Numer Algor
DOI 10.1007/s11075-013-9764-1

ORIGINAL PAPER

A comparison of iterative methods to solve complex valued linear algebraic systems

Owe Axelsson · Maya Neytcheva · Bashir Ahmad

Received: 17 March 2013 / Accepted: 25 August 2013

Abstract Complex valued linear algebraic systems arise in many important applications. We present analytical and extensive numerical comparisons of some available numerical solution methods. It is advocated, in particular for large scale ill-conditioned problems, to rewrite the complex-valued system in real valued form leading to a two-by-two block system of particular form, for which it is shown that a very efficient and robust preconditioned iterative solution method can be constructed. Alternatively, in many cases it turns out that a simple preconditioner in the form of the sum of the real and the imaginary part of the matrix also works well but involves complex arithmetic.

Keywords Linear systems · Complex symmetric · Real valued form · Preconditioning

O. Axelsson · B. Ahmad
King Abdulaziz University, Jeddah, Saudi Arabia

B. Ahmad
e-mail: bashirahmad_qau@yahoo.com

O. Axelsson
Institute of Geonics, AVSR, Ostrava, Czech Republic
e-mail: owe.axelsson@it.uu.se

M. Neytcheva (✉)
Department of Information Technology, Uppsala University, Uppsala, Sweden
e-mail: maya.neytcheva@it.uu.se

Published online: 07 September 2013

1 Introduction

Complex valued linear algebraic systems arise in many important applications, such as in computational electrodynamics (e.g., [1]), in time–dependent Schrödinger equations (e.g., [2]), inverse scattering problems (e.g., [3]), and in the numerical solution of (stiff) systems of ordinary differential equations using implicit Runge–Kutta methods (e.g., [4, 5]).

We pay particular attention to complex symmetric matrices. They arise in important large scale applied problems, such as quantum mechanics, electromagnetism, structural dynamics, electrical power system models, wave propagation, magnetized multicomponent transport etc.

To simplify the data handling and the construction of efficient preconditioners, it can be efficient to rewrite the complex valued system in real valued form to enable the use of real arithmetics. This can, for instance, be done in the following way. Consider the complex linear system

$$\boldsymbol{C}z = \boldsymbol{h}, \tag{1}$$

where $\boldsymbol{C} = A + iB$, $z = \mathbf{x} + i\mathbf{y}$ and $\boldsymbol{h} = \mathbf{f} + i\mathbf{g}$. Thus, $(A + iB)(\mathbf{x} + i\mathbf{y}) = \mathbf{f} + i\mathbf{g}$, where A, B are real matrices, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{f}$, $\mathbf{g}$ are real vectors and $i = \sqrt{-1}$ is the imaginary unit. This system can be rewritten in a matrix form

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{2}$$

As discussed in earlier works, for instance in [6], the real form (2) of the system (1) is not unique. As an example, the form

$$\begin{bmatrix} B & -A \\ A & B \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -\mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \tag{3}$$

is equivalent to (2) and could be the preferred choice for some particular combinations of the matrices A and B. Note that $\boldsymbol{C}$ is nonsingular if and only is $\pm i$ is not a generalized eigenvalue of the matrix pair (A, B).

As noted in [6], if an iterative method is applied directly to a real formulation of (1), the convergence rate may be worse than for the original complex linear system. As the known preconditioners for complex linear systems are relatively limited, one could hope to find a more efficient preconditioner for the real formulation of the problem, with or without utilizing the two-by-two block structure of the matrices in the real formulation. We refer to [7–9] for some examples of preconditioners for the real case, see also [10].

In this work, based on the particular form of the matrix in (2), we show that for solving such a system in two-by-two block form with square block matrices there exists a very efficient preconditioner that requires solution only of matrices that are linear combinations, $A + \alpha B$, of A and B where α is a parameter. Often $A + \alpha B$, is better conditioned than A itself. This method has been published in [11] and [1], where the method was named the 'C–to–R ' method. See also references in these publications.

The preconditioned system can be solved using various iterative solution methods, such as Krylov subspace iteration methods. To lower the solution cost when solving the arising inner systems with matrix $A + \alpha B$ it can be efficient to use inner iteration methods, such as a preconditioned conjugate gradient method, when the matrix is symmetric and positive definite. For the outer iteration method to solve the two-by-two block system it is then advisable to use a flexible version of GMRES, FGMRES ([12, 13]) or a variable preconditioned version of GCG [14], see also [15]. If eigenvalue bounds are known one can even use a Chebyshev semi-iteration method, thereby avoiding computation of inner products and global communication, which becomes of particular interest when applied on multi/many core computers.

Another method, that has more recently gained attention, is based on an operator splitting, alternating direction type of method, described in [16, 17] and [19]. It has the form of a stationary (fixed point) iteration method,

$$\begin{aligned}(\alpha V + A)\mathbf{x}^{k+1/2} &= (\alpha V - iB)\mathbf{x}^k + \mathbf{f} + i\mathbf{g}\\ (\alpha V + B)\mathbf{x}^{k+1} &= (\alpha V + iA)\mathbf{x}^{k+1/2} - i\mathbf{f} + \mathbf{g},\ k = 0, 1, \ldots .\end{aligned}$$

Here V is a preconditioner, chosen as a symmetric and positive definite (spd) matrix if A and B are symmetric and positive semidefinite (spsd). The method is referred to as PMHSS (Preconditioned Modified Hermitian and Skew–Hermitian Splitting method). As before, α is a given preconditioning parameter. It is also possible to rewrite the method as a preconditioner for a GMRES method. This method still involves some complex arithmetics, but to a lesser extent than if a method is applied directly for (1). For the choice $V = I$, the method has been presented and discussed in [18] and [19]. The skew-symmetric splitting method was actually discussed earlier in [20] in a more general framework for non-Hermitian linear systems with a dominant positive definite Hermitian part.

In [7] an iteration method is used, based on the skew–symmetric splitting

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} + \begin{bmatrix} 0 & -B \\ B & 0 \end{bmatrix}.$$

This method can be efficient only if the symmetric part dominates the skew–symmetric part which, however, does not hold in general.

As pointed out in [21], if the complex inner product $\mathbf{x}^*\mathbf{y}$ is replaced by $\mathbf{x}^T\mathbf{y}$, complex symmetric systems can be solved by a classical conjugate gradient method, that is, with short recurrences - a form of the Biconjugate Gradient (BiCG) method. However, this method breaks down if $\mathbf{x}^T\mathbf{x} = 0$ for some complex vector $\mathbf{x} \neq 0$. To some extent, this can be cured by use of a look–ahead strategy, see [22] and [23].

It is also possible to solve complex valued systems with a related algorithm, namely the unpreconditioned or preconditioned Quasi–Minimal–Residual (QMR) method (cf. [24]). The QMR algorithm can be seen as a stabilized version of the Biconjugate Gradient (BiCG) method using look-ahead techniques. Similarly to BiCG, QMR requires matrix-vector multiplications with both the coefficient matrix and its transpose. The main idea behind this algorithm is to solve the arising reduced triangular system in a least square sense, similar to the approach followed in GMRES and GCG. The difference is that the Krylov subspace basis vectors are bi-orthogonal rather than orthogonal as in the GMRES and GCG methods, so the obtained solution

can be viewed as a quasi-minimal residual solution. QMR uses look-ahead techniques to avoid breakdowns in the underlying Lanczos process. The method gives less reduction of the condition number, it may need additional computations due to the look-ahead and requires the action of the transpose of the preconditioner, which could be disadvantageous. However, a transpose-free version of QMR exists [25].

In this paper we focus on methods that are fully robust, i.e., more generally applicable, independent of various problem, discretization and method parameters, and with a nearly optimal order of computational complexity. The purpose of the present paper is to further develop some of the methods and to make a theoretical and a thorough numerical comparison of them.

The remainder of the paper is organized as follows. We present in Section 2 preconditioners for the two-by-two block system (2) with square blocks as arising in the C–to–R method, and derive the rate of convergence when applied for a suitable iterative method, such as a generalized conjugate gradient method or a Chebyshev semi-iterative method. Section 3 contains a description of the Preconditioned Modified Hermitian Skew-Hermitian Splitting (PMHSS) method and the related eigenvalue analysis. In Section 4, various examples where complex valued linear systems arise are presented while Section 5 contains the corresponding numerical results. The final section contains a summary and concluding comparisons of the methods.

2 Preconditioning methods for two-by-two block matrices with square blocks

To solve linear systems with two-by-two block matrices, such as arising from complex valued linear systems and also in other important problems (see, e.g. [6, 7]), we present now two types of preconditioning methods. We assume that $A + \alpha B$, $\alpha > 0$ is nonsingular, where A and B are given in (2). The first is based on a reduction of the system to Schur complement form, where we assume that A is also nonsingular. The Schur complement is then solved by a preconditioned iterative solution method. For this method we present two variants. Both variants involve an inner system to be solved when computing the residuals that arise at each iteration. Since the residuals must be computed sufficiently accurately, the solution of the corresponding inner systems must be done accurately. The second method is not based on a reduction to Schur complement form and involves inner systems only to solve the action of the inverse of the preconditioner, which can take place by iteration and with a not particularly small stopping criteria, thereby saving computational efforts.

2.1 Reduction to Schur complement form

Assume first that A and $A + \alpha B$, $\alpha > 0$ are nonsingular. The system (2) can be reduced to a Schur complement form,

$$(A + BA^{-1}B)\mathbf{x} = \widetilde{\mathbf{f}}, \tag{4}$$

where $\widetilde{\mathbf{f}} = \mathbf{f} + BA^{-1}\mathbf{g}$.

For the iterative solution of the reduced system we use a preconditioner in a form presented already in [5],

$$C_\alpha = (A + \alpha B)A^{-1}(A + \alpha B),$$

where $\alpha > 0$ is a parameter to be chosen.

The convergence of this iterative solution method to solve (4) depends on the eigenvalues λ of $M_\alpha \equiv C_\alpha^{-1}(A + BA^{-1}B)$, i.e., of the generalized eigenvalue problem

$$\lambda C_\alpha \mathbf{z} = (A + BA^{-1}B)\mathbf{z},\ \mathbf{z} \neq 0. \tag{5}$$

Although somewhat restricted, for the analysis we assume that the matrix $\widetilde{B} = A^{-1}B$ is a normal matrix, i.e., with a complete spectrum and eigenvalues μ, $\mu A\mathbf{z} = B\mathbf{z}$, $\mathbf{z} \neq 0$. It follows from (5) that

$$\lambda(I + \alpha\widetilde{B})^2\mathbf{z} = (I + \widetilde{B}^2)\mathbf{z}.$$

Hence,

$$\lambda = \frac{1 + \mu^2}{(1 + \alpha\mu)^2}\,. \tag{6}$$

Clearly, $\lambda \neq 0$ if and only if $\mu \neq \pm i$. A related result for two-by-two block matrices with square blocks is derived in [26] in the context of problems arising in PDE-constrained optimization. The structure of the matrices, that arise there, is similar to that in the C–to–R approach, up to some additional coefficients. There, the authors derive an approximation of the Schur complement of the form

$$S_{PW} = (A + B)A^{-1}(A + B) \tag{7}$$

and show that for symmetric and positive definite A and B, S_{PW} is spectrally equivalent to $S = A + BA^{-1}B$, implying that all the eigenvalues of $S_{PW}^{-1}S$ belong to the interval [0.5, 1], independently of the discretization parameter and the involved scalar coefficients.

