Accepted Manuscript

On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems

Fatemeh Panjeh Ali Beik, Michele Benzi, Sayyed–Hasan Azizi Chaparpordi

PII: S0377-0427(17)30245-5
DOI: http://dx.doi.org/10.1016/j.cam.2017.05.009
Reference: CAM 11140

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 8 September 2016
Revised date : 7 April 2017

Please cite this article as: F.P.A. Beik, et al., On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems, *Journal of Computational and Applied Mathematics* (2017), http://dx.doi.org/10.1016/j.cam.2017.05.009

On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems

Fatemeh Panjeh Ali Beik[†], Michele Benzi[‡] and Sayyed–Hasan Azizi Chaparpordi[†]

[†] *Department of Mathematics, Vali-e-Asr University of Rafsanjan, PO Box 518, Rafsanjan, Iran*

[‡] *Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA*

E-mails: f.beik@vru.ac.ir, benzi@mathcs.emory.edu, m.azizi17@gmail.com

Abstract. We review the use of block diagonal and block lower/upper triangular splittings for constructing iterative methods and preconditioners for solving stabilized saddle point problems. We introduce new variants of these splittings and obtain new results on the convergence of the associated stationary iterations and new bounds on the eigenvalues of the corresponding preconditioned matrices. We further consider inexact versions as preconditioners for flexible Krylov subspace methods, and show experimentally that our techniques can be highly effective for solving linear systems of saddle point type arising from stabilized finite element discretizations of two model problems, one from incompressible fluid mechanics and the other from magnetostatics.

Keywords: iterative methods; convergence; saddle point problems; preconditioning.

AMS Subject Classification: 65F10.

1 Introduction

Consider a stabilized saddle point problem of the form

$$\mathcal{A}u \equiv \begin{pmatrix} A & B \\ -B^T & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ -q \end{pmatrix} \equiv c, \tag{1}$$

where $A \in \mathbb{R}^{m\times m}$ is symmetric positive definite, $B \in \mathbb{R}^{m\times n}$ is of full column rank and C is symmetric positive semidefinite. We are especially interested in the case $C \neq 0$, although our theory also applies to the (non-stabilized) case where $C = 0$. Under these assumptions, (1) admits a unique solution; see, e.g., [4, Lemma 1.1].

Large linear systems of the form (1) arise in several areas of computational science and engineering, including computational fluid dynamics, constrained optimization, regularized least-squares problems, geomechanics, electromagnetics, and so on; see, e.g., [5, 6, 16, 17] and references therein for further details.

Over the years, a number of methods have been proposed in the literature for solving (1): we refer the reader to the survey [5] for developments up to about 2005. More recently, several authors have studied different variants of the classical Uzawa method, mostly for the case $C = 0$; see, for instance [27, 29, 31] and references therein. Moreover, several new preconditioners for Krylov subspace methods have been introduced in recent years, For example, Cao et al. [9] have studied the performance of a preconditioner obtained based on shift-splitting of the saddle point coefficient matrix. Then, Chen and Ma [12] have proposed a general class of preconditioners which incorporates as a special case the preconditioner given in [9]. We mention that the results in [9, 12] have been derived for the case that

the $(2,2)$-block C is zero and the $(1,1)$-block is symmetric positive definite. For a unified treatment of algebraic preconditioners for matrices with a two-by-two block structure, we refer to [1]. We further mention preconditioners for saddle point problems based on the Hermitian/skew-Hermitian splitting; see, e.g., [2, 3, 4]. Recently, Salkuyeh et al. [25] have proposed a modified generalized shift-splitting (MGSS) preconditioner for solving (1) where the $(1,1)$-block is symmetric positive definite and $C \neq 0$. The MGSS preconditioner is obtained based on a splitting of the saddle point matrix which results in an unconditionally convergent stationary iterative method. In addition, a relaxed version of the MGSS preconditioner has been examined.

Despite this activity, techniques based on block diagonal and block triangular splittings and preconditioners remain among the most popular and effective means of solving problems of the form (1), particularly in the field of incompressible fluid mechanics [15]. It should be mentioned that block triangular methods, Uzawa-type methods [8, 13] and the SOR-type method studied by Golub et al. [18] are all closely related. Some of these methods, however, are restricted to the case where $C = 0$.

In this paper we revisit the use of block diagonal and block triangular splittings and preconditioners for problems of the form (1), with special attention to the stabilized case $C \neq 0$. We introduce new variants of these splittings and establisj new results on the convergence of the associated stationary iterations and on the eigenvalues of the corresponding preconditioned matrices. We further consider inexact versions as preconditioners for flexible Krylov subspace methods, and show experimentally that our techniques can be highly effective for solving saddle point linear systems of the form (1) arising from stabilized finite element discretizations of two model problems, one from fluid mechanics and the other from magnetostatics.

Before ending this section, we briefly recall some definitions and properties which will be useful for deriving our main results.

For a given matrix $A \in \mathbb{R}^{n\times n}$, the additive decomposition $A = M - N$ is called a *splitting* if $M, N \in \mathbb{R}^{n\times n}$ and M is nonsingular. For an arbitrary given splitting $A = M - N$, a basic stationary iterative method for solving $Ax = b$ has the following form:

$$x^{(k+1)} = \mathcal{G}x^{(k)} + M^{-1}b, \quad k = 0, 1, 2, \ldots, \tag{2}$$

where $x^{(0)}$ is given and $\mathcal{G} = M^{-1}N$ is called the iteration matrix. It is well-known that the iterative method (2) is convergent for any initial guess $x^{(0)}$ if and only if $\rho(\mathcal{G}) < 1$ where $\rho(\mathcal{G})$ stands for the spectral radius of the iteration matrix $\mathcal{G}$; for further details see [24].

The remainder of this paper is organized as follows. In the next section, the idea of the classical block diagonal (Jacobi) and block triangular (Gauss–Seidel) matrix splittings are utilized to construct three iterative schemes for solving (1). Furthermore, we show that the corresponding iterative methods are convergent under certain (mild) conditions to the solution of (1). In Section 3, we briefly discuss how the proposed iterative methods can be exploited to obtain preconditioners to speed up the convergence of Krylov subspace methods such as the Generalized Minimum Residual (GMRES) method [24]. In Section 4, we report some numerical results for two different model problems to illustrate the efficiency of the proposed preconditioners. Finally, we briefly state our conclusions in Section 5.

Notation. Throughout the paper, for a given matrix A with real eigenvalues, the notations $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of A, respectively. Moreover, the set of all eigenvalues of A is represented by $\sigma(A)$. For an $m \times n$ matrix B, the null space (or kernel) of B is denoted by $\mathrm{Ker}(B)$. For x a complex vector, the conjugate transpose of x is denoted by x^*. For two arbitrary complex vectors x and y of the same dimension, the inner product $\langle x, y\rangle$ is defined by $\langle x, y\rangle = y^*x$. For given vectors x and y of dimensions m and n, $(x; y)$ will denote a column vector of dimension $m + n$.

2 The GJ and GGS-type methods for saddle point problems

The present section consists of two main parts. In Subsection 2.1, the Generalized Jacobi (GJ) block diagonal iterative method for solving (1) is introduced and its convergence properties are discussed. The second subsection is concerned with the Backward and Forward Generalized Gauss–Seidel (BGGS and FGGS) block triangular iterative schemes and their convergence properties.

Let us first consider the following splittings of the coefficient matrix $\mathcal{A}$ in (1):

$$\mathcal{A} = \mathcal{D} - \mathcal{L} - \mathcal{U} = \mathcal{D} - \tilde{\mathcal{L}} - \tilde{\mathcal{U}}, \tag{3}$$

where

$$\mathcal{D} = \begin{pmatrix} A & 0 \\ 0 & M \end{pmatrix}, \quad \mathcal{L} = \begin{pmatrix} 0 & 0 \\ B^T & 0 \end{pmatrix}, \quad \mathcal{U} = \begin{pmatrix} 0 & -B \\ 0 & N \end{pmatrix},$$
$$\tilde{\mathcal{L}} = \begin{pmatrix} 0 & 0 \\ B^T & N \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{U}} = \begin{pmatrix} 0 & -B \\ 0 & 0 \end{pmatrix}.$$

in which $M, N \in \mathbb{R}^{n\times n}$ are given symmetric positive definite and symmetric matrices, respectively, such that $C = M - N$. The underlying assumption is that A is relatively easy to invert (for example, A could be a mass matrix), and so is M (recall that C is generally singular). In the sequel, the kth approximate solution of (1) is denoted by $z^{(k)} = (x^{(k)}; y^{(k)})$.

2.1 The GJ iterative method and its convergence analysis

By extending the idea of the well-known block Jacobi method, the following iterative scheme can be constructed:

$$z^{(k+1)} = \mathcal{G} z^{(k)} + f, \quad k = 0, 1, 2, \ldots, \tag{4}$$

where $z^{(0)} \in \mathbb{R}^{n\times n}$ is arbitrary and

$$\mathcal{G} = \mathcal{D}^{-1}(\mathcal{L} + \mathcal{U}) = \begin{pmatrix} 0 & -A^{-1}B \\ M^{-1}B^T & M^{-1}N \end{pmatrix}, \quad f = \begin{pmatrix} A^{-1}b \\ -M^{-1}q \end{pmatrix}.$$

We observe that $\mathcal{D}^{-1}(\mathcal{L} + \mathcal{U}) = \mathcal{D}^{-1}(\tilde{\mathcal{L}} + \tilde{\mathcal{U}})$. Straightforward computations show that the iterative scheme (4) can be recast as follows:

$$\begin{cases} x^{(k+1)} = A^{-1}(b - By^{(k)}), \\ y^{(k+1)} = M^{-1}Ny^{(k)} + M^{-1}(B^T x^{(k)} - q), \end{cases} \tag{5}$$

$(k = 0, 1, 2, \ldots)$. Next, we recall a lemma which has a key role for analyzing the convergence properties of the iterative scheme (5).

