Electronic Transactions on Numerical Analysis.
Volume 51, pp. 240–261, 2019.
Copyright © 2019, Kent State University.
ISSN 1068–9613.
DOI: 10.1553/etna_vol51s240

ETNA
Kent State University and
Johann Radon Institute (RICAM)

APPROXIMATE RESIDUAL-MINIMIZING SHIFT PARAMETERS FOR THE LOW-RANK ADI ITERATION[*]

PATRICK KÜRSCHNER[†]

Abstract. The low-rank alternating directions implicit (LR-ADI) iteration is a frequently employed method for efficiently computing low-rank approximate solutions of large-scale Lyapunov equations. In order to achieve a rapid error reduction, the iteration requires shift parameters whose selection and generation is often a difficult task, especially for nonsymmetric matrices in the Lyapunov equation. This article represents a follow up of Benner et al. [Electron. Trans. Numer. Anal., 43 (2014–2015), pp. 142–162] and investigates self-generating shift parameters based on a minimization principle for the Lyapunov residual norm. Since the involved objective functions are too expensive to evaluate and hence intractable, objective functions are introduced which are efficiently constructed from the available data generated by the LR-ADI iteration. Several numerical experiments indicate that these residual-minimizing shifts using approximated objective functions outperform existing precomputed and dynamic shift parameter selection techniques, although their generation is more involved.

Key words. Lyapunov equation, alternating directions implicit, low-rank approximation, shift parameters

AMS subject classifications. 15A06, 65F10, 65F30

1. Introduction.

In this paper, we study the numerical solution of large-scale, continuous-time, algebraic Lyapunov equations (CALE)

$$AX + XA^* + BB^* = 0 \tag{1.1}$$

defined by matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$, $s \ll n$, and where $X \in \mathbb{R}^{n\times n}$ is the sought solution. For large sizes n of the problem, directly computing and storing X is infeasible. For dealing with (1.1), it has become common practice to approximate X by a low-rank factorization $X \approx ZZ^*$ with $Z \in \mathbb{R}^{n\times r}$, $\operatorname{rank} Z = r \ll n$. Theoretical evidence for the existence of such low-rank approximations can be found, e.g., in [2, 4, 21, 42]. The low-rank solution factor Z can be computed by iterative methods employing techniques from large-scale numerical linear algebra. Projection-based methods utilizing extended or rational Krylov subspaces, and the low-rank alternating directions implicit (LR-ADI) iteration belongs to the most successful and often used representatives of iterative low-rank methods for (1.1); see, e.g., [8, 10, 16, 17, 30, 44].

Here, we focus on the LR-ADI iteration and a particular important issue thereof. One of the biggest reservations against the LR-ADI iteration is its dependence on certain parameters called shifts, which steer the convergence rate of the iteration. For large problems, especially those defined by nonsymmetric matrices A, generating these shift parameters is a difficult task and often only suboptimal or heuristic shift selection approaches can be employed. In [9], a shift generation approach was proposed where the shifts are chosen dynamically in the course of the LR-ADI iteration and are based on minimizing the Lyapunov residual norm. Unfortunately, although potentially leading to very good shifts, this approach constitutes in its original form only a theoretical concept because employing it is numerically very expensive and thus unusable in practice. This article follows up on [9] and investigates several aspects and modifications of the residual minimization-based shift selection. The main goal is a numerically feasible and efficient generation of high quality shift parameters for the LR-ADI iteration that are based on the residual minimization principle.

[*] Received November 11, 2018. Accepted August 21, 2019. Published online on September 10, 2019. Recommended by D. Szyld.

[†] Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium and Department of Electrical Engineering ESAT/STADIUS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium (patrick.kurschner@kuleuven.be).

1.1. Notation. $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex numbers, and $\mathbb{R}_-$, $\mathbb{C}_-$ refer to the set of strictly negative real numbers and the open left half plane. In the matrix case, $\mathbb{R}^{n\times m}$, $\mathbb{C}^{n\times m}$ denote $n\times m$ real and complex matrices, respectively. For a complex quantity $X=\operatorname{Re}(X)+\jmath\operatorname{Im}(X)$, $\operatorname{Re}(X)$, $\operatorname{Im}(X)$ are its real and imaginary parts, and $\jmath$ is the imaginary unit. The complex conjugate of X is denoted by $\overline{X}$, and $|\xi|$ is the absolute value of $\xi\in\mathbb{C}$. If not stated otherwise, $\|\cdot\|$ is the Euclidean vector- or subordinate matrix norm (spectral norm). The matrix A^* is the transpose of a real or the complex conjugate transpose of a complex matrix A, A^{-1} is the inverse of a nonsingular matrix A, and $A^{-*}=(A^*)^{-1}$. The identity matrix of dimension n is indicated by I_n, and $\mathbf{1}_n:=(1,\dots,1)^T\in\mathbb{R}^n$ is the vector of ones. The spectrum of a matrix A is given by $\Lambda(A)$, and the spectral radius is defined as $\rho(A):=\max\{|\lambda|,\ \lambda\in\Lambda(A)\}$. The symbol $\otimes$ denotes the Kronecker product.

For a multivariate function $f(x_1,\dots,x_d)\ :\ \mathbb{R}^d\mapsto\mathbb{R}$, we employ the typical shorthand notation $f_{x_i}=\frac{\partial f}{\partial x_i}$ and $f_{x_ix_j}=\frac{\partial^2 f}{\partial x_i\partial x_j}$ for the first- and second-order partial derivatives, accumulated in the gradient $\operatorname{grad} f=[f_{x_i}]$ and the Hessian $\operatorname{grad}^2=[f_{x_ix_j}]$, respectively. For a vector-valued function $F(x_1,\dots,x_d)=[f_1,\dots,f_v]^T$, the Jacobian is given by $[\frac{\partial f_i}{\partial x_j}]$. For complex functions $g(z_1,\dots,z_d,\overline{z_1},\dots,\overline{z_d})$ depending on d complex variables and their conjugates, the Wirtinger calculus [38] is used to define complex and complex conjugate derivatives: $\frac{\partial g}{\partial z_j}=\frac{1}{2}\big(\frac{\partial g}{\partial\operatorname{Re}(z_i)}-\jmath\frac{\partial g}{\partial\operatorname{Im}(z_i)}\big)$, $\frac{\partial g}{\partial\overline{z_j}}=\frac{1}{2}\big(\frac{\partial g}{\partial\operatorname{Re}(z_i)}+\jmath\frac{\partial g}{\partial\operatorname{Im}(z_i)}\big)$.

1.2. Problem assumptions. Throughout the article we assume that $\Lambda(A)\subset\mathbb{C}_-$, which ensures a unique positive semidefinite solution X of (1.1). To permit low-rank approximations of X, we shall assume that $s\ll n$. Moreover, we assume that we are able to efficiently solve linear systems of equations of the form $(A+\alpha I)x=b$, $\alpha\in\mathbb{C}$, by either iterative or sparse-direct solvers, where we restrict ourselves for the sake of brevity to the latter type of solvers.

1.3. Overview of this article. We begin by reviewing the low-rank ADI iteration in Section 2, including important structural properties of the method and a brief recapitulation of the ADI shift parameter problem. The residual norm-minimizing shift parameters are discussed in depth in Section 3, where our main contribution, a numerically efficient approach to obtain those shifts, is presented. The main building block is the replacement of the expensive to evaluate and intractable objective functions by approximations that are constructed from the already computed data. Along the way, extensions to the generalized Lyapunov equations

$$AXM^*+MXA^*+BB^*=0,\tag{1.2}$$

with nonsingular $M\in\mathbb{R}^{n\times n}$ will be discussed. Section 4 extends these ideas to the generation of a single shift for the use in more than one LR-ADI iteration steps, which can further reduce the computation times. A series of numerical experiments are given in Section 5, evaluating the performance of the proposed shift generation machinery in comparison with existing selection strategies. Comparisons with other low-rank algorithms for (1.1) are also presented. Section 6 concludes the paper and provides some future research directions.

2. Review of the low-rank ADI iteration.

The low-rank ADI iteration can be derived from the nonstationary iteration

$$\begin{aligned}X_j&=(A-\overline{\alpha_j}I)(A+\alpha_jI)^{-1}X_{j-1}(A+\alpha_jI)^{-*}(A-\overline{\alpha_j}I)^*\\&\quad-2\operatorname{Re}(\alpha_j)(A+\alpha_jI)^{-1}BB^*(A+\alpha_jI)^{-*},\qquad j\geq 1,\ X_0\in\mathbb{R}^{n\times n},\end{aligned}$$

for the CALE (1.1). There, $\alpha_i\in\mathbb{C}_-$, $i=1,\dots,j$, are the previously mentioned shift parameters discussed further in Section 2.1. By introducing the low-rank approximations

Algorithm 1: LR-ADI iteration for computing low-rank solution factors.

```
   Input  : Matrices $A,\ B$ defining (1.1), tolerance $0 < \tau \ll 1$.
   Output : $Z_j \in \mathbb{C}^{n\times sj}$, such that $ZZ^* \approx X$.
1  $W_0 = B,\quad Z_0 = [\,],\quad j = 1$, choose $\alpha_1 \in \mathbb{C}_-$.
2  while $\|W_{j-1}^* W_{j-1}\| \geq \tau \|B^*B\|$ do
3      Solve $(A + \alpha_j I)V_j = W_{j-1}$ for $V_j$.
4      $W_j = W_{j-1} - 2\operatorname{Re}(\alpha_j)V_j$.
5      $Z_j = [Z_{j-1},\ \sqrt{-2\operatorname{Re}(\alpha_j)}V_j]$.
6      Select next shift $\alpha_{j+1} \in \mathbb{C}_-$.
7      $j = j + 1$.
```

$X_j = Z_j Z_j^*$ in each step and assuming $Z_0 = 0$, the above iteration can be rearranged [8, 25, 30, 40] into the low-rank ADI iteration illustrated in Algorithm 1.

For the Lyapunov residual matrix regarding the approximate solution $X_j = Z_j Z_j^*$, we have the following result.