Apart from the matrix-vector multiplications with A and B, the computational cost of the preconditioner, derived in [26], consists of two solutions with $A + B$ and one solution with A. Below we show that for the considered type of matrix structures the solution with A can be avoided and reduction to Schur complement form is unnecessary.

Case: A spd, B spsd

Assume first that A is spd and B is spsd. It follows then that $\mu \geq 0$ and the eigenvalues λ are real and positive. Hence, one can use a preconditioned version of the classical conjugate gradient method (see, e.g. [27]), that is, letting the inner products $(\mathbf{x}, \mathbf{y})$ be defined by the matrix C_α, i.e., $(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T C_\alpha \mathbf{y}$.

The optimal value of the parameter α to minimize the spectral condition number of $C_\alpha^{-1}(A + BA^{-1}B)$ can be determined as follows.

Proposition 1 *Assume that A is spd and B is spsd. Then, the extreme eigenvalues of the preconditioned matrix M_α, defined in (5), satisfy*

$$\lambda_{\min} = \begin{cases} \dfrac{1}{1+\alpha^2}, & \text{if } 0 \le \alpha \le \widehat{\mu} \\ \dfrac{1+\widehat{\mu}^2}{(1+\alpha\widehat{\mu})^2}, & \text{if } \widehat{\mu} \le \alpha \end{cases} \qquad \lambda_{\max} = \begin{cases} 1, & \text{if } \widehat{\alpha} \le \alpha \\ \dfrac{1+\widehat{\mu}^2}{(1+\alpha\widehat{\mu})^2}, & \text{if } 0 \le \alpha \le \widehat{\alpha} \end{cases}$$

where $\widehat{\mu}$ is the maximal eigenvalue of $A^{-1}B$, i.e., $A^{-\frac{1}{2}}BA^{-\frac{1}{2}} \le \widehat{\mu} I$ and

$$\widehat{\alpha} = \frac{\widehat{\mu}}{1+\sqrt{1+\widehat{\mu}^2}}.$$

The spectral condition number of M_α is minimized when $\alpha = \widehat{\alpha}$, in which case

$$\mu_{\min} = \frac{1}{1+\widehat{\alpha}^2}, \quad \mu_{\max} = 1 \quad Cond(M_\alpha) = 1+\widehat{\alpha}^2 = 2\frac{\sqrt{1+\widehat{\mu}^2}}{1+\sqrt{1+\widehat{\mu}^2}}.$$

Proof See [11]. □

General case: A and $A+B$ nonsingular

From (6) one can estimate the location of the eigenvalues for more general problems as follows. Since it may then turn out to be more difficult to estimate the optimal value of α, we choose $\alpha = 1$, which equals the limit of $\widehat{\alpha}$ in Proposition 1 when $\widehat{\mu} \to \infty$. We assume that $\mu = \mu_0 e^{i\phi}$ with $0 \le |\phi| \le \phi_0 < \frac{\pi}{2}$, that is, we assume that the real part of μ is positive. For $\alpha = 1$ it holds that

$$\lambda = \lambda(\mu) = \frac{1+\mu^2}{(1+\mu)^2} = \frac{1}{2} + \frac{1}{2}\left(\frac{\mu-1}{\mu+1}\right)^2,$$

where

$$\frac{\mu-1}{\mu+1} = \frac{(\mu-1)(\overline{\mu}+1)}{(\mu+1)(\overline{\mu}+1)} = \frac{|\mu|^2 - 1 + (\mu-\overline{\mu})}{|\mu|^2 + 1 + (\mu+\overline{\mu})}.$$

The latter shows that the eigenvalues λ are located in a disc in the complex plane with center close to $\frac{1}{2}$ and with radius less than $\frac{1}{2}$. We give now more precise estimates.

Proposition 2 *Let A and $A+B$ be nonsingular and let μ denote eigenvalues of $A^{-1}B$. Let $\mu = \mu_0 e^{i\phi}$, and assume that $0 \le |\phi| \le \phi_0 < \frac{\pi}{2}$. Then the eigenvalues $\lambda(\mu)$ in (6) are located in a disc, centered at $\frac{1}{2}(1+\delta)$ with radius $\frac{1}{2}(1-\delta)$, where $\delta = \dfrac{\cos\phi_0}{1+\cos\phi_0}$. The lower bound of the real part of the eigenvalues equals δ.*

Proof Rewrite $\lambda(\mu)$ as

$$\lambda(\mu) = 1 - \frac{2\mu}{(\mu+1)^2} = 1 - \frac{1}{1+\frac{1}{2}(\mu+\frac{1}{\mu})},$$

where $\frac{1}{2}(\mu + \frac{1}{\mu}) = a\cos\phi + ib\sin\phi$, $a = \frac{1}{2}(\mu_0 + \frac{1}{\mu_0})$, $b = \frac{1}{2}(\mu_0 - \frac{1}{\mu_0})$. Clearly, $a \geq 1$ and $|b| < a$. Hence,

$$\frac{1}{1 + \frac{1}{2}(\mu + \frac{1}{\mu})} = \frac{1 + a\cos\phi - ib\sin\phi}{(1 + a\cos\phi)^2 + (b\sin\phi)^2}.$$

For the real part of λ it holds

$$1 > Re(\lambda) = 1 - \frac{1 + a\cos\phi}{(1 + a\cos\phi)^2 + (b\sin\phi)^2} \geq 1 - \frac{1}{1 + a\cos\phi} \geq \frac{a\cos\phi_0}{1 + a\cos\phi_0} \equiv \widetilde{\delta}.$$

By the assumptions made, for all $a \geq 1$ there holds $\widetilde{\delta} \geq \delta = \dfrac{\cos\phi_0}{1 + \cos\phi_0} > 0$.

For the imaginary part it holds

$$|Im(\lambda)| = \frac{|b\sin\phi|}{(1 + a\cos\phi)^2 + (b\sin\phi)^2} \leq \frac{\widetilde{b}}{(1 + \widetilde{a})^2 + \widetilde{b}^2},$$

with $\widetilde{a} = a\cos\phi_0$ and $\widetilde{b} = |b\sin\phi_0|$. A computation shows that

$$|Im(\lambda)| \leq \frac{1 + \widetilde{a}}{2(1 + \widetilde{a})^2} = \frac{1}{2(1 + a\cos\phi_0)} = \frac{1}{2}\left(1 - \frac{a\cos\phi_0}{1 + a\cos\phi_0}\right) \leq \frac{1}{2}(1 - \delta).$$

□

It follows that the smaller ϕ_0 is, the smaller the convergence factor, $\max\limits_{\mu} |\lambda(\mu)|$, becomes. If $\phi_0 = 0$, then $\delta \geq \frac{1}{2}$ and the eigenvalues λ are contained in the interval $\left[\frac{1}{2}, 1\right]$, which is in accordance with the result from Proposition 1.

When the eigenvalue bounds are known one can use a Chebyshev semi-iteration method instead on the conjugate gradient method, thereby avoiding computation of inner products and global communication thereof to all computer processor cores.

Avoiding the inner systems with the matrix A

The above method involves an inner system with matrix A to be solved at each iteration. In some problems, it might be most efficient to use a direct solution of these systems. However, in other problems A may be less well–conditioned but the linear combination, $A + \alpha B$ may be better conditioned. Note also that the preconditioner involves two solutions with the latter matrix. We show now that the system matrix $C_\alpha^{-1}(A + BA^{-1}B)$ can be rewritten in a form where there is only one preconditioned system with matrix $A + \alpha B$ and the inner system with matrix A has been replaced by $A + \alpha B$. To this end, we first note that using matrix commutativity,

$$\begin{aligned} &A(A + \alpha B)^{-1}BA^{-1}B = A(I + \alpha A^{-1}B)^{-1}A^{-1}BA^{-1}B \\ &= B(I + \alpha A^{-1}B)^{-1}A^{-1}B = B(A + \alpha B)^{-1}B\,. \end{aligned}$$

Further, we rewrite

$$A(A + \alpha B)^{-1}A = A(A + \alpha B)^{-1}(A + \alpha B - \alpha B) = A - \alpha A(A + \alpha B)^{-1}B$$

and

$$I - A(A+\alpha B)^{-1} = (A+\alpha B - A)(A+\alpha B)^{-1} = \alpha B(A+\alpha B)^{-1}.$$

It follows that the preconditioned system

$$C_\alpha^{-1}[(A+BA^{-1}B)\mathbf{x} - \mathbf{f} - BA^{-1}\mathbf{g}] = 0$$

can be rewritten as

$$\begin{aligned}
&(A+\alpha B)^{-1}[A(A+\alpha B)^{-1}(A\mathbf{x} - \mathbf{f} - BA^{-1}\mathbf{g} + BA^{-1}B\mathbf{x})] \\
&\quad = (A+\alpha B)^{-1}[A(A+\alpha B)^{-1}(A\mathbf{x} - \mathbf{f} - BA^{-1}\mathbf{g}) + B(A+\alpha B)^{-1}B\mathbf{x}] \\
&\quad = (A+\alpha B)^{-1}[(A - \alpha A(A+\alpha B)^{-1}B + B(A+\alpha B)^{-1}B)\mathbf{x} \\
&\qquad -(A(A+\alpha B)^{-1} - I)\mathbf{f} - \mathbf{f} - A(A+\alpha B)^{-1}BA^{-1}\mathbf{g}] \\
&\quad = (A+\alpha B)^{-1}[(A - \alpha B + \alpha(I - A(A+\alpha B)^{-1})B + B(A+\alpha B)^{-1}B)\mathbf{x} \\
&\qquad -\mathbf{f} + \alpha B(A+\alpha B)^{-1}\mathbf{f} - BA^{-1}(I+\alpha BA^{-1})^{-1}\mathbf{g}] \\
&\quad = (A+\alpha B)^{-1}[(A - \alpha B + (1+\alpha^2)B(A+\alpha B)^{-1}B)]\mathbf{x} \\
&\qquad -\mathbf{f} - B(A+\alpha B)^{-1}(\mathbf{g} - \alpha\mathbf{f})] = 0\,. \qquad (8)
\end{aligned}$$

This is the same equation that was derived in the following way in [11]. Rewrite

$$\begin{cases} A\mathbf{x} - B\mathbf{y} = \mathbf{f} \\ B\mathbf{x} + A\mathbf{y} = \mathbf{g} \end{cases}$$

in the form

$$\begin{cases} (A - \alpha B)\mathbf{x} + \sqrt{1+\alpha^2}\, B\widetilde{\mathbf{y}} = \mathbf{f} \\ \sqrt{1+\alpha^2}\, B\mathbf{x} - (A+\alpha B)\widetilde{\mathbf{y}} = \widetilde{\mathbf{g}}, \end{cases}$$

where $\widetilde{\mathbf{y}} = \dfrac{\alpha\mathbf{x} - \mathbf{y}}{\sqrt{1+\alpha^2}}$, $\widetilde{\mathbf{g}} = \dfrac{\mathbf{g} - \alpha\mathbf{f}}{\sqrt{1+\alpha^2}}$ and $\alpha > 0$ is a parameter. Here the Schur complement system takes the same form as in (8).

We see that here $A + \alpha B$ arises as inner system when evaluating the residuals ($\mathbf{r}$) for the outer iterative solution method, namely

$$\mathbf{r} = (A - \alpha B)\mathbf{x} + B(A+\alpha B)^{-1}((1+\alpha^2)B\mathbf{x} - \mathbf{g} + \alpha\mathbf{f}) - \mathbf{f}. \qquad (9)$$

The outer iteration preconditioner is $A + \alpha B$.

