SIAM J. MATRIX ANAL. APPL.
Vol. 31, No. 2, pp. 360–374

A GENERALIZATION OF THE HERMITIAN AND SKEW-HERMITIAN SPLITTING ITERATION*

MICHELE BENZI†

To Gene Golub, in memoriam

Abstract. This paper is concerned with a generalization of the Hermitian and skew-Hermitian (HSS) splitting iteration for solving positive definite, non-Hermitian linear systems. It is shown that the new scheme can outperform the standard HSS method in some situations and can be used as an effective preconditioner for certain linear systems in saddle point form. Numerical experiments using discretizations of incompressible flow problems demonstrate the effectiveness of the generalized HSS preconditioner.

Key words. matrix splittings, HSS iteration, preconditioning, saddle point systems, generalized Stokes and Oseen problems, Navier–Stokes equations, convection-diffusion equation

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 15A06

DOI. 10.1137/080723181

1. Introduction.

The Hermitian and skew-Hermitian splitting (HSS) iteration was first introduced by Bai, Golub, and Ng in [5] for the solution of a broad class of non-Hermitian linear systems $Ax = b$. The basic iterative scheme works as follows. Let $\alpha > 0$ be fixed. Denoting by $H = (A + A^*)/2$ the Hermitian part of A and by $S = (A - A^*)/2$ the skew-Hermitian part, the HSS method is the alternating iteration

$$\begin{cases} (H + \alpha I)\, x_{k+\frac{1}{2}} = (\alpha I - S)\, x_k + b, \\ (S + \alpha I)\, x_{k+1} = (\alpha I - H)\, x_{k+\frac{1}{2}} + b \end{cases} \tag{1.1}$$

$(k = 0, 1, \dots)$, where x_0 is an arbitrary initial guess. It was shown in [5] that when H is positive definite, the HSS iteration is unconditionally convergent, i.e., the sequence $\{x_k\}$ converges to the solution $x_* = A^{-1}b$ as $k \to \infty$ for all $\alpha > 0$ and for any choice of x_0. Moreover, it was shown in the same paper that choosing $\alpha = \sqrt{ab}$, where $a = \lambda_{\min}(H)$ and $b = \lambda_{\max}(H)$ are the extreme eigenvalues of H, minimizes an upper bound on the spectral radius of the iteration matrix associated with stationary scheme (1.1).

Due to its promising performance and elegant mathematical properties, the HSS scheme immediately attracted considerable attention, resulting in numerous papers devoted to various aspects of the new algorithm.[1] In one direction, the method was extended to the solution of saddle point problems in [8, 9]; see also [1, 6, 13, 22]. Notice that, in these problems, the Hermitian part of A is generally singular, requiring a different convergence analysis. Another natural development, first considered in [8, 9], was to use the HSS scheme not as a stationary iterative solver but as a preconditioner

*Received by the editors May 5, 2008; accepted for publication (in revised form) by D. P. O'Leary November 3, 2008; published electronically April 16, 2009. This work was supported by National Science Foundation grant DMS-0511336.

http://www.siam.org/journals/simax/31-2/72318.html

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (benzi@mathcs.emory.edu).

[1] It is worth mentioning that the original HSS paper [5] was one of just 22 included in the volume [14] of selected works of Golub.

for Krylov subspace methods [21], resulting in a far more efficient and robust class of solvers. Other significant developments include preconditioned variants of the HSS iteration, extension to certain singular systems, studies on the optimal selection of iteration parameters, and application of the HSS preconditioner to specific problems like convection-diffusion equations, incompressible flow problems, and real formulations of complex linear systems, often with excellent results; see [2, 3, 7, 10, 12] and the references therein.

A potential difficulty with the HSS approach is the need to solve two linear systems at each iteration (or at each application of the preconditioner), one with matrix $H + \alpha I$ and one with matrix $S + \alpha I$; cf. (1.1). The first of these two systems is Hermitian positive definite and often well-conditioned (at least for α not too small), and its solution does not present great difficulties. The shifted skew-Hermitian system, however, can be much more problematic; in some cases, its solution is as difficult as that of the original linear system $Ax = b$. There are situations where matrix S is structured in such a way as to make systems involving $S+\alpha I$ easy to solve; see [11] for an example arising in image processing and [10] for one arising in fluid mechanics, as well as section 4 below. In general, however, this will not be the case. It is, therefore, not surprising that, beginning with the first HSS paper [5], considerable attention has been devoted to the use of *inexact* solves. In a typical situation, the two linear systems in (1.1) are solved to relatively low accuracy, usually by inner iterative schemes which could be some preconditioned Krylov iteration or a multigrid method. There is now considerable evidence that the good convergence properties are preserved even when the inner solves are performed to rather low accuracy, resulting in considerable savings, especially for very large problems; see, e.g., [10].

Again motivated by practical considerations, the HSS method was generalized in [4] by considering splittings of the form $A = P + S$, where P is positive definite (not necessarily Hermitian) and S is skew-Hermitian. In particular, it was shown in [4] that P can be taken to be block triangular, thus making the corresponding alternating iteration (of the form (1.1), with P replacing H) more practical in many cases.

In this paper, a different generalization of the original HSS scheme is presented, and some of its basic properties are studied. The idea is to split H into the sum of two Hermitian positive semidefinite matrices: $H = G + K$, where K is of simple form (e.g., diagonal) and to associate K to the skew-Hermitian portion S of A so that $A = G + (S + K)$. The generalized HSS (GHSS) scheme is based on the splittings

$$A = (G + \alpha I) - (\alpha I - S - K) \quad \text{and} \quad A = (S + K + \alpha I) - (\alpha I - G)\,.$$

When either G or K is positive definite, the resulting scheme, which reduces to the classical HSS method when $K = 0$, is shown to be convergent for all $\alpha > 0$; see the next section. Furthermore, preconditioning can be incorporated into the new scheme in the same manner as in the standard HSS scheme, often resulting in much faster convergence. One advantage of the generalized scheme consists in the fact that the solution of systems with coefficient matrix $S+K+\alpha I$ by inner iterations is made easier, since this matrix is more "diagonally dominant" (loosely speaking) and typically better conditioned than $S + \alpha I$. The new variant (to be used in the preconditioned form or as a preconditioner for a Krylov subspace method) was motivated by the type of matrices that arise in the solution of time-dependent incompressible flow problems, but it is not restricted to them and can be described in a completely algebraic manner.

