ON EFFICIENT NUMERICAL APPROXIMATION OF THE BILINEAR FORM $c^*A^{-1}b$ [‡]

ZDENĚK STRAKOŠ[†] AND PETR TICHÝ[*]

Abstract. Let $A \in \mathbb{C}^{N\times N}$ be a nonsingular complex matrix and b and c complex vectors of length N. The goal of this paper is to investigate approaches for efficient approximations of the bilinear form $c^*A^{-1}b$. Equivalently, we wish to approximate the scalar value c^*x where x solves the linear system $Ax = b$. Here the matrix A can be very large or its elements can be too costly to compute so that A is not explicitly available and it is used only in the form of the matrix-vector product. Therefore a direct method is not an option. For A Hermitian positive definite, $b^*A^{-1}b$ can be efficiently approximated as a by-product of the conjugate-gradient iterations, which is mathematically equivalent to the matching moments approximations computed via the Gauss-Christoffel quadrature. In this paper we propose a new method using the biconjugate gradient iterations which is applicable to the general complex case. The proposed approach will be compared with existing ones using analytic arguments and numerical experiments.

Key words. Bilinear forms, scattering amplitude, method of moments, Krylov subspace methods, CG method, BiCG method, Lanczos algorithm, Arnoldi algorithm, Gauss-Christoffel quadrature, model reduction.

AMS subject classifications. 15A06, 65F10, 65F25, 65G50.

1. Introduction.

Given a nonsingular square matrix $A \in \mathbb{C}^{N\times N}$ and vectors b and c of compatible dimensions, many applications require approximation of the quantity

$$c^*A^{-1}b\,. \tag{1.1}$$

They arise in signal processing under the name scattering amplitude, as well as in nuclear physics, quantum mechanics, computational fluid dynamics; see [44, 20] and the references given there. In numerical linear algebra they arise naturally in computing error bounds for iterative methods, in solving inverse problems, least and total least squares problems etc.; see [19]. This paper presents an approach for approximating $c^*A^{-1}b$ that is designed to be computationally efficient. For context, we also briefly summarize existing techniques for approximating $c^*A^{-1}b$, notably in the special cases when A, b, and c are real or when A is Hermitian positive definite (HPD).

Given the solution x of the linear algebraic system $Ax = b$, (1.1) can be reformulated as

$$c^*A^{-1}b = c^*x\,.$$

In most applications, $c^*A^{-1}b$ need not be computed to a high accuracy; an approximation correct to very few digits of accuracy is sufficient. Therefore direct solution of $Ax = b$ is even for problems of moderate size inefficient. If A is sufficiently large or the elements of A are too costly to compute, then the direct solution is not possible.

[†] Faculty of Mathematics and Physics, Charles University, Sokolovská 83, 186 75 Prague, Czech Republic, email: strakos@karlin.mff.cuni.cz.

[*] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 18207 Prague, Czech Republic, email: tichy@cs.cas.cz.

[‡] This work is a part of the research project MSM0021620839 financed by the Ministry of Education of the Czech Republic (Z. Strakoš) and of the Institutional Research Plan AV0Z10300504 (P. Tichý). The work was also partially supported by the GAAS grant IAA100300802, by the project M100300901 of the institutional support of ASCR, and by the GACR grant 201/09/0917.

A strategy used by several authors is to generate a sequence $\{x_k\}$ of approximate solutions to $Ax = b$ using a Krylov subspace method, and to approximate $c^*A^{-1}b$ by c^*x_n for sufficiently large n. However, even when A is HPD, this approximation may require a large number of iterations as a result of rounding errors affecting x_n; see [52, 53]. A variety of approaches for approximating $c^*A^{-1}b$ have been developed based on quadrature and moments; see, for example, [17]. The extensive literature about connections between moments, iterative methods, and model reduction is too large to summarize here; we mention, as five examples among hundreds, [24, 13, 4, 2] and [11]. The same is true for related literature in the area of physical chemistry and solid state physics computations; for reviews of early papers see [23, 40, 45]. The mathematical roots can be found in the work on orthogonal polynomials and continued fractions by Chebyshev [7] and Stieltjes [49].

The ideas in this paper for the general complex case (which includes also real nonsymmetric case) are based on non-Hermitian generalizations of Vorobyev moment problems [55] (to be defined in Section 2). Algorithmically, this paper extends the results presented in [52, 53] from the HPD case and the conjugate gradient method (CG) to the general complex case and the biconjugate gradient method (BiCG).

2. Matching moments in Krylov subspace methods and the Vorobyev moment problem.

To motivate our approach, Sections 2.1–2.2 summarize some of the well known connections between two Krylov subspace methods, model reduction, and moments. In Krylov subspace methods it might be convenient to consider nonzero initial approximations x_0 and y_0 to the solutions of $Ax = b$ and $A^*y = c$ respectively. That is equivalent to applications of the same methods, with the zero initial approximations, to $A\mathbf{x} = \mathbf{b}$ respectively $A^*\mathbf{y} = \mathbf{c}$, where $\mathbf{b} = b - Ax_0$, $\mathbf{c} = c - A^*y_0$ are the initial residuals and $\mathbf{x} = x - x_0$, $\mathbf{y} = y - y_0$ are unknown. Using

$$c^*A^{-1}b = c^*x_0 + y_0^*\mathbf{b} + \mathbf{c}^*A^{-1}\mathbf{b}\,,$$

$c^*A^{-1}b$ can always be approximated via $\mathbf{c}^*A^{-1}\mathbf{b}$ using zero initial approximations of $\mathbf{x}$, $\mathbf{y}$. Throughout this paper we will therefore consider, with no loss of generality, zero initial approximations.

2.1. Lanczos algorithm as model reduction.

Let $A \in \mathbb{C}^{N\times N}$ be a nonsingular matrix and the vectors v_1 and w_1 of length N satisfy $\|v_1\| = 1$, $w_1^*v_1 = 1$. The nth step of the non-Hermitian Lanczos algorithm applied to A with the starting vectors v_1 and w_1 is associated with the following relations

$$\begin{aligned} AV_n &= V_nT_n + \delta_{n+1}v_{n+1}e_n^T\,, \\ A^*W_n &= W_nT_n^* + \beta_{n+1}^*w_{n+1}e_n^T\,, \end{aligned} \tag{2.1}$$

where $W_n^*V_n = I$, $T_n = W_n^*AV_n$, $\|v_{n+1}\| = 1$, $w_{n+1}^*v_{n+1} = 1$, and the main diagonal, the first subdiagonal and the first superdiagonal of T_n are given by $\gamma_1, \ldots, \gamma_n$, $\delta_2, \ldots, \delta_n$, and $\beta_2, \ldots, \beta_n$ respectively, $\delta_\ell > 0$, $\beta_\ell \neq 0$, $\ell = 2, \ldots, n$; see, e.g., [41, Section 7.1]. Here it is assumed that the algorithm does not break down in steps 1 through n. The columns of V_n form a basis of $\mathcal{K}_n(A, v_1)$,

$$\mathcal{K}_n(A, v_1) \equiv \mathrm{span}\{v_1, Av_1, \ldots, A^{n-1}v_1\} = \mathrm{span}\{v_1, \ldots, v_n\}\,,$$

while the columns of W_n form a basis of $\mathcal{K}_n(A^*, w_1)$. Under the given assumption on existence of the steps 1 through n, the non-Hermitian Lanczos algorithm represents the reduction of the *original model* which consists of the matrix A and two vectors v_1

and w_1 to the *reduced model* which consists of the matrix T_n and two identical vectors e_1 and e_1. The reduced model matches the first $2n$ moments

$$w_1^* A^k v_1 = e_1^T T_n^k e_1 \,, \quad k = 0, 1, \ldots, 2n-1 \,. \tag{2.2}$$

Relation (2.2) can be derived from the Vorobyev moment problem, which is to determine a linear operator A_n on $\mathcal{K}_n(A, v_1)$ such that

$$A_n^j v_1 = A^j v_1, \quad j = 1, \ldots, n-1, \quad \text{and} \quad A_n^n v_1 = V_n W_n^* A^n v_1 \,. \tag{2.3}$$

Defining A_n as the restriction of A to $\mathcal{K}_n(A, v_1)$ projected orthogonally to $\mathcal{K}_n(A^*, w_1)$ (which represents an oblique projection to $\mathcal{K}_n(A, v_1)$)

$$A_n = V_n W_n^* A V_n W_n^* \,, \tag{2.4}$$

it follows from the relation $T_n = W_n^* A V_n$ that

$$A_n = V_n T_n W_n^* \tag{2.5}$$

and

$$w_1^* A^k v_1 = w_1^* A_n^k v_1 = e_1^T T_n^k e_1 \,, \quad k = 0, 1, \ldots, 2n-1 \,; \tag{2.6}$$

see [51]. The matching moment property (2.2) of the non-Hermitian Lanczos algorithm will be linked with the new numerical approximation of the bilinear form (1.1) proposed in Section 3.1.

If A is Hermitian and $w_1 = v_1$, the non-Hermitian Lanczos algorithm reduces to the Hermitian Lanczos algorithm that is associated with the relation

$$A V_n = V_n T_n + \delta_{n+1} v_{n+1} e_n^T \,,$$

where T_n is the Jacobi matrix, and $V_n^* V_n = I$. In this case, the linear operator A_n is the restriction of A to $\mathcal{K}_n(A, v_1)$ projected orthogonally to $\mathcal{K}_n(A, v_1)$. For more details see [55, Chapter III, Sections 2-4], with the summary given in [51].

2.2. Arnoldi algorithm as model reduction. The model reduction represented by the Lanczos algorithm matches the first $2n$ moments (2.2). In the non-Hermitian case, the matrix T_n in (2.2) is determined by oblique projections. This may affect in a negative way conveying information from the original to the reduced model. We therefore need to compare the new numerical approximation proposed in Section 3.1 with the model reduction determined by orthogonal projections. This in the non-Hermitian case leads to long recurrences and the Arnoldi algorithm.

Let $A \in \mathbb{C}^{N \times N}$ be a nonsingular matrix, let v_1 and u_1 be vectors of length N, $\|v_1\| = \|u_1\| = 1$. The nth step of the Arnoldi algorithm applied to A with v_1 is associated with the relation

$$A V_n = V_n H_n + h_{n+1,n} v_{n+1} e_n^T \,, \tag{2.7}$$

where $V_n^* V_n = I_n$, $H_n = V_n^* A V_n$, $V_n^* v_{n+1} = 0$, and H_n is the upper Hessenberg matrix with positive entries on the first subdiagonal; see, e.g., [41, Section 6.3]. The matching moment property of the Arnoldi algorithm can be expressed in the form

$$u_1^* A^k v_1 = u_1^* V_n H_n^k e_1 = t_n^* H_n^k e_1 \,, \quad k = 0, \ldots, n-1 \,, \tag{2.8}$$

where $u_1 \equiv V_n t_n + u_1^{\perp} = V_n(V_n^* u_1) + u_1^{\perp}$, and $u_1^{\perp}$ is the component of u_1 orthogonal to $\mathcal{K}_n(A, v_1)$. With $u_1 = v_1$ we can add one more moment. To derive (2.8), we invoke the Vorobyev moment problem linked with the Arnoldi algorithm, which is to determine a linear operator on $\mathcal{K}_n(A, v_1)$ such that

$$A_n^j v_1 = A^j v_1, \quad j = 1, \dots, n-1, \quad \text{and} \quad A_n^n v_1 = V_n V_n^* A^n v_1\,. \tag{2.9}$$

Defining A_n as the restriction of A to $\mathcal{K}_n(A, v_1)$ projected orthogonally to $\mathcal{K}_n(A, v_1)$

$$A_n = V_n V_n^* A V_n V_n^*\,, \tag{2.10}$$

it follows from the relation $H_n = V_n^* A V_n$ that

$$A_n \;=\; V_n H_n V_n^* \tag{2.11}$$

and

$$u_1^* A^k v_1 = u_1^* A_n^k v_1 = t_n^* H_n^k e_1\,, \quad k = 0, 1, \dots, n-1\,. \tag{2.12}$$

Since A is non-Hermitian, the matching moment property can not in general be extended beyond n moments; see [51].