2.2 The two-by-two block matrix and its preconditioner

The above version of the C–to–R method requires accurate inner solvers to compute the Schur complement residuals. If one instead solves the coupled two-by-two block preconditioner, one avoids this problem as the residuals are computed from the given, unreduced system. Furthermore, for this approach there exists a very efficient preconditioner.

To show this, consider matrices of the form

$$\mathcal{A} = \begin{bmatrix} A & aB^T \\ -bB & A \end{bmatrix},$$

where a, b are real numbers such that $ab > 0$.

Note that A and B are square matrices. We assume that A and $B+B^T$ are positive semidefinite and

$$\ker(A) \cap \ker(B) = \{\emptyset\} \ .$$

It follows readily that under these assumptions, $\mathcal{A}$ is nonsingular. Namely, if $\mathcal{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}$, then

$$A\mathbf{x} + aB^T\mathbf{y} = 0 \ \text{ and } \ -bB\mathbf{x} + A\mathbf{y} = 0$$

so

$$\frac{1}{a}\mathbf{x}^*A\mathbf{x} + \mathbf{x}^*B^T\mathbf{y} = 0 \ \text{ and } \ \frac{1}{b}\mathbf{y}^*A\mathbf{y} - \mathbf{y}^*B\mathbf{x} = 0\,,$$

that is, $\frac{1}{a}\mathbf{x}^*A\mathbf{x} + \frac{1}{b}\mathbf{y}^*A\mathbf{y} = 0$ which, since A is positive semidefinite, implies $\mathbf{x}, \mathbf{y} \in \ker(A)$. But then

$$B\mathbf{x} = 0 \quad \text{and } B^T\mathbf{y} = 0\,, \qquad \text{hence } \mathbf{x} = \mathbf{y} = \mathbf{0},$$

i.e., the singular system has only the trivial solution. We let

$$\mathcal{B} = \begin{bmatrix} A & aB^T \\ -bB & A + \sqrt{ab}(B + B^T) \end{bmatrix},$$

be a preconditioner to $\mathcal{A}$. Clearly $\mathcal{B}$ is also nonsingular.

As shown in [11, 28], its inverse can be written in the explicit form

$$\mathcal{B}^{-1} = \begin{bmatrix} H_1^{-1} + H_2^{-1} - H_2^{-1}AH_1^{-1} & \sqrt{\frac{a}{b}}\left(I - H_2^{-1}A\right)H_1^{-1} \\ -\sqrt{\frac{b}{a}}H_2^{-1}\left(I - AH_1^{-1}\right) & H_2^{-1}AH_1^{-1} \end{bmatrix},$$

where $H_i = A + \sqrt{ab}B_i$, $i = 1, 2$ and $B_1 = B, B_2 = B^T$. Besides some matrix-vector multiplications and vector additions, it follows readily that an action of $\mathcal{B}^{-1}$ involves just one solution of a system with each of the matrices H_1 and H_2, namely, the computation of $\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathcal{B}^{-1}\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\end{bmatrix}$ can take place in the order depicted in Algorithm 1.

Algorithm 1

(i) *Solve* $H_1\mathbf{g} = \mathbf{f}_1 + \sqrt{\frac{a}{b}}\mathbf{f}_2$.
(ii) *Compute* $A\mathbf{g}$ *and* $\mathbf{f}_1 - A\mathbf{g}$.
(iii) *Solve* $H_2\mathbf{h} = \mathbf{f}_1 - A\mathbf{g}$.
(iv) *Compute* $\mathbf{x} = \mathbf{g} + \mathbf{h}$ *and* $\mathbf{y} = -\sqrt{\frac{b}{a}}\mathbf{h}$.

We derive now eigenvalue bounds for $\mathcal{B}^{-1}\mathcal{A}$.

Proposition 3 *Let A and $B+B^T$ be symmetric and positive semi-definite and assume that* $\ker(A) \cap \ker(B) = \{\emptyset\}$.

(i) *Then the eigenvalues* λ *of* $\mathcal{B}^{-1}\mathcal{A}$ *satisfy* $\frac{1}{2} \leq \frac{1}{1+q} \leq \lambda \leq 1$, *where*

$$q = \sup_{\widetilde{\mathbf{x}},\mathbf{y}} \frac{2|\widetilde{\mathbf{x}}^*(B+B^T)\mathbf{y}|}{\widetilde{\mathbf{x}}^*(B+B^T)\widetilde{\mathbf{x}} + \mathbf{y}^*(B+B^T)\mathbf{y}} \leq 1,$$

where $\widetilde{\mathbf{x}} = \sqrt{\frac{b}{a}}\mathbf{x}$ *and* $\mathbf{x}, \mathbf{y}$ *are eigenvectors of the generalized eigenvalue problem* $\lambda\mathcal{B}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathcal{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$. *Here* $\lambda = 1$ *if and only if* $\mathbf{y} \in \mathcal{N}(B+B^T)$.

(ii) *If* A *is symmetric and positive definite, then*

$$\max\left\{\frac{1}{1+q}, \frac{1}{1+\sqrt{ab}\sigma_0}\right\} \leq \lambda \leq 1\,,$$

where $\sigma_0 = \sigma(A^{-1/2}(B+B^T)A^{-1/2})$ *and* $\sigma(.)$ *denotes the spectral radius.*

Proof The generalized eigenvalue problem leads to

$$\left(\frac{1}{\lambda} - 1\right)\mathcal{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}0\\ \sqrt{ab}(B+B^T)\mathbf{y}\end{bmatrix}.$$

Using the similarity (scaling) transformation $D\mathcal{A}\mathcal{D}^{-1}$, where $\mathcal{D} = \begin{bmatrix}\sqrt{\frac{b}{a}}\,I & 0\\ 0 & I\end{bmatrix}$ leads to

$$\left(\frac{1}{\lambda} - 1\right)\begin{bmatrix}A & \sqrt{ab}\,B^T\\ -\sqrt{ab}\,B & A\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{x}}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}0\\ \sqrt{ab}\,(B+B^T)\mathbf{y}\end{bmatrix}, \tag{10}$$

where $\widetilde{\mathbf{x}} = \sqrt{\frac{b}{a}}\,\mathbf{x}$. Multiplying (10) with $(\widetilde{\mathbf{x}}^*, \mathbf{y}^*)$ we obtain

$$\left(\frac{1}{\lambda} - 1\right)(\widetilde{\mathbf{x}}^*A\widetilde{\mathbf{x}} + \sqrt{ab}\,\widetilde{\mathbf{x}}^*B^T\mathbf{y} - \sqrt{ab}\,\mathbf{y}^*B\widetilde{\mathbf{x}} + \mathbf{y}^*A\mathbf{y}) = \sqrt{ab}\,\mathbf{y}^*(B+B^T)\mathbf{y}\,,$$

or

$$\left(\frac{1}{\lambda} - 1\right)(\widetilde{\mathbf{x}}^*A\widetilde{\mathbf{x}} + \mathbf{y}^*A\mathbf{y}) = \sqrt{ab}\,\mathbf{y}^*(B+B^T)\mathbf{y}\,. \tag{11}$$

Since A and $B+B^T$ are positive semidefinite and have no common nullspace vector, it follows that $\lambda \leq 1$. Further it is seen that $\lambda = 1$ if and only if

$$\mathbf{y} \in \mathcal{N}(B+B^T)\,.$$

For $\lambda \neq 1$ and multiplying (10) now with $(\mathbf{y}^*, -\widetilde{\mathbf{x}}^*)$, we find

$$\left(\frac{1}{\lambda} - 1\right)(\mathbf{y}^*A\widetilde{\mathbf{x}} + \sqrt{ab}\mathbf{y}^*B^T\mathbf{y} + \sqrt{ab}\widetilde{\mathbf{x}}^*B\mathbf{x} - \widetilde{\mathbf{x}}^*A\mathbf{y}) = -\sqrt{ab}\widetilde{\mathbf{x}}^*(B+B^T)\mathbf{y}\,,$$

or, since A is symmetric,

$$\left(\frac{1}{\lambda} - 1\right)(2\mathbf{y}^*B^T\mathbf{y} + 2\mathbf{x}^*B\mathbf{x}) = -2\mathbf{x}^*(B+B^T)\mathbf{y}\,.$$

Since $\mathbf{y}^* B^T \mathbf{y} = \mathbf{y}^* B \mathbf{y}$ and $\widetilde{\mathbf{x}}^* B \widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^* B^T \widetilde{\mathbf{x}}$, it follows that

$$\left(\frac{1}{\lambda} - 1\right) (\mathbf{y}^*(B + B^T)\mathbf{y} + \widetilde{\mathbf{x}}^*(B + B^T)\mathbf{x}) = -2\widetilde{\mathbf{x}}^*(B + B^T)\mathbf{y}$$

or

$$\frac{1}{\lambda} - 1 \leq q = \sup_{\widetilde{\mathbf{x}},\mathbf{y}} \frac{2 \mid \widetilde{\mathbf{x}}^*(B + B^T)\mathbf{y} \mid}{\widetilde{\mathbf{x}}^*(B + B^T)\widetilde{\mathbf{x}} + \mathbf{y}^*(B + B^T)\mathbf{y}} \leq 1 .$$

To prove part (ii), if A is spd it follows from (11) that

$$\frac{1}{\lambda} - 1 \leq \sqrt{ab}\, \sigma(\widetilde{B} + \widetilde{B}^T)$$

where

$$\widetilde{B} = A^{-1/2} B A^{-1/2} ,$$

which proves the second lower bound. □

Remark 1 The estimates in Proposition 3 involve coefficients q and σ_0 that are not easily computable. However, the estimates show that the lower bound of the eigenvalues depends on the maximum of two quantities and since $q \leq 1$ it follows that $\lambda \geq \frac{1}{2}$.

Remark 2 The result in Proposition 2 holds also for the two-by-two block preconditioned matrix $B^{-1}A$, i.e., if the roles of A and B are reversed.

2.3 Inner iteration stopping criteria

If residuals in an iterative method are not computed sufficiently accurately, for instance to obtain a reliable stopping criteria, the convergence may stall, i..e. the iteration method many cease to converge after some iteration error bound has been reached. Therefore, the inner systems with the matrix A in (4) and $A+\alpha B$ in (9) must be computed with a sufficiently small stopping criteria. However, the corresponding matrices in Algorithm 1 appear only in the preconditioner and to save computational effort, the systems arising in the preconditioner can be computed by iteration less accurately, i.e., with a larger stopping tolerance.

When the arising inner systems are solved exactly, it follows from Propositions 1 and 3 that the eigenvalues of the preconditioned matrix are contained in an interval $[a, 1], a \geq \frac{1}{2}$, in the case of spsd matrices. For a reasonably small inner iteration stopping criteria, one can show that the eigenvalues are contained in a narrow ellipse about the interval $[a, 1]$, such as an ellipse with foci $(a, 0), (1, 0)$ and with a small eccentricity (ratio of semi-axes). As shown, e.g. in [27], a generalized CG method will then still converge rapidly, typically with just one or two more iterations. The parameters in a Chebyshev semi-iteration can be based on the eccentricity of this ellipse, see e.g. [27]. As remarked previously, the use of Chebyshev semi-iteration avoids computations of inner products and reduces the global communication of them, as otherwise are needed in Krylov subspace methods.

