.10	†	–
5	$\mathcal{P}_{RHSS}$	5	0.20	5	0.22	5	0.19	9	0.44
	$\mathcal{P}_{REHSS}$	5	0.21	4	0.13	3	0.09	3	0.08
	$\mathcal{P}_{HSS}$	14	3.91	53	16.24	†	–	†	–
6	$\mathcal{P}_{RHSS}$	8	2.01	8	2.05	8	2.21	28	8.55
	$\mathcal{P}_{REHSS}$	11	3.15	3	0.48	3	0.43	3	0.4
	$\mathcal{P}_{HSS}$	39	87.20	†	–	†	–	†	–
7	$\mathcal{P}_{RHSS}$	15	29.54	14	27.68	17	35.07	111	250.86
	$\mathcal{P}_{REHSS}$	9	16.61	3	4.11	3	3.81	3	4.4
	$\mathcal{P}_{HSS}$	123	2222.41	–	‡	–	‡	–	‡
8	$\mathcal{P}_{RHSS}$	39	700.77	31	548.88	38	690.25	–	‡
	$\mathcal{P}_{REHSS}$	5	70.94	3	35.25	3	33.79	3	27.73

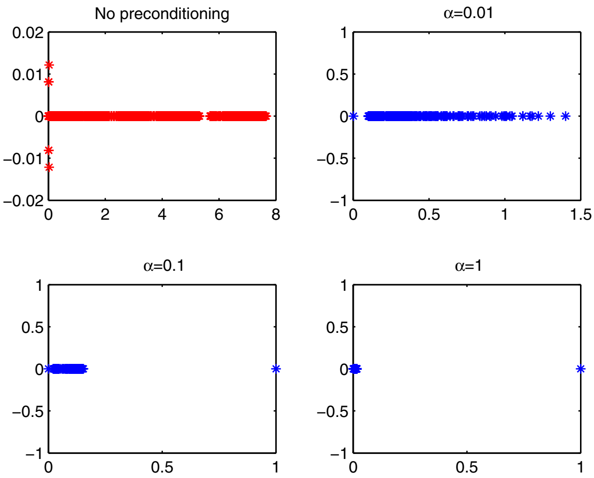

Fig. 1 Eigenvalues distribution of the matrices $\mathcal{A}$ and $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$ for the cavity problem on 32×32 grid with different values of α ($\alpha = 0.01, 0.1, 1$)

tables "IT" stands for the number of restarts in the GMRES(30) algorithm and "CPU" denotes the elapsed CPU time for the convergence. As the numerical results show almost for all the three test problems, the preconditioner $\mathcal{P}_{REHSS}$ is more effective than the preconditioners $\mathcal{P}_{HSS}$ and $\mathcal{P}_{RHSS}$ in terms of the iteration counts and CPU time. The exceptions are the test problems with $\alpha = 10^{-4}$ and $r = 5, 6$ (see Tables 2, 3, and 4) where the results of the P_{RHSS} preconditioner are slightly better than those of the P_{REHSS} preconditioner. As we see, the GMRES(30) method for the preconditioned system with preconditioner $\mathcal{P}_{REHSS}$ always converges, whereas it does not

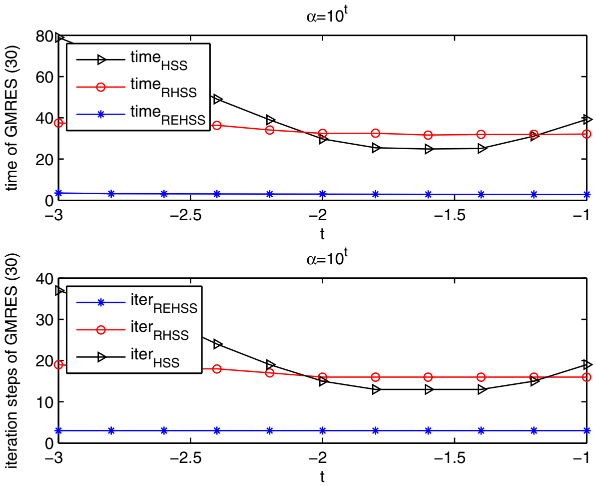

Fig. 2 Number of iterations and CPU time with respect to $t = \log_{10} \alpha$ for the channel domain problem with 128×128 grid

converge for other two preconditioners. Another observation which can be posed here is that, despite preconditioners $\mathcal{P}_{HSS}$ and $\mathcal{P}_{RHSS}$, the behavior of the preconditioned iteration corresponding to the preconditioner $\mathcal{P}_{REHSS}$ is not very sensitive to the choice of α.

In Fig. 1, the eigenvalues distribution of the matrices $\mathcal{A}$ and the preconditioned matrix $\mathcal{P}_{REHSS}^{-1}\mathcal{A}$ for the cavity problem on 32×32 grid, with different values of α ($\alpha = 0.1$, $\alpha = 1$, and $\alpha = 10$) are displayed. We see that the eigenvalues of preconditioned matrices are well-clustered.

In Fig. 2, the number of iterations and the CPU time of GMRES(30) for solving the preconditioned system with the preconditioners $\mathcal{P}_{REHSS}$, $\mathcal{P}_{HSS}$, and $\mathcal{P}_{RHSS}$ for the channel domain problem with 128×128 grid for different values of α are presented. As we see, for this example, the $\mathcal{P}_{REHSS}$ is superior to the preconditioners $\mathcal{P}_{HSS}$ and $\mathcal{P}_{RHSS}$, in terms of the iterations count and the CPU time.

