Technical Report

TR-2006-015

Real-Valued Iterative Algorithms for Complex Symmetric Linear Systems

by

Michele Benzi, Daniele Bertaccini

Mathematics and Computer Science

EMORY UNIVERSITY

REAL-VALUED ITERATIVE ALGORITHMS FOR COMPLEX SYMMETRIC LINEAR SYSTEMS*

MICHELE BENZI† AND DANIELE BERTACCINI‡

Abstract. We consider real-valued preconditioned Krylov subspace methods for the solution of complex linear systems, with an emphasis on symmetric (non-Hermitian) problems. Different choices of the real equivalent formulation are discussed, as well as different types of block preconditioners for Krylov subspace methods. Numerical experiments illustrating the performance of the various approaches are presented.

1. Introduction. In this paper we consider nonsingular linear systems of equations of the form

$$Cz = d, \qquad \text{with} \quad C = A + \mathrm{i}B, \quad z = x + \mathrm{i}y, \quad \text{and} \quad d = b + \mathrm{i}c. \tag{1.1}$$

Here $\mathrm{i} = \sqrt{-1}$, the $n \times n$ matrices A and B are real, and the vectors x, y, b, and c are all in $\mathbb{R}^n$. We are especially interested in the symmetric case: $A = A^T \neq O$ and $B = B^T \neq O$. Note that these conditions rule out the Hermitian and skew-Hermitian cases. Complex symmetric systems arise in a number of applications, including wave propagation (Helmholtz equation), diffuse optical tomography, quantum mechanics (Schrödinger equation), electromagnetism (Maxwell's equations), structural dynamics (frequency response analysis of mechanical systems), circuit analysis, and quantum chromodynamics (lattice QCD). Complex (nonsymmetric) linear systems may also arise from the use of certain shift-and-invert eigenvalue algorithms.

In all these applications the matrices are large, and frequently sparse; hence, preconditioned Krylov subspace methods are a natural choice. There are two possibilities: either tackling the $n \times n$ system (1.1) directly, or working with one of several $2n \times 2n$ equivalent real formulations. Each possibility has its own advantages and disadvantages. In this paper we focus on the second approach. In particular, we claim that the use of a real equivalent formulation is especially viable when either A or B is (semi-)definite, a condition frequently satisfied in practice.

The paper is organized as follows. Section 2 contains some background and motivation for the use of real equivalent formulations. Spectral properties of complex symmetric matrices and their equivalent formulations, with special emphasis on half-plane conditions for the eigenvalues, are briefly reviewed in section 3. Section 4 reviews previous work. Section 5, which constitutes the core of the paper, describes some preconditioners for real formulations. The nonsymmetric case is briefly examined in section 6, while section 7 is devoted to numerical experiments. We present our conclusions in section 8.

2. Background and motivation. We begin this section with some simple observations on various equivalent formulations of a complex linear system. In the complex formulation (1.1), matrices A and B are the real and imaginary part of C, respectively. However, multiplying both sides of (1.1) by i leads to the equivalent

*Manuscript date: October 13, 2006. This work was partially supported by NSF grants DMS-0207599 and DMS-0511336, by INdAM-GNCS, and by MIUR grant 2002014121.

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

‡Dipartimento di Matematica "Guido Castelnuovo", Università di Roma "La Sapienza", 00185 Roma, Italy (bertaccini@mat.uniroma1.it).

system

$$(-B + \mathrm{i}A)(x + \mathrm{i}y) = -c + \mathrm{i}b, \qquad \text{or} \qquad (B - \mathrm{i}A)(x + \mathrm{i}y) = c - \mathrm{i}b. \tag{2.1}$$

Hence, the choice of the real and imaginary part of the coefficient matrix is largely arbitrary. Thus, if we need to assume that the real or the imaginary part of C enjoys a particular property, such as being (semi-)definite, we can always assume that A has the required property. Furthermore, if we denote by C^H the Hermitian conjugate of C, problem (1.1) is equivalent to the system of normal equations

$$C^H C z = C^H d. \tag{2.2}$$

The matrix of this system is Hermitian positive definite (regardless of whether C is symmetric or not). However, the squaring of the condition number makes this approach somewhat unappealing.

As is well-known, the complex $n \times n$ linear system (1.1) also admits several $2n \times 2n$ real equivalent formulations. We begin with

$$\begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}, \qquad \text{or} \quad Mu = d. \tag{2.3}$$

This is in turn equivalent to

$$\begin{pmatrix} A & B \\ B & -A \end{pmatrix} \begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}, \tag{2.4}$$

which is symmetric if A and B are. Other equivalent real-valued forms, which may be regarded as corresponding to the second equation in (2.1), are

$$\begin{pmatrix} B & A \\ -A & B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ -b \end{pmatrix} \tag{2.5}$$

and

$$\begin{pmatrix} B & A \\ A & -B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}. \tag{2.6}$$

Again, the latter system is symmetric if A and B are. One possible disadvantage of the real formulations above is the loss of structure. Consider for instance the case where C is banded with a narrow bandwidth. Then A and B must also have such form, but none of the above real equivalent forms does. This means that a banded direct solver would perform well on the complex form, but poorly on any of the real equivalent ones. Also, the ordering of the real equivalent formulations (2.3)-(2.6) is a poor one from the point of view of other "global" techniques, such as incomplete factorizations. In this case it may be better to consider real equivalent formulations that preserve the structure of C. As described by Day and Heroux in [13], this can be achieved by simply rewriting each entry in the original $n \times n$ complex matrix as an equivalent 2×2 real matrix (and performing analogous replacements in the unknown and right-hand side vectors). For instance, if $c_{ij} = a_{ij} + \mathrm{i}\, b_{ij} \neq 0$ denotes the (i, j)-entry of C, then Day and Heroux explicitly form the real $2n \times 2n$ matrix K whose entries are 2×2 real blocks of the form

$$\begin{pmatrix} a_{ij} & -b_{ij} \\ b_{ij} & a_{ij} \end{pmatrix}.$$

The resulting matrix is just a symmetric permutation of M in (2.3), and the eigenvalues of K are the same as those of M. Day and Heroux call this the *K-formulation*. For instance, let

$$C = \begin{pmatrix} c_{11} & 0 & c_{13} & 0 & c_{15} \\ 0 & c_{22} & c_{23} & 0 & 0 \\ c_{31} & 0 & c_{33} & c_{34} & 0 \\ 0 & 0 & c_{43} & c_{44} & 0 \\ c_{51} & 0 & 0 & 0 & c_{55} \end{pmatrix}, \qquad \text{where} \quad c_{ij} = a_{ij} + \mathrm{i}\, b_{ij}.$$

Then the coefficient matrix of the corresponding K-formulation is

$$K = \left(\begin{array}{cc|cc|cc|cc|cc} a_{11} & -b_{11} & 0 & 0 & a_{13} & -b_{13} & 0 & 0 & a_{15} & -b_{15} \\ b_{11} & a_{11} & 0 & 0 & b_{13} & a_{13} & 0 & 0 & b_{15} & a_{15} \\ \hline 0 & 0 & a_{22} & -b_{22} & a_{23} & -b_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{22} & a_{22} & b_{23} & a_{23} & 0 & 0 & 0 & 0 \\ \hline a_{31} & -b_{31} & 0 & 0 & a_{33} & -b_{33} & a_{34} & -b_{34} & 0 & 0 \\ b_{31} & a_{31} & 0 & 0 & b_{33} & a_{33} & b_{34} & a_{34} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & a_{43} & -b_{43} & a_{44} & -b_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{43} & a_{43} & b_{44} & a_{44} & 0 & 0 \\ \hline a_{51} & -b_{51} & 0 & 0 & 0 & 0 & 0 & 0 & a_{55} & -b_{55} \\ b_{51} & a_{51} & 0 & 0 & 0 & 0 & 0 & 0 & b_{55} & a_{55} \end{array}\right).$$

We note that in some cases it may be possible to avoid storing each entry a_{ij}, b_{ij} twice. For instance, incomplete factorizations of K can be implemented by generating only one row (or column) of K at a time, which avoids doubling the storage for the coefficient matrix. However, the incomplete factors are $2n \times 2n$ matrices and for most dropping strategies the amount of storage required for the incomplete factors will be generally higher than for the original $n \times n$ complex form.