THEOREM 2.1 ([8, 52]). *Assume that j steps of the LR-ADI iteration with the shift parameters $\{\alpha_1, \dots, \alpha_j\} \subset \mathbb{C}_-$ have been applied to* (1.1)*. Then the Lyapunov residual matrix can be factorized via*

$$R_j = AZ_jZ_j^* + Z_jZ_j^*A^* + BB^* = W_jW_j^*, \tag{2.1}$$

where the residual factors $W_j \in \mathbb{C}^{n\times s}$ are given by

$$W_j := (A - \overline{\alpha_j}I)V_j = W_{j-1} - 2\operatorname{Re}(\alpha_j)V_j = W_0 + Z_jG_j, \tag{2.2}$$

with $W_0 := B$, $V_j = (A + \alpha_j I)^{-1}W_{j-1}$, and $G_j := [\gamma_1, \dots, \gamma_j]^* \otimes I_s \in \mathbb{R}^{js\times s}$, where $\gamma_i := \sqrt{-2\operatorname{Re}(\alpha_i)}$, for $i = 1, \dots, j$.

The residual factors $W_j \in \mathbb{C}^{n\times s}$ will play a very important role in this article. As already indicated in line 2 in Algorithm 1, the residual factorization (2.1) greatly helps to cheaply compute the norm of the residual matrix which is useful as a stopping criterion. The low-rank solution factors Z_j generated by the LR-ADI iteration solve certain Sylvester equations. Similar results regarding an older version of the LR-ADI iteration can be found in [29, 30].

COROLLARY 2.2 ([25, Corollary 3.9], [51, Lemma 5.12], [52, Lemma 3.1]). *With the same assumptions and notations as in Theorem 2.1, the low-rank factor Z_j after j steps of the LR-ADI iteration (Algorithm 1) satisfies the Sylvester equations*

$$AZ_j - Z_jS_j = BG_j^*, \tag{2.3a}$$

$$AZ_j + Z_j\overline{S}_j^* = W_jG_j^*, \tag{2.3b}$$

where

$$S_j := \begin{bmatrix} \alpha_1 & \gamma_1\gamma_2 & \cdots & \gamma_1\gamma_j \\ & \ddots & \ddots & \vdots \\ & & \ddots & \gamma_{j-1}\gamma_j \\ & & & \alpha_j \end{bmatrix} \otimes I_s \in \mathbb{C}^{js\times js}.$$

REMARK 2.3. In practice, although (1.1) is defined by real A and B, complex shift parameters can occur. We assume that the set of shifts $\{\alpha_1, \dots, \alpha_j\}$ are closed under complex conjugation and that pairs of complex conjugated shifts occur subsequently, i.e., $\alpha_{i+1} = \overline{\alpha_i}$ if $\mathrm{Im}(\alpha_i) \neq 0$. These complex parameters pairs are in practice dealt within the LR-ADI iteration by a double-step fashion [7, 9, 25] resulting in real low-rank factors Z_j and, important for this study, real low-rank residual factors W_j. Real versions of the above results can be established, but for brevity and clarity we keep the shorter complex versions in the remainder. The real version of the LR-ADI iteration will nevertheless be used in the numerical experiments in the end.

2.1. Shift parameters. The approximation error $X - X_j$ and the residual R_j can be expressed as

$$X - X_j = \mathcal{M}_j(X - X_0)\mathcal{M}_j^*, \quad R_j = \mathcal{M}_j R_0 \mathcal{M}_j^*,$$

$$\text{with} \quad \mathcal{M}_j = \prod_{i=1}^{j} \mathcal{C}(A, \alpha_i), \quad \text{and} \quad \mathcal{C}(A, \alpha) := (A - \overline{\alpha} I)(A + \alpha I)^{-1}$$

is a Cayley transformation of A. Taking norms leads to

$$\frac{\|X - X_j\|}{\|X - X_0\|} \leq c\rho(\mathcal{M}_j)^2, \qquad \frac{\|R_j\|}{\|R_0\|} \leq c\rho(\mathcal{M}_j)^2,$$

where $c \geq 1$ is the squared condition number[1] of the eigenvector matrix of A. Because of $\Lambda(A) \subset \mathbb{C}_-$ as well as $\alpha_i \in \mathbb{C}_-$, it holds that $\rho(\mathcal{C}(A, \alpha_i)) < 1$, for $i = 1, \ldots, j$, and consequently, $\rho(\mathcal{M}_j) < 1$ is getting smaller as the ADI iteration proceeds. This motivates to select the shifts α_i such that $\rho(\mathcal{M}_j)$ is as small as possible leading to the ADI parameter problem

$$\min_{\alpha_1, \ldots, \alpha_j \in \mathbb{C}_-} \left(\max_{\lambda \in \Lambda(A)} |\mu_j(\lambda)| \right), \qquad \mu_j(\lambda) := \prod_{i=1}^{j} \frac{\lambda - \overline{\alpha_i}}{\lambda + \alpha_i}. \tag{2.4}$$

Several shift selection strategies have been developed based on (2.4), e.g., the often used Wachspress [42, 49] and Penzl [36] selection approaches, which precompute a number of shifts before the actual LR-ADI iteration. There, the spectrum $\Lambda(A)$ in (2.4) is replaced by an easy to compute approximation, typically using a small number of approximate eigenvalues generated by Arnoldi and inverse Arnoldi processes. The shifts are then obtained by means of elliptic functions in the Wachspress approach [42, 49] and, respectively, heuristically in the Penzl approach [36]. Starting from (2.4) for selecting shifts has, however, some shortcomings. A disadvantage from the conceptual side is that the min–max problem (2.4) does only take (approximate) eigenvalues of A into account. No information regarding the inhomogeneity BB^* of the CALE (1.1) is incorporated although the low-rank property of BB^* is one significant factor for the singular value decay of the solution and hence for the existence of low-rank approximations [2, 21, 47]. Furthermore, no information regarding the eigenvectors of A enters (2.4). While this might not be a big issue for CALEs defined by symmetric matrices, in the nonsymmetric case the spectrum alone might not be enough to fully explain the singular value decay of the solution; see, e.g., the discussions in [3, 42].

Because only approximate eigenvalues can be used for large-scale problems, Wachspress and Penzl shift strategies can also suffer from poor eigenvalue estimates [42], and the cardinality of the set of approximate eigenvalues (Ritz values) is an unknown quantity the user has to

[1] Alternatively, if the field of values of A lies in $\mathbb{C}_-$, one can use $c = (1 + \sqrt{2})^2$ and replace ρ by the squared maximal magnitude that the rational function μ_j in (2.4) assumes on the field of values; see [13].

provide in advance. Even tiny changes in these quantities can greatly alter the speed of the error or the residual reduction in the ADI iteration. Because the strategies based on (2.4) are usually in general carried out in advance, i.e., shifts are generated before the actual iteration, no information about the current progress of the iteration is incorporated.

Here, we are interested in adaptive shift selection and generation strategies that circumvent these issues. Our goal is that these approaches take the current stage of the iteration into account and that the shifts are generated automatically and in a numerically efficient way during the iteration, i.e., the shift computation should consume only a small fraction of the total numerical effort of the LR-ADI iteration. Next, we review commonly used existing dynamic shift selection approaches and propose some enhancements.

2.1.1. Ritz value-based dynamic shifts. First steps regarding dynamic shift approaches were made in [9] by using Ritz values of A with respect to a subspace $\mathcal{Q}_\ell = \operatorname{range}(Q_\ell) \subseteq \operatorname{range}(Z_j)$, where $Q_\ell \in \mathbb{R}^{n\times\ell}$ has orthonormal columns. The typical choice is to select the most recent block columns of Z_j for spanning $\mathcal{Q}_\ell$:

$$\mathcal{Q}_\ell = \mathcal{Z}(h) \coloneqq \operatorname{range}\left([V_{j-h+1},\ldots,V_j]\right), \tag{2.5}$$

with $h = 1,\ldots,j$, to keep the space dimension small. The Ritz values are given by $\Lambda(H_\ell)$ with $H_\ell \coloneqq Q_\ell^* A Q_\ell$ and can, e.g., be plugged into the Penzl heuristic to select $g \le \ell$ shift parameters. It can happen that $\Lambda(H_\ell)\cap\mathbb{C}_+ \neq \emptyset$ in which case we simple negate all unstable Ritz values. Once these g shifts have been used, the generation and selection process is repeated with Z_{j+g}. Despite its simplicity, this idea already led to a significant speedup of the LR-ADI iteration, in particular for nonsymmetric problems where the a priori computed shifts resulted in a very slow convergence. This approach is the default shift selection routine in the M-M.E.S.S. software package [41]. Further details on an efficient construction of H_ℓ are given later. This basic selection strategy can be modified in the following ways.

2.1.2. Convex hull-based shifts. Motivated by the connection of LR-ADI to rational Krylov subspaces [16, 18, 29, 51, 52] we can borrow the greedy shift selection strategy from [17] which was developed for the rational Krylov subspace method for (1.1). Let $\mathcal{S}\subset\mathbb{C}_-$ be the convex hull of the set of Ritz values $\Lambda(H_\ell)$ and $\partial\mathcal{S}$ its boundary. For a discrete subset $\mathcal{D}\subset\partial\mathcal{S}$ one tries to heuristically find $\alpha\in\mathcal{D}$ that reduces the magnitude of the rational function (cf. (2.4)) connected to the previous LR-ADI steps the most. In contrast to the Ritz value-based shift selection discussed above, the convex hull-based selection will only provide a single shift parameter to be used for the next iteration, and thus, the selection process has to be executed in every iteration step. Note that this approach employed in RKSM uses the Ritz values associated with the full already computed rational Krylov subspace, while in LR-ADI we only use a smaller subspace (2.5).

2.1.3. Residual-Hamiltonian-based shifts. Both strategies mentioned so far select shift parameters on the basis of the eigenvalues of a compressed version H_ℓ of A. A different modification developed for the RADI method [5, 6] for algebraic Riccati equations also takes some eigenvector information into account. For Riccati equations, the core idea is to consider a projected version of the associated Hamiltonian matrix which we can simplify for CALEs. If $[P^*, Q^*]^*$ spans the stable n-dimensional invariant subspace of

$$\mathcal{H}_0 \coloneqq \begin{bmatrix} A^* & 0 \\ BB^* & -A \end{bmatrix},$$

then $X = PQ^{-1}$ solves (1.1). Let $X_j \approx X$ be obtained by LR-ADI, then all later steps can be seen as the application of LR-ADI to the residual Lyapunov equations $A\hat{X} + \hat{X}A^T = -R_j$

(see [25, Corollary 3.8]), where $R_j = W_jW_j^*$ is the residual associated with X_j. The residual equations are connected to the Hamiltonian matrices $\mathcal{H}_j := \begin{bmatrix} A^* & 0 \\ W_jW_j^* & -A \end{bmatrix}$. Following the same motivation as in [6], we set up the projected Hamiltonian $\tilde{\mathcal{H}}_{j,\ell} := \begin{bmatrix} H_\ell^* & 0 \\ Q_\ell^* W_j W_j^* Q_\ell & -H_\ell \end{bmatrix}$, compute its stable eigenvalues λ_k and associated eigenvectors $\begin{bmatrix} p_k \\ q_k \end{bmatrix}$, $p_k,\ q_k \in \mathbb{C}^\ell$, and select the eigenvalue λ_k with the largest $\|q_k\|$ as next ADI shift. As in the convex hull-based selection, this approach delivers only a single shift each time.

