

## PRECONDITIONED CHEBYSHEV BICG METHOD FOR PARAMETERIZED LINEAR SYSTEMS\*

SIOBHÁN CORRENTY<sup>†</sup>, ELIAS JARLEBRING<sup>†</sup>, AND DANIEL B. SZYLD<sup>‡</sup>

**Abstract.** We consider the problem of approximating the solution to  $A(\mu)x(\mu) = b$  for many different values of the parameter  $\mu$ . Here,  $A(\mu)$  is large, sparse, and nonsingular with a nonlinear dependence on  $\mu$ . Our method is based on a companion linearization derived from an accurate Chebyshev interpolation of  $A(\mu)$  on the interval  $[-a, a]$ ,  $a \in \mathbb{R}_+$ , inspired by Effenberger and Kressner [BIT, 52 (2012), pp. 933–951]. The solution to the linearization is approximated in a preconditioned BiCG setting for shifted systems, as proposed in Ahmad et al. [SIAM J. Matrix Anal. Appl., 38 (2017), pp. 401–424], where the Krylov basis matrix is formed once. This process leads to a short-term recurrence method, where one execution of the algorithm produces the approximation of  $x(\mu)$  for many different values of the parameter  $\mu \in [-a, a]$  simultaneously. In particular, this work proposes one algorithm which applies a shift-and-invert preconditioner exactly as well as an algorithm which applies the preconditioner inexactly based on the work by Vogel [Appl. Math. Comput., 188 (2007), pp. 226–233]. The competitiveness of the algorithms is illustrated with large-scale problems arising from a finite element discretization of a Helmholtz equation with a parameterized material coefficient. The software used in the simulations is publicly available online, and thus all our experiments are reproducible.

**Key words.** parameterized linear systems, short-term recurrence methods, Chebyshev interpolation, inexact preconditioning, Krylov subspace methods, companion linearization, shifted linear systems, parameterized Helmholtz equation, time-delay systems

**AMS subject classifications.** 15A06, 65F08, 65F10, 65F50, 65N22, 65P99

**1. Introduction.** In this work, we propose a new approach for computing an efficient approximation of the solution to the parameterized linear system given by

$$(1.1) \quad A(\mu)x(\mu) = b$$

for many values of the parameter  $\mu$  simultaneously. Here,  $A(\mu) \in \mathbb{R}^{n \times n}$  is assumed nonsingular, nonlinear in  $\mu \in [-a, a]$ ,  $a \in \mathbb{R}_+$ , and  $b \in \mathbb{R}^n$ . Specifically, our method finds accurate approximations of  $x(\mu)$  for  $\mu$  in a specified region of the interval  $[-a, a]$ . Parameterized linear systems have been studied previously, for example in [24, 29], where these systems arise in the context of Tikhonov regularization for ill-posed problems, as well as in [31], where the solution was approximated by a tensor of low rank, and in [26], where the right-hand side vector also depended on the parameter.

We assume further that  $A(\mu)$  in (1.1) is large and sparse and can be expressed as the sum of products of matrices and functions, i.e.,

$$(1.2) \quad A(\mu) = C_1 f_1(\mu) + \dots + C_{n_f} f_{n_f}(\mu),$$

where  $n_f \ll n$ . Our method requires an approximation of  $A(\mu)$  via a Chebyshev interpolation. In this way, we compute  $P(\mu) \approx A(\mu)$ , where

$$(1.3) \quad P(\mu) = P_0 \tau_0(\mu) + \dots + P_d \tau_d(\mu)$$

with  $P_\ell \in \mathbb{R}^{n \times n}$  and  $\tau_\ell(\mu)$  the recursively defined Chebyshev polynomials on the interval  $[-a, a]$ . The matrix  $P(\mu)$  is assumed nonsingular throughout this work, and we assume that  $d$ ,

---

\*Received December 8, 2022. Accepted October 2, 2023. Published online on December 7, 2023. Recommended by A. Frommer.

<sup>†</sup>Department of Mathematics, Royal Institute of Technology (KTH), SeRC Swedish e-Science Research Center, Lindstedtsvägen 25, Stockholm, Sweden (`{correnty, eliasj}@kth.se`).

<sup>‡</sup>Department of Mathematics, Temple University, 1805 N. Broad Street, Philadelphia, PA 19122-6094 (`szyld@math.temple.edu`).

the truncation parameter in the Chebyshev approximation (1.3), is large enough such that the error introduced by the Chebyshev interpolation is small.

We present a preconditioned short-term recurrence Krylov subspace method to approximate the solution to

$$(1.4) \quad P(\mu)\tilde{x}(\mu) = b,$$

where  $\tilde{x}(\mu) \approx x(\mu)$ . In practice, our method approximates the solution to a companion linearization of the form

$$(1.5) \quad (K - \mu M)u(\mu) = \tilde{b},$$

with coefficient matrices  $K, M \in \mathbb{R}^{dn \times dn}$  and a constant vector  $\tilde{b} \in \mathbb{R}^{dn}$ . The solution to (1.4) and the companion linearization are equivalent in a certain sense, as shown in Section 2. Here, the bases for two Krylov subspaces are generated via a Lanczos biorthogonalization procedure as in the biconjugate gradient (BiCG) method [19, 33]. Specifically, the method incorporates shift-and-invert preconditioners of the form  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$ , for  $\sigma \in (-a, a)$ , to accelerate convergence for solutions corresponding to values of  $\mu$  close to the chosen target  $\sigma$ . Additionally, the use of such preconditioners leads to a shifted linear system, and shift- and scaling-invariance properties of Krylov subspaces are exploited. In this way, we reuse one Krylov subspace basis matrix to determine approximations of (1.1) for many different values of  $\mu$ .

We propose two variants of our method. The first variant considers an exact application of the preconditioner in a BiCG setting for shifted systems, inspired by [1]. The second variant incorporates an approximate application of the preconditioners in an inexact BiCG setting for shifted systems, based on the prior works [41, 51]. We prove a bound for the residual of the second variant, and the bound is efficient to compute.

The first variant is appropriate only when an LU decomposition of a matrix of dimension  $n \times n$  is feasible, whereas the inexact variant has the potential to solve a wider variety of large-scale problems. While the second variant is based on the first variant, the first variant may be useful in itself in cases where the exact LU decomposition is computable, as discussed in Section 8. Numerical simulations from time-delay systems and a parameterized Helmholtz equation show the performance of our proposed algorithms. Note that the BiCG method with right preconditioning is used throughout this work. A left preconditioned setting would have been possible with the first variant but not the second, as inexact preconditioning requires right preconditioning.

This paper is organized as follows. In Section 2 we present the Chebyshev linearization, and, in Section 3, we describe how an equivalent shifted linear system is obtained. This section also provides preliminaries for the method BiCG for shifted systems. Section 4 shows how the preconditioners are implemented in an efficient manner, and, in Section 5, we derive the preconditioned Chebyshev BiCG method for parameterized linear systems. Section 6 provides a numerical example from a discretized Helmholtz equation, and Section 7 utilizes our method for computing the transfer function from a time-delay system. In Section 8 we derive the inexact variant of the method. Furthermore, we prove a bound for the residual produced by iterates of the inexact method. This section also highlights the performance of our approach for solving large-scale parameterized systems effectively. Conclusions are given in Section 9.

**2. Linearization.** We consider a technique called companion linearization, used in prior works on polynomial eigenvalue problems [34], as well as in works for parameterized linear systems; see, e.g., [26]. Our proposed linearization is of the form (1.5). The solution to (1.5) and  $\tilde{x}(\mu)$  in (1.4) with  $P(\mu)$  (1.3) are equivalent in a certain sense, described as follows.

On the interval  $[-a, a]$ , the Chebyshev polynomials are defined by the well-known three-term recurrence

$$(2.1) \quad \begin{aligned} \tau_0(\mu) &:= 1, \\ \tau_1(\mu) &:= \frac{1}{a}\mu, \\ \tau_{\ell+1}(\mu) &:= \frac{2}{a}\mu\tau_\ell(\mu) - \tau_{\ell-1}(\mu), \end{aligned}$$

and the interpolation condition

$$(2.2) \quad P(\mu_\ell^*) = A(\mu_\ell^*), \quad \ell = 1, \dots, d,$$

holds, where  $\mu_\ell^*$  are the  $d$  roots of the degree  $d$  polynomial  $\tau_d$ . The companion linearization, adapted from the work [15] and utilized here, is given by

$$(2.3) \quad \left( \begin{bmatrix} 0 & I & & & & \\ I & 0 & I & & & \\ & I & 0 & I & & \\ & & & \ddots & & \\ & & & & I & 0 & I \\ P_0 & P_1 & \cdots & P_{d-3} & (-P_d + P_{d-2}) & P_{d-1} \end{bmatrix} - \frac{\mu}{a} \begin{bmatrix} I & & & & & \\ & 2I & & & & \\ & & 2I & & & \\ & & & \ddots & & \\ & & & & 2I & \\ & & & & & -2P_d \end{bmatrix} \right) \begin{bmatrix} u_0(\mu) \\ u_1(\mu) \\ u_2(\mu) \\ \vdots \\ u_{d-2}(\mu) \\ u_{d-1}(\mu) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix}.$$

Specifically,  $u_\ell(\mu) := \tau_\ell(\mu)\tilde{x}(\mu) \in \mathbb{R}^n$ , for  $\ell = 0, \dots, d-1$ , where  $\tilde{x}(\mu)$  is the unique solution in (1.4) and

$$(2.4) \quad \tilde{b} := [0 \ 0 \ 0 \ \cdots \ 0 \ b]^T \in \mathbb{R}^{dn}.$$

In this way, (2.3) is of the form described in (1.5), where we have made the substitution

$$(2.5) \quad P_d u_d(\mu) = P_d \left( \frac{2}{a} \mu u_{d-1}(\mu) - u_{d-2}(\mu) \right)$$

in the last block row, using the recurrence relation (2.1). Note that since  $A(\mu)$  is as in (1.2), the coefficient matrices  $P_\ell$  used in the linearization can be computed efficiently using a discrete cosine transform [47] of the scalar functions  $f_i$ ,  $i = 1, \dots, n_f$ . Specifically, we compute  $P_\ell = C_1 p_\ell^1 + \dots + C_{n_f} p_\ell^{n_f}$ , for  $\ell = 0, \dots, d$ , where the  $i$ th function in (1.2) is approximated by a degree- $d$  polynomial, i.e.,  $f_i(\mu) \approx p_0^i \tau_0(\mu) + \dots + p_d^i \tau_d(\mu)$ . The interpolations are performed using `Chebfun` [13] in Matlab, and the coefficients  $p_\ell^i$  smaller in magnitude than  $10^{-16}$  are set to zero.

Thus, we can consider highly accurate interpolations of  $A(\mu)$  without substantial computation. The following theorem describes the equivalence of the solutions of the approximation in (1.4) and the system (2.3), where the parameter  $\mu$  appears only linearly.

**THEOREM 2.1.** *Let  $A(\mu)$  be as in (1.1) with parameter  $\mu \in \mathbb{R}$  and  $P(\mu)$  as in (1.3) such that (1.4) has a unique solution  $\tilde{x}(\mu)$ . Then the linear system (2.3) has a unique solution of the form  $u(\mu) = [u_0(\mu), \dots, u_{d-1}(\mu)]^T$  with*

$$(2.6) \quad u_\ell(\mu) := \tau_\ell(\mu)\tilde{x}(\mu), \quad \ell = 0, \dots, d-1.$$

*Proof.* Consider the first  $d-1$  block rows of (2.3). An induction using the three-term recurrence of the Chebyshev polynomials on the interval  $[-a, a]$  as in (2.1) implies

$u_\ell(\mu) = \tau_\ell(\mu)\tilde{z}$ ,  $\ell = 0, \dots, d-1$ , for  $\tilde{z} \in \mathbb{R}^n$ . Inserting this parameterized solution into the last block row of (2.3) gives

$$\begin{aligned} \left( P_0\tau_0(\mu) + \dots + P_{d-3}\tau_{d-3}(\mu) + (-P_d + P_{d-2})\tau_{d-2}(\mu) + \frac{2}{a}\mu P_d\tau_{d-1}(\mu) \right) \tilde{z} &= P(\mu)\tilde{z} \\ &= b, \end{aligned}$$

due to (1.3). Note the substitution  $P_d\tau_d(\mu)\tilde{z} = P_d((2\mu/a)\tau_{d-1}(\mu) - \tau_{d-2}(\mu))\tilde{z}$ , analogous to the relation (2.5). Thus,  $\tilde{z} = \tilde{x}(\mu)$ , and the solution (2.6) to the linear system (2.3) is unique since  $\tilde{x}(\mu)$  is unique.  $\square$

REMARK 2.2 (Interpolation coefficients). The linearization (2.3) can be generated from evaluations of  $f_i$  as in (1.2) at the Chebyshev nodes  $\mu_\ell^*$  described in (2.2). In this way, we do not explicitly require the functions  $f_i$  in order to carry out the linearization.

**3. Preliminaries for the Chebyshev BiCG method with exact preconditioning.** We consider a preconditioned short-term recurrence method where two Krylov subspaces are generated via a Lanczos biorthogonalization procedure as in the method BiCG. The biorthogonalization process requires the action of the system matrix, as well as its adjoint. Additionally, our setting uses the shift-and-invert preconditioner  $(K - \sigma M)^{-1}$ , for  $\sigma \in (-a, a)$ , and the corresponding adjoint preconditioner  $(K - \sigma M)^{-T}$ . Some preliminaries are described here as preparation, and the strategy for the application of the preconditioners is shown in Section 4. The proposed method, Algorithm 1, is derived and presented in Section 5, and numerical simulations follow in Section 6 and Section 7.

Our right preconditioned system is of the form

$$\begin{aligned} (3.1a) \quad & (K - \mu M)(K - \sigma M)^{-1}\tilde{u}(\mu) = \tilde{b} \\ (3.1b) \quad & \iff (K - \mu M + \sigma M - \sigma M)(K - \sigma M)^{-1}\tilde{u}(\mu) = \tilde{b} \\ (3.1c) \quad & \iff (I + (-\mu + \sigma)M(K - \sigma M)^{-1})\tilde{u}(\mu) = \tilde{b}, \end{aligned}$$

where  $\tilde{u}(\mu) = (K - \sigma M)u(\mu)$  and

$$(3.2) \quad (K - \mu M)(K - \sigma M)^{-1} \approx I,$$

for  $\mu \approx \sigma$ . The parameter  $\sigma$  in the preconditioners is chosen based on the values of  $\mu$  we are interested in and, thus, can be seen as a target parameter. Specifically, we chose  $\sigma$  such that we approximate (1.1) for many different values of  $\mu$  in a neighborhood of  $\sigma$ .