The remainder of the paper is organized as follows: in section 2, the GHSS scheme is described, and some of its properties are discussed. Preconditioned variants are

considered in section 3. Types of problems for which the GHSS method appears to be well-suited are described in section 4, and a few numerical tests are discussed in section 5. Concluding remarks are given in section 6.

2. The GHSS scheme. First, a brief review of the classical HSS iteration is needed. This method is based on the splitting $A = H + S$, which, in turn, induces the two splittings

$$A = (H + \alpha I) - (\alpha I - S) \qquad \text{and} \qquad A = (S + \alpha I) - (\alpha I - H)\,,$$

with $\alpha > 0$. Alternating between these two splittings yields the HSS iteration (1.1). Elimination of $x_{k+\frac{1}{2}}$ from the second of (1.1) yields

$$x_{k+1} = T_\alpha x_k + P_\alpha^{-1} b$$

$(k = 0, 1, \dots)$, where

$$T_\alpha = I - P_\alpha^{-1} A, \quad P_\alpha = \frac{1}{2\alpha}(H + \alpha I)(S + \alpha I)\,.$$

Here $A = P_\alpha - (P_\alpha - A)$ is the splitting induced by the HSS iteration, and P_α is the HSS preconditioner. The iteration matrix T_α is given by

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

which is similar to

$$\hat{T}_\alpha = (\alpha I - H)(H + \alpha I)^{-1}(\alpha I - S)(S + \alpha I)^{-1}\,.$$

Hence, the following bound for the spectral radius of T_α holds:

$$\rho(T_\alpha) = \rho\left(\hat{T}_\alpha\right) \le \left\|(\alpha I - H)(H + \alpha I)^{-1}\right\| \, \left\|(\alpha I - S)(S + \alpha I)^{-1}\right\|\,,$$

where $\|\cdot\|$ denotes the spectral norm. Since $(\alpha I - S)(S + \alpha I)^{-1}$ is the Cayley transform of a skew-Hermitian matrix, it is unitary (see, e.g., [17]), and therefore, it has norm 1. Denoting by $\sigma(H)$ the spectrum of H, it follows that

$$\rho(T_\alpha) \le \max_{\lambda \in \sigma(H)} \frac{|\lambda - \alpha|}{|\lambda + \alpha|} < 1\,, \tag{2.1}$$

showing that the method converges for all $\alpha > 0$. As a consequence, the spectrum of the preconditioned matrix $P_\alpha^{-1} A$ lies inside the circle of center $(1, 0)$ and radius 1 in the complex plane.

It has been shown in [5] that choosing $\alpha = \alpha_* = \sqrt{\lambda_{\min}(H)\lambda_{\max}(H)}$ leads to the following CG-like convergence bound:

$$\rho(T_{\alpha_*}) \le \rho_* = \frac{\sqrt{\kappa(H)} - 1}{\sqrt{\kappa(H)} + 1}, \tag{2.2}$$

where $\kappa(H)$ is the spectral condition number of H. If the Hermitian part H is only positive semidefinite (i.e., $\lambda_{\min}(H) = 0$), then upper bound (2.1) on the spectral radius becomes $\rho(T_\alpha) \le 1$ and the HSS method may not converge, in general. However, in

the special case of (generalized) saddle point problems, the method can be shown to be unconditionally convergent; see [9, Theorem 3.1].

Numerical experiments show that the HSS iteration is especially effective if either H or S "dominates," i.e., if either part of A is much larger than the other. For instance, assume that $A = H + S = \varepsilon L + S$, where $\varepsilon > 0$ is a (small) parameter, $L = L^*$ is positive definite, and $S = -S^*$. Assume further that L and S have been scaled so that $\|L\| = \|S\| = 1$. Then

$$P_\alpha = \frac{1}{2\alpha}(\varepsilon L + \alpha I)(S + \alpha I) = \frac{1}{2}A + \frac{\varepsilon}{2\alpha}LS + \frac{\alpha}{2}I\,.$$

This shows that, up to a numerical factor, HSS preconditioner P_α can be regarded as an approximate factorization of coefficient matrix A; for fixed α and for $\varepsilon \to 0$, matrix $2P_\alpha$ approaches $A + \alpha I$, which is a good approximation of A when α is small. Taking $\alpha = \sqrt{\varepsilon}$ yields a preconditioner that (apart from the factor $1/2$) approaches A as $\varepsilon \to 0$. This heuristic argument helps explain the good performance of the method on linear systems with dominant skew-Hermitian part; obviously, a similar heuristic argument applies in case the Hermitian part dominates.

Consider now the situation where the Hermitian part H can be split as $H = \varepsilon L + K$ ($\varepsilon > 0$ small), where $L = L^*$ is positive definite and $K = K^*$ is positive semidefinite, possibly singular. For example, K could be a matrix of the form $K = \sigma M$, where σ is related to a time step, e.g., $\sigma = O(1/\Delta t)$, and M is a mass matrix or possibly a scaled identity. In this case, it is easy to see that P_α does not yield a good approximation of A as $\varepsilon \to 0$ (for any α), and, indeed, the performance of HSS on this type of problem is sometimes poor. The following extension of the HSS method may be used instead. Write $G = \varepsilon L$ and consider the splittings

$$A = (G + \alpha I) - (\alpha I - S - K) \quad \text{and} \quad A = (S + K + \alpha I) - (\alpha I - G),$$

together with the corresponding alternating iterative scheme

$$\begin{cases} (G + \alpha I)\, x_{k+\frac{1}{2}} = (\alpha I - S - K)\, x_k + b, \\ (S + K + \alpha I)\, x_{k+1} = (\alpha I - G)\, x_{k+\frac{1}{2}} + b\,, \end{cases} \tag{2.3}$$

($k = 0, 1, \dots$). The convergence of this iterative process is an easy consequence of the following classical result, which is known as *Kellogg's lemma*. See, e.g., [19, page 13] for a proof.