Lemma 2.1. *[28, Section 6.2] Consider the quadratic equation $x^2 - bx + c = 0$, where b and c are real numbers. Both roots of the equation are less than one in modulus if and only if $|c| < 1$ and $|b| < 1 + c$.*

The following theorem supplies sufficient conditions under which the iterative scheme (5) converges to the unique solution of (1).

Theorem 2.2. *Let A and C be symmetric positive definite and semidefinite matrices, respectively, and let B be a full column rank matrix. Suppose that $C = M - N$ where M and N are respectively symmetric positive definite and symmetric matrices. Assume that M and N satisfy the following conditions:*

$$\lambda_{\max}(B^T A^{-1} B) < \lambda_{\min}(M), \tag{6}$$

and

$$\rho(N) < \lambda_{\min}(M) + \lambda_{\min}(B^T A^{-1} B). \tag{7}$$

Then the iterative method (5) converges to the unique solution of (1) for any initial guess.

Proof. Let λ be an arbitrary eigenvalue of $\mathcal{G} = \mathcal{D}^{-1}(\mathcal{L}+\mathcal{U})$ and $(x;y)$ be its corresponding eigenvector. Consequently, we have $(\mathcal{L}+\mathcal{U})(x;y) = \lambda\mathcal{D}(x;y)$, which is equivalent to say that

$$-By = \lambda Ax, \tag{8}$$
$$B^T x + Ny = \lambda My. \tag{9}$$

If $\lambda = 0$, then $|\lambda| < 1$. Therefore, we only need to consider the case that $\lambda \neq 0$. Notice that $y = 0$ is equivalent to $Ax = 0$ which implies $x = 0$ and it is contrary to our assumption that $(x;y)$ is an eigenvector. Hence, we can suppose that $y \neq 0$ and in the sequel, without loss of generality, we assume that $\|y\|_2 = 1$. From (8), it can be seen that

$$x = -\frac{1}{\lambda}A^{-1}By,$$

and substituting the above relation into (9), we get

$$-B^T A^{-1}By + \lambda Ny = \lambda^2 My.$$

Multiplying both sides of the preceding relation on the left by y^*, we obtain

$$-y^*B^T A^{-1}By + \lambda y^*Ny = \lambda^2 y^*My. \tag{10}$$

Next, we set $a = y^*My$, $b = y^*Ny$, and $c = y^*B^T A^{-1}By$. Evidently, a and c are positive constants. In view of (10), λ satisfies the following quadratic equation:

$$\lambda^2 - \frac{b}{a}\lambda + \frac{c}{a} = 0.$$

Notice that $|b| \leq \|N\|_2 = \rho(N)$ which together with (7) implies that $|b| < \lambda_{\min}(M) + \lambda_{\min}(B^T A^{-1}B)$. The result follows from Lemma 2.1, using the assumptions and the fact that

$$0 < \lambda_{\min}(M) \leq a \leq \lambda_{\max}(M) \quad \text{and} \quad 0 < \lambda_{\min}(B^T A^{-1}B) \leq c \leq \lambda_{\max}(B^T A^{-1}B).$$

□

Remark 2.3. We note that the given sufficient condition (7) can be replaced by the more stringent condition $\rho(N) \leq \lambda_{\min}(M)$, which may be easier to check in practice.

Remark 2.4. In view of Eqs. (8) and (9), it can be observed that if $\lambda = 0$ is an eigenvalue of $\mathcal{G}$ then its corresponding eigenvector is of the form $(x;0)$, where $0 \neq x \in \mathrm{Ker}(B^T)$.

It is immediate to derive the following corollary from the proof of the previous theorem.

Corollary 2.5. *Under the same assumptions of Theorem 2.2, let λ be an arbitrary nonzero eigenvalue of the iteration matrix $\mathcal{G}$ and $(x;y)$ its corresponding eigenvector such that $\|y\|_2 = 1$.*

1. *If $\langle Ny, y\rangle^2 \geq 4\langle My, y\rangle \langle B^T A^{-1}By, y\rangle$, then λ is real.*
2. *If $\langle Ny, y\rangle^2 < 4\langle My, y\rangle \langle B^T A^{-1}By, y\rangle$, then λ has non-zero imaginary part.*

Evidently, the inequalities in the above two statements can be respectively restricted as follows:

1. *If $\langle Ny, y\rangle^2 \geq 4\lambda_{\max}(M)\lambda_{\max}(B^T A^{-1}B)$, then λ is real.*
2. *If $\langle Ny, y\rangle^2 < 4\lambda_{\min}(M)\lambda_{\min}(B^T A^{-1}B)$, then λ is complex (i.e., not real).*

As discussed earlier, if (6) and (7) are satisfied, then the iterative scheme (5) converges to the unique solution of (1). In the following remark, we mention three possible choices for the splitting $C = M - N$ and rewrite the sufficient conditions for the convergence of (5) accordingly.

Remark 2.6. Suppose that α is a given positive constant. Let us consider the following three possible instances for choosing the splitting $C = M - N$:

- Case I. If $M = \alpha I + C$ and $N = \alpha I$, then (7) is satisfied immediately. The relation (6) can be rewritten as follows:
$$\alpha > \lambda_{\max}(B^T A^{-1} B) - \lambda_{\min}(C).$$
Hence if we choose the parameter α so that the above inequality holds, then the iterative method (5) is convergent. Notice that since $\lambda_{\min}(C) \geq 0$, the preceding condition in this case can be replaced by the more stringent one $\alpha > \lambda_{\max}(B^T A^{-1} B)$.

- Case II. Suppose $M = \alpha I$ and $N = \alpha I - C$. Clearly, (6) is equivalent to $\alpha > \lambda_{\max}(B^T A^{-1} B)$. Also, condition (7) can be restated as
$$\alpha + \lambda_{\min}(B^T A^{-1} B) > \max\left\{|\alpha - \lambda_{\max}(C)|, |\alpha - \lambda_{\min}(C)|\right\}.$$
With easy manipulations it can be seen that this condition reduces to
$$\alpha > \frac{1}{2}\left(\lambda_{\max}(C) - \lambda_{\min}(B^T A^{-1} B)\right).$$
Consequently, conditions (6) and (7) both hold as soon as the parameter α satisfies the following inequality:
$$\alpha > \max\left\{\lambda_{\max}(B^T A^{-1} B), \frac{1}{2}\left(\lambda_{\max}(C) - \lambda_{\min}(B^T A^{-1} B)\right)\right\}.$$
In view of the fact that $B^T A^{-1} B$ is positive definite, we may consider instead the following more strict condition instead of the preceding one:
$$\alpha > \max\left\{\lambda_{\max}(B^T A^{-1} B), \frac{1}{2}\lambda_{\max}(C)\right\}.$$

- Case III. If $M = \frac{1}{2}(\alpha I + C)$ and $N = \frac{1}{2}(\alpha I - C)$ then conditions (6) and (7) are equivalent to the following inequality:
$$\alpha > \max\left\{2\lambda_{\max}(B^T A^{-1} B) - \lambda_{\min}(C), \frac{1}{2}\left(\lambda_{\max}(C) - \lambda_{\min}(C)\right) - \lambda_{\min}(B^T A^{-1} B)\right\}.$$
From the positive definiteness of $B^T A^{-1} B$ and positive semi-definiteness of C, it follows that the above inequality is certainly satisfied when
$$\alpha > \max\left\{2\lambda_{\max}(B^T A^{-1} B), \frac{1}{2}\lambda_{\max}(C)\right\}.$$

2.2 The BGGS and FGSS iterative methods and their convergence analysis

Before presenting the BGGS and FGGS iterative methods, we give a brief overview of the Uzawa and generalized parameterized inexact Uzawa (GPIU) methods so as to emphasize the differences between the Uzawa-type methods and our proposed two GGS-type methods.

The classical Uzawa method produces the approximate solutions of (1) with the following iterative scheme:

$$\begin{cases} x^{(k+1)} = A^{-1}(b - By^{(k)}), \\ y^{(k+1)} = y^{(k)} + \alpha(B^T x^{(k+1)} - Cy^{(k)} - q), \end{cases} \tag{11}$$

$(k = 0, 1, 2, \ldots)$. Here $\alpha > 0$ is a prescribed parameter. We refer to [13] and the references therein for further details.