Krylov methods have been developed to approximate the solution to shifted linear systems of the same form as (3.1c) in many prior works. See, for example, [6, 20, 23], as well as [5], where multiple shift-and-invert preconditioners were incorporated to build a rich search space in a GMRES framework, and [40, 46], where Krylov recycling techniques were utilized to solve shifted linear systems. Additionally, in [7], preconditioned Krylov subspace methods were considered for the time-harmonic elastic wave equation. Specifically, the solution to an equivalent linearized system was approximated, and shift-and-invert preconditioners with complex linear shifts were considered.

The formulation (3.1c) allows us to take advantage of the shift- and scaling-invariance properties of Krylov subspaces. More concretely, at the  $j$ th iteration of our proposed BiCG method Algorithm 1, we seek approximations of  $\tilde{u}(\mu)$  in (3.1) from the Krylov subspace defined by

$$(3.3) \quad \mathcal{K}_j := \text{span} \left\{ \tilde{b}, M(K - \sigma M)^{-1}\tilde{b}, \dots, (M(K - \sigma M)^{-1})^{j-1}\tilde{b} \right\}.$$

Here,  $\mathcal{K}_j = \tilde{\mathcal{K}}_j$ , where  $\mathcal{K}_j = \mathcal{K}_j(M(K - \sigma M)^{-1}, \tilde{b})$ , and  $\tilde{\mathcal{K}}_j$  is the Krylov subspace of dimension  $j$  generated from the system matrix in (3.1c) and the vector  $\tilde{b}$ . In this way, approximating the solution to (3.1) for many values of  $\mu$  in a BiCG setting requires just one basis matrix for  $\mathcal{K}_j$  and, analogously, one basis matrix for the Krylov subspace  $\mathcal{L}_j$ , defined by

$$(3.4) \quad \mathcal{L}_j := \text{span} \left\{ \tilde{c}, (M(K - \sigma M)^{-1})^T \tilde{c}, \dots, \left( (M(K - \sigma M)^{-1})^T \right)^{j-1} \tilde{c} \right\},$$

for  $\tilde{b}^T \tilde{c} \neq 0$ ,  $\tilde{c} \in \mathbb{R}^{dn}$ . Equivalently, we use one basis matrix for (3.3) to approximate solutions to (1.1) for many values of  $\mu$ , as described in Section 2. To our knowledge, this is the first time a Chebyshev linearization has been combined with a Krylov subspace method to obtain approximate solutions to parameterized linear systems.

In particular, after  $j$  iterations, the Lanczos biorthogonalization generates matrices  $V_j, W_j \in \mathbb{R}^{dn \times j}$ ,  $T_j \in \mathbb{R}^{j \times j}$ , and  $\underline{T}_j, \bar{T}_j^T \in \mathbb{R}^{(j+1) \times j}$  such that the relations

$$(3.5) \quad \begin{aligned} M(K - \sigma M)^{-1} V_j &= V_j T_j + \beta_j v_{j+1} e_j^T = V_{j+1} \underline{T}_j, \\ (M(K - \sigma M)^{-1})^T W_j &= W_j T_j^T + \gamma_j w_{j+1} e_j^T = W_{j+1} \bar{T}_j^T \end{aligned}$$

hold, where the columns of  $V_j$  span the subspace (3.3). Analogously, the columns of  $W_j$  span the subspace (3.4), and the biorthogonalization procedure gives the relation

$$(3.6) \quad W_j^T V_j = I_j,$$

where  $I_j \in \mathbb{R}^{j \times j}$  is the identity matrix of dimension  $j \times j$  and  $e_j$  is the  $j$ th column of  $I_j$ . Here, the square matrix  $T_j$  has the form

$$(3.7) \quad T_j := \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \gamma_{j-1} & \\ & & \beta_{j-1} & \alpha_j & \end{bmatrix} \in \mathbb{R}^{j \times j},$$

and the tridiagonal Hessenberg matrices  $\underline{T}_j$  and  $\bar{T}_j^T$  are given by

$$(3.8) \quad \underline{T}_j := \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \gamma_{j-1} & \\ & & \beta_{j-1} & \alpha_j & \\ & & & \beta_j & \end{bmatrix}, \quad \bar{T}_j^T := \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \beta_{j-1} & \\ & & \gamma_{j-1} & \alpha_j & \\ & & & \gamma_j & \end{bmatrix}.$$

Note that only the  $j \times j$  principal submatrices of  $\underline{T}_j$  and  $\bar{T}_j^T$  are the transpose of each other. An advantage of the Lanczos biorthogonalization procedure is the so-called short-term recurrence of the Krylov basis vectors, i.e., that the matrices  $\underline{T}_j$  and  $\bar{T}_j^T$  in (3.5) are tridiagonal. In this way, the basis vectors are computed recursively at each iteration of the algorithm. This choice of method is different from those in the previous works [11, 28] for parameterized systems based on companion linearization. The resulting linear systems in these works were solved in a GMRES setting, and they used a Gram-Schmidt orthogonalization process, i.e., a long-term recurrence.

The residual of the  $i$ th iterate generated from a BiCG procedure applied to the shifted preconditioned linear system (3.1), denoted  $\tilde{r}_i$ , is orthogonal to the subspace  $\mathcal{L}_i$  as in (3.4),

and  $\tilde{r}_i$  is an element of the Krylov subspace  $\mathcal{K}_{i+1}$  as in (3.3). As established in [22],  $\tilde{r}_i$  is colinear to the residual vector of the  $i$ th iterate resulting from the BiCG method applied to the so-called *seed* system given by

$$(3.9) \quad M(K - \sigma M)^{-1} u^{sd} = \tilde{b},$$

denoted  $r_i$ . This is described entirely in Section 5 and used to derive Algorithm 1. Analogously, we denote the adjoint seed system by

$$(3.10) \quad (K - \sigma M)^{-T} M^T w^{sd} = \tilde{c}.$$

Approximates of (1.1) corresponding to each  $\mu$  in an interval can be obtained with little extra computation using the *exact algorithm* described in Section 5, and this method is based on the colinearity of  $r_i$  and  $\tilde{r}_i$ . Here, the accuracy of the approximations depends on the dimension  $j$  of the Krylov subspace from which these approximates originate. Additionally, the basis matrix for the Krylov subspace does not need to be stored if the values of  $\mu$  are determined in advance. Our method offers a computational advantage over solving each parameterized system individually when we are interested in the solution to many different parameterized systems.

**4. An efficient application of a shift-and-invert preconditioner.** Utilizing preconditioning in the context of Krylov subspace methods can lead to methods that are more efficient overall. This strategy is only suitable when the action of the preconditioner is cheap to apply. Our proposed method incorporates well-established shift-and-invert preconditioning. As we consider a BiCG setting, we require an efficient application of the preconditioner and its adjoint, i.e., we consider applying the action of  $(K - \sigma M)^{-1}$  as well as the action of  $(K - \sigma M)^{-T}$ ,  $\sigma \in (-a, a)$ .

These preconditioners are effective for solving shifted systems of the form (1.5) when the target parameter  $\sigma$  is chosen close to the values of  $\mu$  of interest. This is due to the relation (3.2). The BiCG method tends to experience fast converge when applied to linear systems of this form. In the following, we show how the structure can be exploited such that the action of these preconditioners can be computed efficiently.

Consider approximating the solution to the system (1.5), incorporating right preconditioning with the shift-and-invert preconditioner  $(K - \sigma M)^{-1}$ . The resulting linear system is of the form in (3.1), expressed equivalently as

$$(4.1) \quad (-\mu + \sigma) \left( \frac{1}{(-\mu + \sigma)} I + M(K - \sigma M)^{-1} \right) \tilde{u}(\mu) = \tilde{b}.$$

Note that the formulation in (4.1) was chosen in order to match the notation in the Exact Algorithm 1, presented in Section 5. Specifically, the scalar  $1/(-\mu + \sigma)$  is the coefficient of the identity matrix. This work considers a BiCG setting, and the linear system (4.1) incorporates a shift with a scalar multiple of the identity matrix. Equivalent to (3.5), the shifted relations

$$(4.2) \quad \begin{aligned} V_j + (-\mu + \sigma)M(K - \sigma M)^{-1}V_j &= V_{j+1}(\underline{I}_j + (-\mu + \sigma)\underline{T}_j), \\ W_j + (-\mu + \sigma)(M(K - \sigma M)^{-1})^T W_j &= W_{j+1}(\underline{I}_j + (-\mu + \sigma)\tilde{T}_j^T), \end{aligned}$$

hold, where the matrix  $\underline{I}_j \in \mathbb{R}^{(j+1) \times j}$  is an identity matrix of dimension  $j \times j$  with an extra row of zeros, i.e.,  $V_{j+1}\underline{I}_j = V_j$ . Note that the matrix  $V_j$  is also a basis for the Krylov subspace generated by the matrix in (4.1) and the vector  $\tilde{b}$  by the shift- and scaling-invariance properties of Krylov subspaces described in Section 3.

Approximating the solution to (4.1) in our setting requires a basis for the Krylov subspaces (3.3) and (3.4), obtained from a multiplication with the matrices in (3.9) and (3.10) at each iteration of the Lanczos biorthogonalization. The action of the matrices  $M$  and  $M^T$  require one matrix-vector multiplication of size  $n \times n$  for each product. Additionally, an efficient application of the preconditioner  $(K - \sigma M)^{-1}$  can be performed via a block LU decomposition of the matrix  $(K - \sigma M)\Pi$ , where  $\Pi$  is a permutation matrix, as described in [3, 30]. This process leads directly to an efficient application of  $(K - \sigma M)^{-T}$ . The procedure is described as follows.

Let a permutation of the block columns of the matrix  $(K - \sigma M)$  in (2.3) be given by

$$(4.3) \quad (K - \sigma M)\Pi = \begin{bmatrix} I & & & & & -\frac{\sigma}{a}I \\ -\frac{2\sigma}{a}I & I & & & & I \\ I & -\frac{2\sigma}{a}I & I & & & \\ & & \ddots & & & \\ & & & I & & \\ P_1 & \cdots & P_{d-3} & (-P_d + \frac{2\sigma}{a}I) & (P_{d-1} + \frac{2\sigma}{a}P_d) & P_0 \end{bmatrix},$$

where  $\Pi := \begin{bmatrix} I_n \\ I_{(d-1)n} \end{bmatrix} \in \mathbb{R}^{dn \times dn}$  is an orthogonal matrix, and let  $L_\sigma U_\sigma = (K - \sigma M)\Pi$  be a block LU decomposition, with

$$L_\sigma := \begin{bmatrix} I & & & & & \\ -\frac{2\sigma}{a}I & I & & & & \\ I & -\frac{2\sigma}{a}I & I & & & \\ & & \ddots & & & \\ & & & I & & \\ P_1 & \cdots & P_{d-3} & (-P_d + \frac{2\sigma}{a}I) & (P_{d-1} + \frac{2\sigma}{a}P_d) & P(\sigma) \end{bmatrix}$$

and

$$U_\sigma := \begin{bmatrix} I & & & & -\tau_1(\sigma)I \\ & I & & & -\tau_2(\sigma)I \\ & & I & & -\tau_3(\sigma)I \\ & & & \ddots & \vdots \\ & & & & I & -\tau_{d-1}(\sigma)I \\ & & & & & I \end{bmatrix}.$$

The matrix  $U_\sigma^{-1}$  is identical to  $U_\sigma$ , except for a sign change in the first  $d - 1$  blocks in the last block column. Applying  $L_\sigma^{-1}$  to a vector amounts to recursively calculating the first  $d - 1$  block elements, in addition to one linear solve with the system matrix  $P(\sigma) \in \mathbb{R}^{n \times n}$  when computing the last block row. This process of applying  $L_\sigma^{-1}$  is equivalent to applying Gaussian elimination on a lower block triangular system.

Thus, the action of the preconditioner  $(K - \sigma M)^{-1}$  applied to a vector  $y \in \mathbb{R}^{dn}$  is given by

$$(4.4) \quad (K - \sigma M)^{-1}y = \Pi U_\sigma^{-1} L_\sigma^{-1} y,$$

i.e., the preconditioner in (3.1) can be applied without computing or storing the large matrices  $K$  and  $M$  or  $L_\sigma^{-1}$  and  $U_\sigma^{-1}$ . Note that the action of  $P(\sigma)^{-1}$  in the application of  $L_\sigma^{-1}$  can



be done, for example, via one LU decomposition of  $P(\sigma)$  performed before the start of the algorithm, as it is considered in Algorithm 1 in Section 5. Alternatively, we can apply the action of  $P(\sigma)^{-1}$  approximately via an iterative method, as considered in Algorithm 2 and presented in Section 8.

The action of the adjoint preconditioner  $(K - \sigma M)^{-T}$  can be applied in an analogous way, i.e.,

$$(4.5) \quad (K - \sigma M)^{-T} y = L_\sigma^{-T} U_\sigma^{-T} \Pi^T y,$$

which does not require the storage of the matrices  $K$  and  $M$  or the triangular matrices  $L_\sigma^{-T}$  and  $U_\sigma^{-T}$ . Additionally, the LU factorization of  $P(\sigma)$  required in Algorithm 1 can be reused in the application of the adjoint preconditioner. Thus, the shift-and-invert preconditioner  $(K - \sigma M)^{-1}$  is suitable in a BiCG setting.

**5. Derivation of the Preconditioned Chebyshev BiCG method for parameterized linear systems.** In [22] a BiCG algorithm was derived in order to solve a seed system of the form (3.9), as well as a shifted system of the form (4.1), without requiring additional matrix-vector products for the iterates of the shifted system. Additionally, in [1], a multishift BiCG algorithm with polynomial preconditioning was proposed to approximate the solutions for a family of shifted systems simultaneously. Such approaches are far less costly than solving each system of interest individually in a BiCG setting, without degrading convergence in general.

We summarize below a derivation of the original algorithm from [22], adapted to our shifted system stemming from a companion linearization formed from a Chebyshev interpolation as in (1.5). The method is based on the observation that the residual vectors of the seed system can be used to generate a basis for the Krylov subspace (3.3). The description which follows serves to clarify the steps in Algorithm 1, which we will refer to as the Exact Algorithm 1 or the *exact algorithm*.