Thus, a natural question is which form of the equations to use when solving a complex linear system. Working with the original linear system (1.1) requires using complex arithmetic. There are several reasons why one may wish to avoid this. For instance, it is often the case that most of the entries of C are real, with the nonreal entries localized in just a few positions, typically on the main diagonal; see, e.g., [9, 19, 21, 31]. Using complex arithmetic throughout the code would be wasteful. For instance, when using Krylov subspace methods, the matrix C is used only in the form of matrix-vector products, and savings in the implementation may be possible by keeping the (relatively few) nonreal entries of C separate from the real ones. However, preconditioning is almost always mandatory, and many standard preconditioning techniques will have the undesirable effect of "spreading" nonreal entries to most positions in the preconditioning matrix. This will be the case, for instance, for standard incomplete factorization and sparse approximate inverse preconditioners. Simpler preconditioners, like diagonal or SSOR preconditioning (or polynomial preconditioners) do not suffer from this problem, but they are often not effective enough for complex linear systems of practical interest. Better options are available when C is an imaginary perturbation of a symmetric positive definite (SPD) matrix, e.g., $C = A + \mathrm{i}B$ with A SPD and B diagonal with small norm or of low rank. In this case it is sometimes possible to "update" a preconditioner for A in order to obtain a preconditioner for C; see [9], where good results are reported for certain differential equation problems. Nevertheless, preconditioning complex symmetric matrices remains problematic. Another difficulty, mentioned in [13], is the scarcity of available

preconditioning software supporting complex arithmetic, at least compared to the widespread availability of high-quality packages for the real case.

Because of all these reasons, it makes sense to investigate the iterative solution of complex linear systems using one of the real equivalent formulations. It was shown in [13] that nonsymmetric Krylov subspace methods preconditioned by standard incomplete LU (ILU) factorizations can perform reasonably well with the K-formulation of the equations; in particular, the performance was comparable with that obtained with ILU-preconditioned Krylov methods applied to the original complex form.

Motivated in part by [13], we further consider preconditioning techniques for real equivalent formulations of complex symmetric systems. Rather than applying general-purpose preconditioners to the K-formulation, however, we work with one of the "classical" real equivalent forms (such as (2.3)) with the goal of exploiting the block 2×2 structure. Much work has been done in recent years on preconditioning linear systems with block 2×2 structure, especially in the context of saddle point problems; see, e.g., [4, 8, 22, 29]. Here we propose to investigate the use of such "segregated" block preconditioners in the context of real equivalent formulations of complex linear systems.

We add as a further motivation that while every preconditioner for the complex form (1.1) has a real equivalent formulation, there are infinitely many choices of the preconditioner for the block 2×2 real equivalent formulation that do not have any complex equivalent. Thus, it is possible that the additional "degrees of freedom" available with any of the real equivalent formulations will make it possible to find more effective preconditioning techniques than for the original system in complex form.

3. Spectral properties. The spectrum of a general complex symmetric matrix has no special properties whatever: indeed, it can be shown that *any* square complex matrix is similar to a complex symmetric matrix, see [20, Theorem 4.4.9]. Thus, the eigenvalue distribution of a complex symmetric matrix need not exhibit any type of symmetry. The situation is different for the real equivalent formulations (2.3)–(2.6); this is obvious, since the eigenvalues of a real matrix come in complex conjugate pairs—the spectrum is always symmetric with respect to the real axis. More precisely, let $C = A + \mathrm{i}\, B$ have spectrum $\sigma(C) = \Lambda = \{\lambda_i \,|\, 1 \le i \le n\} \subset \mathbb{C}$. Let $\overline{\Lambda}$ denote the set of conjugates of the elements of Λ. The spectrum of the real equivalent form

$$M = \begin{pmatrix} A & -B \\ B & A \end{pmatrix},$$

is then $\sigma(M) = \Lambda \cup \overline{\Lambda}$.

For the symmetric forms (2.4) and (2.6) the eigenvalues are, of course, all real, but additional structure is present. Consider for example the real equivalent form (2.4). It can be shown that if λ is an eigenvalue, then $-\lambda$ is also an eigenvalue; therefore the spectrum is symmetric with respect to the origin. We refer to [17] for a detailed discussion of the spectral properties of the various real equivalent forms.

It is often the case that for complex linear systems that arise in practice, the eigenvalues lie in a half-plane which excludes the origin. In particular, many complex symmetric matrices satisfy a *half-plane condition* such as

$$\sigma(C) \subset \{\lambda \in \mathbb{C} \,|\, \mathrm{Re}\, \lambda \ge 0\} \qquad \text{or} \qquad \sigma(C) \subset \{\lambda \in \mathbb{C} \,|\, \mathrm{Im}\, \lambda \ge 0\}.$$

This will be the case, in particular, whenever A or B is SPD. In this case, the real equivalent form (2.3) (or (2.5)) will be positive definite, in the sense that its symmetric

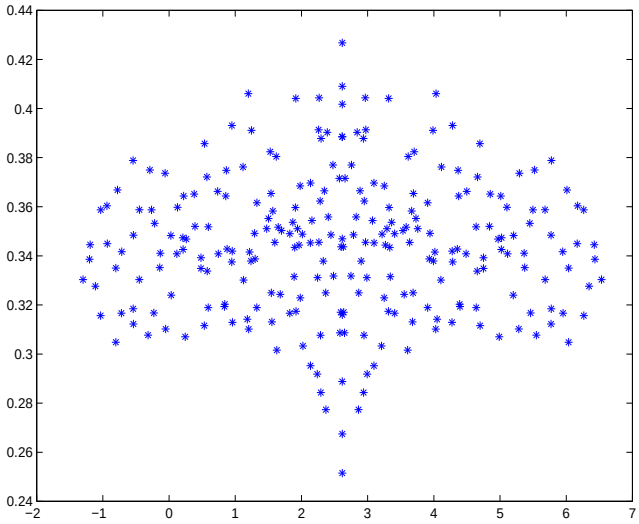


FIG. 3.1. *Eigenvalues of discrete Helmholtz operator on a 16×16 grid.*

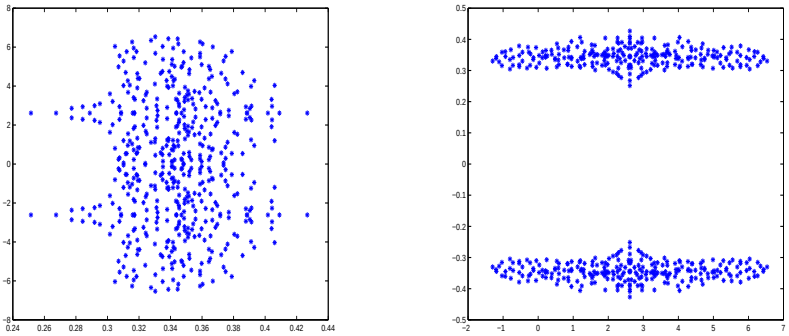
part will be SPD. Thus, eigenvalues of one of the real equivalent formulations (2.3) or (2.5) will also satisfy a half-plane condition. This is a desirable property for many Krylov subspace solvers, like GMRES [34]. Below we list some of the important properties of the real equivalent formulation (2.3) when matrix A in (1.1) is symmetric positive semidefinite:

(i) If B is nonsingular, M in (2.3) is nonsingular (sufficient condition only);
(ii) M is semipositive real, that is, $v^T M v \geq 0$ for all $v \in \mathbb{R}^{2n}$ and positive semistable (i.e., its eigenvalues have nonnegative real part);
(iii) If A is positive definite, then (regardless of B) M is nonsingular, positive real ($v^T M v > 0$ for all $v \neq 0$), and positive stable (all its eigenvalues have positive real part).

Of course, similar properties hold for the nonsymmetric real formulation (2.5) if B is symmetric positive (semi-)definite. In contrast, the symmetric forms (2.4) and (2.6) are necessarily indefinite. For this reason, whenever A (or B) is definite, we prefer to use the nonsymmetric, but positive definite form (2.3) (or (2.5)). Our goal is to find efficient block preconditioners that not only preserve positive-definiteness (whenever it is present), but also achieve a strong clustering of the spectrum away from 0.

As an example we consider the eigenvalues of a complex symmetric matrix $C = A + \mathrm{i}B$ obtained from a finite difference discretization of an Helmholtz-type problem on a 16×16 grid. Fig. 3.1 displays the eigenvalues of C. In this case A is indefinite while B is positive definite. Note that the eigenvalues all lie in the upper half-plane.

In Fig. 3.2(a)-(b) we plot the eigenvalues of the real equivalent formulations (2.5) and (2.3), respectively. Note that the matrix (2.5) is positive real, hence all its eigenvalues lie in the right-half plane. In contrast, the symmetric part of (2.3) is indefinite, and the eigenvalues of (2.3) surround the origin. This often results in very slow convergence of Krylov subspace methods.



(a) Positive real form (2.5) (b) Indefinite form (2.3)

FIG. 3.2. *Eigenvalues for real equivalent formulations of Helmholtz problem.*

4. Previous work. Until about 1990, the most common approaches for solving complex symmetric systems were to use preconditioned conjugate gradients (PCG) on the normal equations (2.2), or real-valued approaches using either nonsymmetric Krylov subspace methods applied to the nonsymmetric forms (2.3) and (2.5), or a symmetric solver like MINRES or SYMMLQ [30] applied to the symmetric forms (2.4) and (2.6). The latter require the preconditioner to be SPD. Krylov methods specifically developed for complex linear systems did not appear until the late Eighties and early Nineties; see, e.g., [11, 17, 23, 24, 26, 38], as well as the recent survey [35, Sections 12 and 13.3].

Influential work by Freund (especially [17]) indicated that the complex-valued formulation is generally better from the point of view of Krylov subspace convergence. As a result, real-valued formulations have generally been shunned. However, preconditioning was not considered in [17]. Of course, preconditioning changes the spectral properties of the matrices involved, and the possible disadvantages of the real formulations may disappear or at least be less of an issue.

Many papers have addressed the problem of finding effective preconditioners for complex linear systems. Besides special solvers developed with a particular application in mind, like for example those studied in [1, 14, 15, 19, 21, 31, 32], there has been some work on the use of general-purpose techniques such as SSOR, incomplete factorizations and sparse approximate inverses; see, e.g., [2, 12, 25, 27, 28]. In spite of much work in this area, it is fair to say that preconditioning complex symmetric matrices remains a challenge—even more so than in the real case.

For reasons already hinted at, in recent years there has been renewed interest in approaches based on real equivalent formulations, and particularly in preconditioning for such forms. Besides the already mentioned paper [13] we mention [3], where an efficient preconditioning method is introduced for problems where both A and B are symmetric positive semidefinite with at least one of the two positive definite. This situation is rather special. In practice it is much more common to find matrices where only one of A or B is (semi-)definite, with the other being strongly indefinite.

5. Preconditioners for real formulations. In this section we consider block preconditioners for the real equivalent formulations. We work primarily with the form (2.3). Ideally, we would like to find preconditioners that are well-defined under minimal assumptions on A and B, do not double storage requirements, are reasonably

efficient and robust, and are fairly straightforward to implement. Concerning storage, we insist that the only arrays of dimension $2n$ should be the vectors needed by the Krylov subspace method. Obviously, matrix-vector products with the coefficient matrix of (2.3) can be implemented using only a single copy of A and B; our preconditioners will only require the (approximate) solution of sparse real linear systems of order n. Therefore, no matrix of order $2n$ is actually ever explicitly formed by the preconditioned Krylov methods applied to (2.3). In other words, we use a "segregated" approach to preconditioning instead of a fully coupled one. This is one point where our approach differs from that taken in [13].

In some applications it happens that one of the two matrices A and B "dominates" the other. For instance, B may have small norm and/or rank compared to A. Another fairly typical situation is when A represents a differential operator (for instance, a discretized Laplacian) and B a bounded operator (such as a mass matrix) or, more generally, a differential operator of lower order than A. Then $A^{-1}B$ represents the discretization of a compact operator and its spectrum will be clustered around the origin in the complex plane; thus, the spectrum of $A^{-1}C = I + \mathrm{i}A^{-1}B$ will be clustered around $(1, 0)$. In this case a good approximation $\widehat{A}$ of A, assumed to be invertible, will be a reasonable choice as a preconditioner for $C = A + \mathrm{i}B$; equivalently, the block diagonal matrix

$$P_A = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{A} \end{pmatrix}, \tag{5.1}$$

can be expected to be a good preconditioner for (2.3).

If B is dominant, there are two possibilities. One is to exchange the roles of A and B (see (2.1) and to use a block preconditioner of the form (5.1) with $\widehat{B} \approx B$ replacing $\widehat{A}$. This may be inconvenient if B is large in norm but has a sizable null space, as it happens in some applications. In this case it may be better to precondition (2.3) with

$$P_\alpha = \begin{pmatrix} \alpha I & -\widehat{B} \\ \widehat{B} & \alpha I \end{pmatrix}, \tag{5.2}$$

where $\alpha > 0$ is a scalar that guarantees the invertibility of P_α. Note that this is a shifted skew-symmetric preconditioner. Linear systems of the form $P_\alpha z = r$ can be solved by a Schur complement reduction, leading to a linear system of order n with coefficient matrix of the form $B^2 + \alpha^2 I$. Note that this matrix is SPD if B is symmetric. Moreover, if B is scaled so that $\lambda_{\max}(B^2) = 1$, then the spectral condition number of $B^2 + \alpha^2 I$ satisfies $\kappa_2(B^2 + \alpha^2 I) \leq 1 + 1/\alpha^2$. If α is not too small then $B^2 + \alpha^2 I$ is well-conditioned and it may be easy to find approximate solutions to linear systems involving $B^2 + \alpha^2 I$, for instance using PCG. Shifted skew-symmetric preconditioning has been introduced, in a different context (convection-dominated flow problems), in [18]. Of course, preconditioning M with P_α is mathematically equivalent to preconditioning the complex matrix $C = A + \mathrm{i}B$ with $\alpha I + \mathrm{i}\widehat{B}$. Note that this is a shifted skew-Hermitian matrix.