In [31], Zhang et al. have proposed the GPIU method for solving a class of two-by-two linear systems which includes (1). More precisely, first, the splitting $\mathcal{A} = \tilde{\mathcal{M}} - \tilde{\mathcal{N}}$ is considered for the coefficient matrix of (1), where

$$\tilde{\mathcal{M}} = \begin{pmatrix} A + P_1 & 0 \\ P_2 & C + P_3 \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{N}} = \begin{pmatrix} P_1 & -B \\ P_2 + B^T & P_3 \end{pmatrix}.$$

Then the GPIU method for solving (1) is given by

$$\begin{cases} x^{(k+1)} &= x^{(k)} + (A + P_1)^{-1}(b - Ax^{(k)} - By^{(k)}), \\ y^{(k+1)} &= y^{(k)} + (C + P_3)^{-1}(-P_2x^{(k+1)} + (P_2 + B^T)x^{(k)} - Cy^{(k)} - q), \end{cases}$$

$(k = 0, 1, 2, \ldots)$, where the P_i are given matrices for $i = 1, 2, 3$, with P_1 and P_2 assumed to be symmetric positive definite. When $P_1 = 0$, then the subsystem $Ax^{(k+1)} = b - Ay^{(k)}$ needs to be solved at each step, similar to the Uzawa method. For the case that the $(2, 2)$-block C is nonzero, the convergence properties of the method and the eigenvalue problem corresponding to the preconditioned matrix $\tilde{\mathcal{M}}^{-1}\mathcal{A}$ have been investigated only for the case that $P_3 = \frac{1}{\delta}C$ where $\delta > 0$ and C is a symmetric positive definite matrix; see [31, Theorems 3.2 and 3.3]. We stress here that in the convergence analysis of the BGGS and FGGS methods, we do not set require C to be nonsingular.

On the basis of (3) and following the idea of the block Gauss–Seidel method [24], the following backward and forward iterative schemes can be constructed to solve (1), respectively.

- The Backward Generalized Gauss–Seidel ($BGGS$) iterative scheme is given by

$$z^{(k+1)} = \tilde{\mathcal{H}}z^{(k)} + \tilde{f}, \qquad k = 0, 1, 2, \ldots, \tag{12}$$

where

$$\tilde{\mathcal{H}} = (\mathcal{D} - \tilde{\mathcal{U}})^{-1}\tilde{\mathcal{L}} = \begin{pmatrix} -A^{-1}BM^{-1}B^T & -A^{-1}BM^{-1}N \\ M^{-1}B^T & M^{-1}N \end{pmatrix}, \tag{13}$$

and

$$\tilde{f} = (\mathcal{D} - \tilde{\mathcal{U}})^{-1}c = \begin{pmatrix} A^{-1}b + A^{-1}BM^{-1}q \\ -M^{-1}q \end{pmatrix}.$$

- The Forward Generalized Gauss–Seidel ($FGGS$) iterative scheme is given by

$$z^{(k+1)} = \mathcal{H}z^{(k)} + f, \qquad k = 0, 1, 2, \ldots, \tag{14}$$

where

$$\mathcal{H} = (\mathcal{D} - \mathcal{L})^{-1}\mathcal{U} = \begin{pmatrix} 0 & -A^{-1}B \\ 0 & -M^{-1}B^TA^{-1}B + M^{-1}N \end{pmatrix}, \tag{15}$$

and

$$f = (\mathcal{D} - \mathcal{L})^{-1}c = \begin{pmatrix} A^{-1}b \\ M^{-1}B^TA^{-1}b - M^{-1}q \end{pmatrix}.$$

Letting $z^{(k)} = (x^{(k)}; y^{(k)})$, where $x^{(0)}$ and $y^{(0)}$ are given, it is not difficult to see that (12) and (14) can be respectively reformulated as follows:

$$\begin{cases} y^{(k+1)} = M^{-1}Ny^{(k)} + M^{-1}(B^Tx^{(k)} - q), \\ x^{(k+1)} = A^{-1}(b - By^{(k+1)}), \quad k = 0, 1, 2, \ldots, \end{cases} \tag{16}$$

and

$$\begin{cases} x^{(k+1)} = A^{-1}(b - By^{(k)}), \\ y^{(k+1)} = M^{-1}Ny^{(k)} + M^{-1}(B^Tx^{(k+1)} - q), \quad k = 0, 1, 2, \ldots. \end{cases} \tag{17}$$

Remark 2.7. The FGSS iterative scheme (17) reduces to the Uzawa iterative method given by (11) when

$$M = \frac{1}{\alpha} I \quad \text{and} \quad N = \frac{1}{\alpha} I - C,$$

where $\alpha > 0$.

Remark 2.8. Let $\bar{\mathcal{H}} = \tilde{\mathcal{L}}(\mathcal{D} - \tilde{\mathcal{U}})^{-1}$. It is straightforward to see that $\bar{\mathcal{H}}$ is similar to the matrix $\tilde{\mathcal{H}}$ in (12), hence $\sigma(\tilde{\mathcal{H}}) = \sigma(\bar{\mathcal{H}})$. Therefore, in order to study the convergence of the BGGS scheme, we can analyze the spectrum of

$$\bar{\mathcal{H}} = \begin{pmatrix} 0 & 0 \\ B^T A^{-1} & -B^T A^{-1} B M^{-1} + N M^{-1} \end{pmatrix}. \tag{18}$$

Now we recall the following useful lemmas, which are crucial for obtaining bounds on the eigenvalues of both $\bar{\mathcal{H}}$ (hence, $\tilde{\mathcal{H}}$) and $\mathcal{H}$. The first one is an immediate consequence of Weyl's Theorem, see [19, Theorem 4.3.1].

Lemma 2.9. *Suppose that A and B are two Hermitian matrices. Then,*

$$\begin{aligned} \lambda_{\max}(A+B) &\leq \lambda_{\max}(A) + \lambda_{\max}(B), \\ \lambda_{\min}(A+B) &\geq \lambda_{\min}(A) + \lambda_{\min}(B). \end{aligned}$$

Lemma 2.10. *[30] Suppose that A is a Hermitian negative definite matrix and B is Hermitian positive semidefinite. Then the eigenvalues of AB are real and satisfy*

$$\lambda_{\min}(A)\lambda_{\min}(B) \leq \lambda_{\max}(AB) \leq \lambda_{\max}(A)\lambda_{\min}(B),$$

$$\lambda_{\min}(A)\lambda_{\max}(B) \leq \lambda_{\min}(AB) \leq \lambda_{\max}(A)\lambda_{\max}(B).$$

Remark 2.11. From (15) and (18), we observe that at least m eigenvalues of $\mathcal{H}$ and $\bar{\mathcal{H}}$ are zero. If we define the matrices G and $\bar{G}$ as follows:

$$G = -M^{-1} B^T A^{-1} B + M^{-1} N \quad \text{and} \quad \bar{G} = -B^T A^{-1} B M^{-1} + N M^{-1}, \tag{19}$$

then the remaining eigenvalues of $\mathcal{H}$ and $\bar{\mathcal{H}}$ are the eigenvalues of G and $\bar{G}$, respectively. On the other hand, G and $\bar{G}$ are similar and therefore have the same spectrum. Consequently, without loss of generality, we only need to analyze the spectrum of G, and all of the obtained results hold for $\bar{G}$ as well. We further observe that since M is positive definite and $N - B^T A^{-1} B$ symmetric, G has only real eigenvalues. In particular, this holds for all choices of M listed in Remark 2.6.

Remark 2.12. Consider now the case where M and N commute: $MN = NM$. Then $M^{-1}N$ is symmetric, and Lemmas 2.9 and 2.10 imply that

$$\begin{aligned} \lambda_{\max}(G) &\leq -\lambda_{\min}(M^{-1})\lambda_{\min}(B^T A^{-1} B) + \lambda_{\max}(M^{-1}N), \\ \lambda_{\min}(G) &\geq -\lambda_{\max}(M^{-1})\lambda_{\max}(B^T A^{-1} B) + \lambda_{\min}(M^{-1}N). \end{aligned}$$

The above two inequalities provide an interval containing the spectrum of $\mathcal{H}$ which is also valid for $\tilde{\mathcal{H}}$ in view of Remark 2.8 and the earlier discussions of this remark. Of course, we may also consider the following larger intervals, which may be more practical to determine as they only require knowledge of the spectral radius of $M^{-1}N$ and of the largest eigenvalues of M and $B^T A^{-1} B$:

- when $M^{-1}N$ is symmetric,

$$[\lambda_{\min}(\tilde{\mathcal{H}}), \lambda_{\max}(\tilde{\mathcal{H}})] = [\lambda_{\min}(\mathcal{H}), \lambda_{\max}(\mathcal{H})] \subset [\eta_1, \eta_2], \tag{20}$$

  where $\eta_1 = -\lambda_{\max}(M^{-1})\lambda_{\max}(B^T A^{-1} B) - \rho(M^{-1}N)$ and $\eta_2 = \max\{0, \lambda_{\max}(M^{-1}N)\}$.

- when $M^{-1}N$ is symmetric positive definite, we can use a smaller interval than (20) given as follows:

$$[\lambda_{\min}(\tilde{\mathcal{H}}), \lambda_{\max}(\tilde{\mathcal{H}})] = [\lambda_{\min}(\mathcal{H}), \lambda_{\max}(\mathcal{H})] \subset [\xi_1, \xi_2], \tag{21}$$

  where $\xi_1 = -\lambda_{\max}(M^{-1})\lambda_{\max}(B^TA^{-1}B)$ and $\xi_2 = \lambda_{\max}(M^{-1}N)$.