Lines 1–8 of the Exact Algorithm 1 correspond to a variant of the standard BiCG method. This implementation is based on a Lanczos biorthogonalization, applied to the seed system (3.9). The coupled two-term recurrence formulation used here is based on an implicitly formed LU decomposition of the tridiagonal matrix  $T_j$  as in (3.5); see [42] for a detailed description. The search direction vectors in this formulation are updated as

$$(5.1) \quad v_{i+1}^* = r_i - \beta_i v_i^*, \quad w_{i+1}^* = s_i - \bar{\beta}_i w_i^*,$$

where the residual vectors

$$(5.2) \quad r_i := \tilde{b} - M(K - \sigma M)^{-1} u_i^{sd}, \quad s_i := \tilde{c} - (K - \sigma M)^{-T} M^T w_i^{sd}$$

are updated recursively (see line 7 of the algorithm), and  $u_i^{sd}, w_i^{sd} \in \mathbb{R}^{dn}$  are approximations of the seed system and adjoint seed systems given in (3.9) and (3.10), respectively, with  $u_0^{sd} := 0, w_0^{sd} := 0$ . The search directions have the property  $(w_i^*)^T M(K - \sigma M)^{-1} v_j^* = 0$ , for  $i \neq j$ , and the residual vectors are orthogonal, i.e.,  $s_i^T r_j = 0$ , for  $i \neq j$ . The approximate solution to the seed system is updated in line 8 as

$$(5.3) \quad u_{i+1}^{sd} = u_i^{sd} + \alpha_i v_{i+1}^*.$$

Let  $\tilde{u}_i(\mu) \in \mathbb{R}^{dn}$  be the  $i$ th approximate solution to the shifted preconditioned linear system (4.1), obtained from the BiCG method. Since  $u_i^{sd}, \tilde{u}_i(\mu)$  are elements of the Krylov subspace  $\mathcal{K}_i$  as in (3.3), we can express these approximate solutions as

$$u_i^{sd} = p_{i-1} \left( M(K - \sigma M)^{-1} \right) \tilde{b}, \quad \tilde{u}_i(\mu) = \tilde{p}_{i-1} \left( \frac{1}{(-\mu + \sigma)} I + M(K - \sigma M)^{-1} \right) \tilde{b},$$



for  $i = 1, 2, \dots$ , where  $p_{i-1}, \tilde{p}_{i-1}$  are polynomials of degree less than or equal to  $i - 1$ . Similarly, the residual  $r_i$  in (5.2) and the residual of the  $i$ th iterate obtained from the BiCG algorithm applied to the shifted system, denoted  $\tilde{r}_i$ , can be expressed as

$$(5.4) \quad r_i = q_i (M(K - \sigma M)^{-1}) \tilde{b}, \quad \tilde{r}_i = \tilde{q}_i \left( \frac{1}{(-\mu + \sigma)} I + M(K - \sigma M)^{-1} \right) \tilde{b},$$

where  $q_i(t) := 1 - tp_{i-1}(t)$ ,  $\tilde{q}_i(t) := 1 - t\tilde{p}_{i-1}(t)$ , and

$$(5.5) \quad q_i(0) = \tilde{q}_i(0) = 1.$$

Here the polynomials  $q_i$  and  $\tilde{q}_i$  are of degree less than or equal to  $i$ . The following lemma characterizes the relation between  $r_i$  and  $\tilde{r}_i$ , originally established in [22].

LEMMA 5.1. *Let  $r_i$  and  $\tilde{r}_i$  as in (5.4) be the residuals at iteration  $i$  resulting from the BiCG method applied to (3.9) and (4.1), respectively. Then, there exists  $\zeta_i \in \mathbb{R}$  such that  $r_i = \zeta_i \tilde{r}_i$ , i.e.,  $r_i$  and  $\tilde{r}_i$  are colinear for  $i = 1, \dots, j$ .*

We omit the proof of Lemma 5.1 as it is analogous to the proof of [22, Theorem 1]. Briefly, the result is shown by noting the shift-invariance of Krylov subspaces as well as the property of the residual of the BiCG iterates:  $r_i, \tilde{r}_i \in (\mathcal{L}_i)^\perp \cap \mathcal{K}_{i+1} =: \mathcal{J}_i$ , where  $\mathcal{L}_i$  is as in (3.4) and  $\mathcal{K}_{i+1}$  is as in (3.3). The biorthogonality condition (3.6) implies that the dimension of  $\mathcal{J}_i$  is 1, and the result follows.

By Lemma 5.1, the relation between the residual of the shifted system and the seed system is given by

$$(5.6) \quad \tilde{r}_i = \frac{1}{\zeta_i} r_i,$$

or, equivalently, with residual polynomials,

$$(5.7) \quad \tilde{r}_i = \tilde{q}_i \left( \frac{1}{-\mu + \sigma} I + M(K - \sigma M)^{-1} \right) \tilde{b} = \frac{1}{\zeta_i} q_i (M(K - \sigma M)^{-1}) \tilde{b}.$$

The equality in (5.7) can be expressed as a function of  $t$ , i.e.,

$$\tilde{q}_i (1/(-\mu + \sigma) + t) = (1/\zeta_i) q_i(t),$$

and, paired with the equality in (5.5), gives

$$(5.8) \quad \zeta_i = q_i \left( \frac{-1}{-\mu + \sigma} \right).$$

Note that the above gives an exact expression for the colinearity coefficient  $\zeta_i$  in Lemma 5.1, completely determined from the residual of the seed system. Thus, from (5.6) and (5.8), we can express the  $i$ th residual resulting from the BiCG method applied to the shifted system via the residual of the  $i$ th iterate obtained from the BiCG procedure applied to the seed system. In other words, we can obtain the residual vectors corresponding to many different shifted systems from one execution of the algorithm and, as a result, update the search vectors in (5.1) to approximate the solution to many shifted systems as in (5.3).

The derivation which follows serves to clarify lines 9–14 of the Exact Algorithm 1, where our algorithm is applied to (4.1) for a set of shifts  $\{\mu_l\}$ ,  $l = 1, \dots, k$ . In the algorithm, we denote the particular  $\zeta_i$  for each shift  $\mu_l$  as  $\zeta_i(\mu_l)$  but use the notation  $\zeta_i := \zeta_i(\mu)$  here for simplicity.

The residual of the BiCG iterates applied to the seed system satisfy the following three-term recursion:

$$(5.9) \quad r_{i+1} = -\alpha_i M(K - \sigma M)^{-1} r_i + \frac{\beta_i \alpha_i}{\alpha_{i-1}} r_{i-1} + \left(1 - \frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) r_i,$$

where we have inserted the update formula  $v_{i+1}^* = r_i - \beta_i v_i^*$  from line 3 of the algorithm into the computation in line 7, i.e.,

$$\begin{aligned} r_i &= r_{i-1} - \alpha_{i-1} M(K - \sigma M)^{-1} v_i^* \\ &= r_{i-1} - \alpha_{i-1} M(K - \sigma M)^{-1} \left( \frac{1}{\beta_i} (r_i - v_{i+1}^*) \right) \\ &= r_{i-1} - \frac{\alpha_{i-1}}{\beta_i} M(K - \sigma M)^{-1} r_i + \frac{\alpha_{i-1}}{\beta_i \alpha_i} (r_i - r_{i+1}) \end{aligned}$$

and used the recursive update formula  $M(K - \sigma M)^{-1} v_{i+1}^* = (1/\alpha_i)(r_i - r_{i+1})$  in the last equality. The relation (5.9) can be expressed with the residual polynomial from (5.4) as

$$(5.10) \quad q_{i+1}(t) = -\alpha_i t q_i(t) + \frac{\beta_i \alpha_i}{\alpha_{i-1}} q_{i-1}(t) + \left(1 - \frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) q_i(t).$$

Specifically, taking  $t = -1/(-\mu + \sigma)$  in (5.10) and using the equality in (5.8), gives

$$(5.11) \quad \zeta_{i+1} = \left(1 - \alpha_i \left(\frac{-1}{-\mu + \sigma}\right) - \frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) \zeta_i + \frac{\beta_i \alpha_i}{\alpha_{i-1}} \zeta_{i-1},$$

i.e., a recurrence for the colinearity coefficients in Lemma 5.1, incorporated in line 10 of the algorithm. Thus, from (5.6), (5.9), and (5.11), the three-term recurrence of the residual vectors resulting from the BiCG method applied to the shifted system is given by

$$\begin{aligned} \tilde{r}_{i+1} &= \frac{1}{\zeta_{i+1}} \left( -\zeta_i \alpha_i M(K - \sigma M)^{-1} \tilde{r}_i + \zeta_{i-1} \left(\frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) \tilde{r}_{i-1} + \zeta_i \left(1 - \frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) \tilde{r}_i \right) \\ &= -\frac{\zeta_i \alpha_i}{\zeta_{i+1}} \left( \frac{1}{-\mu + \sigma} I + M(K - \sigma M)^{-1} \right) \tilde{r}_i + \frac{\zeta_{i-1}}{\zeta_{i+1}} \left(\frac{\beta_i \alpha_i}{\alpha_{i-1}}\right) \tilde{r}_{i-1} \\ &\quad + \left(1 - \frac{\zeta_{i-1}}{\zeta_{i+1}} \left(\frac{\beta_i \alpha_i}{\alpha_{i-1}}\right)\right) \tilde{r}_i, \end{aligned}$$

where we have used the relation (5.11) to obtain the third term of the summation in the last equality. Equivalently, we can express the  $(i+1)$ st residual generated from approximating the solution to (4.1) with the BiCG method through the recurrence

$$\tilde{r}_{i+1} = -\tilde{\alpha}_i \left( \frac{1}{-\mu + \sigma} I + M(K - \sigma M)^{-1} \right) \tilde{r}_i + \frac{\tilde{\beta}_i \tilde{\alpha}_i}{\tilde{\alpha}_{i-1}} \tilde{r}_{i-1} + \left(1 - \frac{\tilde{\beta}_i \tilde{\alpha}_i}{\tilde{\alpha}_{i-1}}\right) \tilde{r}_i,$$

where the coefficients  $\tilde{\alpha}_i$  and  $\tilde{\beta}_i$  are defined as

$$(5.12) \quad \tilde{\alpha}_i := -\alpha_i \left( \frac{\zeta_i}{\zeta_{i+1}} \right), \quad \tilde{\beta}_i := \left( \frac{\alpha_i}{\tilde{\alpha}_i} \right) \left( \frac{\tilde{\alpha}_{i-1}}{\alpha_{i-1}} \right) \frac{\zeta_{i-1}}{\zeta_{i+1}} \beta_i = \left( \frac{\zeta_{i-1}}{\zeta_i} \right)^2 \beta_i$$

and are updated in line 11 of the algorithm. Note that initializing with parameters  $\zeta_0 = \zeta_1 = 1$  in (5.11) ensures that the formulation described above holds for the corresponding seed system

(cf. (5.9)). Analogous to (5.1), we compute the search vectors for solving the shifted systems as

$$(5.13) \quad \tilde{v}_{i+1} = \frac{1}{\tilde{\zeta}_i} r_i - \tilde{\beta}_i \tilde{v}_i,$$

in line 12 and update the approximation of each shifted preconditioned system in line 13 as

$$(5.14) \quad \tilde{u}_{i+1}(\mu) = \tilde{u}_i(\mu) + \tilde{\alpha}_i \tilde{v}_{i+1},$$

i.e., the shifted equivalent of the update described in (5.3). Lines 15–25 of the algorithm ensure that the approximations  $\tilde{x}_j(\mu_l)$  to the linear system (1.1) from the Krylov subspace of dimension  $j$  have relative residual norm below a certain tolerance  $tol$ , for  $l = 1, \dots, k$ .

The Exact Algorithm 1 applies the shift-and-invert preconditioners  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$  via a block LU decomposition as described in Section 4. Note that for each update of the solution to the seed system, only some additional scalar operations and vector additions are required to update the approximations of  $\tilde{u}(\mu_l)$  as in (4.1) for each  $\mu_l$ . This is due to the colinearity of the residuals  $r_i$  and  $\tilde{r}_i$  as described in Lemma 5.1. Furthermore, the Exact Algorithm 1 does not require the storage of the residual vectors in (5.2) at each iteration, as long as the values of  $\mu_l$ ,  $l = 1, \dots, k$ , are determined before the algorithm is executed. This allows for a method with low memory consumption, even when the degree  $d$  of the Chebyshev approximation is large. If  $\rho_i$  in line 2 vanishes, then the algorithm encounters a breakdown. This scenario never occurred in our experiments.

REMARK 5.2 (Adjoint parameterized system). The Exact Algorithm 1 can be used to approximate the shifted right preconditioned adjoint linear system given by

$$(5.15) \quad \begin{aligned} (-\mu + \sigma) \left( \frac{1}{(-\mu + \sigma)} I + (K - \sigma M)^{-T} M^T \right) \tilde{w}(\mu) &= \tilde{c} \\ \iff ((K - \sigma M)^T + (-\mu + \sigma) M^T) \tilde{w}(\mu) &= (K - \sigma M)^T \tilde{c} \\ \iff (K - \mu M)^T \tilde{w}(\mu) &= (K - \sigma M)^T \tilde{c}, \end{aligned}$$

where  $\tilde{w}(\mu) \in \mathbb{R}^{dn}$ . This system can be viewed as the shifted version of the adjoint seed system (3.10). Solutions to the adjoint system  $A(\mu)^T z(\mu) = c$ ,  $c \in \mathbb{R}^n$ , cannot be recovered from the above system due to the structure of the adjoint of the linear system (2.3). Specifically, the solution vector in (5.15) does not contain  $z(\mu)$  since the Chebyshev interpolation coefficients of  $A^T(\mu)$  appear in the last block column of the matrix  $(K - \mu M)^T$  as opposed to the last block row of the matrix  $(K - \mu M)$  in (2.3).

**6. Simulation of a parameterized Helmholtz equation.** To highlight the capabilities of our method, we consider a Helmholtz equation, which describes the propagation of waves. Successful approaches for solving the Helmholtz equation have been considered in prior works such as [8, 16], as well as in [17, 18], where preconditioning was combined with fast iterative solvers. In particular, we consider the parameterized Helmholtz equation given by

$$(6.1) \quad \begin{aligned} (\nabla^2 + \sin^2(\mu)\alpha(x) + \mu^2 + \cos^2(\mu)\beta(x)) u(x) &= h(x), & x \in \Omega, \\ u(x) &= 0, & x \in \partial\Omega, \end{aligned}$$

where  $\alpha(x) = 1 + \sin(x_1)$ ,  $\beta(x) = 1 + \cos(x_2)$ ,  $h(x) = \exp(-x_1 x_2)$ , and  $\Omega \subset ([0, 1] \times [0, 1])$  is as in Figure 6.2. The parameter  $\mu$  in (6.1) can be interpreted as a material parameter. It is of interest to approximate the solution  $u(x)$  for a variety of different values  $\mu$ .