The foregoing discussion suggests that simple choices are available if one of A or B strongly dominates (in some sense) the other matrix. Frequently, however, both A and B need to be taken into account if we want the preconditioner to be effective. If $\widehat{A} \approx A$ and $\widehat{B} \approx B$ are such that $\widehat{A} + \mathrm{i}\widehat{B}$ is easily invertible, then the latter can be used as a preconditioner for the complex system (1.1). This is of course equivalent

to preconditioning the real equivalent formulation (2.3) (say) with the block 2×2 preconditioner

$$P = \begin{pmatrix} \widehat{A} & -\widehat{B} \\ \widehat{B} & \widehat{A} \end{pmatrix}. \tag{5.3}$$

Another possible choice of a preconditioner that includes both A and B is given by the "alternating" Hermitian/skew-Hermitian splitting (HSS) preconditioner

$$P_{hss} = \begin{pmatrix} A + \alpha I & O \\ O & A + \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & -B \\ B & \alpha I \end{pmatrix}, \tag{5.4}$$

where $\alpha > 0$. HSS preconditioning is based on the splitting

$$M = \begin{pmatrix} A & -B \\ B & A \end{pmatrix} = \begin{pmatrix} A & O \\ O & A \end{pmatrix} + \begin{pmatrix} O & -B \\ B & O \end{pmatrix} = H + K. \tag{5.5}$$

The shift $\alpha > 0$ is used to make $H + \alpha I$ positive definite and $K + \alpha I$ invertible. The idea of the preconditioner (5.4) is to alternate between the (shifted) symmetric and skew-symmetric part of M. The HSS preconditioner is well-defined (invertible) for semidefinite A and for any B. It may be well-defined for more general problems as well. As long as A is symmetric positive semidefinite and B is symmetric, its application only requires the solution of SPD systems of order n. If A is positive definite we have $\varrho(I - P_{hss}^{-1} M) < 1$ for all $\alpha > 0$, where $\varrho(\cdot)$ denotes the spectral radius [6]. Moreover, taking

$$\alpha = \sqrt{\lambda_{\min}(A)\lambda_{\max}(A)}$$

minimizes an upper bound on the spectral radius. Also, the result on the spectral radius remains valid under the assumptions that A is positive semidefinite and B nonsingular [7]. In practice, A and B are often replaced by approximations $\widehat{A}$ and $\widehat{B}$. See, e.g., [5, 6, 7, 10] for detailed analyses and variants of this approach. Here we limit ourselves to the following heuristic observation. From the simple identity

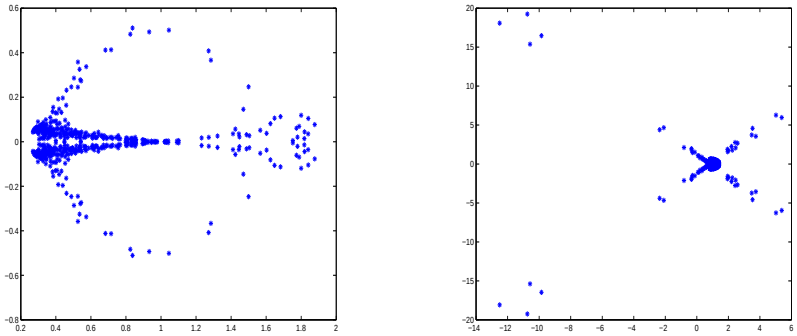
$$P_{hss} = (H + \alpha I)(K + \alpha I) = HK + \alpha M + \alpha^2 I$$

we can see that P_{hss} (with a small α) can be expected to be a good preconditioner if the product HK is small in norm. In the limiting case where $HK = O$ the preconditioner is just a multiple of $M + \alpha I$, which is close to M for α small, and the eigenvalues of $P_{hss}^{-1} M$ will be clustered away from zero. Since

$$HK = \begin{pmatrix} O & -AB \\ AB & O \end{pmatrix},$$

the preconditioner can be expected to work well if the product AB is small in norm. For instance, if A (or B) depends on a parameter, then HSS preconditioning may be a good choice for values of the parameter that make A (or B) small, if any.

In Fig. 5.1(a)-(b) we plot the eigenvalues of the real equivalent form of the discrete Helmholtz operator preconditioned with HSS. The left-hand plot displays the eigenvalues using the positive definite form (2.5) while the right-hand plot displays the eigenvalues using the indefinite form (2.6). We note that in the first case the eigenvalues lie inside the disk of radius 1 centered at $(1, 0)$, as predicted by the theory. On the



(a) Positive real form (2.5) (b) Indefinite form (2.3)

FIG. 5.1. *Eigenvalues for HSS preconditioning of real equivalent formulations of Helmholtz problem.*

other hand, the eigenvalues in the second plot surround the origin. Although the clustering seems stronger in the second plot, the convergence rate of HSS-preconditioned Krylov methods is much worse than for the other case.

All the types of preconditioner indicated so far have equivalent real and complex forms. Next we consider block preconditioners for the real formulation (2.3) that have no equivalent complex form. Clearly, any *block triangular* preconditioner of the form

$$\begin{pmatrix} \widehat{A} & -B \\ O & \widehat{S} \end{pmatrix}, \tag{5.6}$$

with $\widehat{A}$ and $\widehat{S}$ invertible, can be used as a preconditioner for (2.3) but there is no corresponding complex preconditioner for (1.1). The identity

$$\begin{pmatrix} \widehat{A} & -B \\ O & \widehat{S} \end{pmatrix}^{-1} = \begin{pmatrix} \widehat{A}^{-1} & O \\ O & I \end{pmatrix} \begin{pmatrix} I & B \\ O & I \end{pmatrix} \begin{pmatrix} I & O \\ O & \widehat{S}^{-1} \end{pmatrix} \tag{5.7}$$

shows that application of the preconditioner requires one application of $\hat{A}^{-1}$, one of $\widehat{S}^{-1}$, and one sparse matrix-vector multiply with B.

How should the matrix $\widehat{S}$ be chosen? To try to answer this question, we note that if A is invertible then M has the block LU factorization

$$M = \begin{pmatrix} A & -B \\ B & A \end{pmatrix} = \begin{pmatrix} I & O \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & -B \\ O & S \end{pmatrix},$$

where $S = A + BA^{-1}B$ denotes the Schur complement. Note that S is SPD if A is SPD and B is symmetric. It follows that letting

$$P = \begin{pmatrix} A & -B \\ O & S \end{pmatrix}, \tag{5.8}$$

we have $\sigma(MP^{-1}) = \sigma(P^{-1}M) = \{1\}$; furthermore, the minimum polynomial of the preconditioned matrix MP^{-1} (or $P^{-1}M$) has degree 2, and a Krylov method like GMRES is guaranteed to find the solution in at most two steps; see [22, 29].

