

Block preconditioning of real-valued iterative algorithms for complex linear systems

MICHELE BENZI[†]

*Department of Mathematics and Computer Science, Emory University,
Atlanta, GA 30322, USA*

AND

DANIELE BERTACCINI[‡]

Dipartimento di Matematica, Università di Roma “Tor Vergata”, 00133 Roma, Italy

[Received on 16 October 2006; revised on 7 September 2007]

We revisit real-valued preconditioned iterative 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. We argue that if either the real or the symmetric part of the coefficient matrix is positive semidefinite, block preconditioners for real equivalent formulations may be a useful alternative to preconditioners for the original complex formulation. Numerical experiments illustrating the performance of the various approaches are presented.

Keywords: complex symmetric systems; Krylov subspace methods; block preconditioners; Schur complement; Helmholtz equation.

1. Introduction

In this paper, we consider nonsingular linear systems of equations of the form

$$Cz = d, \text{ with } C = A + iB, \quad z = x + iy \quad \text{and} \quad d = b + ic. \quad (1.1)$$

Here $i = \sqrt{-1}$, 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 (Sommerfeld, 1949), diffuse optical tomography (Arridge, 1999, Section 3.3), quantum mechanics (numerical integration of the time-dependent Schrödinger equation by implicit methods) (van Dijk & Toyama, 2007), electromagnetism (Maxwell's equations) (Hiptmair, 2002), molecular scattering (Poirier, 2000), structural dynamics (frequency response analysis of mechanical systems) (Feriani *et al.*, 2000), electrical power system modelling (Howle & Vavasis, 2005) and lattice quantum chromodynamics (Frommer *et al.*, 2000). Another important source of complex symmetric linear systems is the discretization of certain self-adjoint integrodifferential equations arising in environmental modelling (see, e.g. Gambolati & Pini, 1998).

In all these applications, the matrices are large and usually sparse; hence, preconditioned Krylov subspace methods are a natural choice. There are two options: either tackling the $n \times n$ system (1.1)

[†]Corresponding author. Email: benzi@mathcs.emory.edu

[‡]Email: bertaccini@mat.uniroma2.it

directly or working with one of the several $2n \times 2n$ equivalent real formulations. Each option has its own advantages and disadvantages. 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, e.g. on the main diagonal (see, e.g. Bertaccini, 2004; Haber & Ascher, 2001; Howle & Vavasis, 2005; Poirier, 2000). 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 (see, e.g. Bertaccini, 2004). 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. Another difficulty, mentioned in Day & Heroux (2001), 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, there is an interest in iterative methods for real equivalent formulations of complex linear systems. It was shown in Day & Heroux (2001) that nonsymmetric Krylov subspace methods preconditioned by standard incomplete LU (ILU) factorizations can perform reasonably well with the so-called K -formulation of the equations (see Section 2), with a performance comparable to that obtained with ILU-preconditioned Krylov methods applied to the original complex form.

Motivated in part by Day & Heroux (2001), 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) below) 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. Bai, 2006; Benzi *et al.*, 2005; Ipsen, 2001; Murphy *et al.*, 2000). Our main goal is 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. If $f: \mathbb{C}^{n \times n} \rightarrow \mathbb{R}^{2n \times 2n}$ is the homomorphism of the algebra of $n \times n$ complex matrices into the algebra of $2n \times 2n$ real matrices defined by

$$f(A + iB) = \begin{pmatrix} A & -B \\ B & A \end{pmatrix},$$

the range space (image) of f is a subalgebra of dimension $2n^2$ of $\mathbb{R}^{2n \times 2n}$, which has dimension $4n^2$; note that the image of f has Lebesgue measure zero in $\mathbb{R}^{2n \times 2n}$. Thus, it is plausible 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.

While we do not claim that real equivalent formulations should be preferred to the original complex form, we think that there is a role for block preconditioners applied to real formulations, especially when either A or B is (semi) definite, a condition frequently satisfied in practice.

The paper is organized as follows: Section 2 contains a brief description of different 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 presents our own contribution, i.e. block 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. Real equivalent formulations

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 system

$$(-B + iA)(x + iy) = -c + ib \quad \text{or} \quad (B - iA)(x + iy) = c - ib. \quad (2.1)$$

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

$$C^H C z = C^H d. \quad (2.2)$$

The advantage of the normal equations is that the coefficient matrix is Hermitian positive definite. The main disadvantage is that its condition number is the square of the condition number of C . This makes this approach rather unappealing, especially when C itself is ill conditioned.