3. Residual norm-minimizing shifts.

In this section we discuss the main focus of this study: the shift selection strategy originally proposed in [9], where the objective is to find shift parameters that explicitly minimize the Lyapunov residual norm. Assume that step j of the LR-ADI iteration has been completed and that the associated residual factor W_j is a real $n \times s$ matrix (cf. Remark 2.3). By Theorem 2.1 it holds for the next Lyapunov residual that $\|R_{j+1}\| = \|W_{j+1}\|^2$ with

$$W_{j+1} = W_{j+1}(\alpha_{j+1}) = \mathcal{C}(A, \alpha_{j+1})W_j = W_j - 2\operatorname{Re}(\alpha_{j+1})\left((A + \alpha_{j+1}I)^{-1}W_j\right).$$

This motivates to determine the parameter $\alpha_{j+1} \in \mathbb{C}_-$ such that the Lyapunov residual norm is reduced the most from step j to $j+1$. This can be formulated as, e.g., a complex nonlinear least-squares problem (NLS)

$$\begin{aligned} \alpha_{j+1} &= \operatorname*{argmin}_{\alpha \in \mathbb{C}_-} \frac{1}{2}\|\Psi_j(\alpha, \overline{\alpha})\|^2, \\ \Psi_j(\alpha, \overline{\alpha}) &= \mathcal{C}(A, \alpha)W_j = (A - \overline{\alpha}I)(A + \alpha I)^{-1}W_j. \end{aligned} \tag{3.1}$$

The complex function $\Psi_j(\alpha, \overline{\alpha}) : \mathbb{C} \mapsto \mathbb{C}^{n \times s}$ is obviously not analytic in the complex variables α, $\overline{\alpha}$ alone but in the full variable $(\alpha, \overline{\alpha})$, a property typically referred to as polyanalyticity. Furthermore, the residual-minimizing approach can also be considered via the real-valued function $\psi_j^{\mathbb{C}} = \|W_{j+1}\|^2$, leading to the complex minimization problem

$$\alpha_j = \operatorname*{argmin}_{\alpha_j \in \mathbb{C}_-} \psi_j^{\mathbb{C}}(\alpha, \overline{\alpha}), \qquad \psi_j^{\mathbb{C}}(\alpha, \overline{\alpha}) := \|\Psi_j(\alpha, \overline{\alpha})\|^2, \tag{3.2}$$

which corresponds to the original formulation for the residual-minimizing shifts [9].

It is clear that (3.1) and (3.2) essentially encode the same optimization task but differences will occur in the numerical treatment of both formulations. Using $\alpha = \nu + \jmath\xi$ with $0 > \nu \in \mathbb{R}$, $\xi \in \mathbb{R}$, yields that real and imaginary parts of the next shift $\alpha_{j+1} = \nu_{j+1} + \jmath\xi_{j+1}$ can be obtained from solving

$$\begin{aligned} [\nu_{j+1}, \xi_{j+1}] &= \operatorname*{argmin}_{\nu \in \mathbb{R}_-, \xi \in \mathbb{R}} \psi_j(\nu,\ \xi), \\ \psi_j = \psi_j(\nu,\ \xi) &:= \|W_j - 2\nu\left((A + (\nu + \jmath\xi)I)^{-1}W_j\right)\|^2. \end{aligned} \tag{3.3}$$

Since $\|X\|_2^2 = \lambda_{\max}(X^*X)$, the minimization problems (3.2) and (3.3) can also be understood as eigenvalue optimization problems if $s > 1$.

Naturally, if one knows that real shift parameters are sufficient, e.g., when $A = A^*$, then the above minimization problems simplify in the obvious manner by restricting the optimization to $\mathbb{R}_-$. For achieving a reduction of the residual as well as avoiding the singularities at $-\Lambda(A) \subset \mathbb{C}_+$, the constraint $\nu < 0$ is mandatory. Originally, an unconstrained version of (3.3) and derivative-free methods were used in [9], which turned out to be unreliable because, in particular, unusable shifts ($\nu \geq 0$) were frequently generated. In this article, we employ constrained, derivative-based optimization approaches using the complex nonlinear least-squares (3.1) and the real-valued minimization problem (3.3). The underlying objective functions are generally not convex and have potentially more than one minimum in $\mathbb{C}_-$. Here, we will pursue only the detection of local minima because any parameter $\alpha \in \mathbb{C}_-$ will yield at least some reduction of the CALE residual norm such that the substantially larger numerical effort to compute global minima will hardly pay off. The next section gives the structure of the required derivatives of $\Psi_j(\alpha, \overline{\alpha})$, $\psi_j^{\mathbb{R}}$. Afterwards, numerical aspects such as approximating the objective functions, solving the minimization or least-squares problems, and implementing the proposed shift generation framework efficiently in Algorithm 1 are discussed.

3.1. Derivatives of the objective functions. For the least-squares problem (3.1), the Jacobian, and conjugate Jacobian [45] of $\Psi_j(\alpha, \overline{\alpha})$ are

$$\begin{aligned}\frac{\partial \Psi_j(\alpha,\overline{\alpha})}{\alpha} &= -(A-\overline{\alpha}I)(A+\alpha I)^{-2}W_j = \mathcal{C}(A,\alpha)(A+\alpha I)^{-1}W_j,\\ \frac{\partial \Psi_j(\alpha,\overline{\alpha})}{\overline{\alpha}} &= -(A+\alpha I)^{-1}W_j.\end{aligned}$$

The structure of the derivatives for ψ_j in (3.3) is more complicated.

THEOREM 3.1 (Gradient and Hessian of the objective function (3.3)). *Assume that $\alpha = \nu + \jmath\xi \in \mathbb{C}_-$, $W = W_j \in \mathbb{R}^{n\times s}$, and define $L(\nu,\xi) := A + \alpha I$,*

$$\begin{aligned}&S^{(i)} := (L(\nu,\xi)^{-1})^i W, \qquad W_\alpha^{(i)} := S^{(i)} - 2\nu S^{(i+1)}, \qquad \hat{W}^{(i)} := S^{(i)} - \nu S^{(i+1)},\\ &\tilde{R}_\nu := -(W_\alpha^{(0)})^* \hat{W}^{(1)}, \qquad \tilde{R}_\xi := (W_\alpha^{(0)})^* S^{(2)},\end{aligned}$$

for $i = 0, \ldots, 3$. Assume that $(W_\alpha^{(0)})^* W_\alpha^{(0)}$ has s distinct eigenvalues $\theta_1 > \ldots > \theta_s > 0$ and let $(\theta_\ell\,, u_\ell) = (\theta_\ell(\nu,\ \xi)\,, u_\ell(\nu,\ \xi))$ with $\|u_\ell\| = 1$, $\ell = 1, \ldots, s$, be its eigenpairs. Then, the gradient and Hessian of (3.3) are given by

$$\operatorname{grad}\psi_j = 4\begin{bmatrix}\operatorname{Re}(u_1^*\left((W_\alpha^{(0)})^*\hat{W}^{(1)}\right)u_1)\\ -\nu\operatorname{Im}(u_1^*\left((W_\alpha^{(0)})^*S^{(2)}\right)u_1)\end{bmatrix} = 4\begin{bmatrix}\operatorname{Re}(u_1^*\tilde{R}_\nu u_1)\\ -\nu\operatorname{Im}(u_1^*\tilde{R}_\xi u_1)\end{bmatrix},$$

and

$$\begin{aligned}\operatorname{grad}^2\psi_j = 8&\begin{bmatrix}\operatorname{Re}(u_1^*\left((\hat{W}^{(2)})^*W_\alpha^{(0)} + (\hat{W}^{(1)})^*\hat{W}^{(1)}\right)u_1) & h_{12}\\ h_{12} & \nu\operatorname{Re}(u_1^*((S^{(3)})^*W_\alpha^{(0)} + \nu(S^{(2)})^*S^{(2)})u_1)\end{bmatrix}\\ &+\sum_{k=2}^{s}\tfrac{8}{\theta_1-\theta_k}\begin{bmatrix}\left|u_1^*(\tilde{R}_\nu^* + \tilde{R}_\nu)u_k\right|^2 & \tilde{h}_{12}^{(k)}\\ \tilde{h}_{12}^{(k)} & \left|u_1^*(\tilde{R}_\xi^* - \tilde{R}_\xi)u_k\right|^2\end{bmatrix},\end{aligned}\tag{3.4}$$

where

$$\begin{aligned}h_{12} &:= \frac{1}{2}\operatorname{Im}(u_1^*\left((W_\alpha^{(2)})^*W_\alpha^{(0)} - 2\nu(S^{(2)})^*W_\alpha^{(1)}\right)u_1),\\ \tilde{h}_{12}^{(k)} &:= -\operatorname{Re}((u_1^*(\tilde{R}_\nu^* + \tilde{R}_\nu)u_k)(\jmath\nu u_k^*(\tilde{R}_\xi^* - \tilde{R}_\xi)u_1)).\end{aligned}$$

Proof. The results are obtained by building the partial derivatives of $\hat{\mathcal{C}}(A, \alpha = \nu + \jmath\xi)$ and of $\psi_j(\nu,\ \xi) = \sigma_{\max}^2\left(\mathcal{C}(A, \nu + \jmath\xi)W\right) = \lambda_{\max}\left(\Psi^*\Psi\right) = \lambda_{\max}\left((W_\alpha^{(0)})^*W_\alpha^{(0)}\right)$ using results on derivatives of eigenvalues of parameter-dependent matrices; see, e.g., [26, 31]. For a detailed proof the reader is referred to [25, Section 5], where also the formulas adapted to (1.2) are given. □

3.2. Approximating the objective functions. The main issue arising when solving the optimization problems (3.1), (3.3) is that each evaluation of the objective functions Ψ_j, ψ_j and their derivatives at a value α requires additional linear solves with $A + \alpha I$. Thus, each of those evaluations within a derivative-based optimization method will be more expensive than a single LR-ADI iteration step, making the numerical solution of (3.1), (3.3) very costly regardless of the employed optimization algorithm, and consequently, the shift generation would be prohibitively expensive.