In some situations (for instance, if A^{-1} is diagonal or block diagonal with small blocks) it may be possible to form the Schur complement S explicitly and to solve linear systems with S (at least approximately). In general, however, forming S and/or solving systems with it is out of the question. In this case it is necessary to use some approximation. The simplest approach is to let $\widehat{S} = \widehat{A}$. To see when such a choice may work reasonably well, consider the "exact" block preconditioner

$$P_t = \begin{pmatrix} A & -B \\ O & A \end{pmatrix}. \tag{5.9}$$

From the identity

$$P_t^{-1}M = \begin{pmatrix} A & -B \\ O & A \end{pmatrix}^{-1} \begin{pmatrix} A & -B \\ B & A \end{pmatrix} = \begin{pmatrix} I + (A^{-1}B)^2 & O \\ A^{-1}B & I \end{pmatrix}$$

one can see that, again, good results can be expected if the spectrum of $A^{-1}B$ is clustered around the origin, for in this case the preconditioned matrix $P_t^{-1}M$ (or MP_t^{-1}) will have most of its eigenvalues clustered around $(1, 0)$. In other words, this approach can be expected to work well when block diagonal preconditioning (with P_A) works well. Because of the squaring of $A^{-1}B$, block triangular preconditioning can be expected to provide a rate of convergence at least as fast as that obtained with block diagonal preconditioning when the latter works well; by the same token, block triangular preconditioning is likely to be much worse than block diagonal preconditioning when the latter performs poorly. Of course, this informal argument neglects important aspects such as the effects of nonnormality on the behavior of the preconditioned iteration, and therefore it is not to be taken too literally.

If $\widehat{A}$ is an approximation of A such that $\widehat{A}$ has a sparse inverse, then it may be possible to explicitly form the approximate Schur complement $\widehat{S} = \widehat{A} + B\widehat{A}^{-1}B$ (or even $\widehat{S} = A + B\widehat{A}^{-1}B$) for use in a block triangular preconditioner. For instance, if A is diagonally dominant then a diagonal approximation $\widehat{A} = \mathrm{diag}(A)$ may suffice. Also, in some cases it may be possible to construct (either explicitly or implicitly, in the form of some iteration) approximations to the inverse of the Schur complement, $S^{-1} = A^{-1}\left[I + (A^{-1}B)^2\right]^{-1}$. Clearly, there are many possible choices here—the best choice is highly problem-dependent, and a good understanding of the specific underlying application is required in order to derive a good approximation. It may also pay off to switch the roles of A and B in order to have a Schur complement that can be more easily approximated.

We emphasize that if either A or B is SPD, all preconditioners considered so far can be implemented in a way that only requires the solution of SPD systems of order n. This is also true if either one of A or B is symmetric positive semidefinite, for in this case we can always add a small diagonal perturbation to A (or to B) to make it SPD.

6. The nonsymmetric case. The nonsymmetric case has received scant attention in the literature, especially as far as the use of real equivalent formulations is concerned. Most of the block preconditioners discussed in the previous section can be extended to the nonsymmetric case, where at least one of A and B is nonsymmetric. In this case, of course, at least some of the subsystems to be solved when applying the preconditioners will not be symmetric, in general.

As far as the HSS preconditioner is concerned, rather than splitting M as the sum of $\frac{1}{2}(M + M^T)$ and $\frac{1}{2}(M - M^T)$ it is better to use the splitting $A = H + K$ given by

(5.5), even though H will no longer be symmetric (or K skew-symmetric). Similarly, it will no longer be true in general that the eigenvalues of the HSS-preconditioned matrix lie in the disk of radius 1 centered at $(1, 0)$, and furthermore the matrices $H + \alpha I$ and $K + \alpha I$ are no longer guaranteed to be invertible. Nevertheless, the preconditioner may still be well-defined in practice; see section 7.2.

7. Numerical experiments. In this section we present the results of some numerical tests obtained with a MATLAB 7.1.0 implementation of nonsymmetric Krylov subspace methods with the block preconditioners discussed in section 5. We consider real equivalent formulations of complex symmetric problems arising from the discretization of different types of Helmholtz equations, similar to those considered in [17]. Another set of experiments concerns linear systems arising in molecular dynamics (see [13]). Finally, we include a few tests on linear systems arising in frequency analysis of mechanical systems (see [16]).

In all the experiments the initial guess used is the zero vector. The stopping criterion is $\|r_k\|_2 < 10^{-6}\|r_0\|_2$, where r_k denotes the true residual after k iterations. We experiment with full GMRES [34], flexible GMRES (FGMRES [33]), BiCGSTAB [37], and BiCGSTAB(2) [36]. Right preconditioning is used in all cases.

7.1. Helmholtz-type equations. Helmholtz equations are of fundamental importance in the modeling of wave propagation phenomena. Similar to [17], we consider two types of model problems based on the finite-difference discretization of the partial differential equation

$$-\nabla \cdot (c\,\nabla u) + \sigma_1 u + \mathrm{i}\sigma_2 u = f, \tag{7.1}$$

where the coefficients c, σ_1, and σ_2 are real-valued functions. The above equation, supplemented by appropriate boundary conditions, is used to describe the propagation of damped time-harmonic waves. We consider (7.1) on the two-dimensional domain $\Omega = [0, 1] \times [0, 1]$ with different values of σ_1, σ_2 and boundary conditions. For the "diffusivity" coefficient c we used both the constant value $c = 1$ and the discontinuous function defined as follows:

$$c(x, y) = 10,\ (x, y) \in [0.25, 0.75] \times [0.25, 0.75]; \quad c(x, y) = 1 \text{ otherwise.}$$

The first problem is a complex Helmholtz equation with σ_2 strictly positive and $\sigma_1 < 0$, so as to make the real part A of the discrete operator $C = A+\mathrm{i}B$ indefinite and the imaginary part B definite. Note that B is diagonal. The boundary conditions are of Dirichlet type. The second problem is a real Helmholtz equation ($\sigma_2 = 0$) with $\sigma_1 > 0$ and complex boundary conditions (so-called *first-order Sommerfeld conditions*). In this case the real part A of C is positive definite while the imaginary part B is indefinite and rank deficient. The problems were discretized by finite differences using uniform $m \times m$ grids with mesh size $h = 1/(m + 1)$ where m ranges from $m = 32$ to $m = 512$. The size of the corresponding complex linear systems is $n \times n$ with $n = m^2$ ranging from $1,024$ to $262,144$. Right-hand sides were generated at random with components uniformly distributed in $[-1, 1] + \mathrm{i}[-1, 1]$. We remark that for all these problems, Krylov subspace methods either diverge or converge extremely slowly in the absence of preconditioning. Additional details follow.