LEMMA 2.1. *Let $A = H + S$, where $H = \frac{1}{2}(A + A^*)$. If H is positive semidefinite and $\alpha \geq 0$, then*

$$\left\|(\alpha I - A)(\alpha I + A)^{-1}\right\| \leq 1.$$

If, in addition, H is positive definite and $\alpha > 0$, then

$$\left\|(\alpha I - A)(\alpha I + A)^{-1}\right\| < 1.$$

The following convergence result holds.

THEOREM 2.2. *Let $A = (G + K) + S = H + S$, where G and K are Hermitian positive semidefinite and S is skew-Hermitian. If either G or K is positive definite, alternating iteration* (2.3) *converges unconditionally to the unique solution of $Ax = b$.*

Proof. Elimination of $x_{k+\frac{1}{2}}$ from (2.3) leads to the iterative scheme $x_{k+1} = T_\alpha x_k + P_\alpha^{-1} b$, where

$$T_\alpha = (S + K + \alpha I)^{-1}(\alpha I - G)(G + \alpha I)^{-1}(\alpha I - S - K) = I - P_\alpha^{-1} A\,,$$

with $P_\alpha = \frac{1}{2\alpha}(G+\alpha I)(S+K+\alpha I)$. Iteration matrix T_α is similar to

$$\hat{T}_\alpha = (\alpha I - G)(G+\alpha I)^{-1}(\alpha I - S - K)(S+K+\alpha I)^{-1}\,.$$

Clearly,

$$\rho(T_\alpha) = \rho\left(\hat{T}_\alpha\right) \le \left\|(\alpha I - G)(G+\alpha I)^{-1}\right\| \left\|(\alpha I - S - K)(S+K+\alpha I)^{-1}\right\|,$$

and Lemma 2.1 together with the assumption made on G and K readily implies that at least one of the two matrix norms on the left-hand side of this expression is strictly less than 1, with the other being at most 1. Therefore, $\rho(T_\alpha) < 1$ for all $\alpha > 0$. ∎

Some remarks are in order. First of all, it is clear that iterative scheme (2.3) is a generalization of the standard HSS scheme (1.1), to which it reduces whenever $K = 0$.

Secondly, if G and K are both positive definite, then iteration matrix T_α is similar to matrix $\hat{T}_\alpha$, which is the product of two factors $(\alpha I - G)(G+\alpha I)^{-1}$ and $(\alpha I - S - K)(S+K+\alpha I)^{-1}$, both of which are contractions with respect to the spectral norm. In contrast, in the original HSS method, T_α is similar to a matrix $\hat{T}_\alpha$, which is the product of contraction times an isometry. This is not enough to conclude that the new scheme converges (asymptotically) faster than the standard HSS iteration, and, indeed, one can find examples where the spectral radius is smaller for the HSS scheme than for the GHSS one, as well as examples where the converse is true. Hence, no general comparison theorem is possible for the spectral radii associated with the two iterations. Nevertheless, it is possible to give some idea of when GHSS can be expected to perform better than HSS. As a simple example, let $n = 100$, and let $A = G+K+S$, where $G = \varepsilon L$, with $\varepsilon = 10^{-1}$ and

$$L = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 0.1 & & 0 \\ -0.1 & \ddots & \ddots & \\ & \ddots & \ddots & 0.1 \\ 0 & & -0.1 & 0 \end{bmatrix}, \quad K = 10^{-1} I_n\,.$$

Then, for $\alpha = 0.1$, the spectral radius of the iteration matrix is $\rho = 0.5347$ using standard HSS splitting and $\rho = 0.3195$ using GHSS splitting. It is instructive to interpret the corresponding preconditioner

$$P_\alpha = \frac{1}{2\alpha}(\varepsilon L + \alpha I)(S+K+\alpha I)$$

as an approximate factorization of A. Indeed, expanding the product yields

$$P_\alpha = \frac{1}{2}A + \frac{\varepsilon}{2\alpha}L(S+K) + \frac{\alpha}{2}I\,.$$

Again, this shows that, up to a numerical factor, the new preconditioner P_α can be regarded as an approximate factorization of the coefficient matrix A; for fixed α and for $\varepsilon \to 0$, matrix $2P_\alpha$ approaches $A + \alpha I$, which is a good approximation of A when α is small. Again, taking $\alpha = \sqrt{\varepsilon}$ yields a preconditioner that (apart from the factor $1/2$) approaches A as $\varepsilon \to 0$. This is not true, however, for the HSS preconditioner $P_\alpha = \frac{1}{2}A + \frac{1}{2\alpha}(\varepsilon L + K)S + \frac{\alpha}{2}I$.

A few comments are in order concerning the "optimal" choice of the parameter α. In the literature, a great deal of effort has been put into determining the value of α,

which minimizes the spectral radius of the iteration matrix T_α or some upper bound on it. Such estimates on α can be given for the GHSS method using, for example, the approach detailed in [19, Chapter 4]. However, the practical usefulness of such estimates is questionable. First of all, the estimated value of α usually depends on spectral information that may not be accessible, such as the extreme eigenvalues of H (or, in the case of the GHSS method, of G). Second, minimizing the spectral radius or an upper bound on it does not always result in the best choice of α when the stationary iteration is accelerated by a Krylov method; see, e.g., the analysis and examples in [8]. Furthermore, when these techniques are used as preconditioners, the exact value of α is not as important as when the algorithms are used as iterative solvers. Since Krylov acceleration dramatically improves the rate of convergence of these methods (with a largely negligible increase in cost per iteration), there is little reason to try to estimate the "optimal" value of α. Experience suggests that in most applications and for an appropriate scaling of the problem, a "small" value of α (usually between 0.01 and 0.5) gives good results.