3 The preconditioned modified Hermitian and skew-Hermitian splitting method

Following and augmenting the results in [19], we present now the PMHSS method to solve the complex linear system in (1). As pointed out in the introduction, it still involves complex arithmetic but for reasons of comparison we include here a brief description and an analysis of it. The method takes the form of a stationary iterative method,

$$\begin{aligned}(\alpha V + A)\mathbf{x}^{k+1/2} &= (\alpha V - iB)\mathbf{x}^k + \mathbf{b} \\ (\alpha V + B)\mathbf{x}^{k+1} &= (\alpha V + iA)\mathbf{x}^{k+1/2} - i\mathbf{b}\end{aligned} \tag{12}$$

where α is a given positive parameter, V a prescribed symmetric positive definite matrix and i is the imaginary unit. We assume that A is spd and B is spsd. It follows that both $\alpha V + A$ and $\alpha V + B$ are spd. For $V = I$ the method reduces to the modified Hermitian and skew-Hermitian splitting method, presented in [18]. One possible choice of V is A.

The method in (12) can be written in a compact form as

$$\mathbf{x}^{k+1} = \mathcal{L}(V;\alpha)\mathbf{x}^k + \mathcal{R}(V;\alpha)\mathbf{b}\,, \quad k = 0, 1, 2, \ldots \tag{13}$$

where the iteration matrix, $\mathcal{L}(V;\alpha)$, has the form

$$\mathcal{L}(V;\alpha) = (\alpha V + B)^{-1}(\alpha V + iA)(\alpha V + A)^{-1}(\alpha V - iB)$$

and

$$\mathcal{R}(V;\alpha) = (1-i)\alpha(\alpha V + B)^{-1}V(\alpha V + A)^{-1}\,.$$

To analyse the convergence factor we note that

$$\begin{aligned}\sigma(\alpha) = \|\mathcal{L}(V;\alpha)\| &\le \max_{\lambda\in sp(V^{-1}A)} \frac{\sqrt{\alpha^2+\lambda^2}}{\alpha+\lambda} \max_{\mu\in sp(V^{-1}B)} \frac{\sqrt{\alpha^2+\mu^2}}{\alpha+\mu} \\ &\le \max_{\lambda\in sp(V^{-1}A)} \frac{\sqrt{\alpha^2+\lambda^2}}{\alpha+\lambda} < 1\,, \quad \forall\alpha > 0\,.\end{aligned}$$

In particular, for the choice $\alpha = \widehat{\alpha} = \sqrt{\lambda_{\min}\lambda_{\max}}$, where $\lambda_{\min}$, $\lambda_{\max}$ are the extreme eigenvalues of $V^{-1}A$, it follows readily that

$$\sigma(\widehat{\alpha}) \le \frac{\sqrt{\kappa(V^{-1}A)+1}}{\sqrt{\kappa(V^{-1}A)}+1},$$

where $\kappa(V^{-1}A)$ denotes the spectral condition number of $V^{-1}A$.

The smallest convergence factor is achieved for $V = A$, in which case

$$\sigma(\widehat{\alpha}) = \frac{\sqrt{2}}{2}\,. \tag{14}$$

The iteration matrix takes then the form

$$\mathcal{L}(\alpha) := \mathcal{L}(A;\alpha) = \frac{\alpha+i}{\alpha+1}(\alpha A + B)^{-1}(\alpha A - iB) \tag{15}$$

and

$$\mathcal{R}(\alpha) := \mathcal{R}(A;\alpha) = \frac{\alpha(1-i)}{\alpha+1}(\alpha A + B)^{-1}.$$

Here

$$\sigma(\widehat{\alpha}) = \rho(\mathcal{L}(\alpha)) \leq \frac{\sqrt{\alpha^2+1}}{\alpha+1} < 1\,,$$

where $\rho(\cdot)$ denotes the spectral radius. Clearly, the upper bound in (14) is independent of the size of the problem.

From the form of $\mathcal{L}(\alpha)$ in (15) it is seen that the eigenvalues are located in a disc in the complex plane, centered at the unit value and radius $r = \frac{\sqrt{\alpha^2+1}}{\alpha+1}$. The radius is smallest for $\alpha = 1$, when $r = \frac{\sqrt{2}}{2}$.

Since the eigenvalues are contained in a disc with center at unity in the complex plane, it is not possible to improve the rate of convergence by use of a Chebyshev semi-iteration method based on a circumscribing ellipse, see e.g. [27] regarding Chebyshev iterations for complex eigenvalues. Hence, the convergence factor remains equal to $\frac{\sqrt{2}}{2} \approx 0.707$.

In the version designed for solving the complex linear system (1), the PMHSS iteration method deals with real matrices and is a useful preconditioned modification of the HSS iteration method initially introduced in [16] and [17] for solving non-Hermitian positive definite linear systems, see also [20] for an earlier presentation of this method. It naturally results in a preconditioner of a matrix splitting type, called the PMHSS preconditioner, to be used in Krylov subspace iteration methods such as GMRES, employed to solve the complex linear system (1).

When both A and B are symmetric positive semidefinite and satisfy $ker(A) \cap ker(B) = \{\emptyset\}$, with $ker(\cdot)$ being the null space of the corresponding matrix, the PMHSS iteration sequences converge to the unique solutions of the complex and the real linear systems (1) and (2), respectively. For a specific choice of the preconditioning matrix the convergence factor is bounded by $\sigma(\alpha) = \frac{\sqrt{\alpha^2+1}}{\alpha+1}$, and the eigenvalues of the PMHSS-preconditioned matrices are located in a complex disk centered at 1 with radius $\sigma(\alpha)$. Note that the function $\sigma(\alpha)$ is independent of the problem sizes and the input data, and attains the minimum $\frac{\sqrt{2}}{2}$ for $\alpha = 1$.

In the numerical examples we apply PMHSS in the form of (12) with $\alpha = 1$ and $V = A$ as a self-standing solver as well as a preconditioner for the GMRES method. As follows from (13), letting the initial approximation be $\mathbf{x}^0 = \mathbf{0} + i\mathbf{0}$, for this particular choice of the method parameters, the application of the PMHSS method simplifies significantly and becomes

$$\begin{aligned} (A+B)\mathbf{z} &= \mathbf{q} \\ \mathbf{x} &= 0.5 * (1-i)\mathbf{z}, \end{aligned} \tag{16}$$

where $\mathbf{q}$ is the current residual in the iterative solution method. Thus, PMHSS requires only one solution with $A+B$ and complex arithmetic, while C–to–R requires two solutions with $A + B$ and real arithmetic.

We analyse now a simplified version of (16) without the complex factor, namely, the properties of $A + B$ as a preconditioner for $\boldsymbol{C} = A + iB$, where A and B are real-valued matrices. Assume first that A is spd and B is spsd. Then the eigenvalues $\mu = \mu(A^{-1}B)$ are real and nonnegative. Assume that $\mu \leq 1$. If this does not hold, in some cases we can solve $(B - iA)\mathbf{x} = -i\mathbf{f}$ instead of $(A + iB)\mathbf{x} = \mathbf{f}$.

We see that $(A + B)^{-1}(A + iB) = I + (i - 1)(A + B)^{-1}B = I + (i - 1)(I + A^{-1}B)^{-1}(A^{-1}B)$. For the eigenvalues λ of $(A + B)^{-1}(A + iB)$ there holds

$$\lambda - 1 = (i - 1)\frac{\mu}{1 + \mu}.$$

It follows that

$$\frac{1}{2} \leq \frac{1}{1 + \mu_{max}} \leq Re(\lambda) \leq 1 \quad \text{and} \quad |Im(\lambda)| \leq \frac{\mu}{1 + \mu} \leq \frac{\mu_{max}}{1 + \mu_{max}} \leq \frac{1}{2}.$$

Hence, the eigenvalues are contained in a domain in the complex plane, as shown in Fig. 1.

For comparison, we recall, that the eigenvalues for the C–to–R method are real and contained in the interval $[\frac{1}{2}, 1]$.

Consider now the more general case, where the eigenvalues are complex, $\mu = \mu_0 e^{i\phi_0}$, $0 < \mu_0 \leq 1$, $0 \leq |\phi| \leq \phi_0 < \frac{\pi}{2}$.

Proposition 4 *Assume that $A^{-1}B$ is a normal matrix and that $\mu(A^{-1}B) = \mu_0 e^{i\phi_0}$, $0 < \mu_0 \leq 1$, $0 \leq |\phi| \leq \phi_0 < \frac{\pi}{2}$. Then the eigenvalues λ of $(A+B)^{-1}(A+iB)$ satisfy*

$$\frac{1}{2}(1 - \delta_0) \leq Re(\lambda) \leq 1, \quad |Im(\lambda)| \leq \frac{1}{2}(1 + \delta), \text{ where } \delta_0 = \frac{\sin\phi_0}{1 + cos\phi_0} < 1.$$

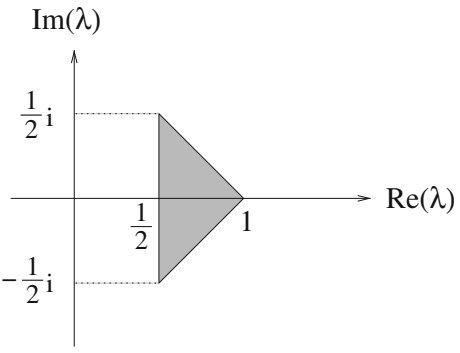

Fig. 1 Eigenvalues of $(A + B)^{-1}(A + iB)$

Proof It holds that

$$\begin{aligned}\lambda - 1 &= (i-1)\frac{\mu_0(\cos\phi + i\sin\phi)}{1+\mu_0(\cos\phi + i\sin\phi)} \\ &= (i-1)\frac{(\cos\phi + i\sin\phi)\left(\frac{1}{\mu_0}+\cos\phi - i\sin\phi\right)}{\left(\frac{1}{\mu_0}+\cos\phi\right)^2 + \sin^2\phi} = (i-1)\frac{\frac{1}{\mu_0}\cos\phi + 1 + i\frac{1}{\mu_0}\sin\phi}{\frac{1}{\mu_0^2}+\frac{2}{\mu_0}\cos\phi + 1} \\ &= \frac{i\left(1+\frac{1}{\mu_0}\cos\phi - \frac{1}{\mu_0}\sin\phi\right) - \left(1+\frac{1}{\mu_0}\cos\phi + \frac{1}{\mu_0}\sin\phi\right)}{1+\frac{1}{\mu_0}\cos\phi + \frac{1}{\mu_0}\left(\frac{1}{\mu_0}+\cos\phi\right)} = \frac{i(1-\widetilde{\delta}) - (1+\widetilde{\delta})}{1+\frac{\frac{1}{\mu_0}+\cos\phi}{\mu_0+\cos\phi}},\end{aligned}$$

where $\widetilde{\delta} = \frac{\sin\phi}{\mu_0+\cos\phi}$. It can be seen that the maximum value of $|Im(\lambda)|$ is taken for $\mu_0 = 1$ and $|Im(\lambda)| \le \frac{1}{2}(1+\delta_0)$. Further, for $\mu_0 = 1$, $Re(\lambda) \ge 1 - \frac{1}{2}(1+\widetilde{\delta}) \ge \frac{1}{2}(1-\delta_0)$. □

Comparing the eigenvalue estimates in Proposition 2 and 4, we see that the eigenvalues for the C–to–R method are contained in a smaller domain in the complex plane than those for the $(A+B)$-preconditioned matrix $\boldsymbol{C}$. In particular, the bound for the maximal modulus of the imaginary part of the eigenvalues is nearly twice larger.