3. Numerical approximation of the bilinear form $c^* A^{-1} b$.

The relationship of CG to the Gauss-Christoffel quadrature, continued fractions and moments was pointed out in the founding paper by Hestenes and Stiefel [29, Sections 14–18]; see also [55, Chapter III, Section 2, pp. 53 and 59] and the summary in [34, pp. 483–484 and p. 493]. In the framework of the Vorobyev moment problem, CG and the Hermitian Lanczos algorithm, the non-Hermitian Lanczos algorithm and the Arnoldi algorithm look for a reduced order operator A_n (see (2.5) and (2.11)), with the property of matching the maximal number of moments; see (2.6) and (2.12). An approximation of the bilinear form $c^* A^{-1} b$ can be then expressed as

$$c^* A_n^{-1} b\,, \tag{3.1}$$

where A_n^{-1} is the matrix representation of the inverse of the reduced order operator A_n which is restricted onto $\mathcal{K}_n(A, b)$; see, e.g., [30, p. 79]. As an example,

$$A_n^{-1} = V_n T_n^{-1} W_n^* \tag{3.2}$$

holds for the non-Hermitian Lanczos algorithm (see (2.5)). Considering the starting vectors $v_1 = b/\|b\|$ and $w_1 = c\|b\|/c^* b$, we get

$$c^* A_n^{-1} b = \frac{c^* b}{\|b\|} w_1^* V_n T_n^{-1} W_n v_1 \|b\| = (c^* b)\, e_1^T T_n^{-1} e_1\,. \tag{3.3}$$

To our knowledge, the formula $e_1^T T_n^{-1} e_1$ was used for the symmetric positive definite case for the first time by Golub and coworkers [8, 15, 9]; for a survey see, e.g., [19], [34, Section 3.3], [16, part V, with the commentary given by Gautschi]. In this section we propose new ways of computing $c^* A_n^{-1} b$ using the BiCG-related methods and relate them to existing approaches.

Our results presented below can be derived without using (3.3) and even without mentioning the Vorobyev moment problem. In order to get an insight into the problem of approximating the bilinear form $c^* A^{-1} b$ (see, e.g., the brief discussion of the Arnoldi algorithm and BiCG in the last section of this paper), this link is, in our opinion, important, similarly as the link with the Gauss-Christoffel quadrature is important for understanding the behavior of the Lanczos algorithm and CG; see, e.g., [29, 25], [21, Section 5 on rounding error analysis].

3.1. Approximation based on the BiCG method. The BiCG method [33, 10] (see Algorithm 1) solves simultaneously the primal and dual systems of linear algebraic equations $Ax = b$ and $A^*y = c$; see [50, 20]. BiCG computes sequences of approximations $\{x_n\}$ and $\{y_n\}$ such that $x_n \in \mathcal{K}_n(A, b)$ and $y_n \in \mathcal{K}_n(A^*, c)$, while

$$r_n \equiv b - Ax_n \perp \mathcal{K}_n(A^*, c), \qquad s_n \equiv c - A^*y_n \perp \mathcal{K}_n(A, b)\,. \tag{3.4}$$

Assuming that there is no breakdown in the first n steps, the sequences of approximate solutions in the BiCG method have the form

Algorithm 1 Biconjugate Gradient Method (BiCG)

```
input $A$, $A^*$, $b$, $c$, $x_0 = 0$, $y_0 = 0$
$r_0 = p_0 = b\,, \quad s_0 = q_0 = c$
for $n = 0, 1, \dots$
  $\alpha_n = \frac{s_n^* r_n}{q_n^* A p_n}$
  $x_{n+1} = x_n + \alpha_n p_n\,, \quad y_{n+1} = y_n + \alpha_n^* q_n$
  $r_{n+1} = r_n - \alpha_n A p_n\,, \quad s_{n+1} = s_n - \alpha_n^* A^* q_n$
  $\eta_{n+1} = \frac{s_{n+1}^* r_{n+1}}{s_n^* r_n}$
  $p_{n+1} = r_{n+1} + \eta_{n+1} p_n\,, \quad q_{n+1} = s_{n+1} + \eta_{n+1}^* q_n$
end
```

$$x_n = V_n f_n \quad \text{and} \quad y_n = W_n g_n, \tag{3.5}$$

for some vectors f_n and g_n. Relation (3.2) which gives an expression for A_n^{-1} suggests using $c^*A_n^{-1}b$ as an approximation of $c^*A^{-1}b$; see (3.3). We now show how this approximation computed from the iterates of the non-Hermitian Lanczos algorithm (described in Section 2.1) with starting vectors $v_1 = b/\|b\|$, $w_1 = c\|b\|/c^*b$, is related to the BiCG method. In order to derive a formula for $c^*A_n^{-1}b$, we invoke two kinds of global biorthogonality conditions associated with the BiCG method:

$$W_n^* r_n = 0 \quad \text{and} \quad V_n^* s_n = 0, \tag{3.6}$$

$$W_n^* b = \|b\| W_n^* v_1 = \|b\| e_1. \tag{3.7}$$

The conditions (3.6) lead to linear systems $\|b\|e_1 = T_n f_n$ and $(v_1^* c)\, e_1 = T_n^* g_n$ for the unknown coordinates f_n and g_n. Consequently, $x_n = \|b\|\, V_n T_n^{-1} e_1$, $y_n = (v_1^* c)\, W_n (T_n^*)^{-1} e_1$. Then, using the global orthogonality relations (3.7), we have

$$c^* A_n^{-1} b \;=\; c^* V_n T_n^{-1} W_n^* b \;=\; c^* x_n\,. \tag{3.8}$$

Analogously, the dual quantity is given by

$$b^* (A_n^{-1})^* c = b^* W_n (T_n^*)^{-1} V_n^* c = b^* y_n. \tag{3.9}$$

The last term in (3.8) gives the well-known *scattering amplitude* approximation to c^*x; see [56, 44, 43]. Please note also that from (3.8)

$$c^* A_n^{-1} b \;=\; c^* b\, (T_n^{-1})_{1,1} \tag{3.10}$$

(see (3.3)), where the value $(T_n^{-1})_{1,1}$ can be easily computed at a negligible additional cost using the algorithm in [17, p. 135]. It is worth pointing out that evaluation of (3.10) does not require explicit computation of x_n.

The global biorthogonality conditions (3.6) and (3.7) needed for the derivation of (3.8) and (3.9) are in general not satisfied in finite precision computations. Due to rounding errors, computing sufficiently accurate approximations using (3.8) (or (3.9)) may require a large number of iterations that are (as shown below) not necessary. Therefore we present a new mathematically equivalent approximation which will be derived using only *local biorthogonality*. Using the expressions for s_{j+1}, r_{j+1} and p_j in Algorithm 1, we have for $j = 0, \dots, n-1$

$$\begin{aligned} & s_j^* A^{-1} r_j - s_{j+1}^* A^{-1} r_{j+1} \\ &= (s_{j+1} + \alpha_j^* A^* q_j)^* A^{-1} (r_{j+1} + \alpha_j A p_j) - s_{j+1}^* A^{-1} r_{j+1} \\ &= \alpha_j^2 q_j^* A p_j + \alpha_j s_{j+1}^* p_j + \alpha_j q_j^* r_{j+1} \\ &= \alpha_j s_j^* r_j + \alpha_j (s_{j+1}^* p_j + q_j^* r_{j+1}) \ = \alpha_j s_j^* r_j \,. \end{aligned} \tag{3.11}$$

For the last equality we used the local biorthogonality between the residuals and the search directions of the primal and dual problem,

$$s_{j+1}^* p_j = 0 \quad \text{and} \quad q_j^* r_{j+1} = 0 \,. \tag{3.12}$$

Consequently, using

$$c^* A^{-1} b - s_n^* A^{-1} r_n = \sum_{j=0}^{n-1} \left(s_j^* A^{-1} r_j - s_{j+1}^* A^{-1} r_{j+1} \right)$$

we finally obtain

$$c^* A^{-1} b = \sum_{j=0}^{n-1} \alpha_j s_j^* r_j + s_n^* A^{-1} r_n \,. \tag{3.13}$$

Relation (3.13) is significant because it provides an exact expression for $c^* A^{-1} b$, the first term of which is a summation involving the (available) inner product of the BiCG primal and dual residuals. As well, (3.13) generalizes the result from the HPD case, in which $b^* A^{-1} b$ and $r_n A^{-1} r_n$ equal, respectively, the squared A-norms of the errors at steps 0 and n; see [52].

If the primal and dual residuals in the BiCG method become small, the second term $s_n^* A^{-1} r_n$ on the right-hand side of (3.13) will also become small. This suggests approximating $c^* A^{-1} b$ by the following quantity:

$$\xi_n^{\mathrm{B}} \equiv \sum_{j=0}^{n-1} \alpha_j s_j^* r_j \tag{3.14}$$

where the superscript "B" means "BiCG". Although, as we show later, ξ_n^{B} is equal to $c^* x_n$ using exact arithmetic, the summation form of ξ_n^{B} in (3.14) is crucial for computational purposes.

Summarizing, $c^* A^{-1} b$ can be approximated using (3.8), (3.10) and by the new ξ_n^{B} defined in (3.14). It remains to prove that these estimates are mathematically (in exact arithmetic) equivalent. A short algebraic manipulation gives

$$\begin{aligned} c^* A^{-1} b - c^* x_n &= c^* A^{-1} r_n \\ &= c^* A^{-1} r_n - y_n^* r_n + y_n^* r_n \\ &= s_n^* A^{-1} r_n + y_n^* r_n \,. \end{aligned} \tag{3.15}$$

Using the global biorthogonality condition (3.6) and $y_n = W_n g_n$ (see (3.5)) we get $y_n^* r_n = 0$ and, consequently,

$$c^*A^{-1}b = c^*x_n + s_n^* A^{-1} r_n \,. \tag{3.16}$$

Comparing (3.13), (3.16), (3.8), and (3.10) we obtain the (exact arithmetic) equivalence

$$\xi_n^{\mathrm{B}} = \sum_{j=0}^{n-1} \alpha_j s_j^* r_j = c^* x_n = c^* b \, (T_n^{-1})_{1,1} \,. \tag{3.17}$$

Although ξ_n^{B} was derived by simple algebraic manipulations without using (3.1), the equivalence (3.17) shows its connection to matching moments model reduction. This connection is, in our opinion, significant for understanding the proposed estimate ξ_n^{B} representing a numerically efficient way of computing (3.1). It is worth pointing out that analogously to the HPD case (see [52]), in finite precision computations (3.17) does not hold, and, as demonstrated below, the individual (mathematically equivalent) approximations can behave very differently.