As already mentioned, the main potential advantage of the GHSS scheme over the classical one is the fact that the solution of linear sytems with coefficient matrix $S + K + \alpha I$, where $K \neq 0$ is positive semidefinite and of simple form (e.g., diagonal or block diagonal with blocks of small size, or perhaps banded) can be expected to be less expensive than in the classical HSS scheme ($K = 0$) in the common situation where inner iterative solvers are used. For example, when K is a nonnegative diagonal matrix, the corresponding system will have a "heavier" diagonal and be better conditioned. Of course, the solution of systems involving matrix $H + \alpha I$ is now replaced by the solution of systems with matrix $G + \alpha I$, which now has a somewhat "weaker" diagonal and could be somewhat less well-conditioned. However, linear systems with matrix $G + \alpha I$ will typically be still fairly easy to solve, since this matrix is Hermitian positive definite and efficient solvers exist for problems of this kind. Some examples are considered in section 4.

3. Preconditioned GHSS iteration. As already mentioned, the HSS iteration is a stationary iteration that can be accelerated by a nonsymmetric Krylov subspace method like GMRES, and the same is true of GHSS. In other words, the matrices P_α induced by HSS and GHSS can be used to precondition system $Ax = b$.

It is also possible, however, to take a different approach and to apply the stationary HSS scheme to the symmetrically preconditioned system $B^{-1}AB^{-*}(B^*x) = B^{-1}b$. Here $A = H + S$ is assumed to be positive real, and the (nonsingular) matrix B should be chosen so that the Hermitian positive definite matrix $P = BB^*$ is spectrally equivalent to H; that is, the condition number of $\hat{H} = B^{-1}HB^{-*}$ is independent of problem parameters, such as mesh size h for linear systems arising from PDEs. It follows then from bounds (2.1)–(2.2) (with $\hat{H}$ replacing H) that the asymptotic rate of convergence of the preconditioned HSS (PHSS) iteration is independent of h; the parameter α can be taken simply to be 1. Remarkably, nonnormality effects do not play a role here [12], and the asymptotic rate of convergence is indeed the actual rate of convergence of the iteration. These properties make the PHSS method especially appealing for the solution of linear systems arising from finite difference or finite element discretizations of convection-diffusion equations, since, for this class of problems, fast Poisson solvers (or multigrid iterations) provide optimal preconditioners P that can be efficiently implemented. It is straightforward to see [9, 12] that the PHSS iteration is mathematically equivalent to the alternating iteration based on the splittings

$$A = (H + \alpha P) - (\alpha P - S) = (S + \alpha P) - (\alpha P - H).$$

See [12] and [6] for detailed analyses and discussions of the PHSS method and its variants.

It is possible to develop a preconditioned GHSS method (PGHSS for short), along the same lines. Let $A = H + S$ and $H = G + K$, with G Hermitian and positive definite. If $P = BB^*$ denotes a preconditioning matrix which is spectrally equivalent to G, then applying the GHSS iteration to the symmetrically preconditioned system $B^{-1}AB^{-*}(B^*x) = B^{-1}b$ results in an iterative process with asymptotic rate of convergence bounded independently of h; once again, one can simply take $\alpha = 1$ for the convergence parameter. The PGHSS scheme is mathematically equivalent to the alternating iteration based on the splittings

$$A = (G + \alpha P) - (\alpha P - K - S) = (S + K + \alpha P) - (\alpha P - H).$$

Much of the analysis and observations in [12] concerning the PHSS method carry over, essentially without changes, to the PGHSS method. Some applications of this scheme to unsteady convection-diffusion equations and to reaction–convection-diffusion equations are considered in the next section.

4. Some applications of the new scheme. In this section, some applications of the GHSS preconditioner and of the preconditioned GHSS method are discussed, with a focus on problems arising in computational fluid dynamics.

4.1. Saddle point problems. As already noted, the HSS preconditioner has been successfully tested on problems from incompressible fluid dynamics; see, in particular, [10, 18]. An important example is given by the *unsteady Navier–Stokes equations*:

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times [0, T], \tag{4.1}$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega \times [0, T], \tag{4.2}$$

where $\nu > 0$ is the kinematic viscosity, $\Omega \subset \mathbb{R}^3$ is an open bounded region with sufficiently smooth boundary $\partial\Omega$, $[0, T]$ is a time interval, $\mathbf{u}$ is the unknown velocity field, and p is the pressure. The unknown functions $u = u(\mathbf{x}, t)$ and $p = p(\mathbf{x}, t)$ are subject to suitable boundary and initial conditions. When implicit methods are used to integrate (4.1)–(4.2) in time and a simple linearization is applied, a sequence of semidiscrete, linear boundary value problems of the form

$$\sigma \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{w} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \tag{4.3}$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega, \tag{4.4}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega \tag{4.5}$$

is obtained. This problem is often referred to as the *generalized Oseen problem*. Here $\mathbf{w}$ is a given divergence-free vector field (the "wind"), and $\sigma > 0$ is related to the time step Δt, for example, $\sigma = O(1/\Delta t)$. In the three-dimensional (3D) case, this is a system of four partial differential equations, to be solved for unknowns $\mathbf{u} = (u, v, w)$ and p. For $\sigma = 0$, the steady-state case is obtained. For $\mathbf{w} = \mathbf{0}$, problem (4.3)–(4.5) reduces to the *generalized Stokes problem*, which is also of considerable importance in the numerical solution of the unsteady Navier–Stokes equations; see [16] or [20] for details.

Space discretization of the generalized Oseen problem (4.3)–(4.5) by finite difference or finite element schemes leads to a large, sparse linear system in *saddle point*

form:

$$\begin{bmatrix} A & B^T \\ -B & C \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{4.6}$$