As a main contribution of this paper, this section proposes shift generation strategies working with cheaper to evaluate approximations of the objective functions $\tilde{\Psi}_j \approx \Psi_j$, $\tilde{\psi}_j \approx \psi_j$. Our main approach is based on a projection framework using a low-dimensional subspace $\mathcal{Q} \subset \mathbb{C}^n$, $\dim(\mathcal{Q}) = \ell \ll n$. Let the columns of $Q_\ell \in \mathbb{C}^{n\times\ell}$ be an orthonormal basis of $\mathcal{Q}$. We employ the usual Galerkin approach to obtain an approximation

$$\begin{aligned}(3.5)\qquad \Psi_j(\alpha,\overline{\alpha}) = \mathcal{C}(A,\alpha)W_j &\approx Q_\ell \mathcal{C}(H_\ell,\alpha)\tilde{W}_{\ell,j} =: \tilde{\Psi}_j(\alpha,\overline{\alpha}),\\ H_\ell := Q_\ell^* A Q_\ell \in \mathbb{C}^{\ell\times\ell}, &\qquad \tilde{W}_{\ell,j} := Q_\ell^* W_j \in \mathbb{C}^{\ell\times s}.\end{aligned}$$

Because of the orthogonality of Q_ℓ it suffices to use the projected objective functions $\tilde{\psi}_j := \|\mathcal{C}(H_\ell,\alpha)\tilde{W}_{\ell,j}\|_2$ and $\hat{\Psi}_j(\alpha,\overline{\alpha}) := \mathcal{C}(H_\ell,\alpha)\tilde{W}_{\ell,j}$. Evaluations of the functions and their derivatives is cheaper because the small dimension of H_ℓ allows easier to solve systems with $H_\ell + \alpha I_\ell$.

In the following we discuss some choices for the projection subspace $\mathcal{Q}$. Our emphasis is that quantities already generated by the LR-ADI iteration are used as much as possible. Since (3.1), (3.3) have to be solved in each iteration step of Algorithm 1 using a different residual factor W_j each time, we also discuss the reuse of approximation data from step j to $j+1$.

3.2.1. Using subspaces spanned by the low-rank factor. In [25] it is suggested to augment the Ritz value-based shifts (Section 2.1.1) by the optimization problem (3.3) using $\tilde{\psi}_j$, i.e., after step j, the space $\mathcal{Q} = \mathcal{Z}(h)$ spanned by the last $h = 1,\ldots,j$ block columns of the already generated low-rank solution factor $Z_j = [V_1,\ldots,V_j]$ is selected as in (2.5). The reduced objective function $\tilde{\psi}_j$ is then defined by H_ℓ and $\tilde{W}_{\ell,j}$. The restriction H_j of A can be build without additional multiplications with A because of (2.3). Let $R_j \in \mathbb{C}^{hs\times hs}$ so that $Q_j = [V_{j-h+1},\ldots,V_j]R_j$ has orthonormal columns. Then,

$$H_j := Q_j^* A Q_j = Q_j^* W_j G_{j,h}^* R_j - R_j^{-1} S_{j,h}^* R_j,$$

where $G_{j,h}$, $S_{j,h}$ indicate the last h block rows (and columns) of G_j, S_j from (2.3). Even though this space selection is rather intuitive, it led to impressive results often outperforming existing shift selecting strategies in [25]. The obtained rate of the residual norm reduction in the LR-ADI iteration was very close to the case when the true objection function was used in (3.3), indicating a sufficiently good approximation of ψ_j at low generation costs. Note that, the concept of approximating an expensive to evaluate objective function by projections onto already built up subspaces can also be found in other areas, e.g., in the context of model order reduction [11].

3.2.2. Krylov and extended Krylov subspace-based approximations. Consider the block Krylov subspace of order p as projection space:

$$\mathcal{Q} = \mathcal{K}_p(A,W_j) := \operatorname{range}\left([W_j, AW_j,\ldots,A^{p-1}W_j]\right).$$

This is a common strategy for approximating the product of a parameter-independent, large-scale matrix function times a block vector $f(A)W_j$; see, e.g., [19, 20, 23, 24, 39]. For our parameter-dependent matrix function $\mathcal{C}(A,\alpha)$, this choice can be motivated by considering the boundary of the stability region, where $\mathcal{C}(A,0)W_j = W_j$, which is the first basis block of $\mathcal{K}_p(A,W_j)$. On the other hand, at $\alpha = 0$ we have for the derivatives, e.g.,

$$\frac{\partial \Psi_j(\alpha,\overline{\alpha})}{\alpha} = -A^{-1}W_j,$$

and moreover, $\mathrm{grad}^2\, \psi_j$ at $\alpha = 0$ involves expressions with $A^{-2}W_j$. In order to get, at least near the origin, a good approximation of Ψ_j, ψ_j and their derivatives, this motivates to also incorporate information from a low-order inverse Krylov subspace $\mathcal{K}_m(A^{-1}, A^{-1}W_j)$ to the projection space $\mathcal{Q}$. Hence, we consider the extended Krylov subspace

$$\begin{aligned}\mathcal{Q} = \mathcal{E}_{p,m}(A,W_j) &\coloneqq \mathcal{K}_p(A,W_j) \cup \mathcal{K}_m(A^{-1},A^{-1}W_j)\\ &= \mathrm{range}\left([W_j, AW_j, \ldots, A^{p-1}W_j, A^{-1}W_j, \ldots, A^{-m}W_j]\right)\end{aligned}$$

as projection subspace. Let the columns of $Q_\ell = Q_{p,m}$ be an orthonormal basis for $\mathcal{Q}$ and recall (3.5). Existing results on such Galerkin approximations using (extended) Krylov subspaces, e.g., [20, 23], dictate for α fixed, $s = 1$, that $\tilde{\Psi}_j(\alpha,\overline{\alpha}) = Q_\ell \mathcal{C}(H_\ell,\alpha)Q_\ell^* W_{\ell,j}$ in (3.5) has degree $p-1$ in the numerator, m in the denominator, and interpolates $\mathcal{C}(z,\alpha)$ at the eigenvalues of H_ℓ. For results regarding $s > 1$ we refer to [19].

Constructing the basis matrix $Q_{p,m}$ and the restrictions H, $\tilde{W}_j$ can be done efficiently by the extended Arnoldi process [43], requiring essentially only matrix vector products and linear solves with A. However, W_j changes throughout the ADI iteration, which would necessitate to construct a new orthonormal basis associated with $\mathcal{E}_{p,m}(A,W_j)$ in each LR-ADI iteration step. As an auxiliary contribution, the next theorem shows that this is not needed for $j > 1$ and shows how the subspaces $\mathcal{E}_{p,m}(A,W_j)$ evolve from an initial subspace $\mathcal{E}_{p,m}(A,B) = \mathcal{E}_{p,m}(A,W_0)$. Note that because of the arising block matrices $q_i \in \mathbb{C}^{n\times s}$, $i = 1,\ldots,\ell$, the expressions $\mathrm{span}\,\{q_1,\ldots,q_\ell\}$ and $\mathrm{range}\,([q_1,\ldots,q_\ell])$ in the theorem are to be understood in the block-wise sense following the framework defined in [19]. In particular, $\mathrm{span}\,\{q_1,\ldots,q_\ell\} = \left\{\sum\limits_{i=1}^{\ell} q_i\Xi_i,\ \Xi_i \in \mathbb{C}^{s\times s}\right\}$ and similarly for $\mathrm{range}\,(\cdot)$.

THEOREM 3.2. *For $j > 1$ and $p,m = 0,\ldots,n$ (with at least one of the orders p,m nonzero) it holds that*

$$\mathcal{E}_{p,m}(A,W_j) \subseteq \mathcal{E}_{p,m}(A,B) \cup \mathrm{range}\,(Z_j)\,.$$

Proof. For simplicity and clarity, we restrict the proof to the case $p > 0,\ m = 0$. The more general situation can be elaborated similarly. Let $\mathcal{K}_p(A,B) = \mathrm{range}\,(K_p(A,B))$, where $K_p(A,B) \coloneqq [B, AB, \ldots, A^{p-1}B] \in \mathbb{R}^{n\times ps}$ is the associated block Krylov matrix. Likewise, $K_p(A,W_j)$ is the Krylov matrix with respect to. $\mathcal{K}_p(A,W_j)$.

We show $\mathrm{span}\left\{A^{p-1}W_j\right\} \subset \mathcal{K}_p(A,B) \cup \mathrm{range}\,(Z_j)$ via induction. For $p = 1$, it holds that $A^0W_j = W_j = K_1(A,B)I_s + Z_jS_j^0G_j = B + Z_jG_j$ because of (2.2). Let the claim be true for all powers up to $p-2$, i.e., it holds that

$$A^{p-2}W_j = K_{p-1}(A,B)M_{p-1} + Z_jN_{p-2}$$

for some matrices $M_{p-1} \in \mathbb{R}^{s(p-1)\times s}$, $N_{p-2} \in \mathbb{R}^{js\times s}$ of rank s. By using (2.2) and (2.3a), we obtain for the induction step from the matrix power $p-2$ to $p-1$

$$\begin{aligned} A^{p-1}W_j &= A(A^{p-2}W_j) = A(K_{p-1}(A,B)M_{p-1} + Z_jN_{p-2}) \\ &= AK_{p-1}(A,B)M_{p-1} + BG_j^*N_{p-1} + Z_jS_jN_{p-2} \\ &= [B, AK_{p-1}(A,B)]\begin{bmatrix} G_j^*N_{p-2} \\ M_{p-1}\end{bmatrix} + Z_jS_jN_{p-2}. \end{aligned}$$