Saylor and Smolarski [44] introduced formally orthogonal polynomials and complex Gauss quadrature as a tool for approximating the quantity $c^*A^{-1}b$ (for an earlier introduction of the Gauss quadratures associated with the non-Hermitian Lanczos algorithm see, e.g., [12]). The paper [44] presents an approximation to $c^*A^{-1}b$ mathematically equivalent to c^*x_n. Its derivation assumes that the matrix A is diagonalizable (which is restrictive). Moreover, the result is computationally less convenient than the new ξ_n^{B} defined by (3.14). Therefore we will not consider the approximation from [44] in further detail.

Apart from the existence of the BiCG iterations in the steps 1 through n, ξ_n^{B} does not require any further assumptions. It can be computed with negligible additional cost from the quantities α_j and $s_j^* r_j$ available during the BiCG run. Please note that in order to compute ξ_n^{B}, the approximate solutions x_n and y_n need not be formed.

3.2. Estimating $c^*A^{-1}b$ using hybrid BiCG methods.

Each step of BiCG requires a matrix-vector product with A and a matrix-vector product with A^*. The idea of Sonneveld [48] was to avoid the multiplication with A^*. The resulting Conjugate Gradient Squared algorithm (CGS) uses two multiplications with A per iteration and it computes approximate solutions only to the primal system. In order to smooth out possible oscillations and to obtain faster convergence, Sonneveld's idea was further developed by Van der Vorst, Gutknecht, their coworkers and other authors to hybrid BiCG methods like BiCG Stabilized (BiCGStab) [54]; see also [27], [47], [3, Chapter 5].

Denoting by $\mathbf{r}_n$ the residual corresponding to the approximate solution $\mathbf{x}_n$ computed by a hybrid BiCG method, we get

$$c^*A^{-1}b = c^*x = c^*\mathbf{x}_n + c^*(x - \mathbf{x}_n) = c^*\mathbf{x}_n + c^*A^{-1}\mathbf{r}_n. \tag{3.18}$$

It is natural to ask whether the inner product $c^*\mathbf{x}_n$ provides a better approximation to $c^*A^{-1}b$ than the BiCG–based c^*x_n. To answer this question, we write the residual vector $\mathbf{r}_n$ in the form

$$\mathbf{r}_n = \psi_n(A)\, r_n \,,$$

where r_n is the BiCG residual and ψ_n is a polynomial of degree n such that $\psi_n(0) = 1$, i.e. $\psi_n(z) = 1 + z\varphi_{n-1}(z)$, where φ_{n-1} is a polynomial of degree $n-1$. The choice of ψ_n determines the particular hybrid BiCG method. From

$$b - A\mathbf{x}_n = \mathbf{r}_n = \psi_n(A)r_n = r_n + A\varphi_{n-1}(A)r_n = b - Ax_n + A\varphi_{n-1}(A)r_n$$

we get

$$\mathbf{x}_n = x_n - \varphi_{n-1}(A)r_n .$$

Since $\varphi_{n-1}(A)^*c \in \mathcal{K}_n(A^*, c)$ and $r_n \perp \mathcal{K}_n(A^*, c)$, we finally get

$$c^*\mathbf{x}_n = c^*x_n - (\varphi_{n-1}(A)^*c)^*r_n = c^*x_n \, . \tag{3.19}$$

In other words, although $\mathbf{x}_n$ can be a better (or worse) approximation to x than the BiCG approximation x_n, both provide the mathematically identical approximations to $c^*A^{-1}b$.

The BiCG coefficients α_j are available in hybrid BiCG methods. The BiCG residuals r_j and s_j are not available, but the inner products $s_j^*r_j$ can be computed as $s_j^*r_j = s_0^*\tilde{\psi}_j(A)r_j \equiv \tau_j$ providing that the leading coefficients in $\tilde{\psi}_j$ and in the polynomial defining s_j are equal. Then

$$\xi_n^{\mathrm{B}} = \sum_{j=0}^{n-1} \alpha_j \tau_j \, . \tag{3.20}$$

Alternatively, we can compute τ_j using the explicitly available coefficients η_j as

$$\tau_0 \equiv c^*b, \quad \tau_j \equiv \eta_j\tau_{j-1} = \prod_{k=0}^{j-1} \frac{s_{k+1}^*r_{k+1}}{s_k^*r_k} = s_j^*r_j \, , \qquad j = 1, \ldots, n-1 \, . \tag{3.21}$$

Although ξ_n^{B} computed via (3.20) using hybrid BiCG methods is mathematically the same as ξ_n^{B} computed via (3.14) using BiCG, results of their numerical evaluation may substantially differ; see Section 7.

3.3. Estimating $c^*A^{-1}b$ via the Arnoldi algorithm. As with the non-Hermitian Lanczos algorithm and the related BiCG, estimating $c^*A^{-1}b$ via the Arnoldi algorithm uses (3.1) where A_n arises from the associated Vorobyev moment problem; see Section 2.2. Taking $u_1 = c$ and $v_1 = b/\|b\|$ and using (2.11), the approximation (3.1) is in the Arnoldi algorithm given by

$$c^*A_n^{-1}b = \|b\| \, t_n^*H_n^{-1}e_1 \, ,$$

where $t_n \equiv V_n^*c$. We therefore denote

$$\xi_n^{\mathrm{A}} \equiv \|b\| \, t_n^*H_n^{-1}e_1 \, , \tag{3.22}$$

where the superscript "A" means "Arnoldi". Note that the same formula can be obtained using the quadrature rules in [5, pp. 776–777].

The significance of ξ_n^{A} (in comparison with ξ_n^{B}) is in the fact that A_n associated with the Arnoldi algorithm is based on orthogonal projections; see Section 2.2. Moreover, although the Arnoldi algorithm matches less moments than the non-Hermitian Lanczos algorithm, it is worth to note that H_n in (3.22) contains $n(n+1)/2 + n - 1$

generally nonzero elements while T_n in (3.17) only $3n-2$ generally nonzero elements. The upper Hessenberg matrix H_n may contain more information about the original model represented by A, b and c than the tridiagonal matrix T_n. Since

$$x_n = A_n^{-1}b = \|b\|\, V_n H_n^{-1} e_1 \tag{3.23}$$

represents the approximate solution of $Ax = b$ in the full orthogonalization method (FOM) (see [41, pp. 159–160]) we can write

$$\xi_n^{\mathrm{A}} = c^* x_n \,, \tag{3.24}$$

where x_n is computed by FOM.

In the HPD case and CG the approximate solution x_n is computed using short recurrences. In finite precision arithmetic computations, short recurrences typically lead to a fast loss of orthogonality due to rounding errors, and, consequently, to *delay of convergence*. Similar behavior can be expected with non-Hermitian Lanczos, BiCG and hybrid BiCG methods due to loss of biorthogonality. Since the Arnoldi algorithm uses long recurrences, the orthogonality among the computed basis vectors is lost in finite precision arithmetic computations only gradually (details of rounding error analysis can be found in [36] and in the earlier literature referenced there). Therefore, unlike in BiCG or in the hybrid BiCG methods, (see (3.8) and (3.19)), in FOM the formula (3.24) can be used in practical computations without delay of convergence due to rounding errors.

4. Transformation to the Hermitian positive definite case.

Numerical approximations of the bilinear form $c^*A^{-1}b$ presented in Section 3 used non-Hermitian Krylov subspace methods applied to the nonsingular complex matrix A. Here we write the bilinear form as

$$c^*A^{-1}b = c^*A^*(AA^*)^{-1}b = c^*(A^*A)^{-1}A^*b \,, \tag{4.1}$$

which suggests deriving its approximation by defining $\tilde{c} = Ac$ and approximating $\tilde{c}^*(AA^*)^{-1}b$. A second possibility is to approximate $c^*(A^*A)^{-1}\tilde{b}$, where $\tilde{b} = A^*b$. In either case, the problem of interest is to approximate $u^*B^{-1}v$ where B is Hermitian and positive definite; see also [17, Section 3.2]. For simplicity we consider only the second choice.

4.1. Using the polarization identity.

If B is real, symmetric and positive definite, it was suggested in [17, pp. 16 and 33] and [21, p. 242] that a polarization identity can be used to approximate $u^*B^{-1}v$, where $u \neq v$. On a complex Hilbert space with the inner product $\langle \cdot, \cdot \rangle$, conjugate linear in the second variable, the polarization identity takes the form (see e.g. [32], [57, p. 23])

$$\begin{aligned} 2\langle v, u\rangle &= (\|v+u\|^2 - \|v-u\|^2 + \mathbf{i}\|v+\mathbf{i}u\|^2 - \mathbf{i}\|v-\mathbf{i}u\|^2)/2 \\ &= \|v+u\|^2 - (1+\mathbf{i})(\|v\|^2 + \|u\|^2) + \mathbf{i}\|v+\mathbf{i}u\|^2 \,. \end{aligned} \tag{4.2}$$

Defining $\langle v, u\rangle \equiv u^*B^{-1}v$, the term $\|u\|^2$ in (4.2) is given by $u^*B^{-1}u$. With $v = A^*b$ and $B = A^*A$, it follows that $v^*B^{-1}v = b^*b$. The remaining three terms that need to be approximated are

$$(v+u)^*B^{-1}(v+u)\,, \quad (v+\mathbf{i}u)^*B^{-1}(v+\mathbf{i}u) \quad \text{and} \quad u^*B^{-1}u \tag{4.3}$$

all of which have the form $w^*B^{-1}w$ with the HPD matrix B. Then BiCG reduces to the standard CG, with (3.13) giving

$$w^*B^{-1}w = \xi_n^{\mathrm{CG}} + r_n^* B^{-1} r_n\,, \qquad \xi_n^{\mathrm{CG}} \equiv \sum_{j=0}^{n-1} \alpha_j \|r_j\|^2\,; \tag{4.4}$$

see [52, relation (3.8)]. Since $B = A^*A$, the quantities α_j, $\|r_j\|^2$ and thus ξ_n^{CG} can conveniently be computed without forming the matrix B using the algorithms CGNR; see [29, Section 10] where "NR" comes from <u>N</u>ormal Equation <u>R</u>esidual [22, Section 10.4]. As an alternative one can consider the HPD analogy of (3.10) with $b = c = w$ and T_n resulting from the n steps of the Hermitian Lanczos algorithm applied to the matrix $B = A^*A$ with the starting vector b. This gives

$$w^*B^{-1}w = \|b\|^*(T_n^{-1})_{1,1} + r_n^* B^{-1} r_n, \tag{4.5}$$

where r_n is as in (4.4). Numerically this can be efficiently computed via the algorithm LSQR proposed by Paige and Saunders [38, 37] which uses the Golub-Kahan bidiagonalization [14] and computes the Cholesky factor of T_n.