Consider a discretization of (6.1) which is of the same form as (1.1), i.e.,

$$(6.2) \quad A(\mu) := A_0 + \sin^2(\mu)A_1 + \mu^2 A_2 + \cos^2(\mu)A_3,$$

---

**Algorithm 1:** Preconditioned Chebyshev BiCG method for parameterized linear systems.

---

**Input:**  $P_\ell$ ,  $\ell = 0, 1, \dots, d$ , as in (1.3) (Chebyshev coefficients)  
 $L_P U_P$ , decomposition of  $P(\sigma) \in \mathbb{R}^{n \times n}$  for  $L_\sigma^{-1}$   
 $\tilde{b} \in \mathbb{R}^{dn}$  as in (2.4),  $\tilde{c} \in \mathbb{R}^{dn}$  such that  $\tilde{b}^T \tilde{c} \neq 0$   
 $\sigma \in \mathbb{R}$  as in (3.1) (target parameter),  $tol$   
 $\{\mu_l\}_{l=1, \dots, k}$ ,  $\mu_l \in \mathbb{R}$ ,  $|\mu_1 - \sigma| \leq |\mu_2 - \sigma| \leq \dots \leq |\mu_k - \sigma|$  (ordered set of shifts)  
 $A(\mu_l)$ ,  $l = 1, \dots, k$ , as in (1.2),  $\zeta_{-1}(\mu_l) = \zeta_0(\mu_l) = 1$ ,  $l = 1, \dots, k$

**Output:** Approx. sol.  $\tilde{x}_j(\mu_l)$ ,  $l = 1, \dots, k$ , to (1.1), from subspace of dim.  $j$

**Initialize:**  $\rho_{-1} = 1$ ,  $\alpha_{-1} = 1$ ,  $\omega(\mu_l) = -1/(-\mu_l + \sigma)$ ,  $l = 1, \dots, k$   
 $v_0^* = w_0^* = 0 \in \mathbb{R}^{dn}$  (search direction, seed system)  
 $\tilde{v}_0(\mu_l) = 0 \in \mathbb{R}^{dn}$ ,  $l = 1, \dots, k$  (search directions, shifted systems)  
 $u_0^{sd} = 0 \in \mathbb{R}^{dn}$  (approx. to seed system)  
 $\tilde{u}_0(\mu_l) = 0 \in \mathbb{R}^{dn}$ ,  $l = 1, \dots, k$  (approx. to shifted systems)  
 $r_0 = \tilde{b}$ ,  $s_0 = \tilde{c}$  (residual vectors)

1 **for**  $i = 0, 1, 2, \dots$  **do**  
2      $\rho_i = (r_i)^T s_i$ ,  $\beta_i = -\rho_i / \rho_{i-1}$   
3      $v_{i+1}^* = r_i - \beta_i v_i^*$ ,  $w_{i+1}^* = s_i - \beta_i w_i^*$   
4     Compute  $\hat{v}_1$  such that  $\hat{v}_1 = M((K - \sigma M)^{-1} v_{i+1}^*)$  as in (4.4)  
5      $\alpha_i = \rho_i / ((w_{i+1}^*)^T \hat{v}_1)$   
6     Compute  $\hat{v}_2$  such that  $\hat{v}_2 = (K - \sigma M)^{-T} (M^T w_{i+1}^*)$  as in (4.5)  
7      $r_{i+1} = r_i - \alpha_i \hat{v}_1$ ,  $s_{i+1} = s_i - \alpha_i \hat{v}_2$   
8      $u_{i+1}^{sd} = u_i^{sd} + \alpha_i v_{i+1}^*$  (approx. sol. to seed system from subspace of dim  $i + 1$ )  
9     **for**  $l = 1, \dots, k$  **do**  
10          $\zeta_{i+1}(\mu_l) = (1 - \alpha_i \omega(\mu_l) - \frac{\beta_i \alpha_i}{\alpha_{i-1}}) \zeta_i(\mu_l) + \frac{\beta_i \alpha_i}{\alpha_{i-1}} \zeta_{i-1}(\mu_l)$  as in (5.11)  
11          $\tilde{\alpha}_i(\mu_l) = -\alpha_i \left( \frac{\zeta_i(\mu_l)}{\zeta_{i+1}(\mu_l)} \right)$ ,  $\tilde{\beta}_i(\mu_l) = \left( \frac{\zeta_{i-1}(\mu_l)}{\zeta_i(\mu_l)} \right)^2 \beta_i$  as in (5.12)  
12          $\tilde{v}_{i+1}(\mu_l) = \left( \frac{1}{\zeta_i(\mu_l)} \right) r_i - \tilde{\beta}_i(\mu_l) \tilde{v}_i(\mu_l)$  as in (5.13)  
13          $\tilde{u}_{i+1}(\mu_l) = \tilde{u}_i(\mu_l) + \tilde{\alpha}_i(\mu_l) \tilde{v}_{i+1}(\mu_l)$  as in (5.14)  
14     **end**  
15     Compute  $\tilde{x}_{i+1}(\mu_k) = \text{POSTPROCESS}(\omega(\mu_k), \tilde{u}_{i+1}(\mu_k))$   
16     Compute  $res = \|A(\mu_k) \tilde{x}_{i+1}(\mu_k) - b\| / \|b\|$   
17     **if**  $res \leq tol$  **then**  
18         Set  $\tilde{x}_j(\mu_k) = \tilde{x}_{i+1}(\mu_k)$ ,  $j = i + 1$   
19         **for**  $l = 1, \dots, k - 1$  **do**  
20              $\tilde{x}_j(\mu_l) = \text{POSTPROCESS}(\omega(\mu_l), \tilde{u}_j(\mu_l))$   
21         **end**  
22         **if**  $\|A(\mu_l) \tilde{x}_j(\mu_l) - b\| / \|b\| \leq tol$ ,  $l = 1, \dots, k - 1$  **then**  
23             return  
24         **end**  
25     **end**  
26 **end**  
27 **function**  $\tilde{x}_{i+1}(\mu) = \text{POSTPROCESS}(\omega(\mu), \tilde{u}_{i+1}(\mu))$   
28     Compute  $\hat{v}_3 = \omega(\mu)(K - \sigma M)^{-1} \tilde{u}_{i+1}(\mu)$  as in (4.4)  
29     Set  $\tilde{x}_{i+1}(\mu) = \hat{v}_3(1 : n)$   
30 **end**

---

where  $A_0, \dots, A_3$  arise from a finite element method (FEM) discretization and  $b$  is the corresponding load vector.<sup>1</sup> Approximating  $A(\mu)$  in (6.2) with a Chebyshev interpolation leads to a parameterized linear system of the form (1.4), where  $P(\mu) \approx A(\mu)$ . We consider an approximate solution to the shifted preconditioned system (3.1) based on a linearization of  $P(\mu)$  as in (2.3). The resulting approximation of the companion linearization and, equivalently, to  $x(\mu)$  is obtained for many values of the parameter  $\mu$  via one execution of the Exact Algorithm 1. The relative residual norm at iteration  $i$ ,

$$(6.3) \quad \frac{\|A(\mu)\tilde{x}_i(\mu) - b\|}{\|b\|},$$

is computed for a variety of  $\mu$  with  $A(\mu)$  in (6.2). The results of this experiment are presented in Figures 6.1–6.5. Here, the nonlinear functions  $\sin^2(\mu)$  and  $\cos^2(\mu)$  are approximated using `Chebfun` in Matlab with truncation parameter  $d$  as in (1.3); see [13]. All simulations in this paper were carried out on a 2.3 GHz Dual-Core Intel Core i5 processor with 16 GB RAM. The software for all examples in this paper were implemented in Matlab, and we made them available online.<sup>2</sup>

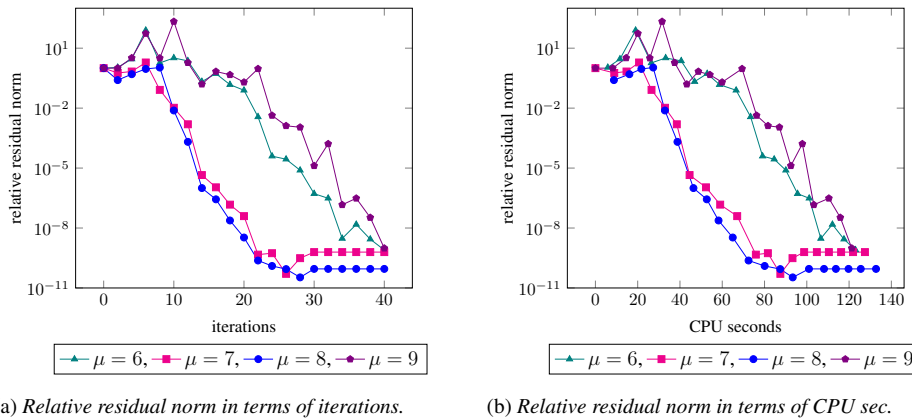


FIG. 6.1. Convergence for approximating the parameterized Helmholtz equation (6.1), generated from one execution of the Exact Algorithm 1 with  $\sigma = 7.5$  and evaluated for different values of  $\mu \in [-10, 10]$ . Here  $n = 243997$ ,  $d = 50$ ,  $tol = 10^{-9}$ , and with the relative residual norm in (6.3).

Our experiments show that the preconditioned BiCG method leads to an accurate approximation of the linear system corresponding to a discretization of (6.1) for values of the parameter  $\mu$  close to the target parameter  $\sigma$ , and one execution of the Exact Algorithm 1 leads to a large variety of solution approximations. Specifically, as the algorithm is run to solve the seed system (3.9), each additional approximation corresponding to values of the parameter  $\mu_l$ ,  $l = 1, \dots, k$ , is updated using only scalar operations and vector additions. Before each of the approximates of (1.1) is returned, a final application of the preconditioner is performed. The basis for the Krylov subspace (3.3) does not need to be stored if the values of the parameter  $\mu$  are determined in advance.

The simulation, visualized in Figure 6.1(a), gives access to the solutions corresponding to all values of  $\mu \in [6, 9]$ , though we visualize just four of these. Analogously, the simulations shown in Figure 6.3(a) and Figure 6.4(a) provide all solutions corresponding to  $\mu \in [10.5, 12]$ .

<sup>1</sup>The matrices and vector were generated using the finite element software FEniCS [2].

<sup>2</sup><https://github.com/siobhanie/ChebyshevBiCG>

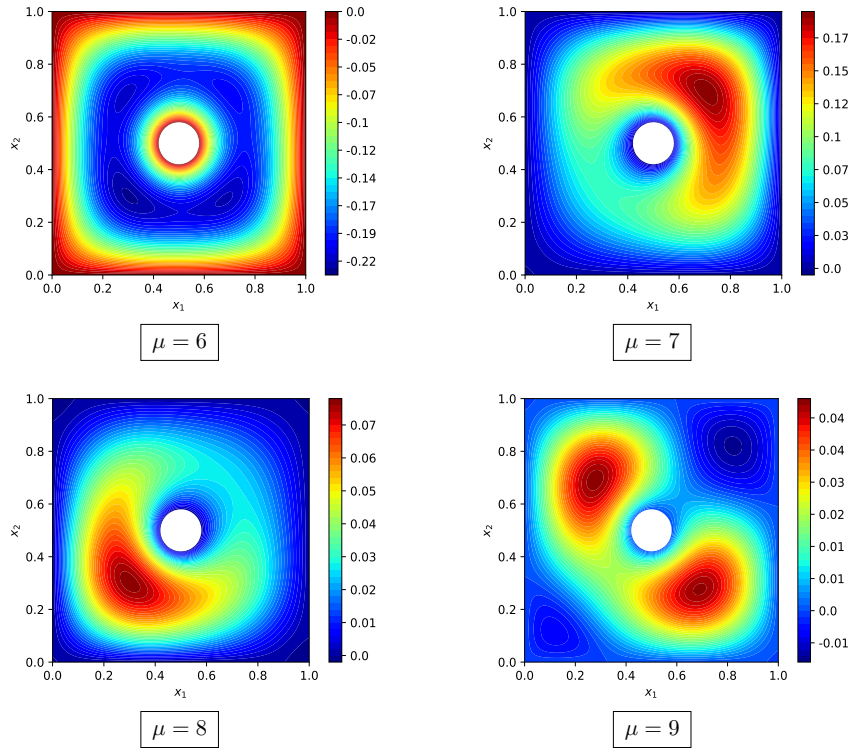


FIG. 6.2. Solutions of a Helmholtz equation corresponding to the simulation in Figure 6.1(a) and one execution of the Exact Algorithm 1 with  $\sigma = 7.5$ .

Note that the relative residual norms of the approximate solutions are all below a prescribed tolerance.

In Figure 6.1(b), Figure 6.3(b), and Figure 6.4(b), we see the benefit of the short-term recurrence property of the Exact Algorithm 1, i.e., the roughly constant cost of each iteration. Specifically, we plot the relative residual norm in terms of CPU seconds, where we measure CPU time after the precomputation steps. This feature is especially useful for simulations which require many iterations until convergence and is not present in methods with a long-term recurrence, e.g., GMRES and the full orthogonal method (FOM). Note that, though one execution of the Exact Algorithm 1 gives approximations for many different values of  $\mu$ , each convergence curve here corresponds to a separate run.

From comparing the simulations in Figure 6.3 and Figure 6.4, we see that the cost of the approach is largely independent of the degree  $d$  of the Chebyshev interpolation. More precisely, the interpolation corresponding to the simulation in Figure 6.3 was performed on the interval  $[-15, 15]$  with  $d = 64$ , leading to a companion linearization of dimension  $64n \times 64n$ . Similarly, the simulation in Figure 6.4 was performed on the interval  $[-40, 40]$  with  $d = 124$ , where the companion linearization had dimension  $124n \times 124n$ . The experiments in Figure 6.3(a) and Figure 6.4(a) converged in roughly the same number of iterations. The simulation in Figure 6.4(b) required approximately twice as many matrix-vector products with a matrix of dimension  $n \times n$  as the one in Figure 6.3(b) and took roughly twice as many CPU seconds as a result. Note that the cost of the application of the preconditioner is the same in both of these simulations as one LU decomposition of  $P(\sigma) \in \mathbb{R}^{n \times n}$  is performed in the

precomputation step. By performing the interpolation on a larger interval, we have access to a greater variety of solutions. However, only the solutions corresponding to values of  $\mu$  close to the target  $\sigma$  converge quickly.