It is easy to see that $N_{p-2} = S_j^{p-2}G_j$ which establishes for $p > 1$

$$A^{p-1}W_j = K_p(A,B)M_k + Z_j(S_j)^{p-1}G_j, \quad M_p := \begin{bmatrix} G_j^*S_j^{p-2}G_j \\ M_{p-1}\end{bmatrix}, \quad M_1 := I_s, \tag{3.6}$$

proving the assertion. For $m > 0$ we have $A^{-1}Z_j = Z_jS_j^{-1} - A^{-1}BG_j^*S_j^{-1}$ by (2.3a) which leads immediately to $A^{-1}W_j = A^{-1}B(I_s - G_j^*S_j^{-1}G_j) + Z_jS_j^{-1}$ and, consequently, $\mathcal{K}_m(A^{-1}, A^{-1}W_j) \subseteq \mathcal{K}_m(A^{-1}, A^{-1}B) \cup \operatorname{range}(Z_j)$ can be shown as for the standard Krylov subspace. The unification yields the claim for $\mathcal{E}_{p,m}$. □

The consequence of Theorem 3.2 is that for every iteration step $j > 1$, a basis for the subspace $\mathcal{E}_{p,m}(A, W_j)$ can be constructed from the initial basis for $\mathcal{E}_{p,m}(A, B)$ and the low-rank factor Z_j. By concatenating the block columns $W_j, AW_j, \ldots, A^{p-1}W_j$ from (3.6) we obtain

$$\begin{aligned} K_p(A,W_j) &= K_p(A,B)T_p^{\mathcal{K}} + Z_jK_p(S_j,G_j), \\ T_p^{\mathcal{K}} &:= \begin{bmatrix} I_s & T_2 & \cdots & T_p \\ & \ddots & \ddots & \vdots \\ & & \ddots & T_2 \\ & & & I_s \end{bmatrix} \in \mathbb{C}^{sp\times sp}, \end{aligned}$$

with $T_i = G_j^*S_j^{i-2}G_j \in \mathbb{C}^{s\times s}$ for $i = 2, \ldots, p$. For $m > 0$ a straightforward generalized expression can be found. Of course, from a numerical point of view it is not wise to work with the explicit (extended) Krylov matrices or the matrix $T_p^{\mathcal{K}}$. Instead, we propose to use

$$Q_{p,m}(A,W_j) = \texttt{orth}[Q_{p,m}(A,B), \omega_j], \quad \omega_j := Z_jQ_{p,m}(S_j,G_j), \tag{3.7}$$

as projection space, where `orth` refers to any stable orthogonalization routine. There, $Q_{p,m}(A,W_j)$, $Q_{p,m}(A,B)$, and $Q_{p,m}(S_j,G_j)$ are orthonormal basis matrices for the extended Krylov spaces $\mathcal{E}_{p,m}(A,W_j)$, $\mathcal{E}_{p,m}(A,B)$, and $\mathcal{E}_{p,m}(S_j,G_j)$, respectively. More details on the numerical implementation are given later in Section 3.2.3.

REMARK 3.3.

1. The result for $m = 0$ indicates a basic framework for acquiring a basis of $\mathcal{K}_p(A,W_j)$ from $\mathcal{K}_p(A,B)$ without new matrix vector products involving A, and thus, it could be useful for iteratively solving the shifted linear systems in LR-ADI by Krylov subspace methods. Since this is beyond the scope of this study, we leave exploiting Theorem 3.2 for iterative linear solves for future work.
2. The motivation for using $\mathcal{E}_{p,m}$ was to improve the approximation of Ψ_j, ψ_j near the origin. However, in practice the origin will be excluded in the actual optimization. One can use shifted spaces defined by $A - \phi I$, $\phi > 0$, e.g., if one can expect that the local minima have $\mathrm{Re}(\alpha) < -\phi < 0$. This only changes the inverse Krylov subspace $\mathcal{K}_m((A-\phi I)^{-1}, (A-\phi I)^{-1}B)$ since the standard Krylov subspaces are shift-invariant.

3. Extensions to rational Krylov subspaces

$$\mathcal{K}_r^{\text{rat}}(A, W_j, \boldsymbol{\beta}) = \text{range}\left\{(A+\beta_1 I)^{-1}W_j, \ldots, \prod_{i=1}^{r}(A+\beta_i I)^{-1}W_j\right\},$$

with shifts $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_r\}$ are also possible, but since this would provoke an additional search for $\boldsymbol{\beta}$, we restrict ourselves in the remainder to the standard and extended Krylov subspace approaches. If $\beta_i = \alpha_i$, $1 \leq i \leq r \leq j$, it is well known that $\text{range}\,(Z_j) \subseteq \mathcal{K}_r^{\text{rat}}(A, B, \boldsymbol{\beta})$ [18, 30, 51, 52]. However, even in this case $\text{span}\,\{(A+\alpha_i I)^{-1}W_j\} \subsetneq \text{range}\,(Z_j)$ such that the construction mentioned in Section 3.2.1 is not a true rational Krylov approximation.

3.2.3. Implementation. We give some remarks on the numerical implementation of the proposed strategy for approximating the objective function Ψ_j, ψ_j within the LR-ADI iteration. Before the LR-ADI iteration is started, a block-extended Arnoldi process [43] with orders p, m is applied to A, B which provides

$$Q_B^* Q_B = I, \quad \text{range}\,(Q_B) = \mathcal{EK}_{p,m}(A, B), \quad P_B := AQ_B,$$

where $Q_B(:, 1:s)\eta = B$, $\eta \in \mathbb{R}^{s\times s}$, and $H_B = Q_B^* A Q_B = Q_B^* P_B \in \mathbb{R}^{(p+m)s\times(p+m)s}$, i.e., the restriction of A with respect to $\mathcal{EK}_{p,m}(A, B)$. For later use, Q_B, P_B, and H_B are stored. If no shift parameter α_1 is provided, then it is computed using H_B and $\tilde{W}_0 := Q_B^* B = [\eta^*, 0, \ldots, 0]^* \in \mathbb{R}^{(p+m)s\times s}$ within (3.3). Suppose that j steps of the LR-ADI iterations have been carried out and α_{j+1} is sought for the next step. A reduced objective function constructed from the approximation space $\mathcal{EK}_p(A, W_j)$ is employed. Motivated by the argumentation in Section 3.2.2, the augmented basis matrix (3.7) with respect to the augmented space $\mathcal{EK}_{p,m}(A, B) \cup \text{span}\,\{\omega_j\}$ is used, where $\omega_j := Z_j Q_{S_j} \in \mathbb{C}^{n\times(p+m)s}$ and $Q_{S_j} \in \mathbb{C}^{js\times(p+m)s}$ is the orthogonal basis matrix spanning $\mathcal{EK}_{p,m}(S_j, G_j)$. Executing the extended Arnoldi process with S_j, G_j is extraordinarily cheap because it involves only quantities of dimension j due to the Kronecker structure of S_j, G_j (cf. (2.2), (2.3a)).

The generation of the shift α_{j+1} is summarized in Algorithm 2 including both subspace choices from Sections 3.2.1 and 3.2.2. The orthogonalization of the long block vectors in lines 7, 9 represents extra computational costs that were not present in the original LR-ADI iteration. In order to keep these costs low, it is advised to use only small values for p, m, e.g., $p, m \leq 2$. This setting appeared to be sufficient in our numerical tests. By cleverly using (2.3a) and similar relations for the extended Arnoldi process [43], the restriction H_j in line 10 can be constructed without explicitly computing additional matrix vector products[2]. Unless $A + A^*$ is negative definite, it can happen that $\Lambda(H_j) \cap \mathbb{C}_- \neq \emptyset$, which would be problematic for the compressed objective functions. A basic counter measure is shown in line 13, where H_j is replaced by its Schur form $H_j \leftarrow Q_{j,H}^* H_j Q_{j,H}$ and any unstable eigenvalues that may appear on the diagonal of H_j are negated. The transformation into the Schur basis $Q_{j,H}$ also simplifies the evaluation of function and derivatives due to the (quasi)triangular structure of the Schur form.

Dealing with the generalized Lyapunov equations. In practice often the generalized Lyapunov equations (1.2) arise with an additional, invertible matrix $M \in \mathbb{R}^{n\times n}$. The LR-ADI iteration for (1.2) is given by

$$V_j = (A + \alpha_j M)^{-1} W_{j-1}, \qquad W_j = W_{j-1} + \gamma_j^2 M V_j, \qquad W_0 := B, \tag{3.8}$$

[2] Details on this can be found in an earlier preprint of this article: https://arxiv.org/abs/1811.05500v1.

Algorithm 2: Construction and solution of reduced minimization problems.

```
Input  : LR-ADI iteration index $j$, low-rank solution factor $Z_j$, residual factor $W_j$,
         previously used shifts $\{\alpha_1, \dots, \alpha_j\}$, orders $p, m$ for extended Krylov
         subspace, matrices $Q_B$, $H_B = Q_B^*(AQ_B)$ of initial space $\mathcal{EK}_{p,m}(A, B)$,
         number $h > 0$ of previous block columns of $Z_j$ if $p = m = 0$.
Output : Next shift $\alpha_{j+1}$ for LR-ADI iteration
if $j > 1$ then
    if $p > 0$ and $m > 0$ then
        if $j \leq p + m$ then
            Set $Q_{S_j} = 1$.
        else
            Generate orthonormal basis $Q_{S_j} \in \mathbb{C}^{sj \times (p+m)s}$ for $\mathcal{E}_{p,m}(S_j, G_j)$ with
              $S_j$, $G_j$ from (2.2), (2.3).
        $Q_j = \texttt{orth}[Q_B, Z_jQ_{S_j}]$, .
    else
        $Q_j = \texttt{orth}[Z_j(:, (j - \min(j, h))s + 1 : js)]$.
    $H_j = Q_j^*(AQ_j)$ , $\tilde{W}_j := Q_j^* W_j$.
else
    $H_j = H_B$, $\tilde{W}_j = Q_B^* W_0 (= Q_B^* B)$.
Compute Schur form $H_j \leftarrow Q_{j,H}^* H_j Q_{j,H}$ (negate unstable eigenvalues on demand),
  $\tilde{W}_j \leftarrow Q_{j,H}^* \tilde{W}_j$
Find local minimizer $\alpha_{j+1} = \nu + \jmath\xi \in \mathbb{C}_-$ by solving compressed optimization
  problems (3.1), (3.3) defined by $H_j$, $\tilde{W}_j$.
```

(see, e.g., [8, 25]) leading to generalizations of the objective functions

$$\Psi_j^M(\alpha, \overline{\alpha}) = (A - \overline{\alpha}M)(A + \alpha M)^{-1} W_j, \qquad \psi_j^M(\alpha, \overline{\alpha}) = \|\Psi_j^M(\alpha, \overline{\alpha})\|^2.$$

Approximating the generalized objective functions by using subspaces of $\operatorname{range}(Z_j)$ as in Section 3.2.1 leads to $\tilde{\Psi}_j^M \approx \Psi_j^M$ defined by

$$N_j := Q_j^* M Q_j, \qquad H_j := Q_j^* A Q_j = Q_j^* W_j G_{j,h}^* R_j - N_j R_j^{-1} S_{j,h}^* R_j, \qquad \tilde{W}_j := Q_j^* W_j,$$

where Q_j, R_j come from a thin QR-factorization of the h newest block columns of Z_j.