The complex $n \times n$ linear system (1.1) also admits several $2n \times 2n$ real equivalent formulations. We begin with

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

This is in turn is equivalent to

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

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

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

and

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

Again, the latter system is symmetric if A and B are. The equivalent real formulations (2.3–2.6) do not preserve the bandwidth or the structure of C . The performance of sparse direct linear solvers applied to

(2.3–2.6) often suffers in comparison to the original complex coefficient case. Also, the effectiveness of preconditioners, like ILU, that are related to sparse direct solvers degrades similarly. For such solvers, it is better to consider real equivalent formulations that preserve the structure of C . As described in Day & Heroux (2001), 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 $c_{ij} = a_{ij} + ib_{ij}$ 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

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

Then, the coefficient matrix of the corresponding K -formulation is

$$K = \begin{pmatrix} a_{11} & -b_{11} & 0 & 0 & a_{13} & -b_{13} & 0 & 0 & a_{15} & -b_{15} \\ b_{11} & a_{11} & 0 & 0 & b_{13} & a_{13} & 0 & 0 & b_{15} & a_{15} \\ \hline 0 & 0 & a_{22} & -b_{22} & a_{23} & -b_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{22} & a_{22} & b_{23} & a_{23} & 0 & 0 & 0 & 0 \\ \hline a_{31} & -b_{31} & 0 & 0 & a_{33} & -b_{33} & a_{34} & -b_{34} & 0 & 0 \\ b_{31} & a_{31} & 0 & 0 & b_{33} & a_{33} & b_{34} & a_{34} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & a_{43} & -b_{43} & a_{44} & -b_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{43} & a_{43} & b_{44} & a_{44} & 0 & 0 \\ \hline a_{51} & -b_{51} & 0 & 0 & 0 & 0 & 0 & 0 & a_{55} & -b_{55} \\ b_{51} & a_{51} & 0 & 0 & 0 & 0 & 0 & 0 & b_{55} & a_{55} \end{pmatrix}.$$

We note that in some cases it may be possible to avoid storing each entry a_{ij} and 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 that for the original $n \times n$ 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 Horn

& Johnson, 1985, 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 $C = A + iB$ have spectrum $\sigma(C) = \Lambda = \{\lambda_i | 1 \leq i \leq n\} \subset \mathbb{C}$. Let $\bar{\Lambda}$ 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 $\sigma(M) = \Lambda \cup \bar{\Lambda}$.

For the symmetric forms (2.4) and (2.6), the eigenvalues are, of course, all real, but additional structure is present. Consider, e.g. 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 Freund (1992) 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

$$\sigma(C) \subset \{\lambda \in \mathbb{C} | \operatorname{Re} \lambda \geq 0\} \quad \text{or} \quad \sigma(C) \subset \{\lambda \in \mathbb{C} | \operatorname{Im} \lambda \geq 0\}.$$

This will be the case, in particular, whenever A or B is symmetric positive definite (SPD). In this case, the real equivalent form (2.3) (or (2.5)) will be positive definite, in the sense that its symmetric 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 (Saad & Schultz, 1986). 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, i.e. $v^T M v \geq 0$ for all $v \in \mathbb{R}^{2n}$, and positive semistable (i.e. its eigenvalues have non-negative 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 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 $C = A + iB$ with A indefinite and B positive definite. The matrix arises from a finite-difference discretization of an Helmholtz-type problem on a 16×16 grid, see Problem 7.1 in Section 7.1. Figure 1 displays the eigenvalues of C . Note that the eigenvalues all lie in the upper half-plane.

In Fig. 2(a, b), we plot the eigenvalues of the real equivalent formulations (2.5) and (2.3), respectively. Note that the eigenvalues of (2.3) surround the origin; this often results in very slow convergence of Krylov subspace methods.

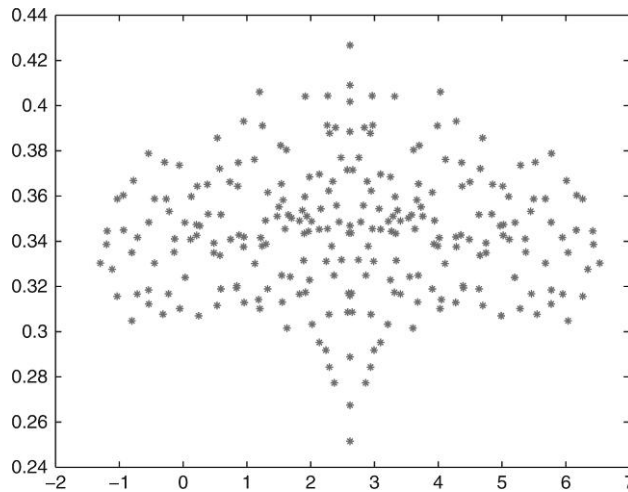


FIG. 1. Eigenvalues of complex symmetric $C = A + iB$ with B being SPD.