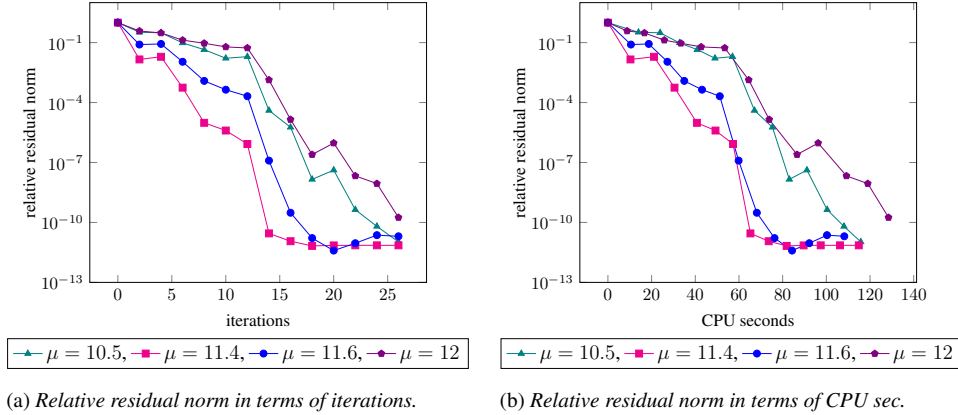


FIG. 6.3. Convergence for approximating the parameterized Helmholtz equation (6.1), generated from one execution of the Exact Algorithm 1 with  $\sigma = 11.25$  and evaluated for different values of  $\mu \in [-15, 15]$ . Here  $n = 243997$ ,  $d = 64$ ,  $\text{tol} = 10^{-9}$ , and the relative residual norm (6.3).

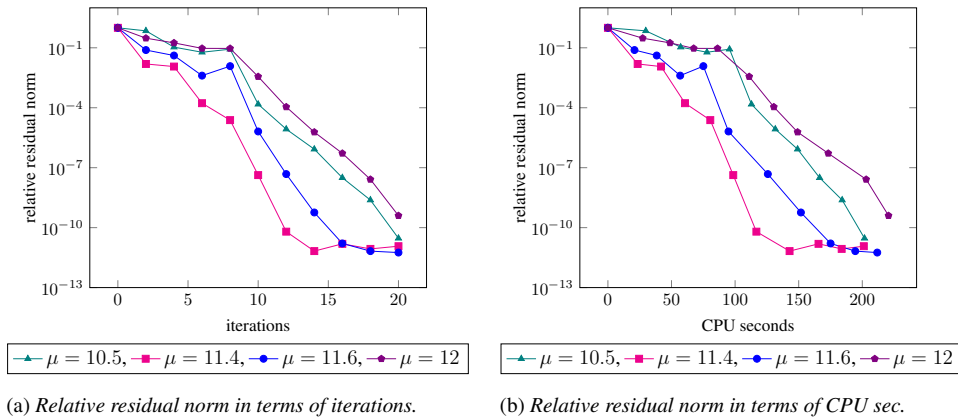


FIG. 6.4. Convergence for approximating the parameterized Helmholtz equation (6.1), generated from one execution of the Exact Algorithm 1 with  $\sigma = 11.25$  and evaluated for different values of  $\mu \in [-40, 40]$ . Here  $n = 243997$ ,  $d = 124$ ,  $\text{tol} = 10^{-9}$ , and the relative residual norm (6.3).

**REMARK 6.1** (Magnitude of the parameter  $\mu$ ). The Chebyshev coefficients can be approximated efficiently using a discrete cosine transform. Thus, we can interpolate  $A(\mu)$  on a large interval with little additional cost. While doing so leads to a larger linearization, it also allows for approximations of (1.1) for large values of  $\mu$ . Considering values of  $\mu$  close enough to  $\sigma$  leads to convergence in  $j$  iterations, where  $j \ll dn$ , with the additional benefit of a constant low cost per iteration. The experiments in this section were designed in order to show a wide variety of solutions.

**7. Simulation of a transfer function of a time-delay system.** Consider the solution to the linear system (1.1), where

$$(7.1) \quad A(\mu) := -\mu I + A_0 + A_1 e^{-\mu}$$



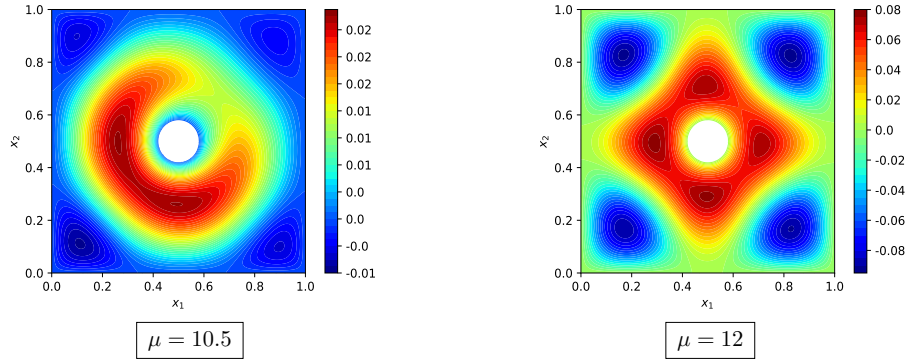


FIG. 6.5. Solutions of a Helmholtz equation corresponding to the simulation in Figure 6.3(a) and one execution of the Exact Algorithm 1 with  $\sigma = 11.25$ .

with random matrices  $A_0, A_1 \in \mathbb{R}^{n \times n}$ , a random vector  $b \in \mathbb{R}^n$ , and  $n = 80$ . The solution to this system is the transfer function of the time-delay system described by

$$\begin{aligned} \dot{x}(t) &= A_0 x(t) + A_1 x(t - \tau) - b v(t), \\ y(t) &= C^T x(t). \end{aligned}$$

Specifically, the transfer function is obtained by applying the Laplace transform to the state equation with  $x(0) = 0$ . In this formulation,  $\mu$  is the Laplace variable; see [27, 35, 36]. The vector  $b \in \mathbb{R}^n$  is the external force,  $x(t) \in \mathbb{R}^n$  is the state vector,  $v(t)$  is the input,  $y(t)$  is the output, and  $\tau > 0$  is the delay. Without loss of generality, we set  $\tau = 1$  and assume the entire state is the output, i.e.,  $C = I \in \mathbb{R}^{n \times n}$ .

Here we use the Exact Algorithm 1 with preconditioner  $K^{-1}$  and the adjoint preconditioner  $K^{-T}$ . The application of the preconditioners is analogous to the implementation of  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$  with  $\sigma = 0$ . The shifted preconditioned system (3.1c) is approximated with one execution of the Exact Algorithm 1, generating approximations of  $x(\mu)$  as in (1.1). The relative residual norm at iteration  $i$  is computed as in (6.3) with  $A(\mu)$  from (7.1).

The results of this experiment are displayed in Figure 7.1. We see that our method is competitive for a variety of positive and negative values of  $\mu$ , and that approximations corresponding to values of  $\mu$  closer to the target parameter  $\sigma = 0$  converge faster than approximations corresponding to values of  $\mu$  farther away. As the Exact Algorithm 1 is run, each approximation is updated using just additional scalar and vector computations. Before each approximate solution to (1.1) is returned, a final application of the preconditioner is performed. The basis for the Krylov subspace (3.3) does not need to be stored if the values of the parameter  $\mu$  are determined in advance. Additionally, the larger the dimension  $j$  of the Krylov subspace from which the approximates come, the more solutions we have access to. As before, the nonlinear function  $e^{-\mu}$  is approximated using `Chebfun` in Matlab with truncation parameter  $d$  as in (1.3).

**8. An inexact variant of the preconditioned Chebyshev BiCG method for parameterized linear systems.** In the short-term recurrence method of the Exact Algorithm 1, the action of the preconditioners  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$  are applied via a block LU decomposition of the matrix product  $(K - \sigma M)\Pi$  in (4.3) when approximating the solution to the shifted preconditioned linear system (3.1). In this way, each application of  $(K - \sigma M)^{-1}$  requires the action of  $P(\sigma)^{-1} \in \mathbb{R}^{n \times n}$ , and each application of  $(K - \sigma M)^{-T}$  requires the

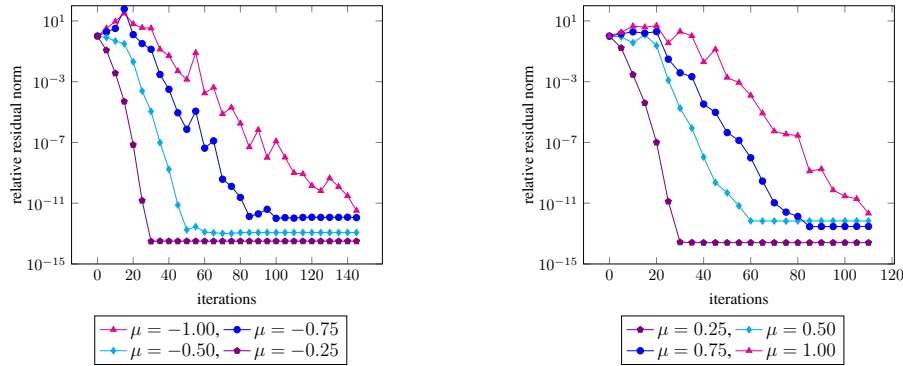


FIG. 7.1. Convergence for approximating (1.1), generated from one (total) execution of the Exact Algorithm 1 with  $\sigma = 0$  and  $A(\mu)$  as in (7.1). Here  $d = 17$ ,  $\text{tol} = 10^{-11}$ , the relative residual norm (6.3), and  $b$  a random vector.

action of  $P(\sigma)^{-T} \in \mathbb{R}^{n \times n}$ , as can be seen in (4.4) and (4.5). An LU decomposition of the matrix  $P(\sigma)$  was performed in the precomputing step and reused at each iteration, though other choices based on a factorization are possible. This approach is only suitable when an LU decomposition of a matrix of dimension  $n \times n$  is feasible.

Consider now an inexact preconditioned BiCG method, where the action of  $(K - \sigma M)^{-1}$  as well as the action of  $(K - \sigma M)^{-T}$  are applied approximately at each iteration. Specifically, the corresponding linear systems with system matrices  $P(\sigma)$  and  $P(\sigma)^T$  are solved iteratively, and the accuracy of the solves varies from one iteration to the next. This approach, inspired by the work [51], eliminates the need for an LU decomposition of  $P(\sigma)$ , offering an improvement in performance for approximating solutions to (1.1). A similar inner-outer approach was investigated in [49], where the effect of the error in the inexact matrix-vector products was analyzed for several different iterative methods. Additionally, in [32], a relaxation strategy for low-rank ADI was studied for solving Lyapunov equations. Both of these works successfully increased the inner tolerance as the error in the outer method decreased, and we apply this strategy to our method.

Similar approaches, where Krylov methods have been used as preconditioners, have been investigated in, for example, [10, 12, 43], as well as [4, 48, 52], where stopping criteria were utilized. To our knowledge, this is the first time that an inexact BiCG method has been used to solve for the solution of multiple shifted systems simultaneously and the first time that such a framework has been paired with a linearization of the form in (1.5) to solve parameterized systems, where the dependence on the parameter is nonlinear.

**8.1. Derivation of the inexact preconditioned Chebyshev BiCG method for parameterized linear systems.** The  $j$ th iteration of the Lanczos biorthogonalization procedure generated by the matrix  $M(K - \sigma M)^{-1}$  and its adjoint appear in (3.5). As we are interested in an inexact algorithm, we consider the analogous relations corresponding to the inexact and iteration-dependent application of the preconditioning matrices  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$ , i.e., at the iteration  $j$ , the relations

$$\begin{aligned}
 (8.1) \quad M\hat{Z}_j &= V_j \hat{T}_j + \hat{\beta}_j v_{j+1} e_j^T = V_{j+1} \hat{T}_j, \\
 \hat{X}_j &= W_j \hat{T}_j^T + \hat{\gamma}_j w_{j+1} e_j^T = W_{j+1} \hat{T}_j^T
 \end{aligned}$$

hold, where

$$(8.2a) \quad \hat{z}_i = \mathcal{P}_{1_i}^{-1} v_i,$$

$$(8.2b) \quad \hat{x}_i = \mathcal{P}_{2_i}^{-1} (M^T w_i)$$

with

$$\mathcal{P}_{1_i}^{-1} \approx (K - \sigma M)^{-1} \quad \text{and} \quad \mathcal{P}_{2_i}^{-1} \approx (K - \sigma M)^{-T}.$$

Here  $\hat{Z}_j = [\hat{z}_1, \dots, \hat{z}_j]$ ,  $\hat{X}_j = [\hat{x}_1, \dots, \hat{x}_j] \in \mathbb{R}^{dn \times j}$ , and the matrices  $\hat{T}_j \in \mathbb{R}^{j \times j}$  and  $\hat{\underline{T}}_j, \hat{\underline{T}}_j^T \in \mathbb{R}^{(j+1) \times j}$  are of the same form as the matrices in (3.7) and (3.8) with entries  $\hat{\alpha}_i$ ,  $\hat{\gamma}_i$ , and  $\hat{\beta}_i$ ,  $i = 1, \dots, j$ , defined below. Note that the choice of notation in (8.2) above is to differentiate the application of the preconditioner in (8.2a) from the application of the adjoint preconditioner in (8.2b). More specifically, the preconditioning matrix  $(K - \sigma M)^{-T}$  is applied inexactly to the vector  $(M^T w_i) \in \mathbb{R}^{dn}$  in the adjoint case.