The approximation error $r_n^* B^{-1} r_n$ in (4.4)–(4.5) is equal to the squared energy norm of the error in CG and therefore it is monotonically decreasing with n. This represents a significant difference in comparison with (3.13), where the error term $s_n^* A^{-1} r_n$ typically oscillates. There are methods for computing the upper and lower bounds for $w^*B^{-1}w$; see [17, 21, 18, 6]. Consequently, using (4.2), one can compute (assuming exact arithmetic) upper and lower bounds for the real and imaginary parts of the bilinear form $c^*A^{-1}b$. Moreover (4.4) holds, up to a small error, also for quantities computed in finite precision arithmetic; see [52]. (It is worth to point out that ξ_n^{CG} computed in finite precision arithmetic can be much larger than its exact arithmetic counterpart computed at the same step.) The price of transforming the non-Hermitian problem to the Hermitian one using the polarization identity (4.2) is, however, substantial. Approximation of three terms (4.3) requires three CG computations with the *same* matrix B and *different* initial vectors, with total *six* matrix-vector multiplications (three with A and three with A^*) per one iteration step. In our experiments, the approach using the polarization identity (4.2) was not competitive with ξ_n^{B}.

4.2. Using the normal equations. Another way to approximate the bilinear form $c^*A^{-1}b$ is to apply CGNR to $A^*Ax = A^*b$. The bilinear form can then be approximated by c^*x_n, where x_n is the nth iterate of CGNR. Unlike in Section 4.1, here only two matrix-vector products (one with A and one with A^*) are needed at each iteration. As with (3.8) in Section 3.1, in finite precision arithmetic computing a sufficiently accurate approximation using c^*x_n may be delayed due to loss of orthogonality caused by rounding errors.

Rewriting the bilinear form using $c^*(A^*A)^{-1}A^*b$ as in (4.1), one can also consider BiCG applied to $B = A^*A$ with *two different* initial vectors $u = A^*b$ and $v = c$; for an analogous approach using the non-Hermitian Lanczos algorithm see [17, Sections 3.2 and 4.2]. BiCG applied to a system with the matrix B and two different initial vectors needs *four* matrix-vector multiplications (two with A and two with A^*) per iteration.

4.3. The GLSQR approach. Saunders et al. suggested in [42] the so called Generalized LSQR method (GLSQR) which is applied to a matrix and two starting vectors. It can be seen as the block-Lanczos algorithm applied to the matrix A^*A with the starting block $[c, A^*b]$; see also [39]. The GLSQR method solves simultaneously the primal and dual systems (similarly to BiCG in Algorithm 1). The nth step is associated with the following relations

$$\begin{aligned} AV_n &= U_n T_n + \zeta_{n+1} u_{n+1} e_n^T \,, \\ A^* U_n &= V_n T_n^* + \theta_{n+1} v_{n+1} e_n^T \,, \end{aligned}$$

where $u_1 = b/\|b\|$, $v_1 = c/\|c\|$, $V_n = [v_1, \ldots, v_n]$ and $U_n = [u_1, \ldots, u_n]$ are orthonormal matrices, $V_n^* v_{n+1} = 0$, $U_n^* u_{n+1} = 0$, T_n is tridiagonal, and ζ_{n+1} and θ_{n+1} are the normalization coefficients. Using GLSQR and applying the block Gauss quadrature rule from [17, Sections 3.3 and 4.3], Golub, Stoll and Wathen derived the following approximation to $c^*A^{-1}b$

$$\xi_n^{\mathrm{G}} = \|b\| \, \|c\| \, e_1^T T_n^{-1} e_1, \tag{4.6}$$

where the superscript “G” means GLSQR; see [20, Section 3.3]. The GLSQR approach requires *two* matrix-vector multiplications (one by A and one by A^*) per iteration.

5. Preconditioning. Let P_L and P_R be nonsingular matrices such that the systems of linear algebraic equations with the matrices P_L and P_R are easily solvable. Clearly

$$c^*A^{-1}b = (P_R^{-*}c)^*(P_L^{-1}AP_R^{-1})^{-1}(P_L^{-1}b) = \mathbf{c}^*\mathbf{A}^{-1}\mathbf{b},$$

where $\mathbf{A} \equiv P_L^{-1}AP_R^{-1}$, $\mathbf{c} \equiv P_R^{-*}c$ and $\mathbf{b} \equiv P_L^{-1}b$. The approximation techniques described above can be applied to the preconditioned problem $\mathbf{c}^*\mathbf{A}^{-1}\mathbf{b}$. Preconditioning should lead to faster convergence. As a side effect, fast convergence can help preventing significant delays due to rounding errors; see the illustrations in Section 7. It is obvious that $\mathbf{A}^{-1}$ need not be formed explicitly.

6. Comments on numerical stability issues. A thorough numerical stability analysis of the approaches for approximating the bilinear form $c^*A^{-1}b$ which are presented in this paper is yet to be done. Here we concentrate on supporting arguments for the claim that the new estimate ξ_n^{B} (see (3.14)) should be preferred to the mathematically equivalent (and commonly used) scattering amplitude estimate c^*x_n; see (3.8).

Using $A^{-1}r_n = x - x_n$, we rewrite for clarity of exposition the formulas which express the errors of the computed approximation (see (3.13)–(3.16))

$$c^*A^{-1}b \;=\; \xi_n^{\mathrm{B}} + s_n^*(x - x_n)\,, \qquad \xi_n^{\mathrm{B}} = \sum_{j=0}^{n-1} \alpha_j s_j^* r_j\,, \tag{6.1}$$

$$\begin{aligned} c^*A^{-1}b \;&=\; c^*x_n + c^*(x - x_n) && (6.2)\\ &=\; c^*x_n + s_n^*(x - x_n) + y_n^*(b - Ax_n)\,. && (6.3) \end{aligned}$$

Mathematically (in exact arithmetic),

$$y_n^*(b - Ax_n) = y_n^* r_n = g_n^* W_n r_n = 0 \tag{6.4}$$

due to the global biorthogonality condition (3.6). Therefore

$$s_n^*(x - x_n) = c^*(x - x_n)\,. \tag{6.5}$$

In computations using finite precision arithmetic the global biorthogonality (3.6) is in general lost, and, subsequently, (6.5) does not hold. Let the quantities computed using finite precision arithmetic are denoted by " ˆ ". Supposing that the BiCG residual $\hat{s}_n$ for the dual problem $A^*y = c$ is small, we may expect

$$|\hat{s}_n^*(x - \hat{x}_n)| \ll |c^*(x - \hat{x}_n)| \,. \tag{6.6}$$

This corresponds to $\hat{\xi}_n^{\mathrm{B}}$ much closer to $c^*A^{-1}b$ than $c^*\hat{x}_n$. In other words, in finite precision arithmetic computations the term $\hat{y}_n^*(b - A\hat{x}_n)$ as well as a possible difference between the true and iteratively computed residuals must be taken into account (for the symmetric positive definite analogy see [52, Section 6]). Providing that the finite precision analogies of (6.1) and (6.3) hold up to a small inaccuracy, the term $\hat{y}_n^*(b - A\hat{x}_n)$ would explain numerical behaviour of the estimate $c^*\hat{x}_n$.

Analogously to (3.15) one can easily derive for the computed approximations $\hat{x}_n$ and $\hat{y}_n$

$$c^*A^{-1}b = c^*\hat{x}_n + (c - A^*\hat{y}_n)^*(x - \hat{x}_n) + \hat{y}_n^*(b - A\hat{x}_n).$$

Therefore, (6.3) holds, up to small inaccuracy, also for results of finite precision computations, until the true residual $b - A\hat{y}_n$ does not differ significantly from the iteratively computed residual $\hat{s}_n$. For more details on the difference between the true and the iteratively computed residuals see the analysis in [47, 26].

Concerning the finite precision analogy of (6.1), the situation is much more complicated. Consider first $A \in \mathbb{R}^{N\times N}$ symmetric positive definite and $c = b \in \mathbb{R}^N$. Then BiCG reduces to CG, $r_n = s_n$ and (6.1) can be rewritten as

$$b^TA^{-1}b = \xi_n^{\mathrm{CG}} + r_n^TA^{-1}r_n \,, \qquad \xi_n^{\mathrm{CG}} = \sum_{j=0}^{n-1} \alpha_j \, \|r_j\|^2 \,, \tag{6.7}$$

or, considering that $r_n^TA^{-1}r_n = (x - x_n)^TA(x - x_n)$, $b^TA^{-1}b = x^TAx$,

$$\|x\|_A^2 = \xi_n^{\mathrm{CG}} + \|x - x_n\|_A^2 \,, \tag{6.8}$$

where the A-norm of a vector z is defined by $\|z\|_A \equiv (z^*Az)^{1/2}$. It was proved in [52] that (6.8) holds also for the results of finite precision arithmetic computations up to a term proportional to $\varepsilon\|x\|_A\|x - \hat{x}_n\|_A$; here ε denotes machine precision unit (we omit some tedious details). Consequently, until $\|x - \hat{x}_n\|_A = (\hat{r}_n^TA^{-1}\hat{r}_n)^{1/2}$ becomes close to $\varepsilon\|x\|_A$, the computed $\hat{\xi}_n^{CG}$ approximates $b^TA^{-1}b = \|x\|_A^2$ with the error of the approximation being close to $\hat{r}_n^TA^{-1}\hat{r}_n = \|x - \hat{x}_n\|_A^2$; see [52, Theorem 10.1]. This result is proved in several steps with two main ingredients. First, it is proved that the iteratively computed residual $\hat{r}_j$ (see Algorithm 1 with $A = A^*$ and $s_0 = r_0 = b$) is sufficiently close to the residual $b - A\hat{x}_n$ computed directly from the approximate solution $\hat{x}_n$. Second, it is proved that the local orthogonality between the residuals and the search vectors $\hat{p}_j^T\hat{r}_{j+1}$ is preserved proportionally to machine precision ε; see [52, Section 9].

For BiCG one can hardly expect results of the same strength. In particular, a close preservation of the local biorthogonality conditions (3.12) can not be proved due to the possible occurrence of the so called breakdowns, when $\hat{q}_jA\hat{p}_j$ or $\hat{s}_j^*\hat{r}_j$ become zero. Note that the breakdowns are not caused by rounding errors; they can occur in exact arithmetic.

Using the technique from [52, 53], one can express the inner product $\hat{q}_j^*\hat{r}_{j+1}$ of the quantities computed in finite precision arithmetic using Algorithm 1 as

$$\hat{q}_j^*\hat{r}_{j+1} \;=\; \frac{\hat{s}_j^*\hat{r}_j}{\hat{s}_{j-1}^*\hat{r}_{j-1}}\,\hat{q}_{j-1}^*\hat{r}_j + \varepsilon\,\vartheta_j\,,$$

and the size of ϑ_j can be bounded by the norms of the computed vectors, the norm of A, and the size of the coefficient $\hat{\alpha}_j$. By induction we obtain, after some algebraic manipulations (cf. [52, p. 74] or [53, p. 805]),

$$\hat{q}_n^*\hat{r}_{n+1} \;=\; \varepsilon\,\hat{s}_n^*\hat{r}_n\sum_{j=0}^{n}\frac{\vartheta_j}{\hat{s}_j^*\hat{r}_j} + \mathcal{O}(\varepsilon^2)\,. \tag{6.9}$$

Now we can clarify the differences between the CG case and the BiCG case.