Here $A = \sigma M + \nu L + N$ is the discrete counterpart of operator $\sigma - \nu\Delta + \mathbf{w}\cdot\nabla$. Matrix L is symmetric positive definite (SPD) and consists of a direct sum of discrete Laplace operators. Skew-symmetric matrix N contains the first-order (convective) terms; note that $N = 0$ for the generalized Stokes problem. Also, M is a mass matrix, possibly a scaled identity. (Rectangular) matrix B^T is the discrete gradient, and $-B$ the discrete divergence operator. Finally, C is a symmetric positive semidefinite pressure stabilization matrix; usually $C = 0$ for div-stable discretizations [16]. The (1,1) block A has dimensions $n \times n$, while (2,2) block C is $m \times m$, with $n > m$. Rather than writing the saddle point system in the equivalent indefinite formulation (with B instead of $-B$ in the (2,1) block and $-C$ instead of C in the (2,2) block), the nonsymmetric positive semidefinite formulation (4.6) is used here in order to easily form the HSS and GHSS splittings. The standard HSS method is based on the splittings

$$\begin{bmatrix} \sigma M + \nu L + N & B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} \sigma M + \nu L + \alpha I & 0 \\ 0 & C + \alpha I \end{bmatrix} - \begin{bmatrix} \alpha I - N & -B^T \\ B & \alpha I \end{bmatrix}$$

and

$$\begin{bmatrix} \sigma M + \nu L + N & B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} N + \alpha I & B^T \\ -B & \alpha I \end{bmatrix} - \begin{bmatrix} \alpha I - \sigma M - \nu L & 0 \\ 0 & \alpha I - C \end{bmatrix}.$$

In contrast, the GHSS method is based on the splittings

$$\begin{bmatrix} \sigma M + \nu L + N & B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} \nu L + \alpha I & 0 \\ 0 & C + \alpha I \end{bmatrix} - \begin{bmatrix} \alpha I - \sigma M - N & -B^T \\ B & \alpha I \end{bmatrix}$$

and

$$\begin{bmatrix} \sigma M + \nu L + N & B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} \sigma M + N + \alpha I & B^T \\ -B & C + \alpha I \end{bmatrix} - \begin{bmatrix} \alpha I - \nu L & 0 \\ 0 & \alpha I - C \end{bmatrix}.$$

Therefore, the difference between HSS and GHSS preconditioning amounts to the structure of the systems to be solved at each application of the preconditioner. HSS requires first solving two decoupled SPD linear systems, one with coefficient matrix $\sigma M + \nu L + \alpha I$ and the other with matrix $C + \alpha I$, followed by a shifted skew-symmetric linear system of the form

$$\begin{bmatrix} N + \alpha I & B^T \\ -B & \alpha I \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ d \end{bmatrix}. \tag{4.7}$$

For 3D problems, the first SPD system decouples into four independent linear systems that are strongly diagonally dominant and rather well-conditioned; in an inexact implementation, just one or two iterations of multigrid or preconditioned CG (PCG) are usually enough to achieve a sufficiently accurate solution, in the sense that the accuracy is enough to preserve the rate of convergence of the outer iteration.

The second system (4.7) is more complicated. For the generalized Stokes problem ($N = 0$), it can be reduced to the solution of a much smaller ($m \times m$) system involving the Schur complement $\alpha I + \frac{1}{\alpha} BB^T$. This system is also SPD and is essentially a

discrete analogue of a (shifted) Poisson-type equation; it can be solved efficiently by CG or by multigrid. When $N \neq 0$, however, such reduction is not possible, since the corresponding Schur complement matrix is now dense, and coupled system (4.7) must be solved by a Krylov method like preconditioned GMRES.

With the GHSS method, the first system to be solved also decouples into a set of independent systems with SPD matrices that are well-conditioned and often diagonally dominant. The second system can also be reduced, in the case $N = 0$, to a much smaller one involving the Schur complement $\alpha I + B(\alpha I + \sigma M)^{-1}B^T$. If M is a diagonal matrix (such as a scaled identity), this matrix can be explicitly formed. If M is not diagonal, for the purpose of constructing a preconditioner, it can be approximated by a diagonal matrix, by either lumping or simply by replacing M with its diagonal. The resulting system is very easy to solve; in an inexact implementation, one or two steps of an inner PCG iteration usually suffice.

As before, when $N \neq 0$, no explicit Schur complement reduction is available, and a coupled system of the form

$$\begin{bmatrix} \sigma M + N + \alpha I & B^T \\ -B & \alpha I \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ d \end{bmatrix} \tag{4.8}$$

must be solved at each iteration. Fortunately, the presence of matrix σM in the (1,1) block tends to make this system easier to solve than the one arising from the standard HSS method (cf. (4.7)), where the term σM is not present (since it is included in the symmetric part). A few numerical experiments in support of this claim are presented in the following section.

It is worth mentioning that the (exact) GHSS iteration can be shown to be unconditionally convergent for saddle point problems of type (4.6), thus generalizing Theorem 3.1 in [9]. The formal statement is as follows.

THEOREM 4.1. *Assume that $A \in \mathbb{R}^{n\times n}$ has positive definite symmetric part $H = (A + A^T)/2$, $B \in \mathbb{R}^{m\times n}$ has full row rank, and that $C = C^T \in \mathbb{R}^{m\times m}$ is positive semidefinite. Assume further that H is split as $H = G + K$, with G SPD and K positive semidefinite. Then the GHSS iteration converges unconditionally to the unique solution of problem* (4.6).

The proof is a straightforward modification of the one given in [9, Theorem 3.1] and is omitted. See also [18, pages 61–65] for a special case.

4.2. Convection-diffusion equations. Here we consider two types of problems, both of which can be solved with the PGHSS method. The first one is the unsteady convection-diffusion equation

$$\frac{\partial u}{\partial t} - \mathrm{div}(a\nabla u) + p\, u_x + q\, u_y = f \quad \text{in } \Omega \times [0,T]\,, \tag{4.9}$$

complemented by suitable initial and boundary conditions. Here the coefficients a, p, q, and the right-hand side f are functions of $x \in \Omega$ and possibly of time t; the diffusion coefficient $a = a(x,t)$ is assumed to be uniformly positive in $\Omega \times [0,T]$, where $[0,T]$ is the time interval of interest. Problem (4.9) is two-dimensional (2D), but 3D problems could be considered as well. The second type of problem is the (steady) convection-diffusion-reaction equation of the form

$$-\mathrm{div}(a\nabla u) + p\, u_x + q\, u_y + r\, u = f \quad \text{in } \Omega\,, \tag{4.10}$$

with suitable boundary conditions. Here a, p, q, r, and f are functions of $x \in \Omega$, with $a = a(x)$ uniformly positive and $r = r(x)$ nonnegative in Ω. Discretization of problems

(4.9) and (4.10) by finite differences or finite elements leads to linear systems of the form $Ax = b$, with A positive real. For both problems (4.9) and (4.10), the symmetric part H of A has natural splittings $H = G + K$, where G represents a discretization of the second-order diffusion term and K stands for a (discrete) zeroth-order operator. Unsteady convection-diffusion-reaction problems combining features of both (4.9) and (4.10) may also be considered, but they are not essentially different from (4.9).