In this setting, an application of the preconditioner  $\mathcal{P}_{1_i}^{-1}$  refers to approximating the action of  $P(\sigma)^{-1}$  within the application of  $L_\sigma^{-1}$  as in (4.4), and an application of  $\mathcal{P}_{2_i}^{-1}$  refers to the application of  $P(\sigma)^{-T}$  in the application of  $L_\sigma^{-T}$  as in (4.5). We denote an exact application of the preconditioners on the vectors  $v_i$  and  $M^T w_i$  as

$$(8.3) \quad \begin{aligned} z_i &:= (K - \sigma M)^{-1} v_i, \\ x_i &:= (K - \sigma M)^{-T} (M^T w_i). \end{aligned}$$

Note that these vectors are not computed in the inexact algorithm, as the preconditioners are applied only approximately. These vectors are defined for the purpose of our analysis. Consider the  $i$ th column in equations (8.1), i.e.,

$$(8.4) \quad \begin{aligned} M \hat{z}_i &= \hat{\gamma}_{i-1} v_{i-1} + \hat{\alpha}_i v_i + \hat{\beta}_i v_{i+1}, \\ \hat{x}_i &= \hat{\beta}_{i-1} w_{i-1} + \hat{\alpha}_i w_i + \hat{\gamma}_i w_{i+1}, \end{aligned}$$

where  $v_0 := 0$ ,  $w_0 := 0$ . Equations (8.4) paired with the biorthogonality condition (3.6) lead to the definition

$$(8.5) \quad \hat{\alpha}_i := w_i^T M \hat{z}_i.$$

Further, we define

$$(8.6a) \quad \hat{r}_i := \hat{\beta}_i v_{i+1} = M \hat{z}_i - \hat{\alpha}_i v_i - \hat{\gamma}_{i-1} v_{i-1},$$

$$(8.6b) \quad \hat{s}_i := \hat{\gamma}_i w_{i+1} = \hat{x}_i - \hat{\alpha}_i w_i - \hat{\beta}_{i-1} w_{i-1},$$

giving

$$1 = w_{i+1}^T v_{i+1} = \begin{pmatrix} \hat{s}_i \\ \hat{\gamma}_i \end{pmatrix}^T \begin{pmatrix} \hat{r}_i \\ \hat{\beta}_i \end{pmatrix},$$

and thus the following definition:

$$(8.7) \quad \hat{\gamma}_i := \frac{\hat{s}_i^T \hat{r}_i}{\hat{\beta}_i}.$$

We define  $\hat{\beta}_i$  using the so-called canonical choice as in [25], i.e.,

$$(8.8) \quad \hat{\beta}_i := \|\hat{r}_i\|_2.$$

We are interested in solving the shifted preconditioned system (3.1) with our inexact BiCG method. Equivalent to (8.1), at iteration  $j$ , the shifted inexact relations

$$(8.9) \quad \begin{aligned} V_j + (-\mu + \sigma)M\hat{Z}_j &= V_{j+1} \left( \hat{I}_j + (-\mu + \sigma)\hat{T}_j \right), \\ W_j + (-\mu + \sigma)\hat{X}_j &= W_{j+1} \left( \hat{I}_j + (-\mu + \sigma)\hat{T}_j^T \right) \end{aligned}$$

hold, where  $\hat{I}_j$  is as in (4.2). In practice, we form the matrices  $\hat{Z}_j$  and  $\hat{T}_j$  in (8.9) once and compute  $\tilde{x}_j(\mu)$ , the approximation of the solution to (1.1) at iteration  $j$  for each value of  $\mu$  as

$$(8.10a) \quad y_j(\mu) = (I_j + (-\mu + \sigma)\hat{T}_j)^{-1}(\beta e_1),$$

$$(8.10b) \quad \hat{u}_j(\mu) = \hat{Z}_j y_j(\mu),$$

$$(8.10c) \quad \tilde{x}_j(\mu) = (\hat{u}_j(\mu))_{1:n},$$

where  $\beta := \|b\|$  and  $(\hat{u}_j(\mu))_{1:n}$  denotes the first  $n$  entries of  $(\hat{u}_j(\mu)) \in \mathbb{R}^{dn}$ . Here  $I_j$  is as in (3.6),  $e_1$  is the first column of the identity matrix of dimension  $j \times j$ , and  $\hat{T}_j \in \mathbb{R}^{j \times j}$  is defined as  $\hat{T}_j$  with the last row removed. We assume the matrix in (8.10a) is nonsingular.

Computing the approximation  $\tilde{x}_j(\mu)$  requires one linear solve with a tridiagonal matrix of dimension  $j \times j$  and one matrix-vector product with a matrix of dimension  $dn \times j$  for each value of  $\mu$ . Note that if at any iteration the inner product of the residual vectors vanishes, i.e.,  $(\hat{s}_i)^T \hat{r}_i = 0$ , then the algorithm encounters a breakdown. This situation never occurred while carrying out the experiments in this work. A full description of the inexact preconditioned Chebyshev BiCG algorithm for parameterized linear systems appears in Algorithm 2. We will refer to Algorithm 2 as the Inexact Algorithm 2 or the *inexact algorithm*.

The Inexact Algorithm 2 terminates when the approximations  $\tilde{x}_j(\mu_l)$  of the linear systems given in (1.1) from the Krylov subspace of dimension  $j$  have relative residual norm below a certain tolerance  $tol$ , for  $l = 1, \dots, k$ . As the Inexact Algorithm 2 stores the basis matrix  $\hat{Z}_j$  for approximations of  $\tilde{x}_j(\mu)$ , as well as the tridiagonal matrix  $\hat{T}_j$ , approximations of (1.1) corresponding to  $\mu \notin \{\mu_l\}$ ,  $l = 1, \dots, k$ , can be computed after the algorithm has been executed once. In particular, it is reasonable to expect accurate approximations corresponding to values of  $\mu$  close to the target  $\sigma$ , i.e., for  $\mu$  such that  $|\mu - \sigma| \leq |\mu_k - \sigma|$ .

**REMARK 8.1** (Storing  $\hat{Z}_j$  and  $\hat{T}_j$ ). In the Exact Algorithm 1, the colinearity of the residuals of the  $i$ th approximations of the BiCG method applied to the seed and shifted systems was used in order to derive a short-term recurrence method for shifted systems. These residual vectors spanned (3.3), and a basis matrix for the Krylov subspace was not stored. Further, updates were computed based on an implicit LU factorization of  $T_j$  in (3.5), i.e., by using a coupled two-term recurrence. This is also the approach in the standard BiCG method.

It was not possible to show an analogous colinearity of the residuals for the inexact method, so the Inexact Algorithm 2 does not update approximations of the shifted systems with the same approach as the Exact Algorithm 1. For reasons of presentation, the Inexact Algorithm 2 stores the basis matrix  $\hat{Z}_j$  and computes the approximations corresponding to each  $\mu$  as described in (8.10). An approach based on an implicit LU factorization of each matrix  $(I_j + (-\mu_l + \sigma)\hat{T}_j)$ ,  $l = 1, \dots, k$ , could have been taken. Note that while  $\hat{Z}_j$  is a basis matrix for the set of approximations given by the Inexact Algorithm 2, it is not a basis matrix for (3.3); see Remark 8.3.

---

**Algorithm 2:** Inexact preconditioned Chebyshev BiCG method for parameterized linear systems.

---

**Input:**  $P_\ell$ ,  $\ell = 0, 1, \dots, d$ , as in (1.3) (Chebyshev coefficients)  
 $\tilde{b} \in \mathbb{R}^{dn}$  as in (2.4),  $\tilde{c} \in \mathbb{R}^{dn}$  such that  $\tilde{b}^T \tilde{c} \neq 0$   
 $\sigma \in \mathbb{R}$  as in (3.1) (target parameter),  $tol$   
 $\{\mu_l\}_{l=1, \dots, k}$ ,  $\mu_l \in \mathbb{R}$ ,  $|\mu_1 - \sigma| \leq |\mu_2 - \sigma| \leq \dots \leq |\mu_k - \sigma|$  (ordered set of shifts)  
 $A(\mu_l)$ ,  $l = 1, \dots, k$ , as in (1.2)  
**Output:** Approx. sol.  $\tilde{x}_j(\mu_l)$ ,  $l = 1, \dots, k$ , to (1.1), from subspace of dim.  $j$ ,  
 matrices  $\hat{Z}_j \in \mathbb{R}^{dn \times j}$ ,  $\hat{T}_j \in \mathbb{R}^{j \times j}$   
**Initialize:**  $\hat{r}_0 = \tilde{b}$ ,  $\hat{s}_0 = \tilde{c}$ ,  $\hat{Z}_0 = []$

```

1 for  $i = 1, 2, \dots$  do
2   Normalize  $v_i = \hat{r}_{i-1} / \hat{\beta}_{i-1}$  using (8.8)
3   Normalize  $w_i = \hat{s}_{i-1} / \hat{\gamma}_{i-1}$  using (8.7)
4   Compute  $\hat{z}_i$  and  $\hat{x}_i$  as in (8.2)
5   Update  $\hat{Z}_i = \begin{bmatrix} \hat{Z}_{i-1} & \hat{z}_i \end{bmatrix}$ 
6   Compute  $\hat{\alpha}_i$  as in (8.5)
7   Compute  $\hat{r}_i$  as in (8.6a)
8   Compute  $\hat{s}_i$  as in (8.6b)
9   Update  $\hat{T}_i$  as in (8.1)
10   $\tilde{x}_i(\mu_k) = \text{POSTPROCESS}(\hat{T}_i, \hat{Z}_i, \mu_k, \sigma)$ 
11   $res = \|A(\mu_k)\tilde{x}_i(\mu_k) - b\| / \|b\|$ 
12  if  $res \leq tol$  do
13    Set  $\tilde{x}_j(\mu_k) = \tilde{x}_i(\mu_k)$ ,  $j = i$ 
14    for  $l=1, \dots, k-1$  do
15       $\tilde{x}_j(\mu_l) = \text{POSTPROCESS}(\hat{T}_j, \hat{Z}_j, \mu_l, \sigma)$ 
16    end
17    if  $\|A(\mu_l)\tilde{x}_j(\mu_l) - b\| / \|b\| \leq tol$ ,  $l = 1, \dots, k-1$  do
18      return
19    end
20  end
21 end
22 function  $\tilde{x}_j(\mu) = \text{POSTPROCESS}(\hat{T}_j, \hat{Z}_j, \mu, \sigma)$ 
23   Compute  $y_j(\mu)$  as in (8.10a)
24   Compute  $\hat{u}_j(\mu)$  as in (8.10b)
25   return  $\tilde{x}_j(\mu)$  as in (8.10c)
26 end

```

---

REMARK 8.2 (Numerical behavior of the Exact Algorithm 1 and the Inexact Algorithm 2). The Exact Algorithm 1 and the Inexact Algorithm 2 with exact preconditioning are equivalent in applying the standard BiCG method to the shifted linear system (4.1) but are based on different approaches. Both methods return the same approximate solution, ignoring roundoff errors, but methods which use the same approach as in the Exact Algorithm 1, i.e., the ones based on a coupled two-term recurrence, are often more robust and have better numerical behavior; see [21].

REMARK 8.3 (Krylov subspaces of the *inexact algorithm*). A result in [44] implies that the relations (8.1) can be expressed equivalently as

$$(8.11) \quad \begin{aligned} M\hat{Z}_j &= (M(K - \sigma M)^{-1} + \mathcal{E}_j) V_j = V_{j+1} \hat{T}_j, \\ \hat{X}_j &= ((K - \sigma M)^{-T} M^T + \mathcal{F}_j) W_j = W_{j+1} \hat{T}_j^T, \end{aligned}$$

where  $\mathcal{E}_j = \sum_{i=1}^j E_i v_i w_i^T$  and  $\mathcal{F}_j = \sum_{i=1}^j F_i w_i v_i^T$ , with  $E_i, F_i \in \mathbb{R}^{dn \times dn}$ . Here  $v_i, w_i$  are the  $i$ th columns of  $V_j$  and  $W_j$ , respectively, and the biorthogonality condition (3.6) holds. The matrices  $E_i$  and  $F_i$  represent the error which is introduced on each inexact application of the preconditioner and its adjoint. Note that the relations (8.11) imply that the columns of  $V_j$  in (8.1) span the Krylov subspace generated from the matrix  $(M(K - \sigma M)^{-1} + \mathcal{E}_j)$  and the vector  $v_1$ , and, analogously, the columns of  $W_j$  in (8.1) span the Krylov subspace generated from the matrix  $((K - \sigma M)^{-T} M^T + \mathcal{F}_j)$  and the vector  $w_1$ . We do not compute the matrices  $E_i$  or  $F_i$  and mention them purely for theoretical reasons.

**8.2. Explicit computation of the residual in the *inexact algorithm*.** The Inexact Algorithm 2 approximates the solution of the linear system (1.5) by solving the shifted preconditioned linear system (3.1), where the preconditioners  $(K - \sigma M)^{-1}$  and  $(K - \sigma M)^{-T}$  are applied approximately in an inexact BiCG setting. In order to better understand the convergence of our approach, we compute a bound for the residual at each iteration of the *inexact algorithm* applied to this system. This bound includes a contribution that is directly related to the error in the application of the preconditioner.

Define the inner residual vector  $p_i$  at iteration  $i$  as

$$(8.12) \quad p_i := (K - \sigma M) \hat{z}_i - v_i,$$

where  $\hat{z}_i$  is as in (8.2). Equivalently, in matrix form,  $\hat{Z}_j := Z_j + (K - \sigma M)^{-1} P_j$ , where  $\hat{Z}_j$  is as in (8.1),  $Z_j = [z_1, \dots, z_j] \in \mathbb{R}^{dn \times j}$ , with  $z_i$  as in (8.3), and  $P_j = [p_1, \dots, p_j] \in \mathbb{R}^{dn \times j}$ . The inner residual vector represents how inexactly we apply the preconditioner in the  $i$ th iteration of the Inexact Algorithm 2.