We comment here that if M is chosen as in Cases I, II, or III of Remark 2.6, then $M^{-1}N$ is symmetric. Moreover, $M^{-1}N$ is positive definite in Case I. Note that for these cases, we only need to know the value of $\lambda_{\min}(C)$ and $\lambda_{\max}(B^TA^{-1}B)$. Note that $\xi_2 = \eta_2 = 1$ when C is singular, i.e., $\lambda_{\min}(C) = 0$.

Next, we establish a sufficient condition for the convergence of the BGGS iterative method in the special case when $M = \alpha I + C$ and $N = \alpha I$, which is of particular interest when solving certain problems, e.g., the stabilized discrete Stokes problem.

Theorem 2.13. *Suppose that $M = \alpha I + C$ and $N = \alpha I$ in (16). If*

$$\alpha > \lambda_{\max}\left(B^TA^{-1}B\right),$$

then the iterative scheme (16) is convergent for any initial choices of $x^{(0)}$ and $y^{(0)}$.

Proof. In view of Remark 2.8, we need to show that $\rho(\bar{\mathcal{H}}) < 1$ (or, equivalently, that $\rho(G) < 1$). Let λ be an arbitrary eigenvalue of $\bar{\mathcal{H}}$. Then either $\lambda = 0$, or λ is a nonzero eigenvalue of

$$G = (N - B^TA^{-1}B)M^{-1}. \tag{22}$$

Substituting $N = \alpha I$ in the above relation, we have $G = -G_1M^{-1}$ where $G_1 = B^TA^{-1}B - \alpha I$ is negative definite under our assumption on α. Now, Lemma 2.10 implies that

$$\frac{\alpha - \lambda_{\max}(B^TA^{-1}B)}{\alpha + \lambda_{\max}(C)} \le \lambda_{\min}(G) \le \frac{\alpha - \lambda_{\min}(B^TA^{-1}B)}{\alpha + \lambda_{\max}(C)}, \tag{23}$$

and

$$\frac{\alpha - \lambda_{\max}(B^TA^{-1}B)}{\alpha + \lambda_{\min}(C)} \le \lambda_{\max}(G) \le \frac{\alpha - \lambda_{\min}(B^TA^{-1}B)}{\alpha + \lambda_{\min}(C)}. \tag{24}$$

From the above bounds and in view of the positive definiteness of $B^TA^{-1}B$, we deduce immediately that all the nonzero eigenvalues of G, and therefore of $\bar{\mathcal{H}}$, fall in the open interval $(0, 1)$. This completes the proof. □

Remark 2.14. From Remark 2.8 and discussions of Remark 2.11, we observe that $\sigma(\tilde{\mathcal{H}}) = \sigma(\mathcal{H})$. As a result, under the same assumptions of Theorem 2.13, we conclude that the FGGS iterative method (17) also converges for any initial guess.

Remark 2.15. In view of (19), if we set $M = C + B^TA^{-1}B$ and $N = B^TA^{-1}B$ then the (2,2)-blocks of $\bar{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ become zero. Consequently, all of the eigenvalues of $\bar{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ are zero and both BGSS and FGSS iterative methods reach the exact solution in two steps. Indeed, both methods reduce to a form of block Gaussian elimination and they become in fact direct, rather than iterative solvers. In general, however, both M and N would be dense matrices and therefore this approach is not a practical one for large problems. Nevertheless, these observations suggest that approximations to such choices of M and N can lead to good preconditioners for Krylov subspace methods; see, e.g., [5] or [15] as well as the following section for additional discussion.

3 Block preconditioners for stabilized saddle point matrices

In general terms, given a splitting $\mathcal{A} = \mathcal{M} - \mathcal{N}$, the matrix $\mathcal{M}^{-1}$ can be used as a preconditioner for the linear system of equations (1); see, e.g, [4, 9, 10, 12] and the references therein for further discussion.

In this paper we investigate the performance of three kinds of preconditioners based on the GJ splitting ($\mathcal{A} = \mathcal{M}_{GJ} - \mathcal{N}_{GJ}$), BGGS splitting ($\mathcal{A} = \mathcal{M}_{BGGS} - \mathcal{N}_{BGGS}$) and FGGS splitting ($\mathcal{A} = \mathcal{M}_{FGGS} - \mathcal{N}_{FGGS}$) where

$$\mathcal{M}_{GJ} = \begin{pmatrix} A & 0 \\ 0 & M \end{pmatrix}, \quad \mathcal{M}_{BGGS} = \begin{pmatrix} A & B \\ 0 & M \end{pmatrix}, \quad \text{and} \quad \mathcal{M}_{FGGS} = \begin{pmatrix} A & 0 \\ -B^T & M \end{pmatrix}.$$

We recall that the matrices $M, N \in \mathbb{R}^{n\times n}$ are given symmetric positive definite and symmetric matrices (respectively) such that $C = M - N$.

While the preconditioner $\mathcal{M}_{GJ}$ is an example of a block diagonal preconditioner, $\mathcal{M}_{BGGS}$ and $\mathcal{M}_{FGGS}$ belong to the class of upper and lower block triangular preconditioners, respectively; see [5, 15, 23] and the references therein. In Subsection 3.1 we briefly discuss some previous work on block triangular preconditioners for stabilized saddle point problems. Subsection 3.2 is mainly devoted to a discussion of how the proposed preconditioners can be implemented in practice.

3.1 Block triangular preconditioners for stabilized saddle point problems

Preconditioners with block triangular structure have been shown to be among the most effective for solving problems of saddle point type, and as a result they have been widely investigated in the literature; see, e.g., [11, 15, 20, 23, 26] and the references therein. In the case of the stabilized system (1), the "ideal" block upper and lower triangular preconditioners are

$$\mathcal{P}_U = \begin{pmatrix} A & B \\ 0 & C + B^T A^{-1} B \end{pmatrix} \quad \text{and} \quad \mathcal{P}_L = \begin{pmatrix} A & 0 \\ -B^T & C + B^T A^{-1} B \end{pmatrix},$$

respectively. It is immediate to see that $\sigma(\mathcal{A}\mathcal{P}_U^{-1}) = \sigma(\mathcal{P}_U^{-1}\mathcal{A}) = \{1\}$ and $\sigma(\mathcal{A}\mathcal{P}_L^{-1}) = \sigma(\mathcal{P}_L^{-1}\mathcal{A}) = \{1\}$. Moreover, the minimal polynomial of these matrices has degree two, which implies that a method like GMRES with these preconditioners will reach the exact solution in at most two steps [21]; see also Remark 2.15. Practical versions of these preconditioners rely on suitable approximations of the action of the inverses of the diagonal blocks A and $S = C + B^T A^{-1} B$.

In [20], symmetric linear systems with coefficient matrix

$$\mathcal{A}_t = \begin{pmatrix} A & B \\ B^T & -t^2 C \end{pmatrix}. \tag{25}$$

(with $t \neq 0$) arising from linear elasticity problems are considered. Note that $\mathcal{A}_t(x; y) = (b; q)$ is equivalent to (1) for $t = 1$. In [20] the matrix C is assumed to be symmetric positive definite (hence, nonsingular) and the following preconditioners are analyzed in combination with an appropriate Krylov subspace method:

$$\hat{\mathcal{B}}_U = \begin{pmatrix} \hat{A} & B \\ 0 & -\hat{C} \end{pmatrix} \quad \text{and} \quad \hat{\mathcal{B}}_L = \begin{pmatrix} \hat{A} & 0 \\ B^T & -\hat{C} \end{pmatrix}.$$

Here $\hat{A}$ and $\hat{C}$ are symmetric positive definite and it is assumed that $A - \hat{A}$ is positive definite (see also [7]). Under these assumptions, it is shown in [20] that the eigenvalues of the preconditioned matrix are all real and positive.

In [26], the author studied the solution of the symmetric saddle point problem (25) with $t = 1$ and C assumed to be symmetric positive semidefinite using the preconditioner

$$\mathcal{P} = \begin{pmatrix} \hat{A} & B \\ 0 & -\hat{C} \end{pmatrix}, \tag{26}$$

where $\hat{A}$ and $\hat{C}$ are symmetric positive definite, but without assuming that $A - \hat{A}$ is positive definite. It is shown in [26] that the eigenvalues of $\mathcal{A}^{-1}\mathcal{P}$, while generally complex, have positive real part. For suitable choices of $\hat{A} \approx A$ and $\hat{C} \approx C$ convergence rates independent of mesh discretization parameters can be obtained for saddle point systems arising from certain types of problems arising from the discretization of partial differential equations (PDEs).

For the same class of symmetric saddle point problems, Cao [11] has investigated block triangular preconditioners of the form

$$\begin{pmatrix} \hat{A} & B \\ 0 & \hat{C} \end{pmatrix}, \tag{27}$$

and established that all eigenvalues of the preconditioned matrix are real (though not all of them positive). Numerical experiments in [11] show that the block upper triangular preconditioner (27) works slightly better than the block upper triangular preconditioner (26). We comment that both Cao and Simoncini have taken the matrix $\hat{A}$ to be an incomplete Cholesky factorization of A in their numerical tests. In [11], the matrix $\hat{C}$ was chosen so that $\hat{C} = C + B^T\hat{A}^{-1}B$. In addition to such $\hat{C}$, Simoncini has also experimented with $\hat{C} = C + B^TB$. The reported numerical experiments, in both of these works, illustrate that these choices for $\hat{A}$ and $\hat{C}$ do not provide "optimal" preconditioners, as the number of iterations for solving the preconditioned system grows as the grid is refined. Nevertheless, these sub-optimal solvers often outperformed, in terms of CPU time, optimal (but expensive) solvers based on spectrally equivalent approximations to A and $C + B^TA^{-1}B$.