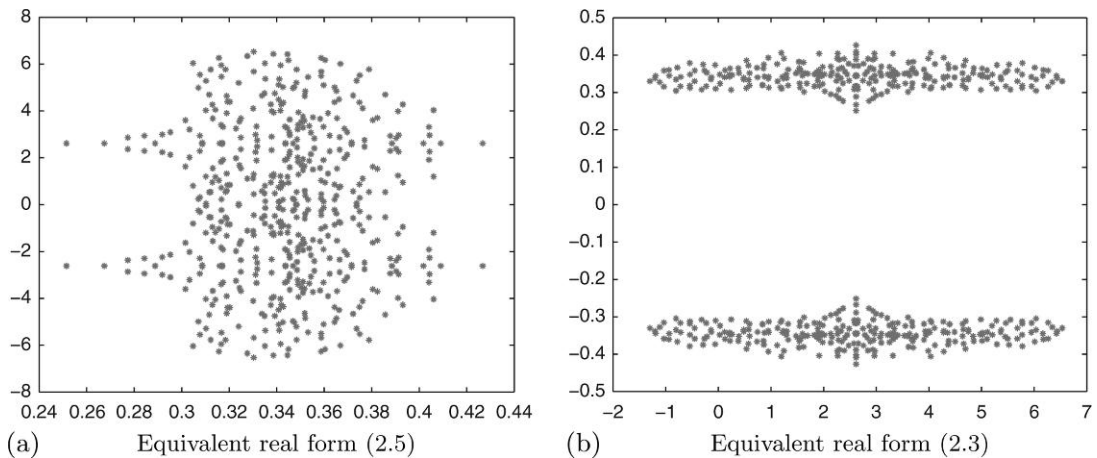


FIG. 2. Eigenvalue plots for two real equivalent formulations. Here, the matrix (2.5) is positive real, hence all its eigenvalues lie in the right-half plane. In contrast, the symmetric part of (2.3) is indefinite and the eigenvalues of (2.3) surround the origin.

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 (Paige & Saunders, 1975) applied to the symmetric forms (2.4) and (2.6). The latter requires 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. Bunse-Gerstner & Stöver (1999), Freund (1990, 1992), Jacobs (1986), Joly & Meurant (1993), Markham (1990) and van der Vorst & Melissen (1990), as well as the recent survey (Simoncini & Szyld, 2007, Sections 12 and 13.3).

Influential work by Freund (especially Freund, 1992) 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 Freund (1992). 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, e.g. those studied in Adams (2007), Elman *et al.* (2001), Erlangga *et al.* (2004), Haber & Ascher (2001), Howle & Vavasis (2005), Poirier (2000) and Reitzinger *et al.* (2003), there has been some work on the use of general-purpose techniques such as SSOR, polynomial preconditioning, incomplete factorizations and sparse approximate inverses (see, e.g. Freund, 1990; Horesh *et al.*, 2006; Mazzia & Pini, 2003; Mazzia & McCoy, 1999). Somewhere in between, we mention the work of Magolu monga Made (2001, e.g. and the references therein) on a class of incomplete factorization preconditioners specifically tailored to the Helmholtz equation. 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 discussed in Section 1, 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 (Day & Heroux, 2001), we mention (Axelsson & Kucherov, 2000) 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 definite (or at least semidefinite), 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 Day & Heroux (2001).

5.1 Preconditioners having both real and complex forms

We begin with preconditioners that have both real and complex forms. 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, $A^{-1}B$ can be thought of as a discretization of a compact operator and most of its eigenvalues will be clustered around the origin in the complex plane. Thus, much of the spectrum of $A^{-1}C = I + iA^{-1}B$ will be clustered around $(1, 0)$. In this case, a good

approximation \widehat{A} of A , assumed to be invertible, might be a reasonable choice as a preconditioner for $C = A + iB$. Equivalently, the block-diagonal matrix

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

might be a good preconditioner for (2.3), except in situations where the preconditioned operator retains a non-negligible number of eigenvalues near the origin. This happens, for instance, if B is very ill conditioned, as is the case when B is a mass matrix for a regular nonuniform finite-element mesh whose element diameters vary over several orders of magnitude. In the case of acoustics and other wave phenomena, however, quasi-uniform meshes must be used in order to maintain sufficient resolution per wave length, and this problem does not typically arise.

We mention that other options are available when C is an imaginary perturbation of a SPD matrix, e.g. $C = A + iB$ with A being 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 Bertaccini (2004), where good results are reported for certain differential equation problems.

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 $\widehat{B}^2 + \alpha^2 I$. Note that this matrix is SPD if \widehat{B} is symmetric. Moreover, if \widehat{B} is scaled so that $\lambda_{\max}(\widehat{B}^2) = 1$, then the spectral condition number of $\widehat{B}^2 + \alpha^2 I$ satisfies $\kappa_2(\widehat{B}^2 + \alpha^2 I) \leq 1 + 1/\alpha^2$. If α is not too small, then $\widehat{B}^2 + \alpha^2 I$ is well conditioned and it may be easy to find approximate solutions to linear systems involving $\widehat{B}^2 + \alpha^2 I$, for instance using PCG. Shifted skew-symmetric preconditioning has been introduced, in a different context (convection-dominated flow problems), in Golub & Vanderstraeten (2000). Of course, preconditioning M with P_α is mathematically equivalent to preconditioning the complex matrix $C = A + iB$ with $\alpha I + i\widehat{B}$. Note that the latter is a shifted skew-Hermitian matrix if \widehat{B} is symmetric.

The foregoing discussion suggests that simple choices are available if one of A or B is (semi) definite and 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 $\widehat{A} + i\widehat{B}$ 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

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

In practice, however, there seem to be very few situations where a preconditioner of the form (5.3) can be effectively employed for the solution of (2.3). Another possible choice of a preconditioner that includes

both A and B is given by the ‘alternating’ Hermitian/skew-Hermitian splitting (HSS) preconditioner

$$P_{\text{HSS}} = \begin{pmatrix} A + \alpha I & O \\ O & A + \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & -B \\ B & \alpha I \end{pmatrix}, \quad (5.4)$$

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

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

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 the skew-symmetric parts of M . The complex equivalent form of this preconditioner is clearly the complex $n \times n$ matrix $(A + \alpha I)(\alpha I + iB)$. The HSS preconditioner is well defined (invertible) for semidefinite A and 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 $\rho(I - P_{\text{HSS}}^{-1}M) < 1$ for all $\alpha > 0$, where $\rho(\cdot)$ denotes the spectral radius (Bai *et al.*, 2003). 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 (Benzi & Golub, 2004). In practice, A and B are often replaced by approximations \hat{A} and \hat{B} . See, e.g. Bai *et al.* (2003), Benzi & Golub (2004) and Bertaccini *et al.* (2005) for detailed analyses and variants of this approach.