For the (extended) Krylov subspace approximations proposed in Section 3.2.2, we define, e.g., $A_M := M^{-1}A$, $W_{M,j} := M^{-1}W_j$, and use

$$\begin{aligned}
&\Psi_j^M(\alpha, \overline{\alpha}) = M\mathring{\Psi}_j(\alpha, \overline{\alpha}), \qquad \mathring{\Psi}_j(\alpha, \overline{\alpha}) = \mathring{\Psi}_j := (A_M - \overline{\alpha}I)(A_M + \alpha I)^{-1} W_{M,j},\\
&\psi_j^M(\alpha, \overline{\alpha}) = \|M\mathring{\Psi}_j\|^2 = \lambda_{\max}(\mathring{\Psi}_j^* M^* M \mathring{\Psi}_j).
\end{aligned}$$

The objective function approximation framework presented before can still be used except that Q_B now spans $\mathcal{EK}_{m,p}(A_M,\ B_M)$ for $B_M := M^{-1}B$. For (3.8) the relations (2.3) hold for A_M, B_M, $W_{M,j}$ such that we can orthogonally augment Q_B by Q_{Z_j} exactly as before to $Q_j = [Q_B,\ Q_{Z_j}]$ and use the approximations

$$\Psi_j^M \approx F_{M,j}\mathring{\tilde{\Psi}}_j(\alpha, \overline{\alpha}), \qquad F_{M,j} := MQ_j, \qquad \mathring{\tilde{\Psi}} := \mathcal{C}(H_j, \alpha) Q_j^* W_{M,j}.$$

The matrix $F_{M,j} \in \mathbb{R}^{n\times 2(m+p)s}$ is independent on the optimization variables and can therefore be easily integrated into the compressed optimization problems via, e.g., a thin QR factorization $F_{M,j} = Q_{M,j}R_{M,j}$.

3.3. Solving the optimization problem. Having constructed the reduced objective function $\tilde{\Psi}_j$, $\tilde{\psi}_j$ by the approaches discussed before, we plan to find a local minimizer with a derivative-based numerical optimization routine. Here, we omit most details on the optimization routines as more information can be found in the given citations and references therein as well as in standard literature on numerical optimization [34].

The constraints in (3.1)–(3.3) can for practical purposes be given by

$$\nu_- \leq \nu \leq \nu_+, \quad 0 \leq \xi \leq \xi_+, \qquad -\infty < \nu_- < \nu_+ < 0, \qquad \xi \in \mathbb{R}_+, \tag{3.9}$$

where the imaginary part was restricted to the nonnegative real numbers because we exclusively consider CALEs defined by real matrices and the generated set of shift parameters is supposed to be closed under complex conjugation. We set $\nu_\pm$, ξ_+ by means of approximate spectral data of A using the extremal eigenvalues of H_j. Often, optimization algorithms require an initial guess, and we use the shift obtained by the Residual-Hamiltonian approach as initial guess since this led to the most promising results.

For solving the polyanalytic, nonlinear least-squares problem in (3.1) with the constraints (3.9), we use the routine `nlsb_gndl` based on a projected Gauss-Newton type method from the TENSORLAB software package [48][3]. For the real-valued constrained minimization problem (3.3) in real variables, a large variety of software solutions is available. The MATLAB Optimization Toolbox provides the general purpose routine `fmincon` that comes with different internal optimization algorithms, e.g., interior-point [50] and trust-region-reflective algorithms [12]. Both allow to specify hard-coded Hessians via (3.4).

However, when $s > 1$ the assumption in Theorem 3.1 that the parameter-dependent matrix $(W_\alpha^{(0)})^* W_\alpha^{(0)}$ has s distinct eigenvalues $\theta_i(\nu,\xi)$, $i = 1,\ldots,s$, can in practice be violated. For instance, it can happen that $\theta_1(\nu,\xi)$ and $\theta_2(\nu,\xi)$ coalesce at certain points ν,ξ. Consequently, at these points the derivatives of ψ_j (or $\tilde{\psi}_j$) do not exist; see, e.g., [14, 27, 28, 31, 32, 35]. Experimental observations show that especially the minima of ψ_j are often attained at those points. A simple approach in [25] to prevent these instances is to transform the eigenvalue optimization to a scalar optimization problem (i.e., $s = 1$) by multiplying the compressed residual factors $\tilde{W}_j$ with an appropriate tangential vector $t \in \mathbb{R}^s$: $\hat{W}_j = \tilde{W}_j t$. An obvious choice for t is the left singular vector corresponding to the largest singular value of $\tilde{W}_j$. The associated modified objective function is then

$$\hat{W}_j = \tilde{W}_j t, \qquad \hat{\psi}_j := \|\hat{W}_j - 2\nu(H_j + (\nu + \jmath\xi)I)^{-1}\hat{W}_j\|^2. \tag{3.10}$$

Although this transformation is an additional approximation step regarding the original function ψ_j, numerical experiments in, e.g., [25] do not indicate a substantial deterioration of the quality of the obtained shift parameters, and moreover, it simplifies the evaluations of the functions and its derivatives a bit further.

Here we also handle the minimization of $\tilde{\psi}_j$ without the modification (3.10). Methods based on the BFGS framework are capable of solving non-smooth optimization problems [14, 28] provided a careful implementation is used. The GRANSO package [14] provides MATLAB implementations of these BFGS type methods and will also be tested in the numerical experiments for (3.3) without the modification (3.10).

[3] In principle, the functionality of TENSORLAB would also allow to solve the complex minimization problem (3.2).

4. Multistep extensions.

Until now a single shift α_{j+1} was generated in each iteration step for reducing the residual norm from the current to the immediate next step. This can be generalized towards the generation of multiple, say $g > 1$, shifts for reducing $\|W_{j+g}\|^2$ the most starting from $\|W_j\|^2$. The NLS formulation for $g \geq 1$ takes the form

$$\{\alpha_{j+1}, \dots, \alpha_{j+g}\} = \underset{\boldsymbol{\alpha} \in \mathbb{C}_-^g}{\operatorname{argmin}} \|\Psi_{j,j+g}(\boldsymbol{\alpha}, \overline{\boldsymbol{\alpha}})\|^2, \quad \Psi_{j,j+g}(\boldsymbol{\alpha}, \overline{\boldsymbol{\alpha}}) := \left(\prod_{i=1}^{g} \mathcal{C}(A, \boldsymbol{\alpha}(i)) \right) W_j.$$

Since we always assumed that if $\alpha_i \in \mathbb{C}_-$, then also its complex conjugate is used (Remark 2.3), this could yield parameters for up to $2g$ future LR-ADI steps. Obviously, this multistep optimization problem is more difficult than the single step one. For instance, since the order in which the shifts are applied is not important we have $\Psi_{j,j+g}(\boldsymbol{\alpha}, \overline{\boldsymbol{\alpha}}) = \Psi_{j,j+g}(\Pi_g \boldsymbol{\alpha}, \Pi_g \overline{\boldsymbol{\alpha}})$ for any permutation $\Pi_r \in \mathbb{R}^{g \times g}$, implying that several local minima always exist. Moreover, the larger g, the harder it will be to approximate $\Psi_{j,j+g}$ by the data available at step j such that potentially better shifts might be obtained from the single shift approach carried out in each step. A similar generalization of (3.3) that can be found in [25] indicated no substantial improvements over $g = 1$. An interesting special situation is when the $g > 1$ future shift parameters are restricted to be equal, $\alpha_{j+i} = \alpha_{j+1}$, $i = 1, \dots, g$. Similar multistep approaches were investigated for Smith-type methods in, e.g., [1, 22, 37, 46]. Although this restriction might slow down the convergence compared to different shifts in each step, a noticeable reduction in the computation time can be gained. In particular, when sparse direct solvers are employed, a sparse LU factorization $LU = A + \alpha_{j+1} I$ is reused in the required forward and backward solves for the linear systems in the next g iteration steps: $V_{j+i} = U^{-1} L^{-1} W_{j+i-1}$, $1 \leq i \leq g$. This can be substantially cheaper compared to solving g different shifted linear systems depending on the value g and the cost for solving a single shifted linear system. Obviously, one could simply use the shift obtained by the single step residual norm minimization framework g times. We hope to obtain a better LR-ADI performance by incorporating the prior knowledge that α_{j+1} is to be used in $g \geq 1$ iteration steps. The associated multi-shift NLS formulation is

$$\alpha_{j+1} = \underset{\alpha \in \mathbb{C}_-}{\operatorname{argmin}} \|\Psi_{j,j+g}(\alpha, \overline{\alpha})\|^2, \qquad \Psi_{j,j+g}(\alpha, \overline{\alpha}) := \mathcal{C}(A, \alpha)^g W_j.$$

Using the product rule, the Jacobian and conjugate Jacobian of $\Psi_{j,j+r}$ are given by

$$\begin{aligned} \frac{\partial \Psi_{j,j+g}(\alpha, \overline{\alpha})}{\alpha} &= -g \mathcal{C}(A, \alpha)^g (A + \alpha I)^{-1} W_j, \\ \frac{\partial \Psi_{j,j+g}(\alpha, \overline{\alpha})}{\overline{\alpha}} &= -g \mathcal{C}(A, \alpha)^{g-1} (A + \alpha I)^{-1} W_j. \end{aligned}$$

By the same reasoning as in Section 3.2.2, these formulas indicate that for approximating the objective function and its derivatives, the orders p, m for the approximation subspace $\mathcal{EK}_{p,m}(A, B)$ should be at least g, but in the numerical experiment smaller orders worked sufficiently well. Extending (3.3) is done in the same way, that is, by defining $\psi_{j,j+g}(\nu, \xi) := \|\mathcal{C}(A, \nu + \jmath \xi)^g W_j\|^2$. Theorem 3.1 for the derivatives of $\psi_{j,j+g}$ can easily be reformulated by using

$$W_\alpha^{(r)} := g \mathcal{C}(A, \alpha)^{g-1} W_j = g \left(I - 2\nu L(\nu, \xi)^{-1} \right)^{g-1} W_j$$

instead of $W_\alpha^{(0)}$.