4 Test examples

The presented methods are tested with matrices, originating from the benchmark problems described below. The arising matrices satisfy the hypothesis in Propositions 2 and 3. Without further knowledge of the eigenvalues of $A^{-1}B$ in Proposition 2 it is not possible to actually compute the eigenvalue bounds for the preconditioner. Nevertheless, for symmetric problems we have shown that there are very tight bounds of the eigenvalues. The propositions should be seen as giving insight into the behaviour of the eigenvalues, for instance as a function of the angle ϕ_0.

Problem 1 (Shifted ω systems) The matrices originate from the discrete parabolic problem (a test example from [11])

$$\frac{\partial v}{\partial t} - \Delta v = f(x,t),\ t > 0, \tag{17}$$

where Δ is the Laplacian operator and the forcing function is periodic in time, $f(x,t) = f_0(x,t)e^{i\omega t}$. Applying the Ansatz $v(x,t) = u(x)e^{i\omega t}$, where u and v are complex-valued functions, we reformulate (17) as

$$\frac{\partial u}{\partial t} - \Delta u + i\omega u = f_0(x,t). \tag{18}$$

Using an implicit time integration method, we must solve a system of the form

$$(K + i\omega\tau M)\mathbf{u} = \mathbf{b},$$

where $K = M + \tau L$, L is the discrete Laplace operator, M is either the identity or the mass matrix, depending on the corresponding space discretization, finite differences (FDM) or finite elements (FEM), and τ is the time-step. In the experiments τ is taken to be equal to the space discretization parameter.

For reasons of comparisons, to match the experiments in [11] and [19], we consider the simplified systems

$$(L + i\omega M)\mathbf{u} = \mathbf{b}. \tag{19}$$

The parameter ω is varied as 0.01, 1, 100.

Problem 2 (**A convection-diffusion problem with a periodic forcing term**) This problem is of the same form as Problem 1, but includes a convection term,

$$\frac{\partial v}{\partial t} - \varepsilon \Delta v + (\mathbf{b} \cdot \nabla) v = f(x, t), \ t > 0, \tag{20}$$

thus, the matrix K is of the form $K = M + \tau(\varepsilon L + B)$, where B originates from the discretization of the convective term $(\mathbf{b} \cdot \nabla)v$ for some given vector field $\mathbf{b}$. To be specific, we choose $\mathbf{b}$ to describe a rotating vortex, $\mathbf{b} = \begin{bmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{bmatrix}$. In the experiments, ε is chosen as 1 and 0.001.

For this test problem we include experiments with the simplified matrices $\boldsymbol{C} = M + i\omega\tau K$ as well as for $\boldsymbol{C} = M + \tau K + i\omega\tau M$ with $\tau = h$ for $\omega = 0.01, 1, 100$.

Problem 3 (**Padé approximations**) The systems to be considered have the following form

$$\left[M + \left(1 + \frac{1}{\sqrt{3}} i \right) \frac{\tau}{4} L \right] \mathbf{u} = \mathbf{b}, \tag{21}$$

The problem parameter τ is varied as $\tau = h$, where h is the characteristic mesh size in space.

As already noted, in Problems 1, 2 and 3 L is the matrix obtained when discretizing the negative Laplace operator $\mathcal{L}u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}$, $x \in \Omega^d$ by either standard central differences or conforming piece-wise linear finite elements on a regular triangular mesh. We consider here two and three space dimensions. For simplicity, the domain of definition is $\Omega = [0, 1]^d$, $d = 2, 3$. Homogeneous Dirichlet boundary conditions are applied on the whole boundary.

Problem 4 (**Schrödinger equation**) Consider the time-dependent Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = \widehat{H} \Psi(x, t) \tag{22}$$

where $\hbar$ is Planck's constant, divided by 2π and $\widehat{H}$ is the (quantum) Hamiltonian, expressing the kinetic and the potential energy operators of the system under

consideration. Without any further description, we follow the methodology from [2] and in particular, Paper VII ([29]). The system to be solved is of the form

$$(M + i\frac{\tau}{2}K)\mathbf{u}^{(n+1)} = (M - i\frac{\tau}{2}K)\mathbf{u}^{n},$$

where τ is the time step and $\mathbf{u}^n$, the solution of the previous time level, is assumed to be known.

The test matrices M and K for this problem are both spd and are taken from [29]. They originate from Radial Basis Functions used for the spatial discretization of (22) and are dense. The matrices are extremely ill-conditioned and the order of the condition number can reach 10^{18}.

Problem 5 (**Matrices from web-available matrix collections**) We test the following complex symmetric matrices available via the UF Sparse matrix collection [30]:

- 'mhd1280b':size 1280, 22778 complex nonzero elements
- 'mplate'; size 5962, 142190 complex nonzero elements
- 'windscreen': car windscreen, size 22 692, 1 482 390 nonzero elements.

5 Numerical experiments

All tests are performed in Matlab 7.12.0 (64 bit) on a laptop Lenovo Thinkpad T500, Intel Core(TM)2 Duo CPU T9400, 2.53GHz with 8GB RAM.

We compare the performance of Matlab's sparse direct solver '\', the preconditioned QMR, GMRES and GCG methods. We use the QMR and GMRES implementations, provided in Matlab and self-implemented GCG. The problem size in all tables is that of the original complex valued linear system. The experiments are done under the following conditions.

1. For a fair comparison we solve the original complex valued system (1). This is done via '\' (the column 'Direct/time'), by unpreconditioned or ILU-preconditioned QMR, PMHSS and PMHSS-preconditioned GMRES. In PMHSS, systems with $A + B$ are solved directly, as in [19].
2. Any time when a direct solver is used, a pre-ordering is first applied in order to avoid unnecessary fill-in and related higher computational cost. We use the symmetric approximate minimal degree (AMD) ordering, as available in Matlab (`symamd`). The same approach is used for the numerical experiments in [19].
3. The solution of the inner systems in PMHSS is done via a direct method (Cholesky decomposition), as in the numerical tests in [19] or ILU for the nonsymmetric matrices.
4. The twice larger real system (2) is solved via C–to–R -preconditioned GCGMR. Due to the included inner solution method with the matrix $A + B$, the preconditioner for the outer solver is slightly varying.

   As the implementation of GMRES, in the currently available version of Matlab, does not incorporate variable preconditioning, GMRES does not always

behave in a stable way for variable preconditioners and therefore it is not used for testing the C–to–R preconditioner.

5. Unless stated otherwise, the (outer) stopping criterion is 10^{-8} and the stopping tolerance for the inner iterative solver in C–to–R , whenever present, is 10^{-3}, both in relative residual norm.
6. As an inner solver for $A + B$, or H_1 and H_2 respectively, we use either a direct method or the aggregation-based algebraic Multigrid method AGMG. The construction, the properties and the implementation of AGMG are described in [31–33] and the references therein. The version of AGMG is precompiled, thus, its performance is comparable with that of Matlab's built-in functions.
7. The time required for constructing the preconditioners (AGMG, LU, LL^T or ILU) is included in the reported total solution time. The average number of inner iterations is reported in brackets. The cases when we use a direct solver for the blocks H_1 and H_2 are indicated by '(0)' inner iterations.
8. In some of the tables we present results for ILU-preconditioned QMR for the complex linear system. The preconditioner is of incomplete factorization type and is obtained via Matlab's function `'ilu'` or `ichol`. The factorization is of type `'ilutp'` for `'ilu'`, i.e., with threshold and pivoting, and with a dropping tolerance 10^{-3} for two-dimensional and 10^{-1} for three-dimensional problems. For `ichol` the type is `'milu'`, i.e., modified incomplete Cholesky factorization. The reported time includes the time for constructing the incomplete factorization and for the QMR iterations.

For each experiment in Tables 1–7 and 9, we have marked with bold the fastest execution time. We note, however, that this criterion alone is not fully characterising the corresponding solution method. In addition, we mention that due to the random right hand size, the execution times might vary slightly.

Tables 1, 2 and 3 present tests with the matrix $\boldsymbol{C} = L + i\omega M$, where $M = I$ in Table 1 and M is a mass matrix in Tables 2 and 3. Tables 4, 5 and 6 illustrate the behaviour of the C–to–R method for nonsymmetric blocks (the block B in Tables 4 and 5, and both blocks A and B in Table 6). In the cases when the convection is stronger ($\varepsilon = 0.001$), we use a direct inner solver. Table 7 shows comparisons for Padé-approximation matrices (Problem 4). Table 8 and Fig. 5 illustrate the performance of the methods for the very ill-conditioned dense matrices arising from Schrödinger equation, discretized using radial basis functions. Finally, Table 9 shows results for some symmetric indefinite matrices from [30].

Figures 2, 3 and 4 illustrate the behaviour of four of the considered solution methods from Table 2 (time vs reduction of the relative residual norm). We see that unpreconditioned versions of QMR and PMHSS may exhibit slow convergence and do not include those in further tests.

Table 2 presents results for Problem 1, discretized by FEM. We see that in this case, for $\omega = 0.01$ and $\omega = 1$ the direct method is slower than C–to–R and PMHSS. Both C–to–R and PMHSS are very robust in all cases and exhibit mesh-independent convergence. PMHSS is sometimes slightly faster since it needs only one solve with $A + B$ per iteration, solved directly, while C–to–R performs two such solves with AGMG(10^{-3}).

Table 1 Problem 1: 2D, FDM, $M = I$, $C = L + i\omega M$; inner solver AGMG

Problem size	Direct time	Unprec. QMR		ILU-QMR		C–to–R -GMRES		C–to–R -GCGMR		PMHSS		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.	iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$													
16384	**0.2323**	286	1.6286	15	0.8753	10(5)	0.6869	10(5)	0.9137	53	0.7601	16	0.4484
			7.0077e-5		2.5711e-5		0.0337		2.4289e-5		4.0862e-7		3.2595e-6
65536	**1.0387**	320	7.7714	17	4.6098	10(5)	1.9056	10(5)	1.758	53	3.8911	20	2.3188
			0.0002		5.1292e-5		0.0597		7.1978e-5		8.2452e-7		8.4799e-6
262144	**6.5873**	323	31.956	17	27.123	10(5)	8.7007	10(5)	7.8604	53	21.99	22	14.973
			0.0004		7.7611e-5		0.0981		0.0002		1.7457e-6		9.6584e-6
$\omega = 1$													
16384	0.1849	36	0.2328	4	0.3505	11(3)	0.3132	11(3)	**0.2268**	43	0.6187	20	0.5707
			7.8635e-7		4.784e-8		0.0024		8.8692e-7		1.0794e-6		2.0895e-7
65536	1.0297	36	**0.8697**	4	2.2528	11(3)	1.1559	11(3)	0.9576	43	3.2193	20	2.1965
			1.6157e-6		9.3011e-8		0.0035		1.886e-6		2.4527e-6		4.3283e-7
262144	6.8656	36	**3.4294**	4	16.423	11(3)	5.8741	11(3)	4.4366	43	18.524	20	13.441
			3.2352e-6		1.7997e-7		0.0058		3.735e-6		5.0839e-6		8.7012e-7
$\omega = 100$													
16384	0.3573	5	**0.0450**	3	0.2304	6(1)	0.1282	7(1)	0.1050	50	0.7104	6	0.1370
			4.4294e-9		1.7938e-12		6.2706e-6		4.6404e-9		1.1714e-8		5.7217e-10
65536	6.5076	5	**0.1572**	3	2.0366	6(1)	0.4339	7(1)	0.3862	50	4.7016	6	0.6203
			9.0105e-9		3.6027e-12		1.1773e-5		9.5005e-9		2.384e-8		1.1659e-9
262144	59.26	5	**0.5446**	3	14.005	6(1)	2.0772	7(1)	1.658	50	29.574	6	3.3102
			1.8041e-8		7.217e-12		2.3517e-5		1.892e-8		4.7705e-8		2.3316e-9