Both PHSS and PGHSS methods can be applied to the solution of the resulting linear systems. Assuming centered finite differences (with a sufficiently fine mesh) are used to discretize problems (4.9) and (4.10), it can be shown that the conditioning of matrix G behaves like h^{-2} for $h \to 0$. The same holds for many standard finite element schemes. The condition number of matrix $H = G + K$ arising from the discretization of (4.10) is also $O(h^{-2})$, since K becomes negligible relative to G when $h \to 0$. A "good" preconditioner P should be spectrally equivalent to $G + K$ or to G. Hence, PHSS and PGHSS require a preconditioner for a "nice" Helmholtz-like equation of type $-\mathrm{div}(a\nabla u) + ru = f$ or a preconditioner for the steady diffusion equation $-\mathrm{div}(a\nabla u) = f$. A multigrid or fast Poisson solver can be used in either case. In alternative, incomplete factorizations can be used, but these will not result in h-independent convergence of the inner iterations. A theoretical comparison of PHSS with PGHSS seems to be very difficult to carry out, also in view of the fact that many different choices are possible for preconditioning matrix P, which could be different for the two methods. If the same preconditioner P is used, the difference between the two approaches can be expected to disappear for $h \to 0$, since, in this limit, the difference between $P^{-1}G$ and $P^{-1}(G + K)$ vanishes. Note that the "size" of reaction term r will also play a role here; the larger r is, the smaller h must be taken before the difference between the two methods becomes negligible.

Next, the linear systems arising from (implicit) time and space discretizations of problem (4.10) are considered. Now G represents a discretization of the differential operator $-\mathrm{div}(a\nabla)$ so that the conditioning of G behaves like h^{-2} for $h \to 0$. On the other hand, K is typically of the form $M/\Delta t$, where M is a mass matrix and Δt denotes the time step. For many of the standard time discretization schemes, $\Delta t = O(h)$; therefore, the conditioning of $H = G + K$ behaves like h^{-1} for $h \to 0$. It is clear, however, that the difference between G and $G+K$ decreases as $h \to 0$. Again, a preconditioner P spectrally equivalent to $G + H$ can be implemented efficiently and can be expected to yield h-independent convergence rates.

Some numerical experiments illustrating the performance of PHSS and PGHSS on a specific example are discussed in the next section.

5. Numerical experiments. In this section, some numerical experiments are discussed, with the goal of showing the potential of the new schemes.

The first set of experiments concerns the generalized Stokes problem, i.e., problem (4.3)–(4.5) with $\mathbf{w} = \mathbf{0}$. In Table 5.1, numerical results are reported for flexible GMRES (FGMRES, see [21]) with an inexact variant of the GHSS preconditioner. The discrete saddle point problems were generated in this case by the marker and cell (MAC) finite difference discretization on a $40 \times 40 \times 40$ grid for different values of σ and ν. This discretization is known to be div-stable so that $C = 0$ in (4.6). Homogeneous Dirichlet boundary conditions were imposed on the velocities. Here $\Omega = [0, 1] \times [0, 1] \times [0, 1]$; the discrete problem has over 250,000 unknowns. The parameter α was set to 0.5, and a zero initial guess was used. The outer iteration was stopped when a reduction of the initial residual by six orders of magnitude was reached. For the inexact inner solves, the CG algorithm with incomplete Cholesky

TABLE 5.1
Iteration count for 3D generalized Stokes problem, inexact variant.

σ	$\nu = 0.1$	$\nu = 0.01$	$\nu = 0.001$	$\nu = 10^{-6}$
1	45	27	16	13
10	32	19	15	12
20	30	18	14	11
50	28	15	13	11
100	25	14	12	10

TABLE 5.2
Results for 3D generalized Stokes problem, $\nu = 0.001$, $\sigma = h^{-1}$.

Grid	n	m	Iterations	CPU time
$10 \times 10 \times 10$	2,700	1,000	12	0.42
$20 \times 20 \times 20$	22,800	8,000	12	4.66
$30 \times 30 \times 30$	78,300	27,000	12	20.97
$40 \times 40 \times 40$	187,200	64,000	13	66.02

preconditioning was employed throughout; the incomplete factorization used drop tolerance $\tau = 0.001$. The inner iterations were stopped as soon as a reduction of the initial residual by one order of magnitude was attained. This required only 1–2 PCG iterations per inner linear solve. The iteration counts, which can be shown to be largely independent of the grid size, improve for increasing σ and decreasing ν.

Table 5.2 shows timings (in seconds) for a generalized Stokes problem with $\nu = 0.001$ for different grids. Here $\sigma = h^{-1}$, where h denotes the grid size. GHSS preconditioning with $\alpha = 0.5$ is used. In the table, the dimensions n and m and the total number of FGMRES iterations are also indicated. The test runs were done on one processor of a SunFire V880 workstation with 8 CPUs and 16 GB of memory. These results show the good performance of the GHSS preconditioner. On this problem, the standard HSS preconditioner gives similar results.

The next experiments show that for more difficult problems, such as the Oseen problem, the GHSS approach can outperform the standard HSS preconditioner. The underlying application is the classical leaky-lid driven cavity problem. The matrices were generated using the IFISS Matlab toolbox [15]. The discretization was obtained using quadrilateral Q1-P0 elements on the unit square (uniform grid). Pressure stabilization was used, hence, $C \neq 0$. For a 32×32 mesh, the matrix dimensions were $n = 2{,}178$ and $m = 1{,}024$ for a total of 3,202 unknowns. The default values $\nu = 0.01$ and $\beta = 0.25$ were used for the viscosity and stabilization parameters, respectively. IFISS scales the resulting matrices so that the symmetric part of the saddle point matrix in (4.6) has norm 1 (using the infinity norm), while the skew-symmetric part has norm 0.25. To simulate a generalized Oseen problem, a matrix of the form σM was added to the (1,1) block, where $\sigma \approx h^{-1}$ (h denotes the mesh size) and M was the mass matrix for the velocity space. The matrix σM has norm 0.1.