In Fig. 3(a, b), we plot the eigenvalues of the real equivalent form of the discrete Helmholtz operator preconditioned with HSS, taking $\alpha = 0.1$. 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 centred at $(1, 0)$, as predicted

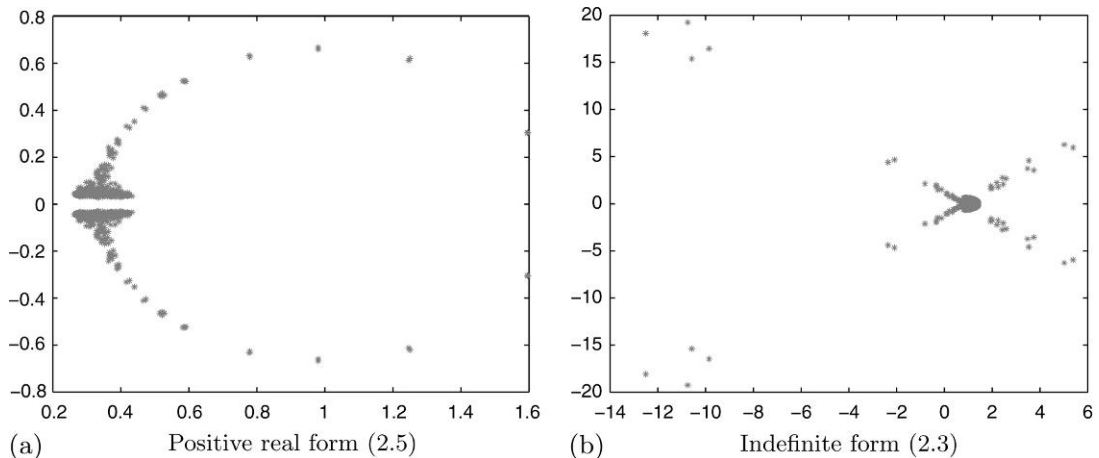


FIG. 3. Eigenvalues for HSS preconditioning of real equivalent formulations of Helmholtz problem.

by the theory. On the 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.

5.2 *Preconditioners that have no complex counterpart*

All the preconditioners discussed in Section 5.1 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 \widehat{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 GMRES is guaranteed to find the solution in at most two steps (see Ipsen, 2001; Murphy *et al.*, 2000).

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 solve linear systems with S (at least approximately). In practical situations, however, forming S and/or solving systems with it is out of question. Instead, 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 (and often better than) 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 non-normality on the behaviour of the preconditioned iteration, and therefore it is not to be taken too literally.

If \hat{A} is an approximation of A such that \hat{A} has a sparse inverse, then it may be possible to explicitly form the approximate Schur complement $\hat{S} = \hat{A} + B\hat{A}^{-1}B$ (or even $\hat{S} = A + B\hat{A}^{-1}B$) for use in a block-triangular preconditioner. For instance, if A is diagonally dominant, then a diagonal approximation $\hat{A} = \text{diag}(A)$ 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, $S^{-1} = [I + (A^{-1}B)^2]^{-1}A^{-1}$. 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 Section 5 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 centred 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 computed with Matlab using various combinations of block preconditioners and iterative methods. We consider real equivalent formulations of complex symmetric problems arising from the discretization of different types of Helmholtz equations, similar to those considered in Freund (1992). Another set of experiments concerns linear systems arising in molecular dynamics (see Day & Heroux, 2001). Finally, we include a few tests with linear systems arising in frequency analysis of mechanical systems (see Feriani *et al.*, 2000).

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 our implementations of full GMRES (Saad & Schultz, 1986), flexible GMRES (FGMRES, Saad, 1993), BiCGSTAB (van der Vorst, 1992) and BiCGSTAB(2) (Sleijpen & Fokkema, 1993). Note that each iteration of BiCGSTAB requires two matrix–vector multiplies and two applications of the preconditioner, and BiCGSTAB(2) requires four of each. Right preconditioning is used in all cases.

7.1 Helmholtz-type equations

Helmholtz equations are of fundamental importance in the modelling of wave propagation phenomena. Similar to Freund (1992), we consider two types of model problems based on the finite-difference discretization of the partial differential equation (PDE)

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

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 2D 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:

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

The first problem is a complex Helmholtz equation with both σ_1 and σ_2 strictly positive, so as to make the real part A of the discrete operator $C = A + iB$ indefinite and the imaginary part B definite. Note that B is diagonal. The boundary conditions are of Dirichlet type. The second problem is a ‘nice’ Helmholtz-like equation with $\sigma_2 = 0$, $\sigma_1 < 0$ and complex boundary 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 1024 to 262 144. Right-hand sides were generated at random with components uniformly distributed in $[-1, 1] + i[-1, 1]$. We remark that for all these problems, Krylov subspace methods either diverge or converge very slowly in the absence of preconditioning. Additional details follow.