We end this subsection with three separate remarks and a brief note.

Remark 3.1. It follows from the proof of Theorem 2.13 that for the choice $M = C + \alpha I$, the eigenvalues of the preconditioned saddle point matrix ${\mathcal{M}_{BGGS}}^{-1}\mathcal{A}$ (and thus those of $\mathcal{A}{\mathcal{M}_{BGGS}}^{-1}$) all fall in the interval $(0, 1)$ for $\alpha > \lambda_{\max}(B^TA^{-1}B)$.

Remark 3.2. Considering the preconditioners (26) and (27) for solving the symmetric saddle point problem (25) with $t = 1$, it is seen that these preconditioners only differ in the sign of $(2, 2)$-block. Similarly, we can consider the situation where the $(2, 2)$-block of $\tilde{\mathcal{P}}_U := \mathcal{M}_{BGGS}$ applied to (1) is replaced by a symmetric negative definite matrix instead of a symmetric positive definite one. In other words, we assume that $C = M - N$ where M is a symmetric negative definite matrix. It is obvious that $\sigma(\mathcal{A}\tilde{\mathcal{P}}_U^{-1}) = \sigma(\tilde{\mathcal{P}}_U^{-1}\mathcal{A})$ and that $\sigma(\mathcal{A}\tilde{\mathcal{P}}_U^{-1}) = \sigma(I - \bar{\mathcal{H}})$ where $\bar{\mathcal{H}}$ is defined by (18). Now, it can be verified that $\mathcal{A}\tilde{\mathcal{P}}_U^{-1}$ has m eigenvalues equal to 1 and its remaining eigenvalues are of the form $1 + \mu$ with $\mu \in \sigma(G)$, where $G = (B^TA^{-1}B - N)M^{-1}$. Evidently, M^{-1} and $B^TA^{-1}B - N$ are, respectively, symmetric negative definite and symmetric matrices. Consequently, Lemma 2.10 ensures that all eigenvalues of $\mathcal{A}\tilde{\mathcal{P}}_U^{-1}$ are real. Notice that one may establish bounds for the eigenvalues of $\mathcal{A}\tilde{\mathcal{P}}_U^{-1}$ with the same arguments utilized in Remark 2.11.

Remark 3.3. As we have seen, there are two different types of preconditioners corresponding to $\tilde{\mathcal{P}}_U = \mathcal{M}_{BGGS}$ and $\tilde{\mathcal{P}}_L = \mathcal{M}_{FGGS}$, i.e., block upper and block lower triangular preconditioners, respectively. From Remark 2.14, we conclude that

$$\sigma(\mathcal{A}\tilde{\mathcal{P}}_U^{-1}) = \sigma(\mathcal{A}\tilde{\mathcal{P}}_L^{-1}).$$

We note that for the symmetric saddle point problem (25) with $t = 1$, Pestana has already established that the spectrum of the preconditioned matrix using the block lower triangular preconditioner coincides with that obtained using the corresponding block upper triangular preconditioner, see [23, Proposition 1]. Of course, the results obtained may not be the same when these preconditioners are applied in conjunction with a given Krylov subspace method because in general the convergence behavior does not depend only on the eigenvalues of the preconditioned coefficient matrix. However, as pointed out in [23], in practice the difference between applying block lower and upper triangular preconditioners should not be very significant. We experimentally study the performance of the preconditioners $\mathcal{P}_U$ and $\mathcal{P}_L$ used with the GMRES and flexible GMRES (FGMRES) methods [24] in Section 4.

Finally, we point out that in practice, unless A is trivial to invert (e.g., diagonal), it is more efficient to use our proposed block triangular preconditioners "inexactly", i.e., in the form

$$\mathcal{M}_{BGGS} \approx \begin{pmatrix} \hat{A} & B \\ 0 & M \end{pmatrix} \quad \text{and} \quad \mathcal{M}_{FGGS} \approx \begin{pmatrix} \hat{A} & 0 \\ -B^T & M \end{pmatrix}, \tag{28}$$

where $\hat{A}^{-1}$ consists of a few iterations of preconditioned CG (PCG), where the preconditioner can be an incomplete Cholesky factorization or one or more (algebraic) multigrid V-cycles . In our numerical experiments we also report the corresponding results for the performance of the exact versions of our proposed preconditioners, in order to illustrate the superiority of using inexact versions over the exact ones. We note that for large problems associated with PDEs in three space dimensions, only these inexact variants are feasible in practice.

3.2 Applying the preconditioners

In this section, we briefly discuss how the GJ, BGGS and FGGS splittings can serve as preconditioners for a Krylov subspace method such as GMRES. Consider the preconditioner $\mathcal{P} = \mathcal{M}^{-1}$ corresponding to a given splitting $\mathcal{A} = \mathcal{M} - \mathcal{N}$. To apply the preconditioner $\mathcal{P}$ within a Krylov subspace method, one needs to compute vectors of the form $z = \mathcal{P}r$. That is, we require to solve the linear system of equations $\mathcal{M}z = r$. In the sequel, we assume that $r = (r_1; r_2)$ and $z = (z_1; z_2)$ where $r_1, z_1 \in \mathbb{R}^m$ and $r_2, z_2 \in \mathbb{R}^n$.

For the preconditioners associated with the GJ, BGGS, and FGGS splittings, the vectors $z = \mathcal{M}_{GJ}^{-1}r$, $z = \mathcal{M}_{BGGS}^{-1}r$, and $z = \mathcal{M}_{FGGS}^{-1}r$ must be computed, respectively. Therefore, we need to solve the following linear systems of equations,

$$\begin{pmatrix} A & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \tag{29}$$

$$\begin{pmatrix} A & B \\ 0 & M \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \tag{30}$$

and

$$\begin{pmatrix} A & 0 \\ -B^T & M \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \tag{31}$$

respectively. In view of (29), (30), and (31), we can derive simple algorithms for computing $z = \mathcal{M}_{GJ}^{-1}r$, $z = \mathcal{M}_{BGGS}^{-1}r$, and $z = \mathcal{M}_{FGGS}^{-1}r$, which are respectively given in Algorithms 1, 2, and 3.

Algorithm 1: Computation of $z = \mathcal{M}_{GJ}^{-1}r$.

Solve $Az_1 = r_1$ by the preconditioned CG method for z_1;
Solve $Mz_2 = r_2$ using the Cholesky factorization of M for z_2.

Algorithm 2: Computation of $z = \mathcal{M}_{BGGS}^{-1}r$.

Solve $Mz_2 = r_2$ using the Cholesky factorization of M for z_2;
Solve $Az_1 = r_1 - Bz_2$ by the preconditioned CG method for z_1.

Remark 3.4. We comment that Algorithms 1–3 are called "inexact" when in at least one line of the algorithms, the PCG method is utilized (with early termination). In contrast, the "exact" version

Algorithm 3: Computation of $z = \mathcal{M}_{FGGS}^{-1} r$.

Solve $Az_1 = r_1$ by the preconditioned CG method for z_1;
Solve $Mz_2 = B^T z_1 + r_2$ using the Cholesky factorization of M for z_2.

of the algorithm is obtained by using the Cholesky factorization in both lines of the algorithms. Of course, if a strict convergence tolerance is used to terminate the PCG iteration, the two approaches become equivalent. In practice, however, the PCG iteration will be terminated quite early (see next section for details).

Remark 3.5. We point out that when we use the PCG method to solve each of the subsystems in the preceding algorithms, a flexible Krylov method (like FGMRES [24]) needs to be used instead of the standard GMRES algorithm for the outer iteration, in view the fact that the CG method is a non-stationary iterative method, unless a very stringent inner tolerance is used to terminate PCG (which would be unnecessarily expensive). It is important to note that if the inner tolerance (in the inner PCG method) is chosen higher than the outer tolerance (in the outer preconditioned GMRES method), then the preconditioned GMRES iterations will converge to an inaccurate solution.

Remark 3.6. Obviously, the convergence theory developed for the stationary iterations based on the GJ, FGGS and BGGS splittings does not apply to the inexact variants of these methods, which are only used as preconditioners for FGMRES. In practice, however, the results on the eigenvalues of the iteration (or preconditioned) matrices remain approximately true in the inexact case. In particular, $\mathcal{A}\mathcal{M}_{FGSS}^{-1}$ and $\mathcal{A}\mathcal{M}_{BGSS}^{-1}$ will have at least m eigenvalues clustered near 1.

4 Numerical experiments

In this section, we experimentally study the performance of the FGMRES method with the proposed inexact GJ and GGS-type preconditioners on a stabilized finite element discretization of the 2D Stokes problem and on a test problem arising from 3D magnetostatics. Moreover, the exact versions of these preconditioners are also examined in conjunction with the standard GMRES method. In this case, for solving the subsystems arising in the preconditioning step, we use a sparse Cholesky factorization with the symmetric approximate minimum degree (SYMAMD) reordering.

All of the reported experiments were performed on a 64-bit 2.45 GHz core i7 processor and 8.00GB RAM using MATLAB version 8.3.0532.