In the CG case $\hat{s}_j = \hat{r}_j$ and $\hat{s}_j^*\hat{r}_j = \|\hat{r}_j\|^2$. As shown in [52], the size of ϑ_j is bounded by $\kappa(A)\,\|\hat{r}_j\|^2$. In summary, the local biorthogonality is bounded by a multiple of $\varepsilon\,\|\hat{r}_j\|^2\,\kappa(A)$; see [52, (9.14) and (9.15)]. In the BiCG case, $\hat{q}_j^*A\hat{p}_j$ and $\hat{s}_j^*\hat{r}_j$ can become zero due to breakdowns. In practice the exact breakdowns are very rare, but near breakdowns can cause the corresponding terms in the sum (6.9) to be large. If near breakdowns appear in BiCG, then preserving the local biorthogonality condition (3.12) up to a small inaccuracy can not be guaranteed in finite precision arithmetic computations. Therefore we were not able to prove that (6.1) holds, up to a small inaccuracy, also in finite precision arithmetic computations. Nevertheless, for ξ_n^{B} there is no need of preserving the *global orthogonality* conditions (3.6)–(3.7), and, in particular, of $y_n^*r_n = 0$, as in the scattering amplitude approximations. This represents a strong numerical argument in favour of the proposed estimate ξ_n^{B}.

7. Application and numerical experiments.

We will illustrate the behaviour of various approaches for approximation of the bilinear form $c^*A^{-1}b$ on several examples of different origin. In this section we omit for simplicity the " ˆ " notation for the computed quantities.

7.1. Test problems.

This paper was practically motivated by the problem of diffraction of light on periodic structures and the RCWA method for its solution; see the monograph [35] and the references given there. Application of the RCWA method can lead to the system of linear algebraic equations, which for the simplest standard 2D model problem has the form (see [28, Section 3.5]),

$$A\,x \equiv \begin{bmatrix} -I & I & e^{\mathbf{i}\sqrt{C}\varrho} & 0 \\ Y_I & \sqrt{C} & -\sqrt{C}e^{\mathbf{i}\sqrt{C}\varrho} & 0 \\ 0 & e^{\mathbf{i}\sqrt{C}\varrho} & I & -I \\ 0 & \sqrt{C}e^{\mathbf{i}\sqrt{C}\varrho} & -\sqrt{C} & -Y_{\mathrm{II}} \end{bmatrix} x \;=\; b\,, \tag{7.1}$$

where Y_{I}, Y_{II} are $(2M+1)\times(2M+1)$ complex diagonal matrices, C is a $(2M+1)\times(2M+1)$ complex Toeplitz plus diagonal matrix, ϱ is a given real and positive parameter, and M is the discretization parameter representing the number of Fourier modes used for approximation of the electric and magnetic fields as well as the material properties. The block structure of (7.1) corresponds to the geometric structure of the physical problem with one slab, where the individual block rows represent the boundary conditions for the electric and magnetic fields on the interface between the slab and the superstrate and the slab and the substrate. For the geometric structure with S slabs the overall number of interfaces is $S+1$, which gives $2(S+1)$ block equations (for (7.1) $2(1+1)=4$). In 3D problems the size of the individual blocks is proportional to the square of the number the Fourier modes.

In real RCWA applications the blocks of the matrix A can not be formed by evaluating the matrix functions. Considering time constraints given by technological restrictions, that would be too slow. Moreover, one does not need the whole solution of the linear algebraic system. For (7.1) one typically needs only the dominant $(M+1)$st component (here e_{M+1} denotes the vector of the compatible dimension with the $(M+1)$st element equal to one and all other elements equal to zero)

$$e_{M+1}^{*}A^{-1}b\,; \tag{7.2}$$

see [28, Section 3.5, relation (3.45)]. Therefore the problem seems to be well suited for an iterative approximation of the bilinear form (1.1) with $c = e_{M+1}$. In our experiments we use $M = 20$, $S = 1$ and $M = 20$, $S = 20$, leading to the resulting RCWA – motivated matrices:

- TE2001 (RCWA, 20 Fourier modes and 1 slab), the matrix $A \in \mathbb{C}^{164\times164}$ is complex nonsymmetric, $\kappa(A) \approx 112$, starting vectors b and c arise from the problem formulation;
- TE2020 (RCWA, 20 Fourier modes and 20 slabs), the matrix $A \in \mathbb{C}^{1722\times1722}$ is complex nonsymmetric, $\kappa(A) \approx 2.9e+03$, starting vectors b and c arise from the problem formulation.

In addition, we use in our illustrations four publicly available matrices from different sources:

- young1c (ACOUST, HB Collection), the matrix $A \in \mathbb{C}^{841\times841}$ is complex symmetric, $\kappa(A) \approx 415$;
- orsirr1 (OILGEN, HB Collection), the matrix $A \in \mathbb{R}^{1030\times1030}$ is real nonsymmetric, $\kappa(A) \approx 7.7e+04$;
- pde2961 (MATPDE, NEP Collection), the matrix $A \in \mathbb{R}^{2961\times2961}$ is real nonsymmetric, $\kappa(A) \approx 642.5$;
- af23560 (AIRFOI, NEP Collection), the matrix $A \in \mathbb{R}^{23560\times23560}$ is real nonsymmetric, the condition number estimate computed via the Matlab command condest(A) gives $\kappa(A) \approx 3.5e+05$.

Except for TE2001 and TE2020 we choose b and c normalized random vectors.

7.2. An overview of compared methods and their implementations. In this paper we presented three approaches for approximating the bilinear form $c^*A^{-1}b$: The non-Hermitian Lanczos approach, the Arnoldi approach and the approach based on transformation to the HPD case. In our numerical experiments we use the standard versions of BiCG [10], CGS [48], BiCGStab(4) [47], modified Gram-Schmidt Arnoldi [41] and GLSQR [20]. For illustration of the behavior of BiCG in exact precision arithmetic we run in some experiments BiCGreo with the rebiorthogonalized basis vectors at each step (at step n, r_n is reorthogonalized against the previously computed $s_0, s_1, \ldots, s_{n-1}$, and s_n is reorthogonalized against the previously computed $r_0, r_1, \ldots, r_{n-1}$). We use a special version of the BiCGStab [54] algorithm with the technique suggested in [46] (we choose the free parametr $\Omega = 0.7$) to improve the accuracy of the computed BiCG coefficients. We compare the approximations ξ_n^{B} (see (3.14) and (3.20)) and c^*x_n computed via BiCG, BiCGreo and the hybrid BiCG methods, ξ_n^{A} (see (3.22)) computed via the Arnoldi algorithm and ξ_n^{G} (see (4.6)) computed via GLSQR. We do not include in our experiments the approximation (3.10)

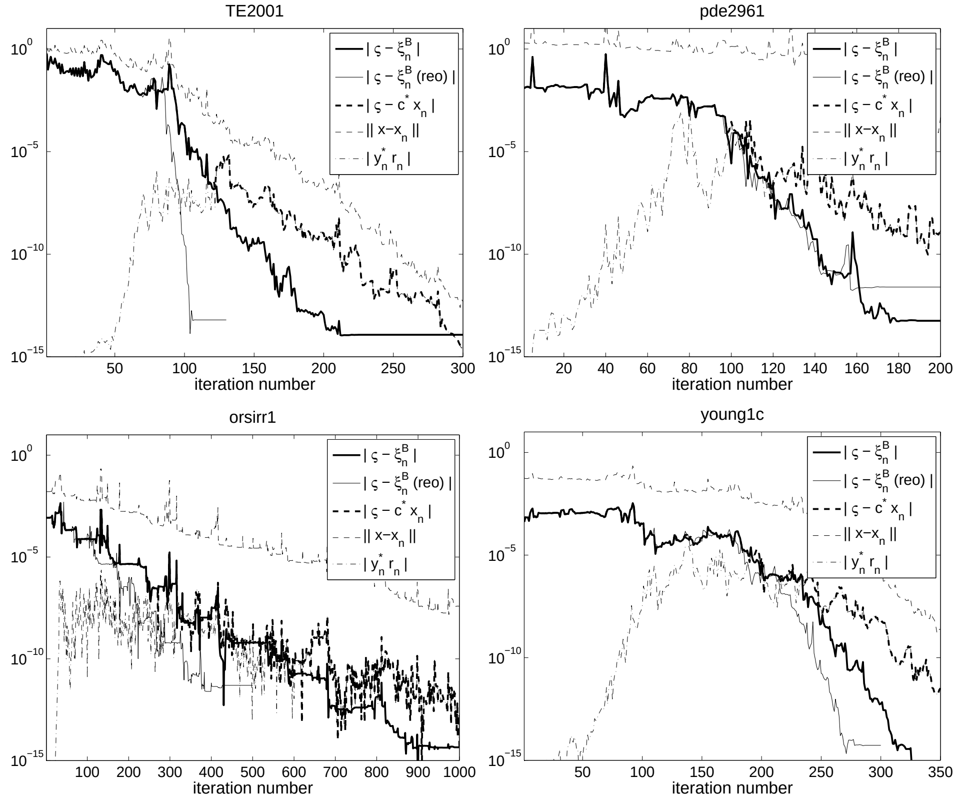

Fig. 7.1. *Comparison of the errors* $|\varsigma-\xi_n^{\mathrm{B}}|$ *(bold solid line) and* $|\varsigma-c^*x_n|$ *(bold dashed line) for the mathematically equivalent approximations computed via BiCG. Both approximations are close to each other until the size of* $|y_n^*r_n|$ *(dash-dotted line) is negligible in comparison to the size of* $|c^*x_n|$*. To simulate the behavior of* ξ_n^{B} *in exact arithmetic, we also plot* $|\varsigma-\xi_n^{\mathrm{B}}\ (\mathrm{reo})|$ *with* $\xi_n^{\mathrm{B}}\ (\mathrm{reo})$ *computed via BiCGreo (solid line).*

computed via the non-Hermitian Lanczos algorithm. It gives very similar results as ξ_n^{B} computed via BiCG. We also do not present results for the approximations introduced in Section 4.1. On our set of problems they do not seem to be competitive with other approximations; see the comment in Section 7.5.

Denote for simplicity of further presentation

$$\varsigma(A,b,c) \equiv \varsigma = c^*A^{-1}b\,.$$

The value ς used for determining the approximation error in all subsequent experiments was computed using the Matlab command `c'(A\b)`.