Problem 1 (Complex Helmholtz equation). The finite difference approximation results in complex symmetric linear systems of the form

$$Ax = b, \quad A = T - h^2\sigma_1 I + \mathrm{i}h^2 D, \quad D = \mathrm{diag}(d_1, ..., d_n), \tag{7.2}$$

α	HSS (GM)	HSS (BiCG)	P_α (GM)	P_α (BiCG)
0.001	38	86	37	75*
0.005	37	44	34	43
0.01	34	31	33	30
0.02	29	21	29	21
0.03	25	16	25	17
0.05	17	9	17	10
0.1	8	4	8	4
0.5	23	13	23	13

TABLE 7.1

Problem 1: Results for real equivalent form (2.5) of complex Helmholtz equation on 32×32 *grid. Number of iterations with HSS and with shifted skew-symmetric preconditioning for different values of* α*. GM=GMRES; BiCG=BiCGSTAB. The* * *indicates early termination of BiCGSTAB.*

where T is the discretization of the diffusion operator $-\nabla \cdot (c\,\nabla u)$, normalized so as to have $\lambda_{\max}(T) = O(1)$, $\lambda_{\min}(T) = O(h^{-2})$ for $h \to 0$. The diagonal entries of D are the values of σ_2 at the grid points. The coefficient σ_1 is chosen so that the real part of the matrix is highly indefinite. For instance, $\sigma_1 \cdot h^2 \approx 1.025$ for $h = 1/33$ (i.e., $m = 32$). The diagonal matrix D has random entries in the interval $(0, 200)$. With this setting of the coefficients, we have (for $h = 1/33$)

$$\|A\|_\infty \approx 78.9, \quad \|B\|_\infty \approx 0.183,$$

making the real part A strongly dominant. Since A is indefinite and B is positive definite, we work with the real equivalent formulation (2.5). Results for the HSS and (shifted) skew-symmetric preconditioners,

$$P_{hss} = \begin{pmatrix} B+\alpha I & O \\ O & B+\alpha I \end{pmatrix} \begin{pmatrix} \alpha I & A \\ -A & \alpha I \end{pmatrix} \quad \text{and} \quad P_\alpha = \begin{pmatrix} \alpha I & A \\ -A & \alpha I \end{pmatrix}, \tag{7.3}$$

with different values of the shift $\alpha > 0$ are reported in Table 7.1 for the 32×32 grid. We then performed experiments on a sequence of increasingly finer grids (up to 256×256) with the fixed value of the parameter $\alpha = 0.1$ and always found that HSS and P_α result in 8 iterations with GMRES. Note that both preconditioners require the solution of SPD linear systems of the form $(A^2 + \alpha^2 I)u = r$ where $A^2 = (T - h^2\sigma_1 I)^2$ can be interpreted as the discretization of a *fourth-order* differential operator. At first sight it may seem strange to transform the initial (complex) second-order PDE into a fourth-order, albeit real, one. It is important to observe, however, that if α is kept constant (as we do here), the condition number of $A^2 + \alpha^2 I$ is uniformly bounded with respect to h. Here we use the Cholesky factorization method to perform "exact" solves; in practice, for h small it is better to use inexact solves obtained with some inner iteration like PCG (using a flexible method like FGMRES for the outer iteration). It is clear from our results that for this problem HSS has no advantage over the shifted skew-symmetric preconditioner, which is perhaps not surprising in view of the fact that B has very small norm relative to A. Likewise, block diagonal or block triangular preconditioners with B on the main diagonal are not effective. Using block diagonal or block triangular preconditioners with A on the main diagonal with the formulation (2.3) is more effective, but requires solving linear systems with the highly ill-conditioned and indefinite matrix A, which is exceedingly expensive.

We also note that the best results are obtained for $\alpha = 0.1$. Furthermore, we note that BiCGSTAB is generally inferior to GMRES for sub-optimal choices of α. Nevertheless, BiCGSTAB has much lower storage requirements than GMRES and this makes it advantageous for large problems. Similar results were obtained with BiCGSTAB(2).

Next, we experiment with block triangular preconditioners of the form

$$\begin{pmatrix} B & A \\ O & \widehat{S} \end{pmatrix}, \tag{7.4}$$

where $\widehat{S}$ is a computationally viable approximation to the Schur complement $S = B + AB^{-1}A$. Note that S is SPD since B is. Forming the Schur complement explicitly is possible since B is diagonal and A is a sparse matrix. Using (7.4) with $\widehat{S} = S$ results in convergence in exactly 2 iterations of GMRES, but it is expensive owing to the high cost of solving linear systems with S. We experimented with an inexact variant using FGMRES as the outer iteration and PCG with incomplete Cholesky preconditioning for the approximate solution of linear systems involving the Schur complement S. We found that using an inner PCG relative residual tolerance of just 10^{-1} resulted in convergence in 7-8 FGMRES iterations, independent of the mesh size h. The total number of inner PCG iterations, however, tends to increase as the mesh is refined. In order to keep the total number of inner PCG iterations from growing it is necessary to reduce the drop tolerance in the incomplete Cholesky factorization (i.e., to accept more fill-ins in the incomplete factor). Here we found that performance is improved if S is first normalized so that its largest diagonal entry is equal to 1, and a small shift ($\approx 10^{-3}$) is added to the diagonal of S prior to computing the incomplete factorization. The shift helps in reducing fill-in and making the incomplete factorization process more stable.

Problem 2 (Real Helmholtz equation with complex boundary conditions). Here $\sigma_2 = 0$ and we imposed a Sommerfeld boundary condition of the form

$$\frac{\partial u}{\partial n} = \mathrm{i}\sigma_2 u \quad \text{on} \quad \{(1, y) \mid 0 < y < 1\}$$

(discretized with forward differences) and Dirichlet boundary conditions on the remaining three sides of the unit square. The resulting linear system is again of the form (7.2) where the diagonal entries of D are given by $d_i = (-1)^s \cdot 100/h$ if i/m is an integer, 0 otherwise, while the sign changes from one corner to the other. Also, we set $\sigma_1 = 200$. Now the real part of the matrix $A + \mathrm{i}B$ is positive definite and diagonally dominant while the imaginary part is indefinite and singular. With this setting, we have (case $m = 32$)

$$\|A\|_\infty \approx 80.2, \quad \|B\|_\infty \approx 3.03.$$

Therefore, we choose to work with the real equivalent form (2.3). Additional numerical tests have been performed with $d_r = (-1)^s \cdot 2000/h$, resulting in matrices A and B having norms of the same order of magnitude:

$$\|A\|_\infty \approx 80.2, \quad \|B\|_\infty \approx 60.6.$$

In the case of the first of these two test problems we found that the shifted skew-symmetric preconditioner P_α results in very slow convergence for all values of α. This

α	HSS (GM)	HSS (BiCG)
0.001	32	70
0.005	33	43
0.01	33	59
0.05	33	26
0.1	32	26
0.5	24	16
1.0	20	11
2.0	21	13
5.0	30	22

TABLE 7.2

Problem 2: Results for real equivalent form (2.3) of Helmholtz equation with complex boundary conditions on 32×32 *grid (indefinite imaginary part; definite real part). Number of iterations with HSS preconditioning for different values of* α*. GM=GMRES; BiCG=BiCGSTAB.*

α	HSS (GM)	HSS (BiCG)
0.001	28	44
0.005	29	28
0.01	29	21
0.05	29	18
0.1	30	19
0.5	23	16
1.0	18	12
2.0	20	12
5.0	30	22

TABLE 7.3

Helmholtz equation model problem 2, same as in Table 7.2 but with the entries of D multiplied by 20*.*

is to be expected since the norm of A is much larger than that of B. In the second case P_α performed much better, but not as well as HSS. This is, again, not surprising since both A and B are of significant size. In Tables 7.2-7.3 we show results obtained with the HSS preconditioner for various values of α. For this problem, $\alpha = 1.0$ appears to be optimal in terms of the number of iterations required to converge.