In all of the following experiments, the initial guess is taken to be the zero vector and the (outer) iterations are stopped once $\|c - \mathcal{A}z^{(k)}\| < 10^{-6}\|c\|$, where $\|\cdot\|$ stands for the Euclidean norm and $z^{(k)} = (x^{(k)}; y^{(k)})$ denotes the current iterate. In the tables, the entries under "Iter" refer to the total number of required iterations for the GMRES method and its exactly preconditioned version. Under "Iters", we report two values, the number of required iterations for the FGMRES method and, in parenthesis, the total number of inner PCG iterations performed in all the FGMRES steps. Note that no restarting is used. Under "CPU" we report the total CPU-time (in seconds) required for the convergence of the various iterative methods. This includes the time needed to compute all the necessary Cholesky factorizations (complete or incomplete). The notation D_C stands for the diagonal matrix consisting of the diagonal entries of C.

Example 4.1. *Consider the following Stokes problem [15],*

$$\begin{cases} -\Delta \mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \end{cases} \tag{32}$$

in a bounded domain $\Omega \subset \mathbb{R}^2$ with suitable boundary conditions. Here $\mathbf{u}$ and p stand for the velocity vector field and the pressure scalar field, respectively.

Discretization of the Stokes problem (32) *using stabilized finite element methods leads to a saddle point system of the form* (1) *with symmetric positive definite* $(1,1)$*-block and symmetric positive semidefinite* $(2,2)$*-block. Here we use the* MATLAB *package IFISS by Elman et al. [14] to produce discretizations of the classic leaky lid driven cavity problem on the unit square using the stabilized Q1-P0 finite element method. The resulting saddle point problems are used to investigate the performance of the proposed preconditioners for improving the convergence rate of GMRES applied to* (1) *with nonzero* $(2,2)$*-block. We comment here that in all of the examined instances, the default value of the stabilization parameter was used.*

For this example, the $(2,2)$-block C is block diagonal with small blocks, see [15, Section 3.3.2]. Therefore, the Cholesky factorization was used in all the block preconditioners to solve the subsystems corresponding to the $(2,2)$-block. In addition, we used the PCG method in Algorithm 1 (Line 1), Algorithm 2 (Line 2), and Algorithm 3 (Line 3). To this end, an incomplete Cholesky factorization of the (1,1)-block of $\mathcal{A}$ is first computed using the MATLAB function `ichol(.,struct('droptol',1e-3,'michol','on'))`. Then, the incomplete factors are utilized as preconditioner in the inner PCG iterations. These iterations were terminated when the residual norm had been reduced by a factor of 100 or when the maximum number of 40 iterations was reached.

First, in order to check tightness of the bounds ξ_i and η_i $(i = 1, 2)$ discussed in Remark 2.11, we report the corresponding computations for the case of a 32×32 grid in Tables 1–3. The matrices M and N were chosen as in Cases I, II and III in Remark 2.6. Notice that the interval (20) is derived for more general cases than in (21), that is, $M^{-1}N$ can be either definite or indefinite when (20) is used. Therefore, it is expected that the established interval (21) will provide a better approximation for the interval containing the spectrum of $\tilde{\mathcal{H}}$ (or $\mathcal{H}$) and our experimental results confirm this. It can be seen that the values of ξ_2 and η_2 are equal to 1, which follows from the fact that in Example 4.1 the $(2,2)$-block C in (1) is singular. As observed, the presented intervals (20) and (21) provide reasonable approximations for $[\lambda_{min}(\mathcal{H}), \lambda_{\max}(\mathcal{H})] = [\lambda_{min}(\tilde{\mathcal{H}}), \lambda_{\max}(\tilde{\mathcal{H}})]$ and they shrink when the value of α increases. Note that for values of α resulting in convergence, the spectral radius of the iteration matrix is very close to 1, indicating slow convergence of the corresponding stationary iteration. The eigenvalue distributions of the preconditioned matrices corresponding to the BGGS (FGGS) preconditioner for different values of α are depicted in Figure 1. Note the increased clustering of the spectrum for increasing values of α.

Next, we examine the performance of exact and inexact GJ, BGGS and FGGS types preconditioners to improve the rate of convergence of GMRES and FGMRES methods. We generate saddle point problems using a sequence of $2^\ell \times 2^\ell$ uniform grids, with $\ell = 4, 5, 6, 7$. The corresponding sizes for the coefficient matrix $\mathcal{A}$ of (1) are given in Table 4. The eigenvalue distributions of the (exactly) preconditioned matrices for the mentioned preconditioners with $\alpha^* = 1/(2^{\ell-1} \times 2^{\ell-1})$ and $\tilde{\alpha}^* = 1/(2^{\ell-2} \times 2^{\ell-2})$ ($2^\ell \times 2^\ell$ grid) are plotted in Figure 2 where $\ell = 5$.

We recall that a preconditioner is said to be *optimal* if the number of preconditioned iterations is independent of the size of the problem (in our case, on the mesh size h) and the amount of work per iteration scales linearly with the size of the problem. This means that the total work (and, approximately, the corresponding CPU-time) should grow by a factor of 4 (for 2D problems) or 8 (for 3D problems) each time the mesh size is halved.

In Tables 5 and 6 we report the number of iterations for the GMRES method without preconditioning and with the block diagonal GJ preconditioner, both for the exact (Table 5) and inexact (Table 6) versions, for the choices $M = \alpha I + C$ and $M = D_C$. We note that strictly speaking, it would make more sense to use the symmetric formulation (25) of the saddle point problem and to use MINRES instead of GMRES, since the exact block diagonal preconditioner is symmetric and positive definite [15]. However, since we are mostly interested in using the inexact versions of the proposed preconditioners, which require the use of FGMRES, we present results for GMRES instead, in order to have a more meaningful comparison of the performance differences between the exact and inexact variants.

The value used for the parameter α has been computed according to the rule $\tilde{\alpha}^* = 1/(2^{\ell-2} \times 2^{\ell-2}) = 16h^2$ for a uniform $2^\ell \times 2^\ell$ grid. This value is very close to the experimentally obtained best value

	$M=\alpha I$ and $N=\alpha I-C$			
α	$\lambda_{\min}(\mathcal{H})$	$\lambda_{\max}(\mathcal{H})$	η_1	η_2
1e-05	-2.0325e+03	0	-2.3211e+03	1
1e-04	-202.3500	0	-231.2137	1
1e-03	-19.3350	0.7187	-22.2214	1
1e-02	-1.0335	0.9719	-1.7596	1
1e-01	-1.0325	0.9719	-1.7596	1
1e+00	0	0.9997	-1.0076	1
1e+01	0	0.9999	-1.0008	1

Table 1: Computations of the established interval (20) for Example 4.1.

	$M=0.5(\alpha I+C)$ and $N=0.5(\alpha I-C)$			
α	$\lambda_{\min}(\mathcal{H})$	$\lambda_{\max}(\mathcal{H})$	η_1	η_2
1e-05	-1.1133e+03	0	-1.5203e+03	1
1e-04	-110.5771	0	-152.9274	1
1e-03	-10.3312	0.4501	-16.1927	1
1e-02	-0.6769	0.9439	-2.5193	1
1e-01	0	0.9944	-1.1519	1
1e+00	0	0.9994	-1.0152	1
1e+01	0	0.9999	-1.0015	1

Table 2: Computations of the established interval (20) for Example 4.1.

	$M=\alpha I+C$ and $N=\alpha I$			
α	$\lambda_{\min}(\mathcal{H})$	$\lambda_{\max}(\mathcal{H})$	ξ_1	ξ_2
1e-05	-556.1622	0	-759.6368	1
1e-04	-54.7885	0.0046	-75.9637	1
1e-03	-4.6656	0.7251	-7.5964	1
1e-02	0	0.9719	-0.7596	1
1e-01	0	0.9972	-0.0760	1
1e+00	0	0.9997	-0.0076	1
1e+01	0	0.9999	-7.5964e-05	1

Table 3: Computations of the established interval (21) for Example 4.1.

ℓ	m	n	$m+n$
4	578	256	834
5	2178	1024	3202
6	8450	4096	12546
7	33282	16384	49666

Table 4: Size of matrices for grid $2^\ell \times 2^\ell$.

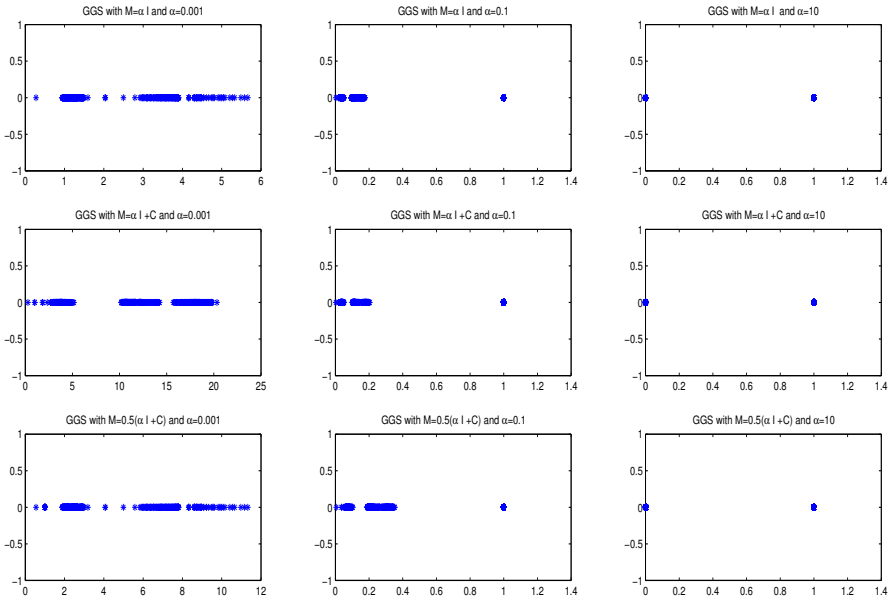

Figure 1: Eigenvalues distribution corresponding to the BGGS (FGGS) preconditioner for Example 4.1 (32×32 grid).