7.3. Comparison of the approximations ξ_n^{B} and c^*x_n. In Figure 7.1 we compare the error $|\varsigma-\xi_n^{\mathrm{B}}|$ of the new approximation ξ_n^{B} (see (3.14)) (bold solid line) with the error $|\varsigma-c^*x_n|$ of the scattering amplitude approximation c^*x_n (see (3.8)) where x_n is computed by Algorithm 1 (dashed line). In order to illustrate the effects of rounding errors to the BiCG algorithm we plot also $|\varsigma-\xi_n^{\mathrm{B}}\ (\mathrm{reo})|$ for $\xi_n^{\mathrm{B}}\ (\mathrm{reo})$ computed via BiCGreo. The comparison is complemented by the upper bound $\|x-x_n\| \geq |c^*(x-x_n)|$ (here $\|c\|=1$) and by the value $|y_n^*r_n|$ (dash-dotted line) which in finite precision arithmetic computations determines the difference between ξ_n^{B} and

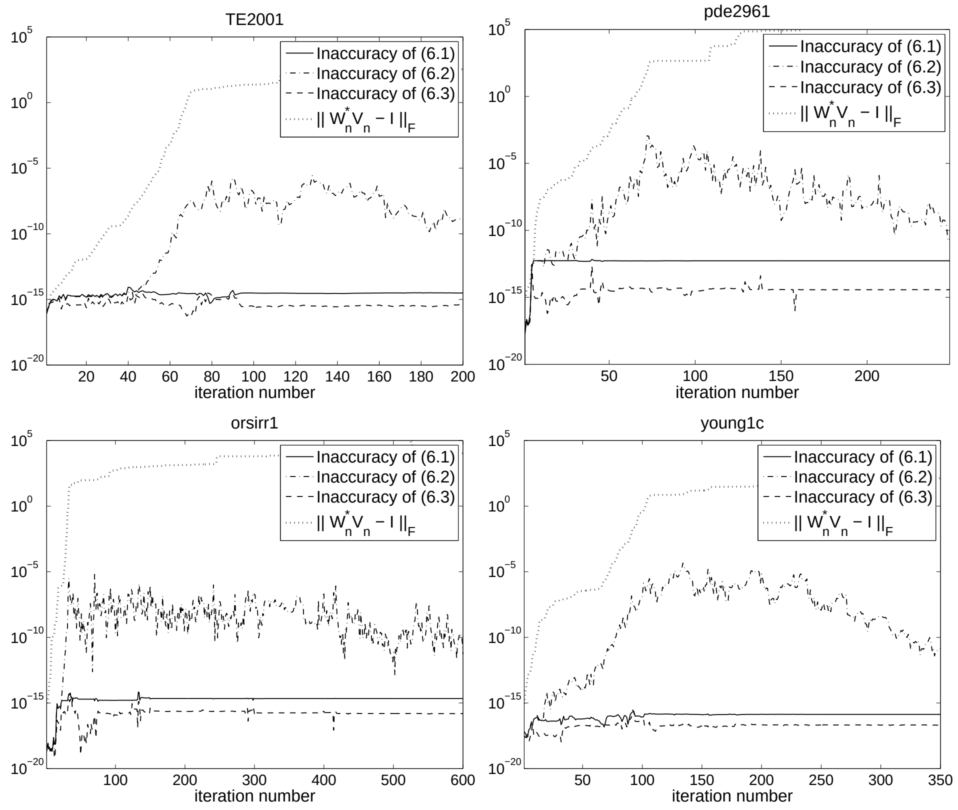

Fig. 7.2. *Inaccuracy in the equations (6.1), (6.2) and (6.3) for the quantities computed in finite precision arithmetic. For each equation we plot the absolute value of the difference of the terms on the left and on the right hand side.*

$c^* x_n$; see (6.1) and (6.3). The dashed line coincides in all figures with the bold solid line until the bold solid line is crossed by the dash-dotted line. It is interesting that for the matrix `pde2961` the approximations ξ_n^{B} (reo) and ξ_n^{B} almost coincide except for the fact that ξ_n^{B} (reo) exhibits larger maximal attainable accuracy (that can be attributed to additional accumulation of roundoff due to rebiorthogonalization). All our experiments confirm that the newly proposed approximation ξ_n^{B} should be preferred to computation of the scattering amplitude $c^* x_n$.

Figure 7.2 shows the inaccuracy of the equations (6.1), (6.2) and (6.3) for the quantities computed in finite precision arithmetic, as well as the loss of global biorthogonality in BiCG. While (6.1) and (6.3) are for all experiments using the matrices `TE2001`, `pde2961`, `orsirr1` and `young1c` satisfied up to the inaccuracy remarkably close to machine precision, (6.2) is considerably violated due to the loss of biorthogonality.

7.4. BiCG and hybrid BiCG methods in approximation of $c^* A^{-1} b$. As explained in Section 3.2, ξ_n^{B} can be computed using hybrid BiCG methods. It is however well known that computing the BiCG coefficients accurately may represent in hybrid BiCG methods a problem. As stated in [46, p. 220], “In order to maintain the convergence properties of the BiCG component in hybrid BiCG methods, it is

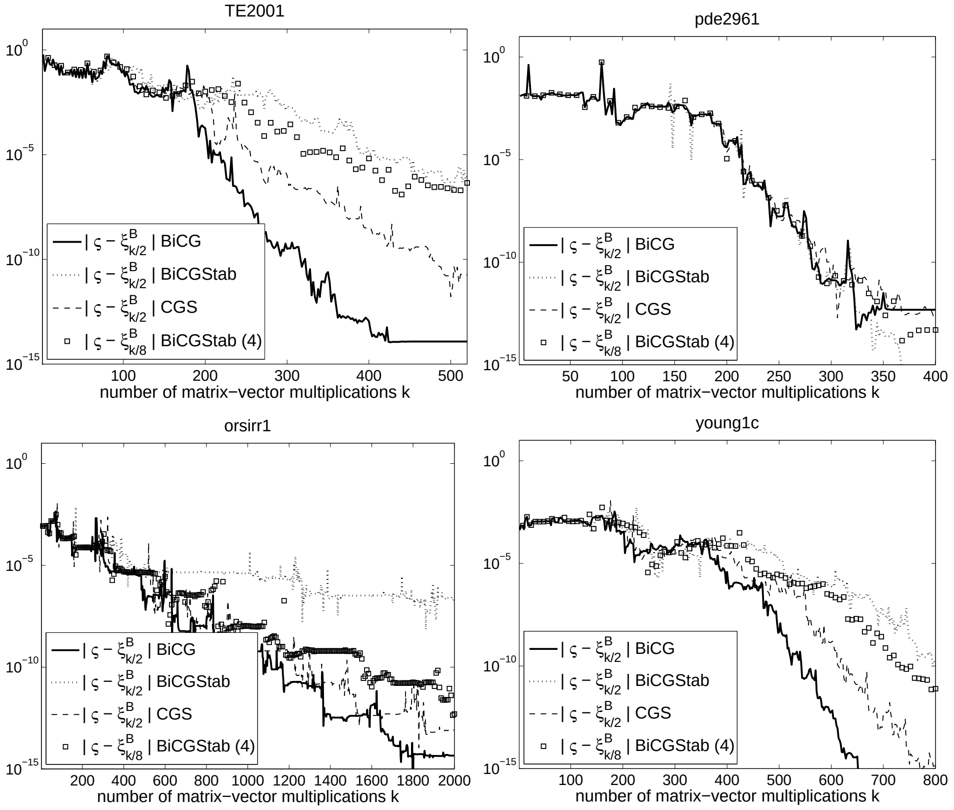

FIG. 7.3. *Comparison of errors $|\varsigma - \xi^{\rm B}_{k/2}|$ for the approximation $\xi^{\rm B}_{k/2}$ computed via BiCG (bold solid line), BiCGStab (dashed line), CGS (dotted line), and the error $|\varsigma - \xi^{\rm B}_{k/8}|$ of the approximation $\xi^{\rm B}_{k/8}$ computed via BiCGStab(4) (squares). The approximations obtained using the hybrid BiCG methods are often significantly more affected by rounding errors than $\xi^{\rm B}_{k/2}$ computed via BiCG. Here k denotes the number of matrix-vector multiplications. For BiCG, BiCGStab and CGS we have $k = 2n$ (two matrix-vector multiplications per iteration). The value $|\varsigma - \xi^{\rm B}_{k/2}|$ is plotted every second value of k. For BiCGStab(4) the value $|\varsigma - \xi^{\rm B}_{k/8}|$ is plotted every eights value of k.*

necessary to select polynomial methods for the hybrid part that permit to compute the BiCG coefficients as accurately as possible". The difficulty in using hybrid BiCG methods for approximating the bilinear form $c^*A^{-1}b$ is illustrated in Figure 7.3 for BiCGStab, CGS and BiCGStab(4). On the x-axis is the number of matrix-vector multiplications, that we denote by k. In all our computations we observed that for the hybrid BiCG methods the computed value $\xi^{\rm B}_n$ (see (3.20)) was always very close to the computed scattering amplitude $c^*\mathbf{x}_n$. This suggests that in hybrid BiCG methods both quantities are affected by rounding errors in a similar way. We observe that none of the hybrid BiCG methods perform in approximating the bilinear form $c^*A^{-1}b$ better than $\xi^{\rm B}_n$ computed via BiCG. On the opposite, in most cases they perform significantly worse. Techniques suggested in [46] applied to BiCGStab did not lead to a substantial improvement of the computed BiCGStab approximations.

In order to get an insight into this observation, we plot (as an example) in the upper part of Figure 7.4 the norm of the error $\|x - x_n\|$ (where x is determined via the Matlab command `A\b`). Note that the approximations to the solution x lie for

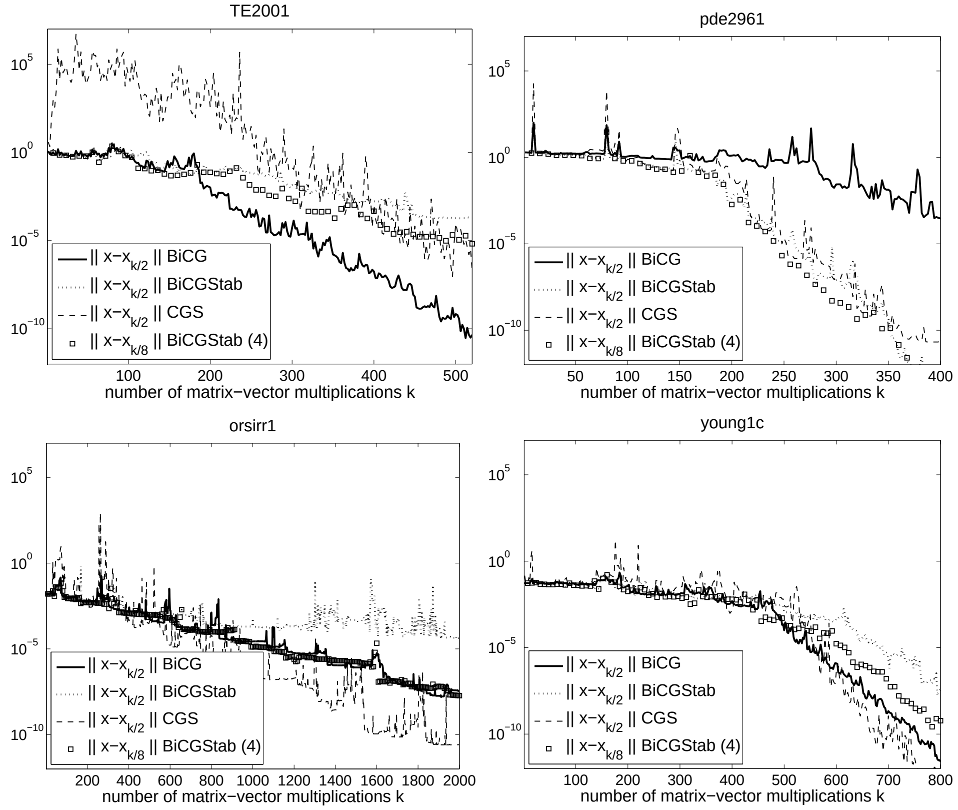

Fig. 7.4. *Euclidean norm of the error of the approximation to the solution of $Ax = b$ computed via BiCG and various hybrid BiCG methods. While BiCG seems to be a winner in approximating the bilinear form $c^*A^{-1}b$ (see Figure 7.3), hybrid BiCG methods are often more efficient in solving the system $Ax = b$.*