PROBLEM 7.1 (Complex Helmholtz equation). The finite-difference approximation results in complex symmetric linear systems of the form

$$Cz = d, \quad C = T - h^2 \sigma_1 I + i h^2 D = A + iB, \tag{7.2}$$

where D is diagonal and T is a standard second-order finite-difference discretization of the diffusion operator $-\nabla \cdot (c \nabla u)$, normalized so as to have $\lambda_{\max}(T) = O(1)$ and $\lambda_{\min}(T) = O(h^{-2})$ for $h \rightarrow 0$.

The diagonal entries of D are the values of σ_2 at the grid points. The coefficient σ_1 is a constant chosen so as to make the real part of the matrix 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 GMRES with the (shifted) skew-symmetric and HSS preconditioners

$$P_\alpha = \begin{pmatrix} \alpha I & A \\ -A & \alpha I \end{pmatrix} \quad \text{and} \quad P_{\text{HSS}} = \begin{pmatrix} B + \alpha I & O \\ O & B + \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & A \\ -A & \alpha I \end{pmatrix} \quad (7.3)$$

with different values of the shift $\alpha > 0$ are shown in Fig. 4 (a, b) for the 32×32 grid. These plots show that the number of iterations is not overly sensitive to the value of α (note the logarithmic scale on the horizontal axis). 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 P_α and HSS result in eight 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 a 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 a sparse Cholesky factorization 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). This would also eliminate the need to explicitly form $A^2 + \alpha^2 I$, which may be expensive. For small α , however, this auxiliary linear system with matrix $A^2 + \alpha^2 I$ will need a preconditioner which may be difficult to find, especially if the matrix itself is not explicitly available.

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 were found to be ineffective. 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 observed that for this problem the performance of BiCGSTAB is generally inferior to GMRES, especially for suboptimal choices of α . In one case, BiCGSTAB suffered a fatal breakdown. 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

$$\begin{pmatrix} B & A \\ O & \widehat{S} \end{pmatrix}, \quad (7.4)$$

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 two 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. Our conclusion is that for this problem, the shifted skew-symmetric preconditioner P_α (5.2) and the block-triangular preconditioner (7.4) are the most effective among the block preconditioners considered in this paper. However, an efficient implementation requires a good approximation to the reduced system matrix $A^2 + \alpha^2 I$ (to the Schur complement $S = B + AB^{-1}A$, respectively). Here, we used an inner iteration preconditioned with a shifted incomplete Cholesky factorization to compute the action of $\widehat{S}^{-1} \approx S^{-1}$, but more work is needed to identify better approximations. Clearly, this is a problem-dependent task.

PROBLEM 7.2 (Real Helmholtz equation with complex boundary conditions). Here, $\sigma_1 = -200$, $\sigma_2 = 0$ and we impose a boundary condition of the form

$$\frac{\partial u}{\partial n} + \beta u = 0 \quad \text{on } \{(1, y) | 0 < y < 1\}$$

and Dirichlet boundary conditions on the remaining three sides of the unit square, with the complex function β chosen so as to make the imaginary part of C indefinite (cf. Magolu monga Made, 2001). We take β such that the resulting linear system is again of the form (7.2) where the diagonal entries of

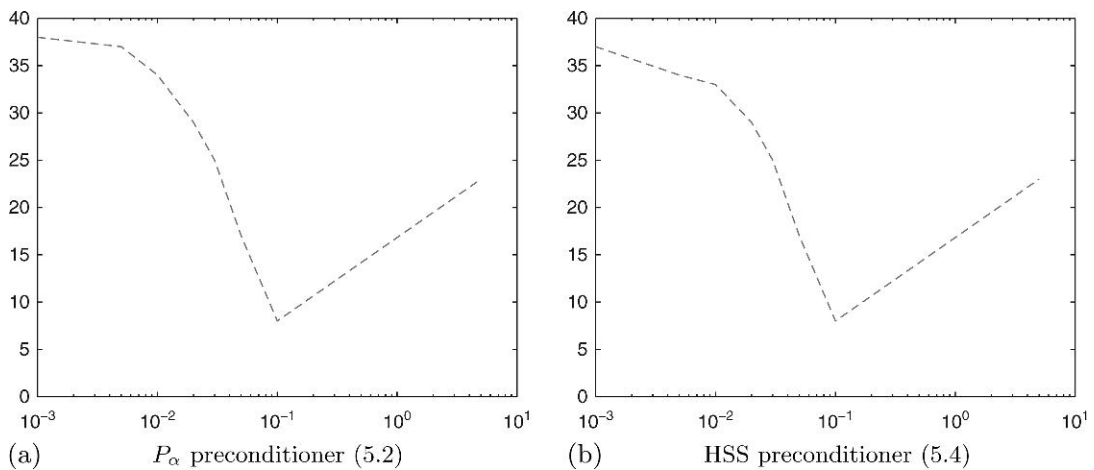


FIG. 4. Problem 7.1: results for real equivalent form (2.5) of complex Helmholtz equation on 32×32 grid. Number of GMRES iterations for P_α and HSS preconditioning as a function of α . Note the logarithmic scale on the horizontal axis.