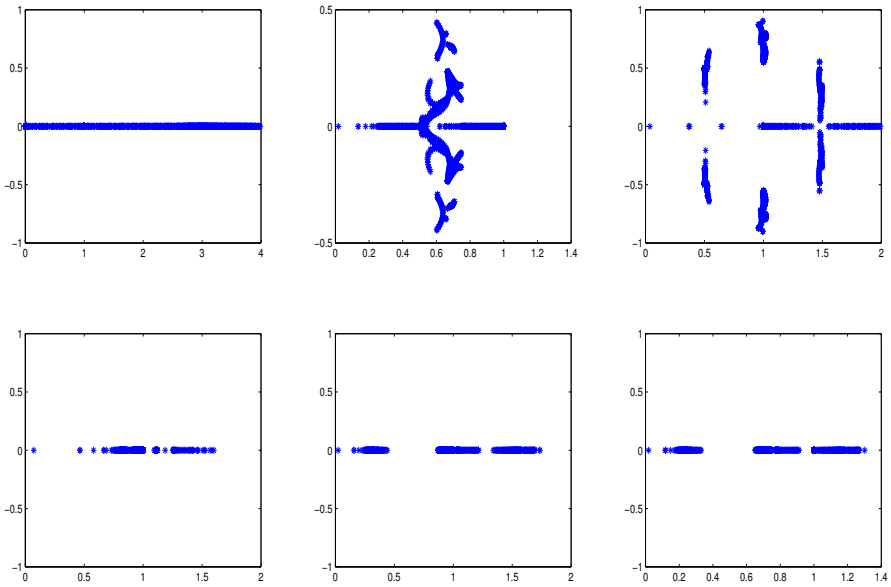

Figure 2: Eigenvalues distribution of $\mathcal{A}$ (top left) and the preconditioned matrices using GJ with $M = \tilde{\alpha}^* I + C$ (top center), $M = D_C$ (top right), and GGS with $M = \alpha^* I + C$ (bottom left), GGS with $M = \alpha^* I + D_C$ (bottom center) and GGS with $M = \tilde{\alpha}^* I$ (bottom right) for Example 4.1 (32×32 grid).

ℓ	GMRES		GMRES ($M = \tilde{\alpha}^* I + C$)		GMRES ($M = D_C$)	
	Iter	CPU	Iter	CPU	Iter	CPU
4	86	0.089	20	0.015	26	0.018
5	182	0.591	24	0.029	31	0.036
6	365	6.953	28	0.169	36	0.205
7	691	115.826	31	0.985	40	1.261

Table 5: Numerical results for solving Example 4.1 by GMRES and its preconditioned version with exact GJ preconditioner and $\tilde{\alpha}^* = 1/(2^{\ell-2} \times 2^{\ell-2})$.

ℓ	FGMRES ($M = \tilde{\alpha}^* I + C$)		FGMRES ($M = D_C$)	
	Iters	CPU	Iters	CPU
4	19(74)	0.029	27(107)	0.037
5	20(117)	0.059	26(153)	0.071
6	22(178)	0.207	29(232)	0.269
7	23(267)	1.096	30(348)	1.372

Table 6: Numerical results for solving Example 4.1 by FGMRES with inexact GJ preconditioner on a $2^\ell \times 2^\ell$ grid with $\tilde{\alpha}^* = 1/(2^{\ell-2} \times 2^{\ell-2})$.

of α which yields the smallest number of required iterations in the GMRES and FGMRES methods. The experimental results show that the examined GJ type preconditioners, while not quite optimal, scale quite well with increasing size, especially for the choice $M = \alpha I + C$. It is also interesting to note that for this test problem and preconditioner type, the inexact variant requires slightly higher CPU times than the exact one.

In Tables 7, 9, 11 and 13, we report the results obtained with the block triangular preconditioners corresponding to the BGGS and FGGS splittings in their exact (Tables 7 and 11) and inexact (Tables 9 and 13) versions.

We recall that for the Stokes problem, problem, Elman et al. [15] suggest using a block diagonal preconditioner of the form

$$\begin{pmatrix} \hat{A} & 0 \\ 0 & M \end{pmatrix}, \tag{33}$$

to be used in conjunction with the MINRES method (applied to the symmetric form of the saddle point problem). In (33), $\hat{A}$ is an optimal preconditioner for A (such as a multigrid V-cycle) and M an approximation to the (negative) Schur complement $C + B^T A^{-1} B$. In practice, using $M = \mathrm{diag}(Q) \approx h^2 I$ (for a uniform mesh) where $\mathrm{diag}(Q)$ denotes the diagonal part of pressure mass matrix Q yields an essentially optimal preconditioner. Motivated by this, we examined the performance of BGGS and FGGS preconditioners for the case that $M \approx h^2 I$; the results obtained are reported for both exact (Tables 8 and 12) and inexact (Tables 10 and 14) versions, respectively. As seen, the numerical experiments illustrate that these types of preconditioners are almost perfectly scalable. We also observe that the inexact block triangular preconditioners (with FGMRES) are faster than the corresponding exact version (with GMRES) and about twice as fast as the block diagonal preconditioner. The reported results appear to be quite competitive with those reported (for the same problem) in [15, Table 4.1], where MINRES with block diagonal preconditioning is used. Finally, we note that the block upper triangular (BGGS) variant appears to be slightly superior to the block lower triangular one (FGGS).

Example 4.2. *In this example we consider a saddle point problem which arises from a mixed finite element discretization of a 3D electromagnetics application. Specifically, the problem consists in computing the magnetic field in a system given by a cube of iron of relative magnetic permeability* 10^3 *surrounded by air, immersed in an imposed uniform field. For more details about the problem and the properties of the coefficient matrix of the resulting saddle point system we refer to Example 2 in*

ℓ	GMRES ($M=\alpha^*I+C$)		GMRES ($M=\alpha^*I+D_C$)	
	Iter	CPU	Iter	CPU
4	10	0.098	17	0.115
5	13	0.155	21	0.177
6	15	0.497	24	0.513
7	16	1.813	26	2.139

Table 7: Numerical results for solving Example 4.1 by GMRES with exact BGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$.

ℓ	GMRES ($M=\alpha^*I$)		GMRES ($M=\tilde{\alpha}^*I$)	
	Iter	CPU	Iter	CPU
4	18	0.104	17	0.095
5	22	0.188	21	0.175
6	25	0.493	23	0.525
7	28	2.139	26	2.159

Table 8: Numerical results for solving Example 4.1 by GMRES with exact BGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$ and $\tilde{\alpha}^*=1/(2^{\ell-2}\times 2^{\ell-2})$.

ℓ	FGMRES ($M=\alpha^*I+C$)		FGMRES ($M=\alpha^*I+D_C$)	
	Iters	CPU	Iters	CPU
4	10(39)	0.025	14(55)	0.027
5	9(52)	0.038	14(76)	0.046
6	9(70)	0.117	15(112)	0.152
7	10(107)	0.544	15(156)	0.717

Table 9: Numerical results for solving Example 4.1 by FGMRES with inexact BGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$.

ℓ	FGMRES ($M=\alpha^*I$)		FGMRES ($M=\tilde{\alpha}^*I$)	
	Iters	CPU	Iter	CPU
4	17(67)	0.031	14(55)	0.028
5	15(78)	0.048	14(76)	0.046
6	15(112)	0.154	17(129)	0.166
7	15(156)	0.719	17(180)	0.799

Table 10: Numerical results for solving Example 4.1 by FGMRES with inexact BGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$ and $\tilde{\alpha}^*=1/(2^{\ell-2}\times 2^{\ell-2})$.

ℓ	GMRES ($M=\alpha^*I+C$)		GMRES ($M=\alpha^*I+D_C$)	
	Iter	CPU	Iter	CPU
4	9	0.106	17	0.129
5	12	0.167	20	0.173
6	14	0.466	23	0.519
7	15	1.768	25	2.129

Table 11: Numerical results for solving Example 4.1 by GMRES with exact FGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$.

ℓ	GMRES ($M=\alpha^*I$)		GMRES ($M=\tilde{\alpha}^*I$)	
	Iter	CPU	Iter	CPU
4	18	0.105	16	0.096
5	21	0.181	20	0.171
6	24	0.516	23	0.491
7	27	2.110	25	2.047

Table 12: Numerical results for solving Example 4.1 by GMRES with exact FGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$ and $\tilde{\alpha}^*=1/(2^{\ell-2}\times 2^{\ell-2})$.