various methods in Krylov subspaces of various dimensions. In particular, the BiCG approximation x_n lies in $\mathcal{K}_n(A, b)$, the CGS and BiCGStab approximations $\mathbf{x}_n$ lie in $\mathcal{K}_{2n}(A, b)$ and the BiCGStab(4) approximation $\mathbf{x}_n$ lies in $\mathcal{K}_{8n}(A, b)$. For the matrix `TE2001` BiCG outperforms the other methods even in computing the approximate solution to $Ax = b$, while for the matrix `pde2961` it performs much worse than the hybrid BiCG methods, with BiCGStab(4) the winner. For `orsirr1` and `young1c` there is no clear winner (a more detailed comparison of BiCG and hybrid BiCG methods as linear algebraic solvers is out of the scope of this paper). Despite the fact that $\|x-x_n\|$ is for `pde2961` worst for the BiCG algorithm, the behaviour of $|s_n^*(x-x_n)|$ still causes ξ_n^{B} to behave even in this case about as well as the approximations computed via the hybrid BiCG methods.

In conclusion, in our experiments (this paper gives a small sample of them) the ξ_n^{B} computed via BiCG was not outperformed by the approximations computed via the hybrid BiCG methods. In most examples ξ_n^{B} computed via BiCG performed significantly better.

7.5. Transformation to the Hermitian positive definite case. From the approaches described in Section 4, GLSQR performed in our experiments best both in terms of iteration count and in the number of matrix-vector multiplications. However,

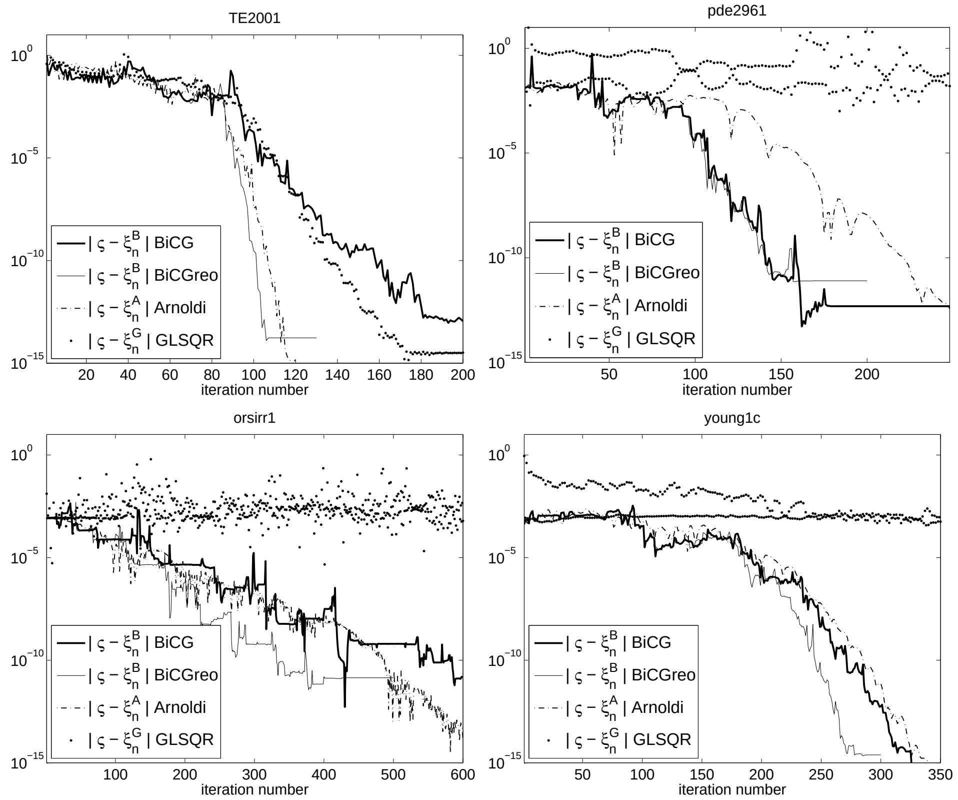

FIG. 7.5. *Comparison of errors for different approaches: $|\varsigma - \xi_n^{\mathrm{B}}|$ (bold solid line) from Section 3.1 with ξ_n^{B} computed via BiCG, $|\varsigma - \xi_n^{\mathrm{B}}(reo)|$ (solid line) with $\xi_n^{\mathrm{B}}(reo)$ computed via BiCGreo, $|\varsigma - \xi_n^{\mathrm{A}}|$ (dash-dotted line) with ξ_n^{A} computed via the modified Gram-Schmidt Arnoldi algorithm from Section 3.3 and $|\varsigma - \xi_n^{\mathrm{G}}|$ (dots) with ξ_n^{G} computed via GLSQR from Section 4.3.*

even GLSQR was in most cases rather slow, as documented below. This observation can not be explained by an effect of ill-conditioning of the matrix A^*A (in most of our experiments we used matrices with a moderate condition number). Results of further investigation of this topic will be reported elsewhere.

7.6. Comparison of approaches using different Krylov subspace methods. Figure 7.5 compares $|\varsigma - \xi_n^{\mathrm{B}}|$ with ξ_n^{B} (see (3.14)) from Section 3.1 computed via BiCG (bold solid line), $|\varsigma - \xi_n^{\mathrm{B}}\ (\mathrm{reo})|$ with $\xi_n^{\mathrm{B}}\ (\mathrm{reo})$ computed via BiCGreo (solid line), the error $|\varsigma - \xi_n^{\mathrm{A}}|$ with ξ_n^{A} (see (3.24)) computed via the modified Gram-Schmidt Arnoldi algorithm from Section 3.3 (dash-dotted line) and the error $|\varsigma - \xi_n^{\mathrm{G}}|$ of the GLSQR approximation ξ_n^{G} (see (4.6)) from Section 4.3 (dotted line).

We observe that the methods behave differently for different problems. Among the methods using short recurrences, the newly proposed approximation ξ_n^{B} wins except for `TE2001` where ξ_n^{G} performs slightly better (ξ_n^{B} (reo) is not considered a practical alternative). For other problems GLSQR approximation ξ_n^{G} performs rather poorly (please notice the “double lines” for the problems `pde2961` and `young1c`). The approximation ξ_n^{A} computed via the MGS Arnoldi algorithm converges faster than the approximations based on short recurrences (except for `young1c`), but slower than ξ_n^{B} (reo). We emphasize that the cost of the Arnoldi iteration increases with the iter-

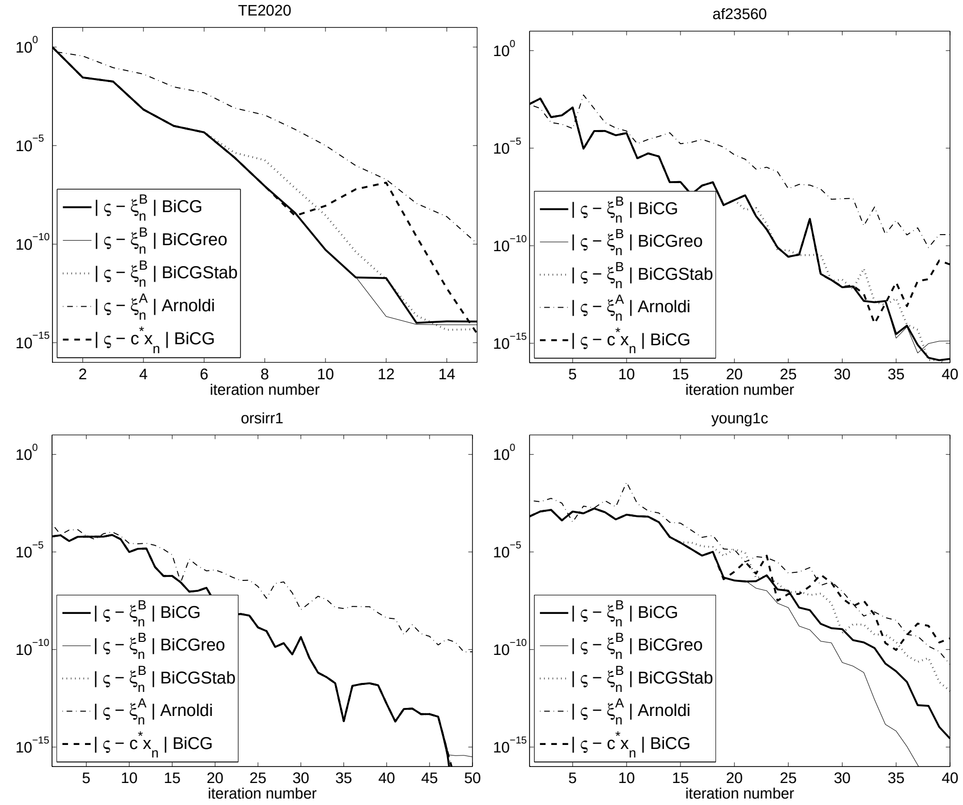

Fig. 7.6. *Comparison of errors for various approaches with preconditioning. The error of approximation computed via BiCG (bold solid line), BiCGStab (dashed line) and BiCGreo (solid line) are for some problems very close to each other.*

ation number n. The Arnoldi algorithm matches n moments, while the BiCG method $2n$ moments; see Sections 2.1, 2.2 and 3.1. Since the Arnoldi algorithm uses orthogonal projections while the BiCG method oblique projections, the smaller number of the matched moments alone does not explain the observed behaviour. The cost of computations can not be evaluated using matrix-vector products due to the fact that the other costs are for methods based on short recurrences (BiCG and GLSQR) and long recurrences (the Arnoldi algorithm) significantly different. In practical applications, the cost should be measured by computer time. At any case, our experiments suggest that the newly proposed ξ_n^{B} is highly competitive.