D are given by

$$d_{i+(j-1)m} = \begin{cases} (-1)^{j-1} 100/h, & \text{if } i = m, \\ 0, & \text{otherwise} \end{cases} \quad (7.5)$$

($i, j = 1, \dots, m$). Now, the real part of the matrix $A + iB$ 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 the value 2000 replacing 100 in (7.5), resulting in matrices A and B having norms of the same order of magnitude. For example, for $m = 32$, we have

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

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 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 Fig. 5(a, b), we show results obtained for GMRES with the HSS preconditioner for various values of α ; similar results were obtained with BiCGSTAB. Note, again, the mild dependence of the rate of convergence on α .

In Table 1, we show results for three fine-grid discretizations of the same problem of Fig. 5(b). 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 1, 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

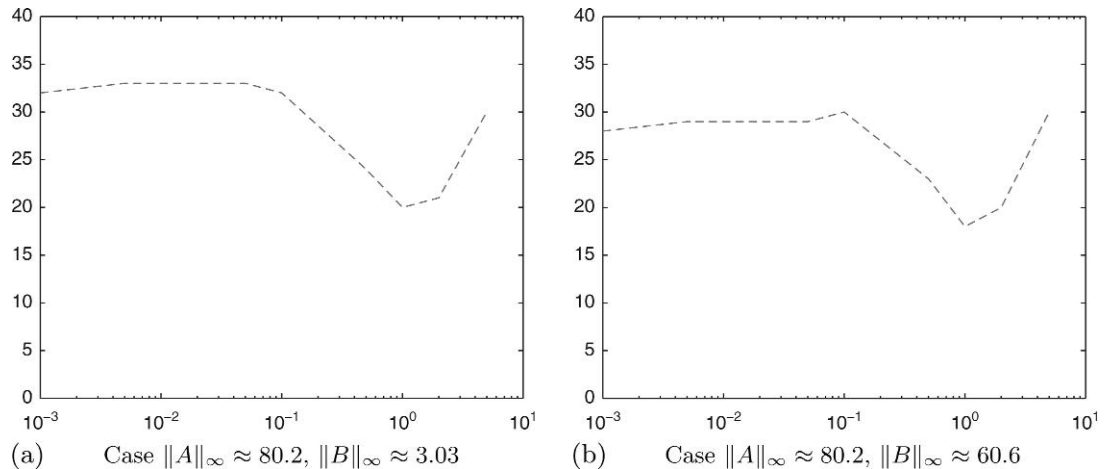


FIG. 5. Problem 7.2: results for real equivalent form (2.3) of real Helmholtz equations with complex boundary conditions on 32×32 grid. Number of GMRES iterations with HSS preconditioning as a function of α . Note the logarithmic scale on the horizontal axis.

TABLE 1 *Helmholtz equation model problem 7.2, same as in Fig. 5(b) but with finer grids. BiCGSTAB with HSS preconditioning, number of iterations as a function of α . Elapsed times (in seconds) are given in parentheses*

α	128 \times 128 grid	256 \times 256 grid	512 \times 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)

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; note that the mesh is uniform. We conclude that for this (relatively easy) problem, simple block-triangular preconditioners of the form (5.9) are the most effective among the block preconditioners considered in this paper.

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 Day & Heroux (2001). These problems, denoted M3D2 and M4D2, are nonsymmetric. The eigenvalues of M3D2 satisfy the half-plane condition $\text{Im } \lambda > 0$, but neither the real part A nor 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 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 Fig. 6 and Table 2, we report results obtained with the analogues of the shifted skew-symmetric and HSS preconditioners (7.3). Since A and B are nonsymmetric, these are not truly shifted skew-symmetric and HSS 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 .

We conclude that matrix M4D2 is a possible example of a problem for which block preconditioners of the type studied in this paper are not effective. To put things in perspective, however, it should be mentioned that this is a very difficult problem. In Day & Heroux (2001), the authors observed that in order to have rapid convergence of ILU-preconditioned GMRES, large amounts of fill-in in the incomplete factors have to be allowed, both for the original complex formulation and when using the equivalent K -formulation.

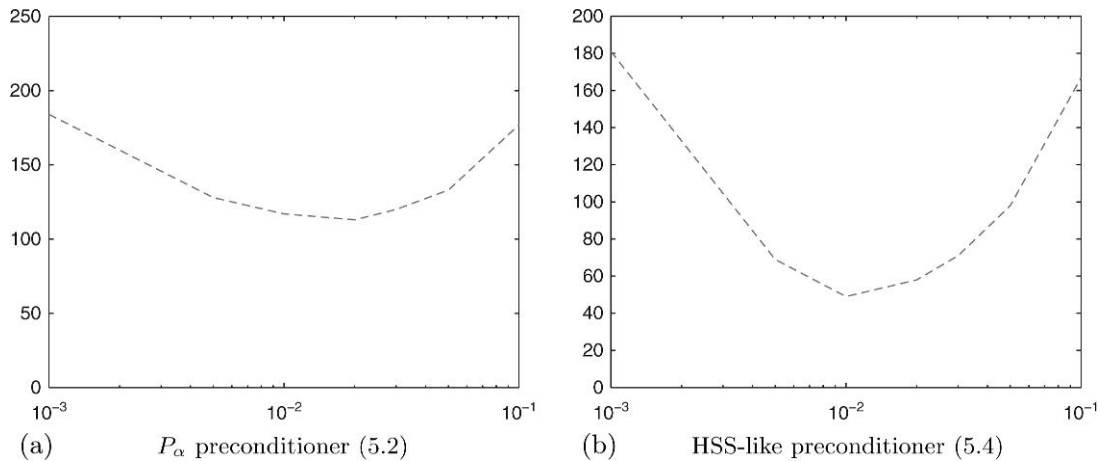


FIG. 6. Problem M3D2: number of GMRES iterations as a function of α for solving the real equivalent form (2.3) with preconditioners P_α and HSS. Note the logarithmic scale on the horizontal axis.