Table 2 Problem 1: 2D, FEM, M-mass matrix, $\boldsymbol{C} = L + i\omega M$;inner solver AGMG

Problem size	Direct time	ILU-QMR		C–to–R -GCGMR		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4225	0.0594	14	0.2601	3(6)	0.0620	3	**0.0196**
			1.3617e-6		4.9205e-6		1.4001e-8
16641	0.2652	26	1.8462	3(6)	0.22638	3	**0.1084**
			2.9051e-5		6.7551e-5		6.5565e-8
66049	1.5852	52	13.311	3(6)	0.88911	3	**0.5652**
			1.2776e-4		1.8912e-4		1.7679e-7
263169	9.735	103	98.19	3(6)	4.0115	3	**2.5433**
			1.4075e-3		6.1502e-4		8.5698e-7
$\omega = 1$							
4225	0.1120	14	0.2473	5(7)	**0.0976**	5	0.1476
			1.1306e-6		6.5157e-6		2.5291e-6
16641	0.2584	26	1.7549	5(7)	0.3514	5	**0.15714**
			3.4661e-5		7.7713e-5		7.5906e-6
66049	1.5239	51	13.049	5(8)	1.6808	5	**0.87701**
			1.7116e-4		3.7116e-4		5.4428e-5
263169	9.9332	101	99.411	5(8)	7.2114	5	**4.5982**
			1.8803e-3		1.8262e-3		1.6029e-4
$\omega = 100$							
4225	**0.0521**	10	0.2366	10(6)	0.1589	17	0.1290
			1.6627e-6		6.2009e-6		1.0996e-6
16641	**0.2918**	19	1.6017	10(7)	0.6094	17	0.6866
			6.8632e-6		2.8814e-5		3.6833e-6
66049	**1.6299**	36	10.675	10(7)	2.6054	17	3.1166
			8.0457e-5		1.3499e-4		1.6439e-5
263169	**9.9345**	70	73.228	9(8)	11.803	17	15.154
			8.2749e-4		3.5128e-3		6.7764e-5

Table 3 shows the corresponding result for 3D. We see again the mesh-independent convergence of C–to–R and PMHSS. The direct method and PMHSS are not tested for the largest problem size due to very long factorization times.

The experiments in Table 8 deserve special attention. The matrices correspond to RBF discretizations with tuned shape parameters of the radial basis functions that optimize the underlying discretization error. We note that size 900 is considered fairly large in the context of RBF. Due to the very ill-conditioned matrices, the direct solution method fails to solve the systems. Again, due to the ill-conditioning, the norm of the residual is not representative for the achieved accuracy in the solution and therefore is not included. In Table 8 we present results for 30 iterations of C–to–R -GCG

Table 3 Problem 1: 3D, FEM, M-mass matrix, $\boldsymbol{C} = L + i\omega M$; inner solver AGMG

Problem size	Direct time	ILU-QMR		C–to–R -GCGMR		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4913	0.5755	69	0.2761	3(4)	0.2133	3	**0.1226**
			1.3967e-4		8.1647e-05		7.5096e-07
35937	34.608	138	4.522	3(5)	**1.2087**	3	4.1468
			3.4811e-4		1.2793e-3		4.7588e-06
274625	–	278	87.563	3(6)	**12.982**	–	–
$\omega = 1$							
4913	0.6597	72	0.2958	6(4)	0.3673	6	**0.2202**
			5.9542e-05		2.5354e-05		2.0718e-06
35937	33.771	136	4.6284	6(6)	**2.1454**	6	7.3506
			4.9821e-4		1.9643e-4		1.3665e-05
274625	–	267	80.072	6(7)	**24.335**	–	–
$\omega = 100$							
4913	0.5374	42	**0.1818**	11(3)	0.2125	18	0.6515
			6.0743e-06		4.3167e-06		7.5454e-06
35937	33.129	78	2.5282	10(5)	**2.3168**	19	20.839
			8.9689e-05		2.5178e-4		1.5175e-05
274625	–	149	46.315	10(6)	**30.578**	–	–

and PMHSS-GMRES. We see that for the largest problem the time for C–to–R is about twice larger than that for PMHSS, due to the fact that we solve two systems with $A + B$. However, the convergence of PMHSS-GMRES stagnates already from the first iterations, as illustrated in Fig. 5a.

In the lower part of Table 8 we present results when the preconditioners are based on sparsified A and B blocks. The sparsification is performed by moving all positive entries in A and B that are smaller than $5\,10^{-3}$ to the corresponding main diagonal, preserving in this way the positive definiteness of the blocks. A portrait of the sparsified matrix is shown in Fig. 5b.

Note that for Problem 4 PMHSS stagnates, so the time is not representative.

Table 9 shows results for three complex symmetric matrices from [30]. The outcome is aligned with that of the other test problems.

The comparisons of the performance of the various solution techniques can be summarized as follows.

(1) The AMG-preconditioned method C–to–R shows full robustness and convergence, independent of the problem size. The PMHSS-preconditioned method shows also robustness and optimal rate of convergence for nearly all tested problems. It is seen that for $\varepsilon = 0.001$ and $\omega = 100$ the number of iterations

Table 4 Problem 2: 2D, FEM, $\boldsymbol{C} = K + i\,\omega\tau M$, $\varepsilon = 1$, M-mass matrix; inner solver AGMG

Problem size	Direct time	C–to–R		C–two–R		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4225	0.0470	3(5)	0.0717	7(7)	0.1589	2	**0.0164**
			2.0896e-8		4.4788e-12		1.0497e-9
16641	0.2476	3(6)	0.2259	7(7)	0.51517	2	**0.0377**
			5.4488e-8		1.0898e-11		6.2437e-10
66049	1.5274	3(5)	0.8565	5(7)	1.5929	2	**0.1998**
			7.3991e-8		7.7311e-9		2.8014e-10
263169	9.7766	3(6)	3.9865	5(7)	7.2498	2	**0.9414**
			7.8179e-8		1.9918e-8		6.5723e-10
$\omega = 1$							
4225	0.0461	3(6)	0.0723	6(8)	0.2089	3	**0.0135**
			1.3746e-6		1.4369e-11		3.6543e-9
16641	0.2480	3(6)	0.2280	6(8)	**0.4714**	3	0.0534
			6.4117e-7		2.1348e-11		6.2376e-10
66049	1.5623	3(6)	0.8694	6(7)	1.7909	3	**0.3468**
			5.7496e-7		1.7939e-10		1.7732e-10
263169	9.6558	3(6)	3.8978	5(7)	7.2043	3	**1.1893**
			2.4242e-7		5.1919e-8		5.1323e-10
$\omega = 100$							
4225	0.0489	6(7)	0.1168	11(8)	0.2270	6	**0.0272**
			3.4502e-7		2.4806e-10		2.1618e-8
16641	0.2440	5(7)	0.3699	10(9)	0.75855	5	**0.0946**
			6.834e-7		2.7446e-10		8.6175e-8
66049	1.5722	4(7)	1.3181	8(9)	2.7422	5	**0.3286**
			1.6873e-5		4.6072e-8		6.8803e-9
263169	9.6356	4(8)	6.0517	8(9)	13.021	4	**1.5163**
			4.9863e-6		7.0361e-9		1.0156e-7

increase for larger meshsize h. This is due to the small viscosity parameter ε and a relatively larger imaginary part of the complex matrix.

(2) For certain simple problems unpreconditioned QMR is very efficient and fast, however its behaviour is not robust in general.

(3) The ILU-preconditioned QMR for $M = I$ (Problem 1) exhibits mesh-independent rate of convergence. The solution time, however, is not competitive, compared to the other preconditioners. In the general case it shows the expected behaviour of an ILU-preconditioned method. We see, for instance

Table 5 Problem 2: 2D, FEM, $\boldsymbol{C} = K + i\,\omega\tau M$, $\varepsilon = 0.001$, M-mass matrix; direct inner solver

Problem size	Direct time	C–to–R		C–two–R		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4225	**0.0800**	3(0)	0.2473	6(0)	0.3966	3	0.2352
			5.9737e-5		1.2126e-8		1.9313e-5
16641	**0.2595**	3(0)	0.5696	6(0)	1.1328	3	0.3235
			9.8495e-6		3.0229e-9		2.5607e-6
66049	1.5449	2(0)	0.8490	5(0)	2.0191	3	**0.7897**
			0.0059		5.527e-8		5.2968e-7
263169	29.187	2(0)	**1.0878**	5(0)	2.7409	3	1.2342
			0.0118		2.4269e-7		2.5907e-7
$\omega = 1$							
4225	**0.0484**	6(0)	0.3517	12(0)	0.6980	7	0.2430
			2.6281e-4		1.7353e-8		6.8298e-7
16641	**0.25744**	6(0)	1.0028	11(0)	1.8666	6	0.5806
			3.0986e-5		3.1909e-8		1.8314e-6
66049	1.5598	5(0)	1.9795	10(0)	3.4714	5	**1.0717**
			1.8155e-4		5.6199e-8		2.9491e-5
263169	27.831	4(0)	1.937	8(0)	4.2821	4	**1.8631**
			2.4054e-3		1.4156e-6		2.4289e-4
$\omega = 100$							
4225	**0.1012**	24(0)	1.2535	40(0)	2.8732	28	0.9310
			4.4962e-5		3.0782e-8		8.2283e-6
16641	**0.3831**	15(0)	2.2864	25(0)	3.7851	20	2.261
			1.6401e-4		2.4324e-7		5.5178e-6
66049	**1.6727**	11(0)	3.5482	19(0)	6.1852	15	3.6467
			1.8430e-3		3.9131e-6		1.5487e-5
263169	27.448	9(0)	**4.5502**	16(0)	7.9045	12	4.9988
			3.6719e-3		4.2998e-5		5.4274e-5

from Tables 2 and 3, that the iterations increase twice when the mesh is refined from h to $h/2$.

(4) Modern sparse direct solvers are implemented very efficiently and can be outperformed by preconditioned iterative methods only for large enough problems. The notion 'large enough' is computer- and implementation-dependent. The experiments show, however, that the C–to–R method with AGMG as an inner solver can be faster than the sparse direct method in Matlab even in 2D for not very large problems.