In Table 7.4 we show results for three fine grid discretizations of the same problem of Table 7.3. We report the number of BiCGSTAB iterations with HSS preconditioning for different values of α. Now the best value of α depends on h, with $\alpha = 0.1$ being optimal for the finest grid. Note, however, that the number of preconditioned iterations is not h-independent. In Table 7.4 we also report elapsed times using one processor of an AMD Opteron. The superlinear scaling of the timings is due in part to the growth in the number of iterations, and in part to the sparse Cholesky factorizations used to solve the linear systems arising from the application of the preconditioner: better scaling (per iteration) may be obtained using multigrid.

For this problem, however, we found that both block diagonal (5.1) and block triangular (5.9) preconditioning result in h-independent convergence rates. Using exact solves, we found these two preconditioners to require about 50 and 32 BiCGSTAB iterations, respectively. This can be easily explained observing that the preconditioned matrix corresponds to a compact perturbation of the identity operator.

α	128×128 grid	256×256 grid	512×512 grid
0.001	111 (20.4)	169 (158.)	133 (638.)
0.005	56 (10.3)	78 (72.6)	121 (580.)
0.01	51 (9.37)	68 (63.6)	83 (399.)
0.02	49 (8.99)	65 (60.7)	87 (417.)
0.03	43 (7.97)	52 (49.1)	61 (296.)
0.05	38 (6.96)	50 (47.2)	54 (262.)
0.1	36 (6.70)	40 (37.4)	52 (248.)
0.2	24 (4.48)	33 (31.3)	60 (290.)
0.3	22 (4.13)	34 (32.2)	77 (372.)
0.5	26 (4.78)	42 (39.3)	107 (511.)
1.0	36 (6.62)	59 (55.1)	140 (673.)

TABLE 7.4

Helmholtz equation model problem 2, same as in Table 7.3 but with finer grids. BiCGSTAB iterations for different values of α. Elapsed times (in seconds) are given in parentheses.

7.2. Molecular dynamics. Here we consider two instances of a computational chemistry model problem proposed by Sherry Li of NERSC and also used in [13]. These problems, denoted M3D2 and M4D2, are nonsymmetric. The eigenvalues of M3D2 verify the half-plane condition $\mathrm{Im}(\lambda) > 0$, but neither the real part A or the imaginary part B is definite (i.e., they have indefinite symmetric part). The matrices have been normalized so that $\max\{\|H\|_\infty, \|K\|_\infty\} = 1$, where H and K denote the symmetric and skew-symmetric part of the real equivalent form (2.3), respectively. Upon this normalization, the relative size of A and B is given by

$$\|A\|_\infty \approx 0.49, \quad \|B\|_\infty \approx 2.26$$

for M3D2 and by

$$\|A\|_\infty \approx 0.35, \quad \|B\|_\infty \approx 2.23$$

for M4D2. We considered the real equivalent form (2.5) with the dominant block B on the main diagonal. For both M3D2 and M4D2 we found that preconditioning is essential in order to achieve convergence in less than 300 iterations of GMRES or BiCGSTAB. Block diagonal and block triangular preconditioners with the B on the main diagonal (inverted exactly) resulted in full GMRES converging in 216 and 113 iterations, respectively, for M3D2; for M4D2 these preconditioners proved insufficient to achieve convergence within 300 iterations. Being unfamiliar with the details of this application, we did not attempt to construct block triangular preconditioners based on approximate Schur complements.

In Tables 7.5-7.6 we report results obtained with the analogues of the HSS and shifted skew-symmetric preconditioners (7.3). Since A and B are nonsymmetric, these are not truly HSS and shifted skew-symmetric preconditioners. However, in both cases the preconditioners turned out to be well-defined (i.e., nonsingular). Sparse LU factorization was used to solve linear systems with matrices $B + \alpha I$ and $A^2 + \alpha^2 I$.

For the first of these two problems we can see that the HSS-like preconditioner (5.4) results in faster convergence than the preconditioner (5.2). We also note that now $\alpha = 0.01$ is optimal (or nearly so) for both preconditioners. For the second problem the rate of convergence is about the same, and unfortunately not very good, with either approach, regardless of α.

α	HSS (GM)	P_α (GM)
0.001	181	184
0.005	69	128
0.01	49	117
0.02	58	113
0.03	71	120
0.05	98	133
0.1	167	177

TABLE 7.5

M3D2: number of GMRES iterations for solving the real equivalent form (2.3) with the preconditioners (5.4) and (5.2). Original matrix is complex nonsymmetric 1024×1024.

α	HSS (GM)	HSS (BiCG)	P_α (GM)	P_α (BiCG)
0.0005	248	nc	162	202
0.001	175	nc	163	257
0.005	133	136	nc	nc
0.01	187	237	nc	nc

TABLE 7.6

M4D2: number of iterations for solving the real equivalent form (2.3) with the preconditioners (5.4) and (5.2). Original matrix is complex nonsymmetric 10000×10000. *GM=GMRES; BiCG=BiCGSTAB; nc=no convergence.*

7.3. Structural dynamics. Finally, we consider some complex symmetric linear systems coming from direct frequency analysis in structural dynamics. The typical form of these (linearized) problems is

$$(\sigma^2 M + \sigma C + K_*)x = f, \tag{7.5}$$

where M is the mass matrix, which is typically real symmetric, possibly singular; C is the damping matrix, $C = \mathrm{i}C_V$ with C_V real, diagonal, and "highly" singular and possibly zero; K_* is the sum of two parts: the (real) stiffness matrix and the hysteretic damping matrix $K_* = K + \mathrm{i}C_H$, which is complex symmetric. When $C_H = \mu K$, where μ is a damping coefficient, we have $K_* = (1 + \mathrm{i}\mu)K$. Also $\sigma = \mathrm{i}\omega$, where ω is the circular frequency (measured in rad/sec). Thus, the real part A is SPD. For the right-hand side we use a vector f such that the solution x has all entries equal to $1 + \mathrm{i}$.

The first test problem considered is referred to as case "b" (with $\omega = 1$Hz) in [16]. The second one is referred to as case "c_1" (also with $\omega = 1$Hz) in [16]. For the first problem we have

$$\|A\|_\infty \approx 2.60, \quad \|B\|_\infty \approx 0.25$$

and for the second problem

$$\|A\|_\infty \approx 1.50, \quad \|B\|_\infty \approx 0.06.$$

Because of the properties of A and B, it is natural to choose the real equivalent form (2.3). It is also reasonable to expect that block diagonal or triangular preconditioners of the form (5.1) or (5.9) will perform well. Indeed, we found that for both test problems, just two iterations of Bi-CGSTAB(2) (with either (5.1) or (5.9) for the preconditioner, using exact solves) suffice to reach convergence. Without preconditioning,

the two problems require 13 and 243 Bi-CGSTAB(2) iterations, respectively. Recall that each iteration of Bi-CGSTAB(2) involves four matrix-vector products and four applications of the preconditioner, if present. For the first of the two test problems we also had good results (5-6 iterations with $\alpha = 0.1$) with the shifted skew-symmetric and HSS preconditioners applied to the real equivalent form (2.5). The same approach can also be used to solve the second test problem, but convergence is much slower, requiring around 50 iterations of Bi-CGSTAB(2).