TABLE 2 Problem 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 $10\,000 \times 10\,000$. GM, GMRES; BiCG, BiCGSTAB; nc, no convergence

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

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

$$(\sigma^2 M + \sigma C_V + K_*)x = f, \tag{7.6}$$

where M is the inertia (mass) matrix, which is typically real symmetric, possibly singular; C_V is the viscous damping matrix (real, diagonal, ‘highly singular’ and possibly zero) and $K_* = K + iC_H$, where K is the (real symmetric) stiffness matrix and C_H is the hysteretic damping matrix, which is also real symmetric. When $C_H = \mu K$, where μ is a damping coefficient, we have $K_* = (1 + i\mu)K$. Also $\sigma = i\omega$, where ω is the circular frequency (measured in rad/sec). For moderate values of ω , the real part $A = K - \omega^2 M$ is SPD. For the right-hand side, we use a vector f such that the solution x has all entries equal to $1 + i$.

The first test problem considered is referred to as case ‘b’ (with $\omega = 1$ Hz) in [Feriani et al. \(2000\)](#). The second one is referred to as case ‘c₁’ (also with $\omega = 1$ Hz) in [Feriani et al. \(2000\)](#). 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 BiCGSTAB(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 BiCGSTAB(2) iterations, respectively. Recall that each iteration of BiCGSTAB(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 BiCGSTAB(2).

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. Not surprisingly, no single preconditioner was found to be superior to the remaining ones on all problems considered here. Nevertheless, we found that in several cases block-triangular preconditioners, for which no equivalent complex preconditioner exists, are superior to block preconditioners that have a complex counterpart. Choosing a good preconditioner requires a knowledge of the nature and properties of the real and imaginary parts A and B of the original complex coefficient matrix. In this paper, we have given some guidelines for the choice of the real equivalent formulation and corresponding block preconditioners based on the spectral properties and relative size (in norm) of A and B .

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 regard the block preconditioners for equivalent real forms and the preconditioners for the original complex systems as

complementary, not as competitors, with the former being a potentially useful tool when one wishes to avoid the complex form, whatever the reason may be.

An important practical issue with the block preconditioners is to find good approximations for the corresponding reduced systems (Schur complements); clearly, this is a problem-dependent aspect. One caveat that should be kept in mind is that problems with both A and B indefinite are hard to solve with any of the methods considered here and possibly with any existing method and formulation (whether real or complex).

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 the referees for their many useful comments and suggestions. Thanks also to Valeria Simoncini for making the structural dynamics test problems available to us.

Funding

National Science Foundation (DMS-0207599, DMS-0511336); INdAM-GNCS; MIUR (2004015437, 2006017542).

REFERENCES

- ADAMS, M. F. (2007) Algebraic multigrid methods for direct frequency response analyses in solid mechanics. *Comput. Mech.*, **39**, 497–507.
- ARRIDGE, S. R. (1999) Optical tomography in medical imaging. *Inverse Probl.*, **15**, R41–R93.
- AXELSSON, O. & KUCHEROV, A. (2000) Real valued iterative methods for solving complex symmetric linear systems. *Numer. Linear Algebra Appl.*, **7**, 197–218.
- BAI, Z.-Z. (2006) Structured preconditioners for nonsingular matrices of block two-by-two structures. *Math. Comput.*, **75**, 791–815.
- BAI, Z.-Z., GOLUB, G. H. & NG, M. K. (2003) Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, **24**, 603–626.
- BENZI, M. & GOLUB, G. H. (2004) A preconditioner for generalized saddle point problems. *SIAM J. Matrix Anal. Appl.*, **26**, 20–41.
- BENZI, M., GOLUB, G. H. & LIESEN, J. (2005) Numerical solution of saddle point problems. *Acta Numer.*, **14**, 1–137.
- BERTACCINI, D. (2004) Efficient preconditioning for sequences of parametric complex symmetric linear systems. *Electron. Trans. Numer. Anal.*, **18**, 49–64.
- BERTACCINI, D., GOLUB, G. H., SERRA-CAPIZZANO, S. & TABLINO-POSSIO, C. (2005) Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation. *Numer. Math.*, **99**, 441–484.
- BUNSE-GERSTNER, A. & STÖVER, R. (1999) On a conjugate gradient-type method for solving complex symmetric linear systems. *Linear Algebra Appl.*, **287**, 105–123.
- DAY, D. D. & HEROUX, M. A. (2001) Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, **23**, 480–498.