ℓ	FGMRES ($M=\alpha^*I+C$)		FGMRES ($M=\alpha I+D_C$)	
	Iters	CPU	Iter	CPU
4	11(43)	0.027	15(59)	0.029
5	12(70)	0.045	17(100)	0.052
6	12(102)	0.141	18(145)	0.181
7	13(155)	0.703	19(222)	0.935

Table 13: Numerical results for solving Example 4.1 by FGMRES with inexact FGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$.

ℓ	FGMRES ($M=\alpha^*I$)		FGMRES ($M=\tilde{\alpha}^*I$)	
	Iters	CPU	Iter	CPU
4	20(79)	0.034	15(59)	0.029
5	21(124)	0.056	16(94)	0.051
6	20(163)	0.205	18(145)	0.183
7	21(245)	1.101	18(208)	0.878

Table 14: Numerical results for solving Example 4.1 by FGMRES with inexact FGGS preconditioner with $\alpha^*=1/(2^{\ell-1}\times 2^{\ell-1})$ and $\tilde{\alpha}^*=1/(2^{\ell-2}\times 2^{\ell-2})$.

Method Preconditioner	GMRES No		GMRES GJ		GMRES BGGS	
size	Iter	CPU	Iters	CPU	Iter	CPU
1119	231	0.414	25	0.039	13	0.026
2208	223	0.642	21	0.054	11	0.054
4371	350	2.743	25	0.416	13	0.223
8622	432	7.574	29	1.716	15	0.895
22675	968	82.593	43	11.329	22	5.816

Table 15: Numerical results for solving Example 4.2 by GMRES method.

Method Preconditioner	FGMRES GJ (Case A)		FGMRES GJ (Case B)	
size	Iters	CPU	Iters	CPU
1119	28(122)	0.063	28(129)	0.062
2208	26(119)	0.095	26(121)	0.077
4371	28(143)	0.185	28(146)	0.164
8622	32(168)	0.396	34(190)	0.427
22675	48(316)	1.722	48(324)	1.731

Table 16: Numerical results for solving Example 4.2 by FGMRES method.

[22, page 610]; see also [26, Example 10]. For this problem the matrix B in (1) is not full rank and $C \neq 0$ is symmetric positive semidefinite and such that $B^TB + C$ is symmetric positive definite. The nonzero matrix C arises from the need to stabilize the discrete saddle point problem.

First, we consider the GJ and BGGS preconditioners with two different choices of the splitting $C = M - N$, given as follows:

- Case a. $M = B^TB + C$.
- Case b. $M = -(B^TB + C)$.

We utilize the PCG method in both lines of Algorithms 1 and 2. Here, incomplete Cholesky factorizations of the (1,1)-block A in (1) and $B^TB + C$ in Cases A and B are computed using the MATLAB functions `ichol(`A, `struct('droptol',1e-2)`" and `ichol(`$B^TB + C$, `struct('droptol',1e-4)`", respectively. Then, the incomplete factors are utilized as preconditioners in the inner PCG iterations for solving subsystems in both lines of the algorithms. Similar to the previous example, the iterations in the PCG method are stopped when the residual norm is reduced by a factor of 100 or when the maximum number of 40 iterations is reached.

In Table 15 we report the results corresponding to GMRES and its preconditioned versions where the subsystems are solved exactly via sparse Cholesky factorization with SYMAMD reordering. With "size" we denote the total number ($= n + m$) of unknowns in the saddle point system. Notice that when the subsystems are solved exactly, the results for Cases A and B are identical.

In Tables 16 and 17 we report results for FGMRES with the inexact GJ and BGGS preconditioners, respectively. For each preconditioner we show the results for both Cases A and B. It can be seen that

Method Preconditioner	GMRES BGGS (Case A)		FGMRES BGGS (Case B)	
size	Iters	CPU	Iters	CPU
1119	18(80)	0.047	13(58)	0.047
2208	17(77)	0.069	12(53)	0.051
4371	19(96)	0.134	13(65)	0.093
8622	21(114)	0.284	16(87)	0.236
22675	32(214)	1.325	23(157)	1.016

Table 17: Numerical results for solving Example 4.2 by FGMRES method.

for FGMRES with inexact BGGS preconditioning, case B appears to provide superior performance. When the problem size is scaled by a factor of about 21 (from smallest to largest), the CPU time increases by a factor of 22 — a nearly optimal scaling.

Moreover, it can be seen that the inexact approach is, again, much faster than the exact one. We also tried replacing the incomplete Cholesky preconditioner for the inner PCG iterations with a standard algebraic multigrid (AMG) solver, but we found that this approach was far more expensive and not competitive in terms of both CPU time and storage with the IC-based one.

We conclude this section by mentioning that our approach (particularly the inexact BGSS preconditioner, Case B) was found to outperform the block diagonal and block triangular preconditioners proposed in [22] and [26] which, to the best of our knowledge, represented the fastest methods available in the literature up to now for this problem.

5 Conclusions

In this paper we have revisited the use of block diagonal and block triangular splittings and preconditioners for solving stabilized saddle point problems.

At the outset we considered the case where the $(1,1)$ block is assumed to be inverted exactly, while the (generally singular) $(2,2)$ block is split as $C = M - N$. Various choices of M have been considered. We have addressed the theoretical question of which choices of M can be expected to provide convergent splittings, obtaining some new results. In particular, bounds on the eigenvalues of the (exactly) preconditioned matrices were derived.

Next, we investigated the performance of both exact and inexact variants of these block diagonal and block triangular preconditioners for methods like GMRES and FGMRES. We have tested these methods on two types of test problems, a 2D Stokes problem and a 3D magnetostatics problem. Our experiments show that certain inexact variants of the upper triangular block preconditioner work very well in practice, outperforming some of the best methods previously available for solving these two problems. For the Stokes problem these preconditioners contain a user-defined parameter, however, it is easy to find (nearly) optimal values of the parameter in terms of the mesh size.

Although in this paper we have focused primarily on saddle point problems arising from the use of stabilized finite elements, the convergence theory and eigenvalue bounds apply to any saddle point problem of the form (1), including those arising from regularized least-squares and other optimization problems, as long as the conditions on the matrices A, B, and C are satisfied. Whether the solvers studied in this paper would prove competitive on such problems is, however, a different matter, which we leave for future investigation.

Acknowledgments

The authors are grateful to Ilaria Perugia for providing the test matrices for Example 4.2. The work of MB was supported in part by DOE (Office of Science) grant ERKJ247.

References

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

[2] 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.

[3] Z.-Z. Bai, G.H. Golub, and C.-K. Li, *Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices*, SIAM J. Sci. Comput., 28 (2006), 583-603.

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

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

[6] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods, Springer–Verlag, New York, 1991.

[7] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), 1-17.

[8] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal. 34 (1997), 1072-1092.

[9] Y. Cao, J. Du, and Q. Niu, *Shift-splitting preconditioners for saddle point problems*, J. Comput. Appl. Math., 272 (2014), 239-250.

[10] Y. Cao, S. Li, and L. Q. Yao, *A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems*, Appl. Math. Lett., 49 (2015), 20-27.

[11] Z.-H. Cao, *Positive stable block triangular preconditioners for symmetric saddle point problems*, Appl. Numer. Math., 57 (2007), 889-990.

[12] C. Chen and C. Ma, *A generalized shift-splitting preconditioners for nonsymmetric saddle point problems*, Appl. Math. Lett., 43 (2015), 49-55.

[13] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), 1645-1661.

[14] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: A Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Software., 33 (2007), Article 14.

[15] H. C. Elman, D. J. Silvester, and A. J. Wathen, Finite Elements and Fast Iterative Solvers, Second Edition, Oxford University Press, Oxford, 2014.

[16] V. Girault and P. A. Raviart, Finite Element Methods for Navier–Stokes Equations, Springer–Verlag, Berlin, 1986.

[17] R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator–splitting Methods in Nonlinear Mechanics, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[18] G. H. Golub, X. Wu and J. Y. Yuan, *SOR-like methods for augmented systems*, BIT Numer. Math., 41 (2001), 71-85.

[19] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, UK, 1985.

[20] A. Klawonn, *Block–triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), 172-184.

[21] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (1999), 1969-1972.

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

[23] J. Pestana, *On the eigenvalues and eigenvectors of block triangular preconditioned block matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), 517-525.

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

[25] D. K. Salkuyeh, M. Masoudi, and D. Hezari, *On the generalized shift-splitting preconditioner for saddle point problems*, Appl. Math. Lett., 48 (2015), 55-61.

[26] V. Simoncini, *Block triangular preconditioners for symmetric saddle–point problems*, Appl. Numer. Math., 49 (2004), 63-80.

[27] A. L. Yang and Y. J. Wu, *The Uzawa–HSS method for saddle–point problems*, Appl. Math. Lett., 38 (2014), 38-42.

[28] D. M. Young, Iterative Solution or Large Linear Systems, Academic Press, New York, 1971.

[29] J. H. Yun, *Variants of the Uzawa method for saddle point problems*, Comput. Math. Appl., 65 (2013), 1037-1046.

[30] F. Zhang and Q. Zhang, *Eigenvalue inequalities for matrix product*, IEEE Trans. Automat. Control, 51 (2006), 1506-1509.

[31] G. F. Zhang, J. L. Yang, and S. S. Wang, *On generalized parameterized inexact Uzawa method for a block two-by-two linear system*, J. Comput. Appl. Math., 255 (2014), 193-207.