6 Concluding remarks

We have presented a new relaxed version of the Hermitian and skew-Hermitian splitting preconditioner say REHSS for the saddle point problem (1.1). Some properties of the preconditioner have been presented. From numerical point of view, the proposed preconditioner has been compared with two recently proposed preconditioners HSS and RHSS. Numerical results showed that the REHSS preconditioner is in general superior to the HSS and RHSS preconditioners. Moreover, the REHSS preconditioner is not very sensitive to the involving parameter.

Acknowledgments This work is supported by the Iran National Science Foundation (INSF) under Grant No. 93050251. The work is also partially supported by University of Guilan. The authors are grateful to the anonymous reviewers for their useful comments.

References

1. Bai, Z.-Z.: A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations. Adv Comput. Math **10**, 169–186 (1999)
2. Bai, Z.-Z.: Modified block SSOR preconditioners for symmetric positive definite linear systems. Ann. Oper. Res **103**, 263–282 (2001)
3. Bai, Z.-Z.: Structured preconditioners for nonsingular matrices of block two-by-two structures. Math. Comput **75**, 791–815 (2006)
4. Bai, Z.-Z., Golub, G.H.: Accelerated and Hermitian 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.: Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. Math. Comput. **76**, 287–298 (2007)
6. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput **26**, 844–863 (2004)
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 skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math **98**, 1–32 (2004)

9. Bai, Z.-Z., Ng, M.K., Wang, Z.-Q.: Constraint preconditioners for symmetric indefinite matrices. SIAM J. Matrix Anal. Appl **31**, 410–433 (2009)
10. Bai, Z.-Z., Parlett, B.N., Wang, Z.-Q.: On generalized successive overrelaxation methods for augmented linear systems. Numer. Math **102**, 1–38 (2005)
11. Bai, Z.-Z., Wang, Z.-Q.: On parametrized inexact Uzawa methods for generalized saddle point problems. Linear Algebra Appl **428**, 2900–2932 (2008)
12. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 418–477 (2002)
13. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 20–41 (2004)
14. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer **14**, 1–137 (2005)
15. Benzi, M., Guo, X.-P.: A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations. Appl. Numer. Math **61**, 66–76 (2011)
16. Benzi, M., Ng, M.K., Niu, Q., Wang, Z.: A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations. J. Comput. Phys **230**, 6185–6202 (2011)
17. Cao, Y., Yao, L.-Q., Jiang, M.-Q.: A modified dimensional split preconditioner for generalized saddle point problems. J. Comput. Appl. Math **250**, 70–82 (2013)
18. Cao, Z.-H.: Positive stable block triangular preconditioners for symmetric saddle point problems. Appl. Numer. Math **57**, 899–910 (2007)
19. Cao, Y., Yao, L.-Q., Jiang, M.-Q., Niu, Q.: A relaxed HSS preconditioner for saddle point problems from mesh free discretization. J. Comput. Math **31**, 398–421 (2013)
20. Elman, H.C., Ramage, A., Silvester, D.J.: IFISS: a Matlab toolbox for modelling incompressible flow. ACM Trans. Math. Software **33**, Article 14 (2007)
21. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite elements and fast iterative solvers, oxford university press oxford (2003)
22. Fan, H.-T., Zheng, B., Zhu, X.-Y.: A relaxed positive semi-definite and skew-Hermitian splitting preconditioner for non-Hermitian generalied saddle point problems. Numer. Algor. doi:10.1007/s11075-015-0068-5. In press
23. Jiang, M.-Q., Cao, Y., Yao, L.-Q.: On parametrized block triangular preconditioners for generalized saddle point problems. Appl. Math. Comput **216**, 1777–1789 (2010)
24. Golub, G.H., Wu, X., Yuan, J.-Y.: SOR-Like methods for augmented systems. BIT **55**, 71–85 (2001)
25. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indenite linear systems. SIAM J. Matrix Anal. Appl **21**, 1300–1317 (2000)
26. Murphy, M.F., Golub, G.H., Wathen, A.J.: A note on preconditioning for indenite linear systems. SIAM J. Sci. Comput **21**, 1969–1972 (2000)
27. Pan, J.-Y., Ng, M.K., Bai, Z.-Z.: New preconditioners for saddle point problems. Appl. Math. Comput **172**, 762–771 (2006)
28. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
29. Simoncini, V.: Block triangular preconditioners for symmetric saddle-point problems. Appl. Numer. Math **49**, 63–80 (2004)
30. Sturler, E.D., Liesen, J.: Block-diagonal and constraint preconditioners for nonsymmetric indenite linear systems. SIAM J. Sci. Comput **26**, 1598–619 (2005)
31. Wu, X.-N., Golub, G.H., Cuminato, J.A., Yuan, J.-Y.: Symmetric-triangular decomposition and its applications Part II: preconditioners for inde?nite systems. BIT **48**, 139–162 (2008)
32. Xie, Y.-J., Ma, C.-F.: A modified positive-definite and skew-Hermitian splitting preconditioner for generalized saddle point problems from the Navier-Stokes equation. Numer. Algor. doi:10.1007/s11075-015-0043-1. In press