We also experimented with different values of the driving circular frequency ω and we found the number of preconditioned iterations to be scarcely sensitive to it. Analogous observations hold for the results of tests performed on the so-called *inverse formulation* (see [16]).

8. Concluding remarks. In this paper we have investigated several block preconditioners for real-equivalent formulations of complex linear systems, with an emphasis on the complex symmetric case. We have also given guidelines for the choice of the real equivalent formulation and corresponding block preconditioners based on the properties and relative size (in norm) of the real and imaginary part of the coefficient matrix.

We have not attempted to carry out any systematic comparison of preconditioners for the real equivalent formulations with approaches based on the original complex form. We note, however, that the block preconditioners considered in this paper, while not always optimal, showed a good deal of robustness. In particular, HSS-type preconditioning succeeded in solving all of the test problems. Block diagonal and block triangular preconditioners based on approximate Schur complements can also be expected to be quite reliable. In contrast, we found that general-purpose preconditioners (like incomplete factorizations and sparse approximate inverses) applied to the original complex linear system (1.1) failed to solve some of the test problems for any reasonable value of the tuning parameters.

While more work is necessary, especially in order to find efficient (inexact) implementations of block preconditioning for real equivalent formulations for specific complex linear systems, we believe that such preconditioners can be useful tools and should be considered when solving large, sparse, complex systems of linear equations. This is especially true if one wishes to avoid the use of complex arithmetic and if efficient solvers are available for the (real) $n \times n$ linear subsystems that arise from block preconditioning.

Acknowledgements. We would like to thank Valeria Simoncini for making the structural dynamics test problems available to us.

REFERENCES

[1] M. F. Adams, *Algebraic multigrid methods for direct frequency response analyses in solid mechanics*, Comput. Mech., to appear (2006).

[2] G. Alléon, M. Benzi, and L. Giraud, *Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics*, Numer. Alg., 16 (1997), pp. 1–15.

[3] O. Axelsson and A. Kucherov, *Real valued iterative methods for solving complex symmetric linear systems*, Numer. Linear Algebra Appl., 7 (2000), pp. 197–218.

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

[5] Z.-Z. Bai, G. H. Golub, and C.-K. Li, *Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices*, Math. Comp., in press (2006).

[6] Z.-Z. BAI, G. H. GOLUB, AND M. K. NG, *Hermitian and Skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.

[7] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.

[8] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer.,14 (2005), pp. 1–137.

[9] D. BERTACCINI, *Efficient preconditioning for sequences of parametric complex symmetric linear systems*, Electr. Trans. Numer. Anal., 18 (2004), pp. 49–64.

[10] 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.

[11] A. BUNSE-GERSTNER AND R. STÖVER, *On a Conjugate Gradient-type method for solving complex symmetric linear systems*, Linear Algebra Appl., 287 (1999), pp. 105–123.

[12] B. CARPENTIERI, I. S. DUFF, L. GIRAUD, AND M. MAGOLU MONGA MADE, *Sparse symmetric preconditioners for dense linear systems in electromagnetism*, Numer. Linear Algebra Appl., 11 (2004), pp. 753–771.

[13] D. D. DAY AND M. A. HEROUX, *Solving complex-valued linear systems via equivalent real formulations*, SIAM J. Sci. Comput., 23 (2001), pp. 480–498.

[14] H. C. ELMAN, O. ERNST, AND D. P. O'LEARY, *A multigrid algorithm method enhanced by Krylov subspace iterations for discrete Helmholtz equations*, SIAM J. Sci. Comput. 23 (2001), pp. 1291–1315.

[15] Y. A. ERLANGGA, C. VUIK AND C. W. OOSTERLEE, *On a class of preconditioners for solving the Helmholtz equation*, Appl. Numer. Math., 50 (2004), pp. 409–425.

[16] A. FERIANI, F. PEROTTI, AND V. SIMONCINI, *Iterative system solvers for the frequency analysis of linear mechanical systems*, Comput. Methods Appl. Mech. Engrg., 190 (2000), pp. 1719–1739.

[17] R. W. FREUND, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 425–448.

[18] G. H. GOLUB AND D. VANDERSTRAETEN, *On the preconditioning of matrices with skew-symmetric splittings*, Numer. Alg., 25 (2000), pp. 223–239.

[19] E. HABER AND U. M. ASCHER, *Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 22 (2001), pp. 1943–1961.

[20] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[21] V. E. HOWLE AND S. A. VAVASIS, *An iterative method for solving complex-symmetric systems arising in electrical power modeling*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1150–1178.

[22] I. C. F. IPSEN, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[23] D. A. H. JACOBS, *A generalization of the Conjugate-Gradient method to solve complex systems*, IMA J. Numer. Anal., 6 (1986), pp. 447–452.

[24] P. JOLY AND G. MEURANT, *Complex Conjugate Gradient methods*, Numer. Algorithms, 4 (1993), pp. 379–406.

[25] M. MAGOLU MONGA MADE, *Incomplete factorization based preconditioning for solving the Helmholtz equation*, Int. J. Numer. Meth. Engng., 50 (2001), pp. 1077–1101.

[26] G. MARKHAM, *Conjugate Gradient type methods for indefinite, asymmetric, and complex systems*, IMA J. Numer. Anal., 10 (1990), pp. 155–170.

[27] A. MAZZIA AND G. PINI, *Numerical performance of preconditioning techniques for the solution of complex sparse linear systems*, Comm. Numerical Methods Engrg., 19 (2003), pp. 37–48.

[28] F. MAZZIA AND R. A. MCCOY, *Numerical experiments with a shifted SSOR preconditioner*, in D. R. Kincaid and A. C. Elster, eds., *Iterative Methods in Scientific Computation V*, IMACS Series in Computational and Applied Mathematics, vol. 5, New Brunswick, NJ (1999), pp. 195–209.

[29] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[30] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[31] B. POIRIER, *Efficient preconditioning scheme for block partitioned matrices with structured sparsity*, Numer. Linear Algebra Appl., 7 (2000), pp. 715–726.

[32] S. REITZINGER, U. SCHREIBER, AND U. VAN RIENEN, *Algebraic multigrid for complex symmetric matrices and applications*, J. Comput. Appl. Math., 155 (2003), pp. 405–421.

[33] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[34] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[35] V. SIMONCINI AND D. B. SZYLD, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., to appear (2006).

[36] G. L. G. SLEIJPEN AND D. R. FOKKEMA, *BiCGStab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum*, Electr. Trans. Numer. Anal., 1 (1993), pp. 11–32.

[37] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 10 (1992), pp. 631–644.

[38] H. A. VAN DER VORST AND J. B. M. MELISSEN, *A Petrov–Galerkin type method for solving $Ax = b$, where A is symmetric complex*, IEEE Trans. Magnetics, 26 (1990), pp. 706–708.