Table 6 Problem 2: 2D, FEM, $\boldsymbol{C} = M + \tau K + i\,\omega\tau M$, $\varepsilon = 0.001$, M-mass matrix; direct inner solver

Problem size	Direct time	C–to–R		C–two–R		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4225	0.0483	3(0)	0.0514	5(0)	0.0917	2	**0.0243**
			2.8285e-8		1.1641e-10		1.0454e-4
16641	0.2387	2(0)	0.0978	5(0)	0.1527	2	**0.0397**
			1.7111e-4		3.8835e-10		6.6854e-5
66049	1.6961	2(0)	0.2385	5(0)	0.5956	2	**0.1995**
			1.4181e-4		2.1835e-9		5.7479e-5
263169	65.123	2(0)	1.1008	5(0)	2.7705	2	**0.9556**
			1.1121e-4		1.2507e-8		4.814e-5
$\omega = 1$							
4225	0.0538	5(0)	0.0649	9(0)	0.127	4	**0.0389**
			4.9564e-6		3.2426e-10		8.7065e-5
16641	0.2574	4(0)	0.0909	7(0)	0.1861	4	**0.0678**
			7.9416e-5		1.4279e-7		1.8878e-5
66049	1.6234	4(0)	0.4225	7(0)	0.7924	3	**0.2456**
			1.1904e-5		1.1906e-8		1.2970e-3
263169	60.632	3(0)	1.5047	6(0)	3.1972	3	**1.2285**
			1.3834e-3		5.2784e-7		5.1520e-4
$\omega = 100$							
4225	**0.0501**	20(0)	0.1123	32(0)	0.2111	25	0.1899
			1.9517e-4		5.7998e-8		5.2046e.4
16641	**0.24019**	16(0)	0.3529	26(0)	0.57012	16	0.3358
			4.3439e-4		3.1132e-7		8.3680e-4
66049	1.5825	12(0)	1.3102	20(0)	2.0338	11	**0.8935**
			1.4694e-3		1.7939e-6		1.6232e-3
263169	60.102	9(0)	4.2618	15(0)	7.2431	8	**3.2465**
			3.5112e-3		1.4380e-5		4.4112e-3

(5) With the particular choices of the method parameters α and V made here, the PMHSS uses a sparse direct solver once per each iteration.

The theory for the convergence of PMHSS was previously developed for complex symmetric matrices, but has been successfully extended and applied to matrices that are not complex symmetric, cf. Tables 4, 5 and 6.

(6) C–to–R and PMHSS possess good parallelization properties. Both require solutions of inner systems with real matrices of similar type, even identical for $\alpha = 1$ and $V = A$, for which efficient multilevel techniques are applicable.

Table 7 Problem 3: 2D, FEM, $\tau = h$, M-mass matrix; inner solver AGMG

Problem size	Direct time	ILU-QMR		C–to–R		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
Random right-hand side							
4225	**0.0696**	8	0.2262	9(6)	0.1641	13	0.0801
			3.3923e-4		6.3274e-4		1.1498e-4
16641	**0.2534**	11	1.303	9(6)	0.4912	14	0.4221
			2.5876e-3		1.2436e-2		6.4089e-4
66049	**1.6024**	16	7.2485	9(6)	2.1323	15	2.1898
			8.9767e-3		9.6769e-2		1.6452e-3
263169	**10.013**	21	38.663	9(6)	10.422	15	12.86
			0.2114		0.6531		0.01264
Right-hand side: $(1+i)(A[1, 1 \cdots 1]^T)$							
4225	0.0958	9	0.2237	9(6)	0.1409	12	**0.0743**
			2.2792e-8		8.1938e-7		4.5851e-7
16641	**0.2772**	11	1.3131	9(6)	0.5119	13	0.4474
			1.5922e-6		1.4688e-5		1.4903e-6
66049	**1.6526**	16	7.2463	9(7)	2.2284	14	2.0032
			5.4538e-6		2.7184e-5		1.9176e-6
263169	**10.279**	21	37.926	9(7)	10.506	14	12.166
			4.1101e-5		1.0334e-4		5.4956e-6

(7) It is seen from the numerical results that, while having the same outer stopping criterion, the norm of the error in the iterative solution obtained by PMHSS-GMRES is in some cases noticeably smaller than the error of the iterative solution, obtained by C–to–R -GCG. The reason for that is that even though the residuals have been equally reduced, the error in the imaginary part of the solution is larger than that in the real part of the solution. To compensate for

Table 8 Problem 4: Schrödinger equation, direct inner solver

Problem size	C–to–R -GCGMR		PMHSS-GMRES	
	iter	time	iter	time
Dense preconditioner				
400	30(0)	0.1399	30	0.1109
625	30(0)	0.4094	30	0.1560
900	30(0)	0.7755	30	0.3041
Sparsified preconditioner				
400	30(0)	0.2252	30	0.2244
625	30(0)	0.3801	30	0.2339
900	30(0)	0.8344	30	0.3867

Table 9 Problem 5: inner solver AGMG

Problem size	Direct time	ILU-QMR		C–to–R -GCGMR		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
'mhd1280b'							
1280	**0.0037**	1280	0.7764	2(0)	0.01875	2	0.0079
			21.089		1.9986e-7		4.6214e-8
'mplate'							
5962	**1.3891**	–	–	8(0)	9.7639	12	8.0599
			–		3.6131e-3		3.8309e-4
'windscreen'							
22692	1.6853	–	–	1(1)	**0.2995**	1	4.2994
			–		1.8718e-13		9.7802e-13

this effect, we can perform a few additional iterations with C–to–R , applied to the system (2), where the roles of A and B have been interchanged, thereby using the already obtained solution as an initial guess, and continue with a number of iterations with C–to–R applied to the system (3). The resulting method is referred to as C–two–R . In Tables 4, 5 and 6 one can see the effect of C–two–R , for a reduced relative outer stopping criterion 10^{-6}.

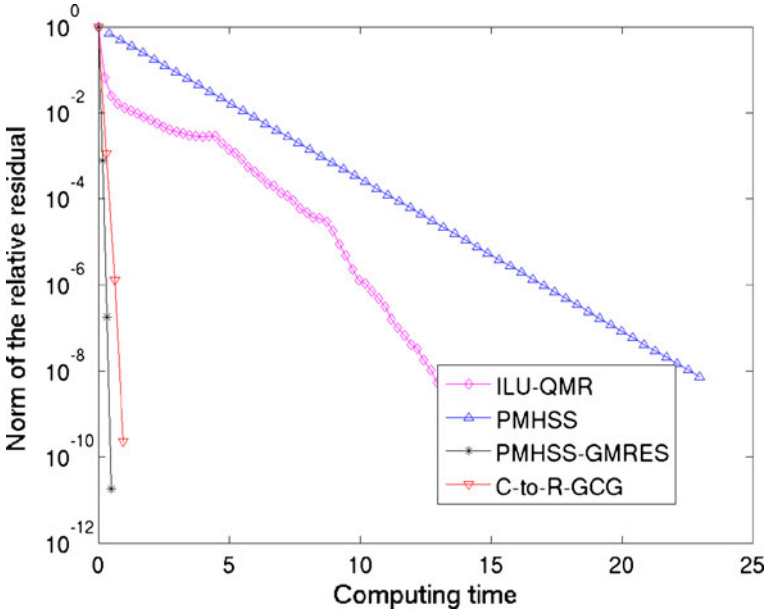

Fig. 2 Problem 1: Convergence comparisons, FEM, problem size 66049, $\omega = 0.01$

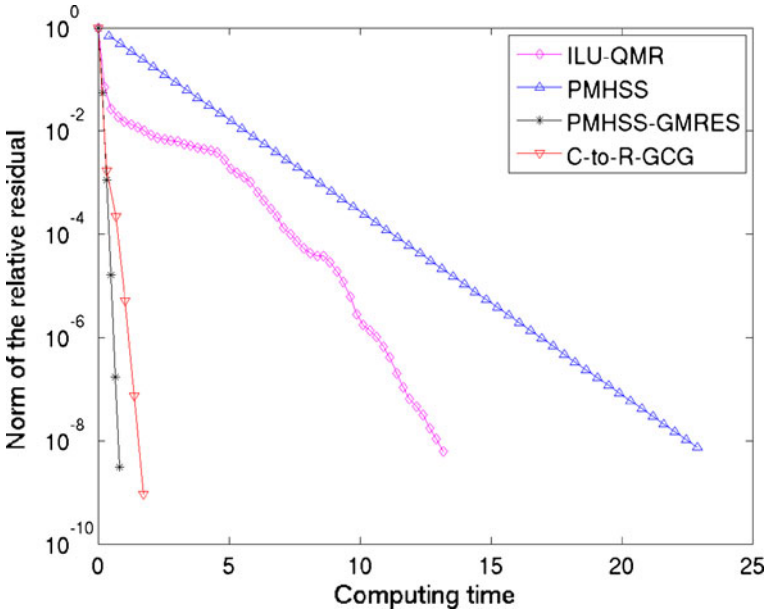

Fig. 3 Problem 1: Convergence comparisons, FEM, problem size 66049, $\omega = 1$

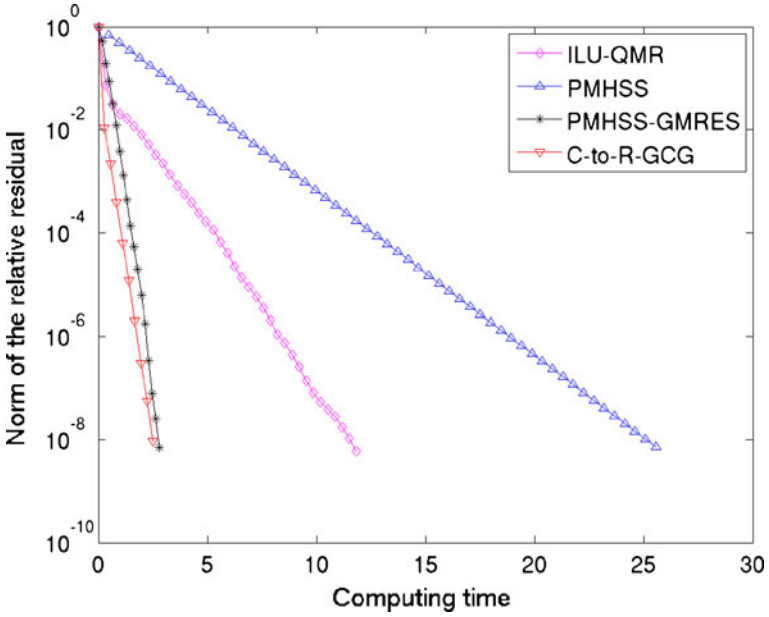

Fig. 4 Problem 1: Convergence comparisons, FEM, problem size 66049, $\omega = 100$