5. Numerical experiments.

In this section we execute several numerical examples to evaluate different aspects of the residual norm-minimizing shift selection techniques. All experiments were done in MATLAB 2016a using a Intel Core 2 i7-7500U CPU @ 2.7 GHz with 16 GB RAM. We wish to obtain an approximate solution such that the scaled Lyapunov residual norm satisfies

$$\mathfrak{R} := \|\mathcal{R}^{\text{true}}\|/\|B\|^2 \leq \varepsilon, \qquad 0 < \varepsilon \ll 1.$$

Table 5.1 summarizes the four different test examples.

TABLE 5.1
Overview of the numerical examples.

Example	n	s	Origin of matrix A (and M)	ε
`cd2d`	40000	$\{1, 5\}$	Finite difference discretization of the 2d operator $\mathcal{L}(u) = \Delta u - 100x\frac{\partial u}{\partial x} - 1000y\frac{\partial u}{\partial y}$ on $[0,1]^2$ with homogeneous Dirichlet b.c.	10^{-8}
`cd3d`	27000	10	Finite difference discretization of the 3d operator $\mathcal{L}(u) = \Delta u - 100x\frac{\partial u}{\partial x} - 1000y\frac{\partial u}{\partial y} - 10z\frac{\partial u}{\partial z}$ on $[0,1]^3$ with homogeneous Dirichlet b.c.	10^{-8}
`lung`	109460	10	A model of temperature and water vapor transport in the human lung from the suitesparse collection [15].	10^{-8}
`chip`	20082	1	Finite element model of a chip cooling process [33], $M = M^* \neq I$.	10^{-10}

The right-hand side factors B for all examples except `chip` are generated randomly with uniformly distributed entries, where the random number generator is initialized by `rand('state', 0)`. The maximal allowed number of LR-ADI steps is restricted to 150. In all experiments, we also emphasize the numerical costs for generating shift parameters by giving shift generation times t_{shift} next to the total run times t_{total} of the LR-ADI iteration. Before we compare the proposed residual-minimizing shifts against other existing approaches, some tests with respect to certain aspects of this shift selection framework are conducted.

5.1. Approximation of the objective function.

At first, we evaluate different approximation approaches from Section 3.2 for the objective functions, i.e., we test the influence of different choices for the projection subspace to the overall performance of the LR-ADI iteration. This experiment is carried out on the `cd2d` example (see Table 5.1) with a single vector in B and $\|B\| = 1$. The NLS formulation (3.1) is employed and dealt with by the TENSORLAB routine `nlsb_gndl`. As approximation subspaces the last $h = 8$ columns of Z_j from Section 3.2.1 (denoted by $\mathcal{Z}(8)$) and the extended Krylov approximations from Section 3.2.2 with different orders p, m are used such that the dimension of the basis is in all cases at most 8. Moreover, the experiment is carried out in the single step ($g = 1$) as well as multistep fashion ($g = 5$) from Section 4. The obtained results are presented in Figure 5.1 and Table 5.2.

Apparently, for the single step optimization approach, using $\mathcal{Z}(h)$ as approximation space seems to yield shifts leading to the fastest convergence compared to the other subspace choices. The required iteration numbers and timings are the smallest among all tested settings. In particular, the pure Krylov ($m = 0$) and inverse Krylov subspaces ($p = 0$) lag behind the other choices. The picture changes when considering the multistep optimization from Section 4

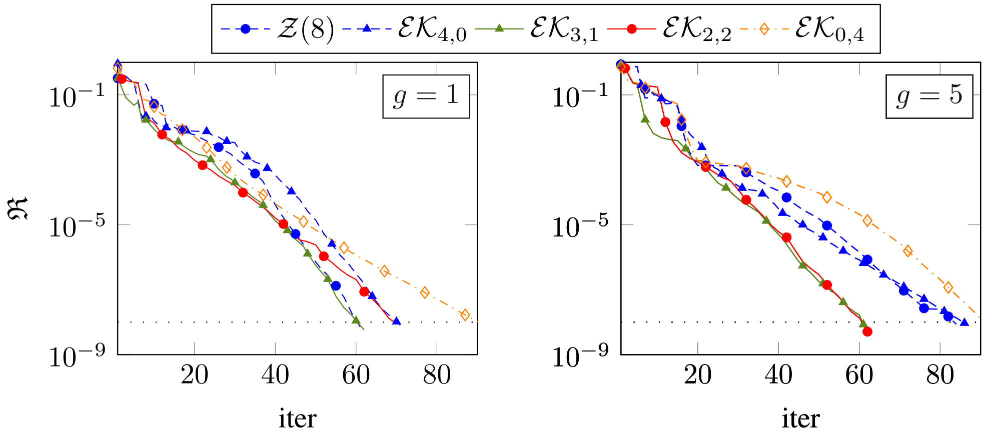

FIG. 5.1. *Residual norm history of LR-ADI iteration using different objective function approximations for the* `cd2d` *example. Left: single step shift selection ($g = 1$). Right: multistep shift selection ($g = 5$).*

TABLE 5.2

Results with different projection subspaces for the objective function approximation using the `cd2d` *example. For single and multistep approaches ($g = 5$), listed are the executed iteration numbers (iters), the column dimension of the built up low-rank factors (dim), the estimated rank (rk) of the approximate solution, the total and shift computation times (t_{total}, t_{shift}) in seconds, and the final residual norm $\mathfrak{R}$ (res).*

proj.	Single step ($g = 1$)						Multistep ($g = 5$)					
space	iters	dim	rk	t_{total}	t_{shift}	res	iters	dim	rk	t_{total}	t_{shift}	res
$\mathcal{Z}(8)$	61	61	58	8.6	1.3	$7.2 \cdot 10^{-9}$	84	84	63	3.6	0.5	$8.6 \cdot 10^{-9}$
$\mathcal{EK}_{4,0}$	71	71	58	12.3	2.9	$7.9 \cdot 10^{-9}$	86	86	63	4.6	0.9	$9.4 \cdot 10^{-9}$
$\mathcal{EK}_{3,1}$	62	62	59	10.5	2.4	$5.7 \cdot 10^{-9}$	61	61	61	3.2	0.7	$8.5 \cdot 10^{-9}$
$\mathcal{EK}_{2,2}$	70	70	64	12.3	3.0	$8.7 \cdot 10^{-9}$	62	62	62	3.0	0.7	$5.1 \cdot 10^{-9}$
$\mathcal{EK}_{0,4}$	91	91	62	18.7	3.1	$9.1 \cdot 10^{-9}$	92	92	63	4.2	1.0	$8.0 \cdot 10^{-9}$

over $g = 5$ steps, where the extended Krylov approximations with $m, p \geq 1$ yield better shifts, i.e., less iteration steps compared to using $\operatorname{range}(Z_j)$. Interestingly, in some cases the number of iteration steps is even lower compared to the single step optimization. Due to the reuse of LU factorizations over $g = 5$ steps and since the optimization problem has to be solved less frequently, the savings in the computation times reported in Table 5.2 are substantial. To conclude, while the standard objective function approximation using $\operatorname{range}(Z_j)$ seems to work satisfactory in most cases for the single step shift selection, for $g > 1$ better results might be obtained by the proposed extended Krylov approximations with $m, p \geq 1$.

5.2. Choice of the optimization routine. Now we test different optimization problem formulations (3.1), (3.3) as well as different optimization routines for the `cd3d` example (see Table 5.1) having $s = 10$ columns in B. As in the previous experiment, the TENSORLAB routine `nlsb_gndl` is used for the NLS problem, but for the function minimization problem (3.3) we employ GRANSO and `fmincon`. In `fmincon` the interior-point and trust-region-reflective methods are used as optimization routines. Since $s > 1$, we also use the tangential approximation (3.10) to avoid the potential non-smoothness of ψ_j in (3.3) and test this modification also within the NLS framework. The projection subspaces for the objective function approximations are constructed from the last $h = 4$ block columns of Z_j. Table 5.3 summarizes the results.

TABLE 5.3
Results with different optimization routines for the `cd3d` *example.*

Opt. problem	Opt. routine	iters	dim	rk	t_{total}	t_{shift}	res
NLS (3.1)	`nlsb_gndl`	52	520	520	88.6	2.1	$2.9 \cdot 10^{-9}$
NLS (3.10)	`nlsb_gndl`	47	470	470	78.0	1.7	$7.8 \cdot 10^{-9}$
fun.min. (3.3)	`fmincon`+int. point	48	480	480	67.8	2.0	$5.6 \cdot 10^{-9}$
fun.min. (3.10)	`fmincon`+int. point	50	500	500	77.6	2.1	$4.4 \cdot 10^{-9}$
fun.min. (3.3)	`fmincon`+thrust region reflective	49	490	490	82.0	1.9	$4.7 \cdot 10^{-9}$
fun.min. (3.10)	`fmincon`+thrust region reflective	51	510	510	80.1	1.7	$2.0 \cdot 10^{-9}$
fun.min. (3.3)	GRANSO	51	510	510	111.3	20.4	$2.0 \cdot 10^{-9}$

Judging from the number of required LR-ADI steps, the usage of different optimization routines appears to have less impact than working with different objective function approximations. The additional tangential approximation (3.10) seems to slow down the LR-ADI iteration only marginally. The exception, as seen in Table 5.3, is when comparing to the NLS formulation (3.1) and `nlsb_gndl` is used, where five less LR-ADI steps, and consequently less computation time, are required. Using GRANSO resulted in comparatively high computational times for this shift generation. The main computational bottleneck in this method are the arising quadratic optimization problems. Apparently, the non-smoothness of the function ϕ_j (in the sense of coalescing eigenvalues of $W(\alpha)^*W(\alpha)$) did hardly occur or appear to be problematic for methods for smooth optimization problems so that the application of non-smooth optimizers or using the tangential approximation (3.10) might not be necessary in most cases. Although not reported here, tests using `fmincon` without explicitly provided Hessians led to similar results regarding the required number of steps of the LR-ADI iteration but to marginally longer shift generation times since the inherent optimization algorithms (interior-point or trust-region-reflective) required more steps. The performance of the optimization routines appeared to be also noticeably influenced by the choice of the initial guess. Using the heuristic instead of the residual-Hamiltonian selection for determining the initial guess led to higher shift generation times due to longer runs of the optimization routines. Setting up the constraints (3.9) for the optimization variables by using the computed Ritz values (eigenvalues of H_j) led in a few cases to difficulties for the solution of the optimization problems. Especially the upper bounds for the imaginary parts of the shift parameters appeared to be of strong influence. Further adjustments are necessary in this directions, also with respect to deciding in advance if the optimization problems can be safely restricted to real variables. Currently, this is only done for problems with real spectra (e.g., $A = A^*$).