7.7. Preconditioning. In practice, iterative methods can not be used without efficient preconditioning. In Figure 7.6 we illustrate results of computations for the same approaches as in Figure 7.5 except for GLSQR which was skipped due to uncompetitive performance. (This does not mean, however, that GLSQR is in general uncompetitive. We were unable to make it work for our problems; the matter needs further investigation.) For `TE2020` we used a special preconditioning tailored to the problem, for `af23560` and `young1c` we used the incomplete Cholesky preconditioning with the drop tolerance 5×10^{-2} and 10^{-2} respectively (they were found experimentally as good compromises between performance and fill-in). For the problem `orsirr1` we used the incomplete Cholesky preconditioning with zero fill-in. We can observe that all approaches based on short recurrences, except for the scattering amplitude approximation c^*x_n computed via BiCG, are comparable (except for `young1c` they are very close or almost coincide with ξ_n^{B}). They clearly outperform ξ_n^{A} in terms of iterations. If the number of iterations is small, the comparison on a real-world problem with a significant cost of the matrix-vector multiplication might, however, be more in favour of ξ_n^{A} computed via the Arnoldi algorithm. It is worth to point out that due to long recurrences ξ_n^{A} can safely be computed via the FOM method using c^*x_n; see (3.24).

8. Concluding remarks. This paper proposes the new approximation ξ_n^{B} for the bilinear form $c^*A^{-1}b$ and compares it to the existing approaches. We have linked the presented approximations to the matching moment properties of the Krylov subspace methods. While the maximal number of moments matched at step n of the Hermitian and non-Hermitian Lanczos algorithm and BiCG is $2n$, the Arnoldi algorithm matches at step n only n moments. Matching $2n$ moments using oblique projections, however, does not necessarily mean an advantage over using the Arnoldi algorithm with orthogonal projections (at the price of computing long recurrences) and matching n moments only. In practice, the cost evaluation must take into account specifics of the given application problem which determine, e.g., the cost of the matrix-vector products in relation to the cost of the iteration updates. Therefore the choice of an optimal approach (including a choice of stopping criteria) is application-dependent. Nevertheless, the newly proposed approximation ξ_n^{B} (see (3.14)) is, in our opinion, highly competitive and it can be considered a good reference standard for any other possible approach. The approximation error can be estimated using techniques based on computing d additional iterations analogously to CG [52, Section 4]; see also [34, Section 5.3] and, in the context of constructing stopping criteria in numerical solution of PDEs, e.g., [1, 31]. The approximation ξ_n^{B} clearly outperforms the mathematically equivalent scattering amplitude approximation c^*x_n. Scattering amplitude approximations computed via short recurrences rely upon preserving global biorthogonality among the computed vectors. Their convergence is delayed due to rounding errors much more than convergence of the approximation ξ_n^{B}, and therefore they should not be used in practical computations.

Acknowledgement. The authors are indebted to Paul Van Dooren and Volker Mehrmann for their useful comments and to Gerard Meurant for his valuable suggestions. The presentation also benefits from the comments on the previous version of the manuscript given by Karen Willcox, and by the very valuable comments of three anonymous referees. The authors express their thanks to Margaret Wright for extraordinary detailed comments and advice.

REFERENCES

[1] M. Arioli, *A stopping criterion for the conjugate gradient algorithms in a finite element method framework*, Numer. Math., 97 (2004), pp. 1–24.

[2] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, Appl. Numer. Math., 43 (2002), pp. 9–44. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[3] C. Brezinski, *Projection Methods for Systems of Equations*, vol. 7 of Studies in Computational Mathematics, North-Holland Publishing Co., Amsterdam, 1997.

[4] A. Bultheel and M. Van Barel, *Linear algebra, rational approximation and orthogonal polynomials*, vol. 6 of Studies in Computational Mathematics, North-Holland Publishing Co., Amsterdam, 1997.

[5] D. Calvetti, S.-M. Kim, and L. Reichel, *Quadrature rules based on the Arnoldi process*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 765–781 (electronic).

[6] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, *Computable error bounds and estimates for the conjugate gradient method*, Numer. Algorithms, 25 (2000), pp. 75–88. Mathematical journey through analysis, matrix theory and scientific computation (Kent, OH, 1999).

[7] P. Chebyshev, *Sur les fractions continues*, (1855). Reprinted in Oeuvres I, 11 (Chelsea, New York, 1962), pp. 204–229.

[8] G. Dahlquist, S. C. Eisenstat, and G. H. Golub, *Bounds for the error of linear systems of equations using the theory of moments*, J. Math. Anal. Appl., 37 (1972), pp. 151–166.

[9] G. Dahlquist, G. H. Golub, and S. G. Nash, *Bounds for the error in linear systems*, in Semi-infinite programming (Proc. Workshop, Bad Honnef, 1978), vol. 15 of Lecture Notes in Control and Information Sci., Springer, Berlin, 1979, pp. 154–172.

[10] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, pp. 73–89. Lecture Notes in Math., Vol. 506.

[11] R. W. Freund, *Model reduction methods based on Krylov subspaces*, Acta Numer., 12 (2003), pp. 267–319.

[12] R. W. Freund and M. Hochbruck, *Gauss quadratures associated with the Arnoldi process and the Lanczos algorithm*, in Linear algebra for large scale and real-time applications, M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., vol. 232 of NATO Advanced Science Institutes Series E: Applied Sciences, Dordrecht, 1993, Kluwer Academic Publishers Group, pp. 377–380.

[13] K. Gallivan, E. Grimme, and P. Van Dooren, *Asymptotic waveform evaluation via a Lanczos method*, Appl. Math. Lett., 7 (1994), pp. 75–80.

[14] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[15] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[16] ———, *Milestones in Matrix Computation: Selected Works of Gene H. Golub, with Commentaries*, Oxford Science Publications, Oxford University Press, Oxford, 2007. Edited by Raymond H. Chan, Chen Greif and Dianne P. O'Leary.

[17] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical analysis 1993 (Dundee, 1993), vol. 303 of Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, 1994, pp. 105–156.

[18] G. H. Golub and G. Meurant, *Matrices, moments and quadrature. II. How to compute the norm of the error in iterative methods*, BIT, 37 (1997), pp. 687–705. Direct methods, linear algebra in optimization, iterative methods (Toulouse, 1995/1996).

[19] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature With Applications*, Princeton University Press, USA, 2010.

[20] G. H. Golub, M. Stoll, and A. Wathen, *Approximation of the scattering amplitude and linear systems*, Electron. Trans. Numer. Anal., 31 (2008), pp. 178–203.

[21] G. H. Golub and Z. Strakoš, *Estimates in quadratic formulas*, Numer. Algorithms, 8 (1994), pp. 241–268.

[22] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[23] R. G. Gordon, *Error bounds in equilibrium statistical mechanics*, J. Math. Physics, 9 (1968), pp. 655–663.

[24] W. B. Gragg, *Matrix interpretations and applications of the continued fraction algorithm*, in Proceedings of the International Conference on Padé Approximants, Continued Fractions and Related Topics (Univ. Colorado, Boulder, Colo., 1972; dedicated to the memory of H. S. Wall), vol. 4, 1974, pp. 213–225.

[25] A. Greenbaum, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.

[26] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[27] M. H. Gutknecht, *Variants of BICGSTAB for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033.

[28] J. J. Hench and Z. Strakoš, *The RCWA method—a case study with open questions and perspectives of algebraic computations*, Electron. Trans. Numer. Anal., 31 (2008), pp. 331–357.

[29] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436 (1953).

[30] K. Hoffman and R. Kunze, *Linear algebra*, Second edition, Prentice-Hall Inc., Englewood Cliffs, N.J., 1971.

[31] P. Jiránek, Z. Strakoš, and M. Vohralík, *A posteriori error estimates including algebraic error and stopping criteria for iterative solvers*, SIAM J. Sci. Comput., 32 (2010), pp. 1567–1590.

[32] P. Jordan and J. Von Neumann, *On inner products in linear, metric spaces*, Ann. of Math. (2), 36 (1935), pp. 719–723.

[33] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[34] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.

[35] M. Neviere and E. Popov, *Light Propagation in Periodic Media*, Marcel Dekker Ltd., New York, 2002.

[36] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284 (electronic).

[37] C. C. Paige and M. A. Saunders, *Algorithm 583: LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 195–201.

[38] ———, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[39] L. Reichel and Q. Ye, *A generalized LSQR algorithm*, Numer. Linear Algebra Appl., 15 (2008), pp. 643–660.

[40] W. P. Reinhardt, *l^2 discretization of atomic and molecular electronic continua: Moment, quadrature and j-matrix techniques*, Comp. Phys. Comm., 17 (1979), pp. 1–21.

[41] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed., 2003.

[42] M. A. Saunders, H. D. Simon, and E. L. Yip, *Two conjugate-gradient-type methods for unsymmetric linear equations*, SIAM J. Numer. Anal., 25 (1988), pp. 927–940.

[43] P. E. Saylor and D. C. Smolarski, *Addendum to: "Why Gaussian quadrature in the complex plane?" [Numer. Algorithms* **26** *(2001), no. 3, 251–280]*, Numer. Algorithms, 27 (2001), pp. 215–217.

[44] ———, *Why Gaussian quadrature in the complex plane?*, Numer. Algorithms, 26 (2001), pp. 251–280.

[45] L. Schlessinger and C. Schwartz, *Analyticity as a useful computational tool*, Phys. Rev. Lett., 16 (1966), pp. 1173–1174.

[46] G. L. G. Sleijpen and H. A. van der Vorst, *Maintaining convergence properties of BiCGstab methods in finite precision arithmetic*, Numer. Algorithms, 10 (1995), pp. 203–223.

[47] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema, BiCGstab(l) *and other hybrid Bi-CG methods*, Numer. Algorithms, 7 (1994), pp. 75–109.

[48] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

[49] T. J. Stieltjes, *Recherches sur les fractions continues*, Ann. Fac. Sci. Toulouse Sci. Math. Sci. Phys., 8 (1894), pp. J. 1–122. Reprinted in Oeuvres II (P. Noordhoff, Groningen, 1918), pp. 402–566. English translation *Investigations on continued fractions* in Thomas Jan Stieltjes, Collected Papers, Vol. II (Springer-Verlag, Berlin, 1993), pp. 609–745.

[50] M. Stoll, *Solving linear systems using the adjoint*, PhD thesis, University of Oxford, Oxford, UK, 2009.

[51] Z. Strakoš, *Model reduction using the Vorobyev moment problem*, Numer. Algorithms, 51 (2009), pp. 363–379.

[52] Z. Strakoš and P. Tichý, *On error estimation in the conjugate gradient method and why it works in finite precision computations*, Electron. Trans. Numer. Anal., 13 (2002), pp. 56–80.

[53] ———, *Error estimation in preconditioned conjugate gradients*, BIT, 45 (2005), pp. 789–817.

[54] H. A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[55] Y. V. Vorobyev, *Methods of moments in applied mathematics*, Translated from the Russian by Bernard Seckler, Gordon and Breach Science Publishers, New York, 1965.

[56] K. F. Warnick and W. C. Chew, *Numerical simulation methods for rough surface scattering*, Waves Random Media, 11 (2001), pp. R1–R30.

[57] N. Young, *An introduction to Hilbert space*, Cambridge Mathematical Textbooks, Cambridge University Press, Cambridge, 1988.