Table 5.3 contains results for the HSS and GHSS preconditioners. The outer iteration was (full) GMRES, with a convergence tolerance of 10^{-6}. The symmetric and nonsymmetric linear systems arising at each application of the preconditioner were solved by PCG and by GMRES, respectively, with a rather tight convergence tolerance (also set to 10^{-6}). The inner PCG iteration was preconditioned by an incomplete Cholesky factorization, and the inner GMRES iteration was preconditioned by an incomplete LU factorization; in both cases, the drop tolerance was set to 0.01. The table reports the number of outer preconditioned GMRES iterations, the average

TABLE 5.3
Results for 2D generalized Oseen problem, $\nu = 0.01$.

Preconditioner	Outer its.	PCG/outer	GMRES/outer	Total inner
HSS($\alpha = 0.04$)	47	3	5	376
GHSS($\alpha = 0.01$)	25	2	4	150
HSS($\alpha = 0.01$)	67	2	12	938
GHSS($\alpha = 0.04$)	44	2	5	308

TABLE 5.4
Number of nonzeros in the incomplete factors.

Preconditioner	Incomplete Cholesky	Incomplete LU
HSS($\alpha = 0.04$)	8,438	84,214
GHSS($\alpha = 0.01$)	15,166	132,295
HSS($\alpha = 0.01$)	11,546	179,437
GHSS($\alpha = 0.04$)	11,546	82,221

number of inner PCG and ILU-GMRES iterations, and the total number of inner iterations. The values of α that result in the fastest convergence, determined experimentally using 10^{-2} increments, were found to be $\alpha = 0.04$ for HSS and $\alpha = 0.01$ for GHSS. Results for GHSS with the value $\alpha = 0.04$ and for HSS with the value $\alpha = 0.01$ are also included in Table 5.3 (bottom half), so as to allow a comparison of the two preconditioners for the same value of α. It is clear that, for this problem, the GHSS preconditioner is better, both in terms of rate of convergence and in terms of work per iteration. Even with a suboptimal value of α, the GHSS preconditioner can outperform the optimally tuned HSS preconditioner.

The results in Table 5.3 correspond to what are essentially exact variants of the two preconditioners. Substantial savings in overall work can be obtained by solving the subproblems arising at each iteration inexactly rather than exactly. Numerical experiments indicate that an inner convergence tolerance of 10^{-2} is generally enough to preserve the rate of convergence of the outer iteration. Of course, GMRES must now be replaced by flexible GMRES. With the inexact variants, the number of outer iterations is essentially unchanged from those reported in Table 5.3, while the average number of inner PCG and ILU-GMRES is reduced to about four (= 1 PCG +3 ILU-GMRES) for the HSS method and to about three (= 1 PCG +2 ILU-GMRES) for GHSS when the value $\alpha = 0.04$ is used. The total number of inner iterations is about 188 for inexact HSS preconditioning and about 132 for inexact GHSS preconditioning.

Of course, the total cost of the preconditioned iterations also depends on the amount of fill-in in the incomplete factors. This information is provided in Table 5.4, where the number of nonzeros in the incomplete Cholesky and ILU factors for the two methods is shown. The ILU factors of the nonsymmetric matrices arising from both HSS and GHSS preconditioning tend to incur rather high fill-in with the original ordering; the numbers reported in the table are for the reverse Cuthill–McKee ordering. In contrast, the incomplete Cholesky factorizations incurred modest amounts of fill-in, so the original ordering was used. Note that the system matrix itself contains 35,266 nonzero entries. The results in Table 5.4 show that when the value $\alpha = 0.04$ is used for both preconditioners, the overall storage requirements are very similar; shifting the mass matrix from one side to the other of the splitting leads to a somewhat higher fill-in in the incomplete Cholesky factor, which is nearly compensated by a corresponding decrease in the fill-in for the ILU factors. On the other hand, for this problem, the "optimal" value of α (in terms of number of outer iterations)

TABLE 5.5
PGHSS results for unsteady 2D convection-diffusion equation (5.1), $\Delta t = h$.

Grid	PGHSS its.	PCG/outer	GMRES/outer
16×16	6	21	8
32×32	5	18	8
64×64	6	12	7
128×128	5	9	8

TABLE 5.6
PHSS results for unsteady 2D convection-diffusion equation (5.1), $\Delta t = h$.

Grid	PHSS its.	PCG/outer	GMRES/outer
16×16	57	16	11
32×32	34	13	12
64×64	23	10	12
128×128	12	7	12

is smaller for GHSS than for HSS, and this translates to higher fill-in in the GHSS incomplete factors overall; see the second row of Table 5.4. This higher fill-in, however, is more than compensated by the considerably smaller number of total inner iterations required by the optimal GHSS method; cf. Table 5.3, second row.

We conclude this section with a few numerical experiments illustrating the solution of discretized convection-diffusion equations by means of the preconditioned GHSS method. The underlying equation is of the form

$$\frac{\partial u}{\partial t} - 100\,\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 100\,u_x + 100\,u_y = f \quad \text{in } \Omega \times [0,T], \tag{5.1}$$

with $\Omega = [0,1] \times [0,1]$ and with zero Dirichlet boundary conditions and suitable initial condition. A uniform Q1 finite element scheme with streamline-upwind Petrov–Galerkin (SUPG) stabilization [15, 16] is used for the spatial discretization. Discretization in time is achieved by a simple backward Euler method with time step $\Delta t = h$, where h is the mesh size. This leads to linear systems of the form $Ax = b$, where $A = H + S = (G + K) + S$. Here $G = G^T$ is the discretization of the anisotropic Laplacian, $S = -S^T$ is the discretization of the first-order terms, and $K = K^T = \sigma M$, where $\sigma = h^{-1}$ and M is the mass matrix. A good preconditioner for this problem is $P = L + K$, where L is the isotropic Neumann Laplacian. As discussed in [12], the parameter α can be simply chosen to be 1.