- ELMAN, H. C., ERNST, O. & O'LEARY, D. P. (2001) A multigrid algorithm method enhanced by Krylov subspace iterations for discrete Helmholtz equations. *SIAM J. Sci. Comput.*, **23**, 1291–1315.
- ERLANGGA, Y. A., VUIK, C. & OOSTERLEE, C. W. (2004) On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, **50**, 409–425.
- FERIANI, A., PEROTTI, F. & SIMONCINI, V. (2000) Iterative system solvers for the frequency analysis of linear mechanical systems. *Comput. Methods Appl. Mech. Eng.*, **190**, 1719–1739.
- FREUND, R. W. (1990) On conjugate gradient type methods and polynomial preconditioners for a class of complex non-Hermitian matrices. *Numer. Math.*, **57**, 285–312.
- FREUND, R. W. (1992) Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Stat. Comput.*, **13**, 425–448.
- FROMMER, A., LIPPERT, T., MEDEKE, B. & SCHILLING, K. (eds) (2000) *Numerical Challenges in Lattice Quantum Chromodynamics*. Lecture Notes in Computational Science and Engineering, vol. 15. Berlin: Springer.
- GAMBOLATI, G. & PINI, G. (1998) Complex solution to nonideal contaminant transport through porous media. *J. Comput. Phys.*, **145**, 538–554.
- GOLUB, G. H. & VANDERSTRAETEN, D. (2000) On the preconditioning of matrices with skew-symmetric splittings. *Numer. Algorithms*, **25**, 223–239.
- HABER, E. & ASCHER, U. M. (2001) Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients. *SIAM J. Sci. Comput.*, **22**, 1943–1961.
- HIPTMAIR, R. (2002) Finite elements in computational electromagnetism. *Acta Numer.*, **11**, 237–339.
- HORESH, L., SCHWEIGER, M., BOLLHÖFER, M., DOUIRI, A., HOLDER, D. S. & ARRIDGE, S. R. (2006) Multilevel preconditioning for 3D large-scale soft-field medical applications modelling. *Int. J. Inf. Syst. Sci.*, **2**, 532–556.
- HORN, R. A. & JOHNSON, C. R. (1985) *Matrix Analysis*. Cambridge: Cambridge University Press.
- HOWLE, V. E. & VAVASIS, S. A. (2005) An iterative method for solving complex-symmetric systems arising in electrical power modeling. *SIAM J. Matrix Anal. Appl.*, **26**, 1150–1178.
- IPSEN, I. C. F. (2001) A note on preconditioning nonsymmetric matrices. *SIAM J. Sci. Comput.*, **23**, 1050–1051.
- JACOBS, D. A. H. (1986) A generalization of the conjugate-gradient method to solve complex systems. *IMA J. Numer. Anal.*, **6**, 447–452.
- JOLY, P. & MEURANT, G. (1993) Complex conjugate gradient methods. *Numer. Algorithms*, **4**, 379–406.
- MAGOLU MONGA MADE, M. (2001) Incomplete factorization based preconditioning for solving the Helmholtz equation. *Int. J. Numer. Methods Eng.*, **50**, 1077–1101.
- MARKHAM, G. (1990) Conjugate gradient type methods for indefinite, asymmetric, and complex systems. *IMA J. Numer. Anal.*, **10**, 155–170.
- MAZZIA, A. & PINI, G. (2003) Numerical performance of preconditioning techniques for the solution of complex sparse linear systems. *Commun. Numer. Methods Eng.*, **19**, 37–48.
- MAZZIA, F. & MCCOY, R. A. (1999) Numerical experiments with a shifted SSOR preconditioner. *Iterative Methods in Scientific Computation V* (D. R. Kincaid & A. C. Elster eds). IMACS Series in Computational and Applied Mathematics, vol. 5. New Brunswick, NJ: International Association for Mathematics and Computers in Simulation, pp. 195–209.
- MURPHY, M. F., GOLUB, G. H. & WATHEN, A. J. (2000) A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, **21**, 1969–1972.
- PAIGE, C. C. & SAUNDERS, M. A. (1975) Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, **12**, 617–629.
- POIRIER, B. (2000) Efficient preconditioning scheme for block partitioned matrices with structured sparsity. *Numer. Linear Algebra Appl.*, **7**, 715–726.
- REITZINGER, S., SCHREIBER, U. & VAN RIENEN, U. (2003) Algebraic multigrid for complex symmetric matrices and applications. *J. Comput. Appl. Math.*, **155**, 405–421.

- SAAD, Y. (1993) A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, **14**, 461–469.
- SAAD, Y. & SCHULTZ, M. H. (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869.
- SIMONCINI, V. & SZYLD, D. B. (2007) Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.*, **14**, 1–59.
- SLEIJPEN, G. L. G. & FOKKEMA, D. R. (1993) BiCGStab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum. *Electron. Trans. Numer. Anal.*, **1**, 11–32.
- SOMMERFELD, A. (1949) *Partial Differential Equations*. New York, NY: Academic Press.
- VAN DER VORST, H. A. (1992) Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **10**, 631–644.
- VAN DER VORST, H. A. & MELISSEN, J. B. M. (1990) A Petrov–Galerkin type method for solving $Ax = b$, where A is symmetric complex. *IEEE Trans. Magn.*, **26**, 706–708.
- VAN DIJK, W. & TOYAMA, F. M. (2007) Accurate numerical solutions of the time-dependent Schrödinger equation. *Preprint*. arXiv:physics/0701150v1 [physics.comp-ph].