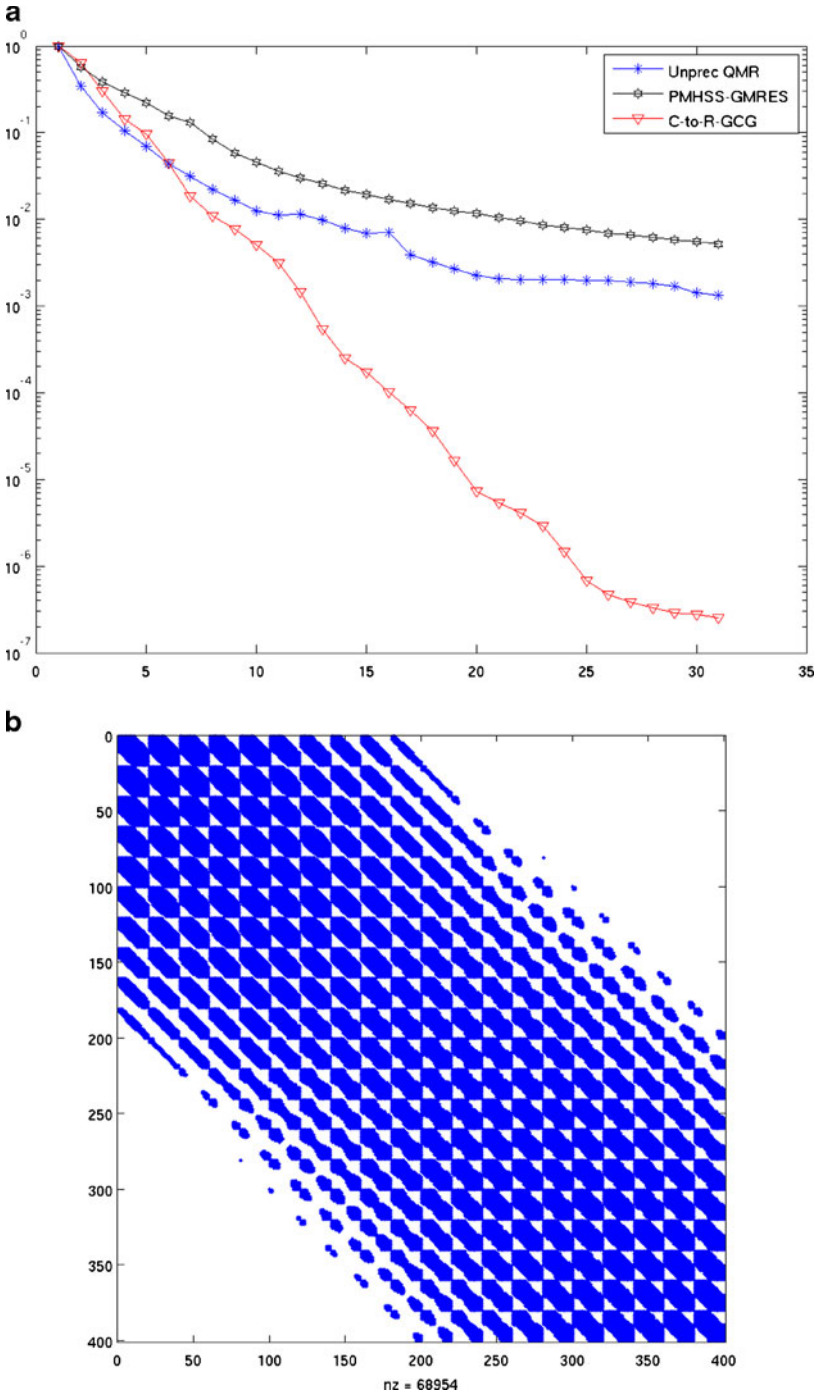

Fig. 5 Problem 4. **a** Dense preconditioner: number of iterations on the 'x-axes and Euclidean norm of the computed residual on the 'y'-axes. **b** A portrait of a sparsified block A, matrix size 400

6 Conclusions

For test problems of somewhat simpler kind and of relatively small size, it has been seen that direct solution methods are competitive and the QMR method gives smallest elapsed computer times. This holds for 2D problems of not too large problem size and for problems involving the identity matrix as a dominating part. For large scale problems, in particular, for 3D problems, direct methods can not compete with iterative solution methods.

The QMR method and its ILU-preconditioned version turn out not to be robust, i.e., not applicable for all test problems. Since we aim at a fully robust method, we turn to the C–to–R method and the PMHSS method. For exact solution of the arising inner systems of equations in a problem, where A and B are spsd, it has been shown that the C–to–R method gives eigenvalues in an interval $[a, 1]$, where $a \geq \frac{1}{2}$ and the number a depends on the eigenvalues of $B^{-1}A$ and the preconditioning parameter α. Hence, the condition number is bounded by 2. This holds if $B^{-1}A$ have real, nonnegative eigenvalues. When an inner iteration method with a finite stopping criterion is used, the resulting eigenvalues will be somewhat perturbed, depending on the stopping criteria for the inner iterations. It is then advisable to use a flexible GMRES ([13]) or a variable preconditioned GCG ([14]) iteration method. Since there are only few iterations, in practice mostly less than 12, there is no need to use a restarted version of these methods.

Alternatively, based on the given eigenvalue bounds, one can use a Chebyshev iteration method, thereby avoiding the computation of inner products and global communication of them to all computer processor cores.

The PMHSS method results in complex eigenvalues located in a disc with center at the unit value $\lambda = 1$, and with radius $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$. For $\alpha = 1$, the radius equals $1/\sqrt{2}$. Hence, a condition number just based on the real eigenvalues, equals $(1 + 1/\sqrt{2})/(1 - 1/\sqrt{2}) = 3 + 2\sqrt{2} \approx 6$. Since the eigenvalues are complex, the actual rate of convergence will be relatively slower than the method, based on this condition number. The special case of the method, which is equivalent to using $A + B$ as a preconditioner to $\boldsymbol{C}$ has more favourable properties and converges fast when the arising systems with $A + B$ are solved by a direct method. This, however, makes the method less competitive for 3D and ill-conditioned problems. In general, the method can be expected to converge slower than the C–to–R method due to the following two reasons.

(i) The eigenvalues in the PMHSS method are complex even for spd (spsd) matrices A, B.
(ii) The eigenvalues are located in a slightly larger domain than for the C–to–R method.

Additionally, the PMHSS method still needs some complex arithmetic computations in the iterations.

The numerical tests in [19], the arising inner systems are solved via a direct solution method. Since the matrices are assumed to be symmetric and positive definite, a Cholesky factorization method was used. The implementation in [19] is

computationally less efficient since there the parameter α is not equal to 1 and two inner systems must be solved. The use of the special version of the method, as implemented here, is however quite efficient. Then, for $\alpha = 1$ only one inner system must be solved.

In the implementation in this paper, a direct method for the solution of the inner system in PMHSS is used. It is unclear how sensitive the eigenspectrum will be for inexact (iterative) solution of these systems. However, if the inner systems are solved with a multilevel or AMG preconditioned method, under certain considerations, one can obtain a nearly optimal order, $O(h^{-d}), d = 2, 3$ of computational complexity. Such tests are not included here. We note that in latest release of AGMG handles also complex matrices.

Both the special version of the PMHSS method and the C–to–R method are applicable also in the case where the matrices A and B are nonsymmetric, that is, when $A^{-1}B$ has complex eigenvalues.

As a general conclusion, both the theoretical and numerical results show that the iterative methods, C–to–R and PMHSS are most generally applicable and that the preconditioned C–to–R method has a significant potential, in that it shows robustness and numerical stability for broader classes of problems. Also, for not too simple problems, it may outperform other methods of choice for complex symmetric matrices. The use of the C–two–R version of the method can in some cases significantly improve the accuracy of the iterative solution at a relatively small amount of additional computations and no additional effort to implement it.

Acknowledgments This work was funded by King Abdulaziz University (KAU), under grant No. (35-3-1432/HiCi). The technical and financial support of KAU is hereby gratefully acknowledged.

The authors are indebted to professor Zhong-Zhi Bai for the numerous discussions and help with implementing the PMHSS preconditioner, as well as to Dr. Fang Chen for the permission to access the codes, employed in the numerical experiments in [19]. The authors also thank Dr. Katharina Kormann for providing the matrices for Problem 4.

We also thank the anonymous reviewers for their careful reading and constructive criticism.

References

1. van Rienen, U.: Numerical methods in computational electrodynamics. Linear systems in practical applications. Springer–Verlag, Berlin Heidelberg (2001)
2. Kormann, K.: Efficient and reliable simulation of quantum molecular dynamics. Ph.D. Thesis, Uppsala University. http://uu.diva-portal.org/smash/record.jsf?pid=diva2:549981
3. Novikov, S., Manakov, S.V., Pitaevskiĭ, L.P., Zakharov, V.E.: Theory of Solitons. The Inverse Scattering Method. Translated from Russian. Contemporary Soviet Mathematics. Consultants Bureau [Plenum], New York (1984)
4. Butcher, J.C.: Integration processes based on Radau quadrature formulas. Math. Comp. **18**, 233–244 (1964)
5. Axelsson, O.: On the efficiency of a class of A–stable methods. BIT **14**, 279–287 (1974)
6. Day, D., Heroux, M.A.: Solving complex-valued linear systems via equivalent real formulations. SIAM J. Sci. Comput. **23**, 480–498 (2001)
7. Benzi, M., Bertaccini, D.: Block preconditioning of real–valued iterative algorithms for complex linear systems. SIAM J. Numer. Anal. **28**, 598–618 (2008)
8. Giovangiglia, V., Grailleb, B.: Projected iterative algorithms for complex symmetric systems arising in magnetized multicomponent transport. Linear Algebra Appl. **430**, 1404–1422 (2009)

9. Howle, V.E., Vavasis, S.A.: An iterative method for solving complex-symmetric systems arising in electrical power modeling. SIAM J. Matrix Anal. Appl. **26**, 1150–1178 (2005)
10. Bai, Z.-Z., Benzi, M., Chen, F., Wang, Z.-Q.: Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA J. Numer. Anal. **33**, 343–369 (2013)
11. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**, 197–218 (2000)
12. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**, 856–869 (1986). doi:10.1137/0907058
13. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM, J. Sci. Comp. **14**, 461–469 (1993)
14. Axelsson, O., Vassilevski, P.S.: A black box generalized conjugate gradient solver with inner iterations and variable–step preconditioning. SIAM J. Matrix Anal. Appl. **12**, 625–644 (1991)
15. Vassilevski, P.S.: Multilevel Block Factorization Preconditioners: Matrix-Based Analysis and Algorithms for Solving Finite Element Equations. Springer, New York (2008)
16. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)
17. Bai, Z.-Z., Golub, G.H., Ng, M.K.: On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. Numer. Linear Algebra Appl. **14**, 319–335 (2007)
18. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
19. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algoritm. **56**, 297–317 (2011)
20. Axelsson, O., Bai, Z.-Z., Qiu, S.-X.: A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part. Numer. Algoritm. **35**, 351–372 (2004)
21. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., van der Vorst, H.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd edn. SIAM, Philadelphia (1994)
22. Freund, R.W.: Conjugate gradient–type methods for linear systems with complex symmetric coefficient matrices. SIAM J. Sci. Statist. Comput. **13**, 425–448 (1992)
23. van der Vorst, H., Melissen, J.: A Petrov–Galerkin type method for solving $Ax = b$ where A is symmetric complex. IEEE Trans. Magn. **26**, 706–708 (1990)
24. Freund, R.W., Nachtigal, N.M.: QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numer. Math. **60**, 315–339 (1991)
25. Freund, R.W.: A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems. SIAM J. Sci. Comput. **14**, 470–482 (1993)
26. Pearson, J.W., Wathen, A.J.: A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer. Linear Algebra Appl. **19**, 816–829 (2012)
27. Axelsson, O.: Iterative Solution Methods. Cambridge University Press, Cambridge (1994)
28. Axelsson, O., Boyanova, P., Kronbichler, M., Neytcheva, M., Wu, X.: Numerical and computational efficiency of solvers for two-phase problems. Comput. Math. Appl. (2012). Published on line at doi:10.1016/j.camwa.2012.05.020
29. Kormann, K., Larsson, E.: An RBF-Galerkin Approach to the Time-Dependent Schrödinger Equation. Department of Information Technology, Uppsala University, TR 2012–024 (2012)
30. The University of Florida Sparse Matrix Collection, maintained by T. Davis and Y. Hu, http://www.cise.ufl.edu/research/sparse/matrices/
31. Notay, Y.: AGMG software and documentation; see http://homepages.ulb.ac.be/~ynotay/AGMG
32. Napov, A., Notay, Y.: An algebraic multigrid method with guaranteed convergence rate. SIAM J. Sci. Comput. **34**, A1079–A1109 (2012)
33. Notay, Y.: Aggregation-based algebraic multigrid for convection-diffusion equations. SIAM J. Sci. Comput. **34**, A2288–A2316 (2012)