The relations (8.1) and (8.9) are equivalent to the shifted relations given by

$$(8.13) \quad \begin{aligned} V_j + (-\mu + \sigma) M \hat{Z}_j &= V_j \left( I_j + (-\mu + \sigma) \hat{T}_j \right) + (-\mu + \sigma) \hat{\beta}_j v_{j+1} e_j^T, \\ W_j + (-\mu + \sigma) \hat{X}_j &= W_j \left( I_j + (-\mu + \sigma) \hat{T}_j^T \right) + (-\mu + \sigma) \hat{\gamma}_j w_{j+1} e_j^T, \end{aligned}$$

and  $r_i^{in}$ , the residual of the  $i$ th iterate of the Inexact Algorithm 2 applied to the linear system (1.5), is expressed as

$$(8.14) \quad \begin{aligned} r_i^{in} &= \tilde{b} - (K - \mu M) \hat{u}_i(\mu) \\ &= \tilde{b} - (K - \mu M) \hat{Z}_i y_i(\mu) \\ &= \tilde{b} - (K - \mu M + \sigma M - \sigma M) \hat{Z}_i y_i(\mu) \\ &= \tilde{b} - ((-\mu + \sigma) M + (K - \sigma M)) \hat{Z}_i y_i(\mu) \\ &= \tilde{b} - (-\mu + \sigma) M \hat{Z}_i y_i(\mu) - V_i y_i(\mu) - P_i y_i(\mu) \\ &= \left( V_i \left( \beta e_1 - \left( I_i + (-\mu + \sigma) \hat{T}_i \right) \right) - (-\mu + \sigma) \hat{\beta}_i v_{i+1} e_i^T \right) y_i(\mu) - P_i y_i(\mu) \\ &= (\mu - \sigma) \hat{\beta}_i v_{i+1} e_i^T y_i(\mu) - P_i y_i(\mu) \\ &= r_i^{app} - P_i y_i(\mu), \end{aligned}$$

where  $\hat{u}_i(\mu)$ ,  $y_i(\mu)$  are as in (8.10),  $\beta := \|\tilde{b}\|$  with  $\tilde{b}$  is as in (2.3), and we have used the shifted relations (8.13). The vector  $r_i^{app}$  approximates  $r_i^{ex}$ , the residual of the Exact Algorithm 1 applied to (1.5). This residual is defined as  $r_i^{ex} := (\mu - \sigma)\beta_i v_{i+1} e_i^T y_i(\mu)$ , where  $\beta_i$  and  $v_{i+1}$ , as well as the computation of  $y_i(\mu) = (I_i + (-\mu + \sigma)T_i)^{-1}(\beta e_1)$ , similar to (8.10), stem from the relations (3.5); see [42]. Note that the quantity  $r_i^{app}$  above is efficient to compute. An analogous result regarding the application of the adjoint preconditioner  $(K - \sigma M)^{-T}$  holds for the residual of the adjoint linear system given in (5.15).

**8.3. Convergence of the inexact algorithm.** Preconditioned Krylov subspace methods are only suitable when the action of the preconditioner is cheap to apply. It is, therefore, of interest to apply the preconditioners in the Inexact Algorithm 2 in the most efficient way possible. Inspired by work in [44], we prove a computable bound for the residual of the *inexact algorithm* applied to the system (3.1). The error in the inexact application of the preconditioner contributes to this bound.

Consider first the shifted relations (8.9) after  $j - 1$  iterations of the inexact Lanczos biorthogonalization process. Let a QR decomposition of the shifted tridiagonal upper Hessenberg matrix be such that

$$(8.15) \quad \underline{T}_{j-1} := \left( I_{j-1} + (-\mu + \sigma)\hat{T}_{j-1} \right) = Q_{j-1} \left[ R_{j-1}^T \quad 0 \right]^T = Q_{j-1} \hat{R}_{j-1},$$

where the entries of  $\underline{T}_{j-1} \in \mathbb{R}^{j \times (j-1)}$  are denoted as  $\underline{t}_{l,\ell}$ ,  $l = 1, \dots, j$ ,  $\ell = 1, \dots, j-1$ , the matrix  $Q_{j-1}^T$  is defined as

$$(8.16) \quad Q_{j-1}^T := \Omega_{j-1} \Omega_{j-2} \cdots \Omega_1 \in \mathbb{R}^{j \times j},$$

i.e., the product of Givens rotation matrices  $\Omega_i$ ,  $i = 1, \dots, j-1$ , given by

$$\Omega_i := \begin{bmatrix} I_{i-1} & & & & \\ & c_i & s_i & & \\ & -s_i & c_i & & \\ & & & & I_{j-1-i} \end{bmatrix} \in \mathbb{R}^{j \times j},$$

and  $R_{j-1} \in \mathbb{R}^{(j-1) \times (j-1)}$  is upper triangular. Here  $s_i$ ,  $c_i$  are the sines and cosines of the Givens rotations constructed to eliminate the nonzero elements  $\underline{t}_{i+1,i}$  on the subdiagonal of  $\underline{T}_{j-1}$ .

At lines 10–14 of the Inexact Algorithm 2, we update our approximation  $\tilde{x}_j(\mu)$  as in (8.10) by first performing a linear solve with the square matrix  $\mathbf{T}_j := (I_j + (-\mu + \sigma)\hat{T}_j) \in \mathbb{R}^{j \times j}$ . Using the QR factorization in (8.15), the matrix  $\mathbf{T}_j$  can be expressed as

$$\mathbf{T}_j = [\underline{T}_{j-1} \quad \mathbf{t}_j] = [Q_{j-1} \hat{R}_{j-1} \quad \mathbf{t}_j] = Q_{j-1} \left[ \hat{R}_{j-1} \quad Q_{j-1}^T \mathbf{t}_j \right] = Q_{j-1} \tilde{R}_j,$$

where  $\mathbf{t}_j \in \mathbb{R}^j$  is the  $j$ th column of  $\mathbf{T}_j$ ,  $\tilde{R}_j := \left[ \hat{R}_{j-1} \quad Q_{j-1}^T \mathbf{t}_j \right] \in \mathbb{R}^{j \times j}$ , and  $Q_{j-1}^T$  is as in (8.16). We rewrite the linear solve in (8.10a) as

$$(8.17) \quad y_j(\mu) = \mathbf{T}_j^{-1}(\beta e_1) = \tilde{R}_j^{-1} Q_{j-1}^T(\beta e_1)$$

and define  $\hat{g}_j := Q_{j-1}^T(\beta e_1) \in \mathbb{R}^j$ , giving  $y_j(\mu) = \tilde{R}_j^{-1} \hat{g}_j$ , where  $\tilde{R}_j^{-1}$  is an upper triangular matrix as the inverse of an upper triangular matrix. As shown in [42], the entries of  $\hat{g}_j$  are



given by  $\hat{g}_j = [\gamma_1 \ \dots \ \gamma_j]^T$ , where  $\gamma_i := |c_i s_1 \cdots s_{i-1} \beta|$ , for  $i = 1, \dots, j-1$ , and the  $j$ th component is equal to

$$(8.18) \quad \gamma_j := |s_1 \cdots s_{j-1} \beta|.$$

Let  $\eta_i^{(j)} = (y_j(\mu))_i$  denote the  $i$ th component of  $y_j(\mu)$ . Then,

$$(8.19) \quad \begin{aligned} |(y_j(\mu))_i| &= |\eta_i^{(j)}| = |(\tilde{R}_j^{-1})_{i,1:j} \hat{g}_j| \leq \left\| (\tilde{R}_j^{-1})_{i,1:j} \right\| \|(\hat{g}_j)_{1:j}\| \\ &= \left\| e_i^T \tilde{R}_j^{-1} \right\| \|(\hat{g}_j)_{1:j}\| = \left\| \tilde{R}_j^{-1} \right\| \|(\hat{g}_j)_{1:j}\| = \frac{1}{\sigma_j(\mathbf{T}_j)} \|(\hat{g}_j)_{1:j}\|, \end{aligned}$$

where  $\sigma_j(\mathbf{T}_j)$  denotes the largest singular value of  $\mathbf{T}_j$  and

$$\begin{aligned} \|(\hat{g}_j)_{1:j}\|^2 &= \gamma_1^2 + \gamma_2^2 + \cdots + \gamma_j^2 \\ &= \beta^2 (|c_1 s_1 s_2 \cdots s_{j-1}|^2 + |c_2 s_1 s_2 \cdots s_j|^2 + \cdots + |s_1 s_2 \cdots s_{j-1}|^2) \\ &= \beta^2 (s_1 \cdots s_{j-1})^2 (|c_1|^2 + |c_2 s_1|^2 + \cdots + |s_1 \cdots s_{j-1}|^2). \end{aligned}$$

The equality  $\left\| [c_i, c_{i+1} s_i, \dots, s_i \cdots s_{j-1}] \right\| = \|\Omega_{j-1} \cdots \Omega_i e_i\| = 1$  holds, and thus,

$$\|(\hat{g}_j)_{1:j}\| = \beta |s_1 \cdots s_{j-1}|.$$

Note that the norm of  $r_i^{app}$ , as computed in (8.14), is given by

$$\|r_i^{app}\| = \left\| (-\mu + \sigma) \hat{\beta}_i v_{i+1} e_i^T y_i(\mu) \right\| = \frac{|\underline{t}_{i+1,i}|}{|\tilde{r}_{i,i}|} |s_1 \cdots s_{i-1} \beta|,$$

where  $(-\mu + \sigma) \hat{\beta}_i = \underline{t}_{i+1,i}$ ,  $\tilde{r}_{i,i}$  denotes the entry in the  $i$ th row,  $i$ th column of  $\tilde{R}_i \in \mathbb{R}^{i \times i}$ , and we have computed the  $i$ th entry of  $y_i(\mu) \in \mathbb{R}^i$  using (8.17) and (8.18). This computation is analogous to [9, Equation (5.2)]. We define the quantity

$$(8.20) \quad \Delta_i := \frac{|\tilde{r}_{i,i}|}{|\underline{t}_{i+1,i}|} \|r_i^{app}\| = |\tilde{r}_{i,i}| |(y_i(\mu))_i| = \beta |s_1 \cdots s_{i-1}|$$

and thus obtain the bound

$$(8.21) \quad |(y_j(\mu))_i| = |\eta_i^{(j)}| \leq \frac{1}{\sigma_j(\mathbf{T}_j)} \Delta_i$$

from (8.19) and (8.20). The following theorem expresses a computable bound for the norm of the residual of the *inexact algorithm* applied to the system (3.1).

**THEOREM 8.4.** *Let  $r_i^{in}$  be the residual of the  $i$ th iterate of the Inexact Algorithm 2 applied to the linear system (1.5), and define*

$$r_i^{app} := (\mu - \sigma) \hat{\beta}_i v_{i+1} e_i^T y_i(\mu),$$

for  $i = 1, \dots, j$ , where  $\hat{\beta}_i, v_{i+1}$  are as in (8.13) and  $y_i(\mu) = [\eta_1^{(i)}, \dots, \eta_i^{(i)}] \in \mathbb{R}^i$  as in (8.10). If at each iteration  $i \leq j$  the inner residual vector  $p_i$  (8.12) satisfies

$$(8.22) \quad \|p_i\| \leq \frac{1}{j} \frac{\sigma_j(\mathbf{T}_j)}{\Delta_i} \varepsilon := \varepsilon_{inner}^{(i)},$$

with  $\Delta_i$  (8.20), then

$$\|r_i^{in}\| \leq \|r_i^{app}\| + \varepsilon.$$

*Proof.* The proof follows directly from the reasoning above paired with (8.14), i.e.,

$$\begin{aligned} \|r_j^{in}\| &\leq \|r_i^{app}\| + \|P_j y_j(\mu)\| = \|r_i^{app}\| + \left\| \sum_{i=1}^j p_i \eta_i^{(j)} \right\| \\ &\leq \|r_i^{app}\| + \sum_{i=1}^j \|p_i\| |\eta_i^{(j)}| \leq \|r_i^{app}\| + \varepsilon, \end{aligned}$$

where we have used the bounds in (8.21) and (8.22).  $\square$

**8.4. Approximation of a parameterized Helmholtz equation by the *inexact algorithm*.**

Consider approximating the solution of the parameterized Helmholtz equation in (6.1) with the Inexact Algorithm 2. In the simulation shown in Figure 8.1, the action of  $P(\sigma)^{-1}$  in the application of the preconditioner is approximated via the iterative method BiCG. Here the tolerance in the BiCG procedure, referred to as the inner tolerance, is varied at each iteration. Specifically, we set the inner tolerance at iteration  $i$ , denoted  $tol_i$ , to  $tol_1 = 10^{-14}$  and

$$(8.23) \quad tol_i = \frac{1}{|(y_{i-1}(\mu^*))_{i-1}|} \varepsilon \approx \frac{1}{\Delta_{i-1}} \varepsilon,$$

for  $i = 2, \dots, j$ , where  $(y_{i-1}(\mu^*))_{i-1}$  denotes the  $(i - 1)$ st component of  $y_{i-1}(\mu^*) \in \mathbb{R}^{i-1}$  as in (8.10a) and  $\Delta_{i-1}$  is as in (8.20). The parameter  $\mu^*$  is equal to the  $\mu_l$  furthest from the target parameter  $\sigma$ , i.e., the parameter with corresponding approximation from which we expect the slowest convergence (see (3.2)). Note that the inner tolerance as computed in (8.23) uses information from the previous iteration.

The experiment shown in Figure 8.1 was produced with one execution of the Inexact Algorithm 2. Though we display just four solutions, accessing each of the corresponding approximations of  $\mu \in [2.50, 3.50]$  requires the solution of a tridiagonal system of dimension  $j \times j$ , where  $j$  is the dimension of the subspace from which the approximates come. The relative residual norms of these solutions are below a prescribed tolerance. Figure 8.2 displays the convergence of the same simulation as a function of CPU time, omitting a comparison to an exact application of the preconditioner. In this way, we see the cost in CPU seconds of each iteration of the algorithm. The roughly constant cost of each iteration is due to the short-term recurrence feature of the method. The CPU times here are measured after the precomputation steps and, though one execution of the Inexact Algorithm 2 gives approximations for many different values of  $\mu$ , each convergence curve here corresponds to a separate run.

Theorem 8.4 provides a bound for the residual obtained from applying the *inexact algorithm* to the linearized system (3.1), under the assumption that the bound in (8.22) is fulfilled for  $i = 1, \dots, j$ . As the quantity  $\Delta_i$  is proportional to  $r_i^{app}$ , an approximation of the residual assuming exact preconditioning, we can expect to solve the inner linear systems with growing inaccuracy as the outer residual decreases. Increasing the inner tolerance as the algorithm proceeds improves the performance without destroying the accuracy of the method. For comparison, we apply the preconditioner exactly via an LU decomposition of  $P(\sigma)$  and display the convergence for a variety of values  $\mu$ .