Results for the PGHSS method using a sequence of grids are given in Table 5.5. The corresponding linear systems range in size from 289×289 to $16{,}641 \times 16{,}641$; SUPG stabilization is needed only for the two coarser meshes. The table reports the number of (stationary) PGHSS iterations needed to reduce the initial residual by six orders of magnitude, the average number of inner PCG iterations (with preconditioner P), and the average number of GMRES iterations (also with preconditioner P). The inner systems are solved to a relative residual tolerance of 10^{-7}. The expected h-independent behavior of the outer iteration is clear from these results. The inner preconditioned GMRES iteration is also h-independent, whereas the number of inner PCG iterations actually decreases as the mesh is refined.

For completeness, results for the PHSS iteration are given in Table 5.6, using the same preconditioner P. The number of PHSS iterations, rather than being constant, decreases as $h \to 0$. For coarse grid problems, the number of iterations is much

higher than for PGHSS, but the difference between the two methods decreases as the mesh is refined. The theory predicts that asymptotically as $h \to 0$, the number of iterations for PHSS should settle around a constant value, independent of h [12]; although numerical experiments with very fine meshes could not be performed due to storage limitations, it is likely that the number of iterations in the limit should be about 6, the same as for PGHSS, since for h sufficiently small, the difference between G and $H = G + K$ becomes negligible.

Note the behavior of the inner PCG and GMRES iterations for PHSS. While the number of inner PCG iterations per outer iteration is somewhat smaller compared to PGHSS, the number of inner GMRES iterations is somewhat higher. This is also as expected, since the SPD systems arising from HSS are somewhat better conditioned than those for GHSS, while the situation is the opposite for the nonsymmetric systems.

Finally, it should be observed that the preconditioner $P = L + K$ is inexpensive to apply. On the finer grid, it was found that just 3–4 iterations of PCG (with preconditioner K) are enough to solve linear systems involving P to a relative residual tolerance of 10^{-7}.

Although the PGHSS method appears to be superior to GHSS for this particular test problem, it is difficult to compare the two approaches in general. The behavior of the two methods is obviously problem dependent, and it can be expected to be different for different choices of preconditioner P. Further work is needed to better understand when one method can be expected to outperform the other. However, it would seem that PGHSS has excellent potential for solving linear systems arising from time-dependent problems.

6. Conclusions. In this paper, a generalization of the HSS splitting method of Bai, Golub, and Ng has been described. The exact version of the new scheme has been shown to be unconditionally convergent. Similar to the PHSS method, the convergence rate of the GHSS scheme can be greatly improved by preconditioning. The new method can also be accelerated by a (flexible) Krylov subspace method with inexact inner solves. The new scheme has the advantage of replacing the solution of the shifted skew-symmetric system in the HSS method by the solution of an easier (i.e., more diagonally dominant, better conditioned) system. Numerical tests with unsteady incompressible flow problems and with convection-diffusion equations have been presented showing the effectiveness of the new approach. The proposed preconditioner should also be applicable, in principle, to other types of equations, including time-dependent problems in electromagnetics and systems of diffusion-convection-reaction equations.

Acknowledgments. I would like to thank the anonymous referees for helpful suggestions and Jia Liu for writing the Matlab code used in the numerical experiments for the generalized Stokes problem.

REFERENCES

[1] 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), pp. 1–23.

[2] 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), pp. 583–603.

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

[4] Z. Z. Bai, G. H. Golub, L.-Z. Lu, and J.-F. Yin, *Block triangular and skew-Hermitian splitting methods for positive-definite linear systems*, SIAM J. Sci. Comput., 26 (2005), pp. 844–863.
[5] Z. Z. Bai, G. H. Golub, and M. K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.
[6] Z. Z. Bai, G. H. Golub, and J. Y. Pan, *Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems*, Numer. Math., 98 (2004), pp. 1–32.
[7] M. Benzi and D. Bertaccini, *Block preconditioning of real-valued iterative algorithms for complex linear systems*, IMA J. Numer. Anal., 28 (2008), pp. 598–618.
[8] M. Benzi, M. J. Gander and G. H. Golub, *Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems*, BIT, 43 (2003), pp. 881–900.
[9] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
[10] M. Benzi and J. Liu, *An efficient solver for the incompressible Navier–Stokes equations in rotation form*, SIAM J. Sci. Comput., 29 (2007), pp. 1959–1981.
[11] M. Benzi and M. K. Ng, *Preconditioned iterative methods for weighted Toeplitz least squares problems*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1106–1124.
[12] D. Bertaccini, G. H. Golub, S. Serra-Capizzano, and C. Tablino-Possio, *Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation*, Numer. Math., 99 (2005), pp. 441–484.
[13] L. C. Chan, M. K. Ng, and N. K. Tsing, *Spectral analysis for HSS preconditioners*, Numer. Math. Theor. Methods Appl., 15 (2006), pp. 1–18.
[14] R. H. Chan, C. Greif, and D. P. O'Leary (Eds.), *Milestones in Matrix Computations. The Selected Works of Gene H. Golub With Commentaries*, Oxford Science Publications, Oxford University Press, Oxford, UK, 2007.
[15] H. C. Elman, A. Ramage, and D. J. Silvester, *Algorithm* 866: *IFISS, a Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Software, 33 (2007), article 14.
[16] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Numer. Math. Sci. Comput., Oxford University Press, Oxford, UK, 2005.
[17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[18] J. Liu, *Preconditioned Krylov Subspace Methods for Incompressible Flow Problems*, Ph.D. thesis, Department of Mathematics and Computer Science, Emory University, Atlanta, 2006; also available online at http://www.uwf.edu/jliu/research.htm.
[19] G. I. Marchuk, *Methods of Numerical Mathematics*, Springer-Verlag, New York, 1984.
[20] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, Berlin, 1994.
[21] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*, SIAM, Philadelphia, 2003.
[22] V. Simoncini and M. Benzi, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 377–389.