Figure 8.3 shows the convergence of the *inexact algorithm* applied to the same discretization of the parameterized Helmholtz equation in (6.1), for larger values of the parameter  $\mu$ . Here we consider a problem of a larger dimension, where an LU decomposition of  $P(\sigma)$  is no

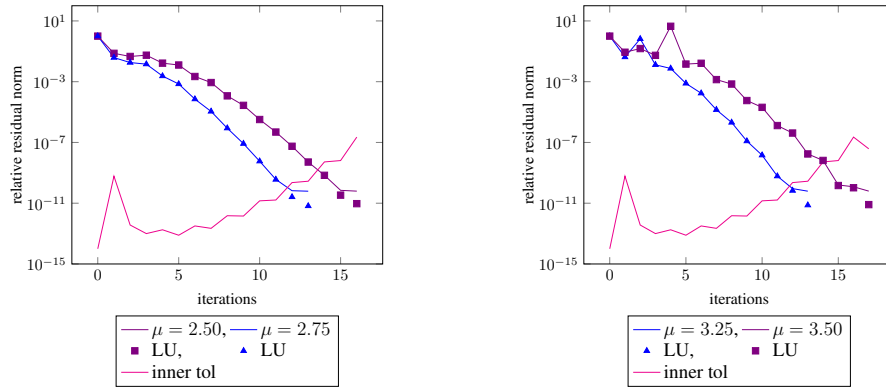


FIG. 8.1. Convergence for approximating the parameterized Helmholtz equation (6.1), generated from one (total) execution of the Inexact Algorithm 2 with  $\sigma = 3$  and evaluated for different values of  $\mu \in [-5, 5]$ . Here  $n = 243997$ ,  $d = 34$ , (outer)  $\text{tol} = 10^{-10}$ , and the relative residual norm (6.3). BiCG method with variable inner tol for application of  $P(\sigma)^{-1}$ , according to (8.23) with  $\varepsilon = 10^{-12}$ . Compare to LU of  $P(\sigma)$ .

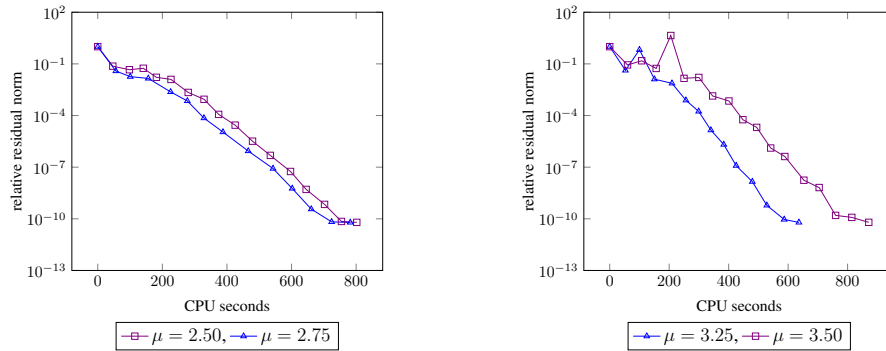
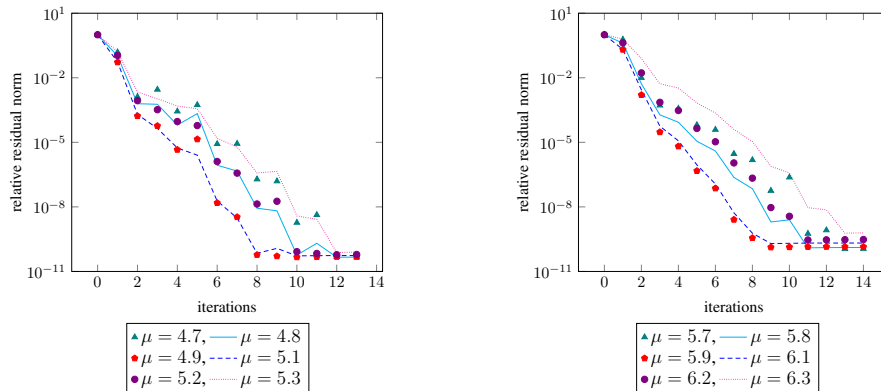


FIG. 8.2. Relative residual norm in terms of CPU sec for the simulation displayed in Figure 8.1. Note that no LU decomposition of  $P(\sigma)$  is performed.

longer feasible, and we apply the action of  $P(\sigma)^{-1}$  with the iterative method Aggregation-based algebraic multigrid<sup>3</sup> (AGMG) [37, 38, 39]. For this simulation, the inner tolerance is as described in (8.23), i.e., we solve the inner systems with increasing inaccuracy as the outer residual of the method decreases. We see that for larger values of the parameter  $\mu$ , the target parameter  $\sigma$  should be chosen closer to the values of  $\mu$  which are of interest. As before, executing the algorithm once allows us to compute the corresponding approximates for all  $\mu$  on a given interval in a cheap way, though we display just six of these solutions in the interval [4.7, 5.3] in Figure 8.3(a) and six solutions in [5.7, 6.3] in Figure 8.3(b).

In summary, the inexact framework successfully eliminates the need for an LU decomposition of the matrix  $P(\sigma)$  in the precomputing steps while maintaining the accuracy of the method for many values of  $\mu$ . The short-term recurrence property of the Inexact Algorithm 2 offers a constant cost per iteration. Thus, the *inexact algorithm* is suitable for a wide range of large-scale simulations where an LU decomposition of a matrix is not feasible.

<sup>3</sup>Yvan Notay, AGMG software and documentation; see <http://agmg.eu>.



(a) One execution of the Inexact Algorithm 2 with  $\sigma = 5$  and  $(\text{outer}) \text{ tol} = 10^{-10}$ .

(b) One execution of the Inexact Algorithm 2 with  $\sigma = 6$  and  $(\text{outer}) \text{ tol} = 10^{-9}$ .

FIG. 8.3. Convergence for approximating the parameterized Helmholtz equation (6.1), generated from the Inexact Algorithm 2 and evaluated for different values of  $\mu \in [-8, 8]$ . Here  $n = 976076$ ,  $d = 44$ , and the relative residual norm (6.3). AGMG with variable  $\text{tol}$  in application of  $P(\sigma)^{-1}$  according to (8.23) with  $\varepsilon = 10^{-12}$ . Note that no LU of  $P(\sigma)$  is feasible.

**9. Conclusions and future work.** This work proposes two variants of a novel Krylov subspace method to approximate the solution to parameterized linear systems of the form (1.1). Both algorithms return a function  $\tilde{x}_j(\mu)$  at iteration  $j$  which is cheap to evaluate for many different values of the parameter  $\mu$ . These algorithms are constructed by considering the approximate solution to a companion linearization based on an accurate Chebyshev interpolation of  $A(\mu)$ , where shift-and-invert preconditioners are used. The approximation of the resulting shifted preconditioned system is found in a shifted BiCG setting.

Here both the preconditioner and its adjoint are applied via an efficient block LU decomposition of the matrix  $(K - \sigma M)\Pi$  as in (4.3). The first variant considers exact applications of the preconditioners, and the second variant applies an approximation of the preconditioners in an inexact setting. A computable bound for the residual obtained from iterates of the *inexact method* was shown, and a contribution in the bound is directly related to the error in the application of the preconditioner. Additionally, both algorithms offer a short-term recurrence, resulting in a constant cost per iteration. Numerical results confirm that the algorithms proposed here are suitable for large-scale simulations.

The methods IDR(s) [45] and IDR(s) for shifted systems [14] have proven effective for solving nonsymmetric linear systems. Another successful short-term recurrence method for shifted systems was developed in [22], based on the method Bi-CGSTAB [50]. Using these methods to solve the linearization (3.1) would likely result in new robust methods, though further research would be needed. Furthermore, this work considered only real-valued preconditioners, though complex-valued shift-and-invert preconditioners have successfully been incorporated in several previous works for solving the Helmholtz equation; see, for instance, [17, 18]. While such a strategy would likely work here as well, it would require additional analysis.

REFERENCES

[1] M. I. AHMAD, D. B. SZYLD, AND M. B. VAN GIJZEN, *Preconditioned multishift BiCG for  $\mathcal{H}_2$ -optimal model reduction*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 401–424.  
 [2] M. S. ALNAES, J. BLECHTA, J. HAKE, A. JOHANSSON, B. KEHLET, A. LOGG, C. RICHARDSON, J. RING,

- M. E. ROGNES, AND G. N. WELLS, *The FEniCS project version 1.5*, Arch. Numer. Softw., 3 (2015).  
<https://doi.org/10.11588/ans.2015.100.20553>
- [3] A. AMIRASLANI, R. M. CORLESS, AND P. LANCASTER, *Linearization of matrix polynomials expressed in polynomial bases*, IMA J. Numer. Anal., 29 (2009), pp. 141–157.
  - [4] O. AXELSSON AND P. S. VASSILEVSKI, *A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 625–644.
  - [5] T. BAKHOS, P. K. KITANIDIS, S. LADENHEIM, A. K. SAIBABA, AND D. B. SZYLD, *Multipreconditioned GMRES for shifted systems*, SIAM J. Sci. Comput., 39 (2017), pp. S222–S247.
  - [6] M. BAUMANN AND M. B. VAN GIJZEN, *Nested Krylov methods for shifted linear systems*, SIAM J. Sci. Comput., 37 (2015), pp. S90–S112.
  - [7] ———, *An efficient two-level preconditioner for multi-frequency wave propagation problems*, Appl. Numer. Math., 135 (2019), pp. 316–332.
  - [8] A. BAYLISS, C. I. GOLDSTEIN, AND E. TURKEL, *An iterative method for the Helmholtz equation*, J. Comput. Phys., 49 (1983), pp. 443–457.
  - [9] P. N. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.
  - [10] A. CHAPMAN AND Y. SAAD, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.
  - [11] S. CORRENTY, E. JARLEBRING, AND K. M. SOODHALTER, *Preconditioned infinite GMRES for parameterized linear systems*, Preprint on arXiv, 2022. Accepted for Publication in SISC.  
<https://arxiv.org/abs/2206.05153>.
  - [12] E. DE STURLER, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.
  - [13] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
  - [14] L. DU, T. SOGABE, AND S.-L. ZHANG, *IDR(s) for solving shifted nonsymmetric linear systems*, J. Comput. Appl. Math., 274 (2015), pp. 35–43.
  - [15] C. EFFENBERGER AND D. KRESSNER, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT Numer. Math., 52 (2012), pp. 933–951.
  - [16] H. ELMAN, O. ERNST, AND D. O’LEARY, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM J. Sci. Comput., 23 (2001), pp. 1291–1315.
  - [17] Y. A. ERLANGGA, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Arch. Comput. Methods Eng., 15 (2008), pp. 37–66.
  - [18] Y. A. ERLANGGA, C. VUIK, AND C. W. OOSTERLEE, *On a class of preconditioners for solving the Helmholtz equation*, Appl. Numer. Math., 50 (2004), pp. 409–425.
  - [19] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, G. A. Watson, ed., Lecture Notes in Math., Vol. 506, Springer, Berlin, 1976, pp. 73–89.
  - [20] R. W. FREUND, *Solution of shifted linear systems by quasi-minimal residual iterations*, in Numerical Linear Algebra, L. Reichel, A. Ruttan, and R. S. Varga, eds., de Gruyter, Berlin, 1993, pp. 101–121.
  - [21] R. W. FREUND AND N. M. NACHTIGAL, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. Sci. Comput., 15 (1994), pp. 313–337.
  - [22] A. FROMMER, *BiCGStab(l) for families of shifted linear systems*, Computing, 70 (2003), pp. 87–109.
  - [23] A. FROMMER AND U. GLÄSSNER, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26.
  - [24] A. FROMMER AND P. MAASS, *Fast CG-based methods for Tikhonov-Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.
  - [25] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
  - [26] G.-D. GU AND V. SIMONCINI, *Numerical solution of parameter-dependent linear systems*, Numer. Linear Algebra Appl., 12 (2005), pp. 923–940.
  - [27] K. GU, V. L. KHARITONOV, AND J. CHEN, *Stability of Time-Delay Systems*, Birkhäuser, Boston, 2003.
  - [28] E. JARLEBRING AND S. CORRENTY, *Infinite GMRES for parameterized linear systems*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 1382–1405.
  - [29] M. E. KILMER AND D. P. O’LEARY, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221.
  - [30] D. KRESSNER AND J. E. ROMAN, *Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis*, Numer. Linear Algebra Appl., 21 (2014), pp. 569–588.
  - [31] D. KRESSNER AND C. TOBLER, *Low-rank tensor Krylov subspace methods for parameterized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.
  - [32] P. KÜRSCHNER AND M. A. FREITAG, *Inexact methods for the low rank solution to large scale Lyapunov equations*, BIT, 60 (2020), pp. 1221–1259.
  - [33] C. LANCZOS, *Solution of systems of linear equations by minimized-iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

- [34] D. S. MACKEY, N. MACKEY, C. MEHL, AND V. MEHRMANN, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.
- [35] W. MICHIELS, E. JARLEBRING, AND K. MEERBERGEN, *Krylov-based model order reduction of time-delay systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1399–1421.
- [36] W. MICHIELS AND S.-I. NICULESCU, *Stability and Stabilization of Time-Delay Systems*, SIAM, Philadelphia, 2007.
- [37] A. NAPOV AND Y. NOTAY, *An algebraic multigrid method with guaranteed convergence rate*, SIAM J. Sci. Comput., 34 (2012), pp. A1079–A1109.
- [38] Y. NOTAY, *An aggregation-based algebraic multigrid method*, Electron. Trans. Numer. Anal., 37 (2010), pp. 123–146.  
<https://etna.ricam.oeaw.ac.at/vol.37.2010/pp123-146.dir/pp123-146.pdf>
- [39] ———, *Aggregation-based algebraic multigrid for convection-diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2288–A2316.
- [40] M. L. PARKS, E. DE STURLER, G. MACKEY, D. D. JOHNSON, AND S. MAITI, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
- [41] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
- [42] ———, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
- [43] V. SIMONCINI AND D. B. SZYLD, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2002), pp. 2219–2239.
- [44] ———, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.
- [45] P. SONNEVELD AND M. B. VAN GIJZEN, *IDR( $s$ ): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Comput., 31 (2008/09), pp. 1035–1062.
- [46] K. M. SOODHALTER, D. B. SZYLD, AND F. XUE, *Krylov subspace recycling for sequences of shifted linear systems*, Appl. Numer. Math., 81 (2014), pp. 105–118.
- [47] G. STRANG, *The discrete cosine transform*, SIAM Rev., 41 (1999), pp. 135–147.
- [48] D. B. SZYLD AND J. A. VOGEL, *FQMR: a flexible quasi-minimal residual method with inexact preconditioning*, SIAM J. Sci. Comput., 23 (2001), pp. 363–380.
- [49] J. VAN DEN ESHOF, G. L. G. SLEIJPEN, AND M. B. VAN GIJZEN, *Iterative linear system solvers with approximate matrix-vector products*, in QCD and Numerical Analysis III, A. Boriçi, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, eds., vol. 47 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2005, pp. 133–142.
- [50] 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.
- [51] J. A. VOGEL, *Flexible BiCG and flexible Bi-CGSTAB for nonsymmetric linear systems*, Appl. Math. Comput., 188 (2007), pp. 226–233.
- [52] C. VUIK, *Further experiences with GMRESR*, Supercomputer, 55 (1993), pp. 13–27.