5.3. Comparison with other shift selection routines and methods. Now the LR-ADI performance obtained with the approximate residual norm-minimizing shifts is compared with the other shift selection strategies reviewed in Section 2.1. All employed shift generation and selection approaches are used and abbreviated as in Table 5.4.

We also run a few tests with the multistep approach with $g = 5$. For each example, we also compare the LR-ADI to the rational Krylov subspace method [16] (RKSM) equipped with the convex hull-based shift selection [17] and the extended Krylov subspace method [43] (EKSM). The reduced Lyapunov equations in RKSM, EKSM are solved in every fifth step if $s > 1$.

Figure 5.2 displays the history of the scaled Lyapunov residual norms for some selection approaches and the `cd3d`, `chip` examples. Table 5.5 summarizes the results of the experiment.

TABLE 5.4
Overview of employed shift selection strategies.

Type	Abbreviation	Description of strategy	Information
precomputed	heur(J, p, m)	Heuristic selection of $J \in \mathbb{N}$ shifts from Ritz values associated with $\mathcal{EK}_{p,m}(A, B\mathbf{1}_s)$, cyclic usage.	[36, 40]
	Wachs$(\epsilon,\ p,\ m)$	Wachspress selection using Ritz values associated with $\mathcal{EK}_{p,m}(A, B\mathbf{1}_s)$ and tolerance $0 < \epsilon \ll 1$, cyclic usage.	[40, 42, 49]
adaptive	$\mathcal{Z}(h)$+heur	Projection-based shifts using newest h block columns of Z_j and selection via heuristic.	[9, 25], Section 2.1.1
	$\mathcal{Z}(h)$+conv	Projection-based shifts as above, but convex hull-based selection.	[17], Section 2.1.2
	$\mathcal{Z}(h)$+Hres	Projection-based shifts as above, but residual Hamiltonian-based selection.	[6], Section 2.1.3
	resmin+$\mathcal{Q}$+OR	Residual norm-minimizing shifts with $\mathcal{Q}$ as approximation space and OR as optimization routine.	[9, 25], Section 3

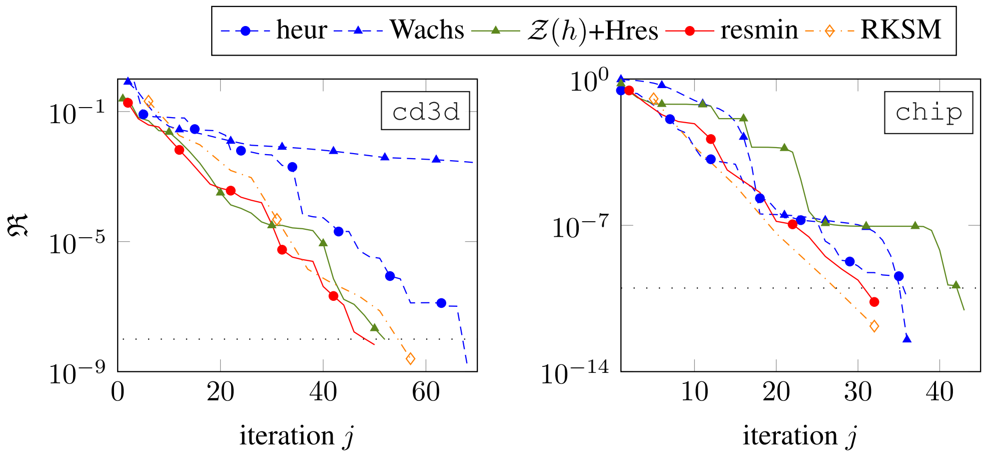

FIG. 5.2. *Residual norm history of LR-ADI iteration and RKSM using different shift selection strategies. Left:* `cd3d` *example. Right:* `chip` *example.*

The proposed residual norm-minimizing shift generation strategy based on reduced objective functions leads to the smallest number of required iteration steps compared to the other selection approaches. The obtained rate of residual norm reduction is very close to the one obtained by RKSM, but LR-ADI required in all tests less computation time than RKSM. Hence, taking both the iteration numbers as well as the computation times in account, with the right set of shift parameters, the LR-ADI iteration is competitive to RKSM. Note that RKSM is theoretically expected to converge faster than LR-ADI [16]. EKSM generates in all experiments larger subspaces but still manages to deliver competitive computation times against some of the other methods. This is due to the efficient way the linear systems can be solved in EKSM. For `chip`, EKSM is the fastest method in terms of run time. Among

TABLE 5.5
Comparison of different shift routines and against RKSM, EKSM. A superscript † *denotes* $g = 5$.

Ex.	Method	Shift selection strategy	iters	dim	rk	t_{total}	t_{shift}	res
`cd2d`	LR-ADI	heur(20, 30, 20)	137	685	269	30.3	0.6	$9.0 \cdot 10^{-9}$
		Wachs(10^{-8}, 30, 20)	93	465	254	22.0	0.6	$7.2 \cdot 10^{-9}$
		$\mathcal{Z}(4)$+heur	74	370	257	16.3	0.5	$9.2 \cdot 10^{-9}$
		$\mathcal{Z}(4)$+conv	80	400	265	18.5	1.8	$6.1 \cdot 10^{-9}$
		$\mathcal{Z}(4)$+Hres	74	370	259	19.7	1.6	$2.8 \cdot 10^{-9}$
		resmin+$\mathcal{Z}(4)$+`fmincon`	58	290	251	19.6	6.4	$6.2 \cdot 10^{-9}$
	RKSM	convex hull	61	305	277	35.9	3.9	$4.3 \cdot 10^{-9}$
	EKSM		60	600	269	38.7	–	$9.8 \cdot 10^{-9}$
`cd3d`	LR-ADI	heur(20, 40, 30)	68	680	504	85.4	1.7	$1.7 \cdot 10^{-9}$
		Wachs(10^{-8}, 30, 20)	150	1500	253	190.1	1.5	$5.2 \cdot 10^{-4}$
		$\mathcal{Z}(4)$+heur	71	710	511	89.1	0.3	$1.9 \cdot 10^{-9}$
		$\mathcal{Z}(4)$+conv	57	570	487	71.8	1.3	$2.3 \cdot 10^{-9}$
		$\mathcal{Z}(4)$+Hres	52	520	448	66.3	1.1	$9.7 \cdot 10^{-9}$
		resmin+$\mathcal{Z}(4)$+`fmincon`	50	500	444	67.9	2.3	$6.7 \cdot 10^{-9}$
		resmin+$\mathcal{EK}_{1,1}$+`nlsb_gndl`†	59	590	498	24.5	0.9	$6.2 \cdot 10^{-9}$
	RKSM	convex hull	57	570	517	113.3	4.2	$2.5 \cdot 10^{-9}$
	EKSM		60	1200	480	72.0	–	$2.3 \cdot 10^{-9}$
`lung`	LR-ADI	heur(20, 30, 20)	150	1500	332	72.6	4.2	$1.7 \cdot 10^{-8}$
		Wachs(10^{-8}, 30, 20)	150	1500	261	66.1	4.3	$2.8 \cdot 10^{-2}$
		$\mathcal{Z}(2)$+heur	94	940	343	42.8	1.0	$4.6 \cdot 10^{-10}$
		$\mathcal{Z}(2)$+conv	80	800	348	44.5	7.1	$3.6 \cdot 10^{-9}$
		$\mathcal{Z}(2)$+Hres	71	710	349	42.4	5.5	$8.5 \cdot 10^{-9}$
		resmin+$\mathcal{Z}(2)$+`fmincon`	65	650	351	37.5	4.6	$8.6 \cdot 10^{-9}$
		resmin+$\mathcal{Z}(2)$+GRANSO†	69	690	350	9.1	1.3	$9.5 \cdot 10^{-9}$
	RKSM	convex hull	61	610	353	135.9	12.4	$6.1 \cdot 10^{-9}$
	EKSM		45	900	345	104.4	–	$2.1 \cdot 10^{-9}$
`chip`	LR-ADI	heur(10, 20, 10)	33	33	28	24.1	1.2	$7.1 \cdot 10^{-11}$
		Wachs(10^{-12}, 20, 10)	34	34	28	24.4	1.2	$5.0 \cdot 10^{-13}$
		$\mathcal{Z}(4)$+heur	70	70	45	49.3	0.2	$5.6 \cdot 10^{-11}$
		$\mathcal{Z}(4)$+conv	55	55	44	38.9	0.3	$5.6 \cdot 10^{-11}$
		$\mathcal{Z}(4)$+Hres	43	43	43	30.4	0.2	$8.6 \cdot 10^{-12}$
		resmin+$\mathcal{Z}(4)$+`fmincon`	32	32	29	27.5	1.0	$2.2 \cdot 10^{-11}$
		resmin+$\mathcal{EK}_{1,1}$+`nlsb_gndl`†	32	32	28	10.2	0.2	$7.0 \cdot 10^{-11}$
	RKSM	convex hull	28	28	26	22.0	1.6	$7.5 \cdot 10^{-11}$
	EKSM		34	68	28	5.4	–	$4.7 \cdot 10^{-11}$

the Ritz value-based shift selection techniques (Section 2.1.1) for LR-ADI, the Residual-Hamiltonian selection (Section 2.1.3, $\mathcal{Z}(h)$+Hres) appears to perform best, leading to iteration numbers close to the ones obtained with the residual-minimizing shifts. The precomputed shift approaches (heur(J, p, m), Wachs(ϵ, p, m)) could in several cases not compete with the dynamic shift generation approaches, which again underlines the superiority of an adaptive selection of shift parameters. The generation times t_{shift} of the adaptive shifts were in all cases only a small fraction of the total computation times t_{total}. The quickest reduction of the residual norm and the smallest number of iteration steps was achieved by the (single step) residual minimization-based approaches. They appear to be especially effective for the `cd2d` and `lung` examples, where up to 20 percent less iteration steps are required. These two examples have more nonnormal A than the other two examples `cd3d`, `chip` such that the more involved residual minimization framework seems to be the safer choice for acquiring high quality shifts. For the examples `cd3d`, `chip`, already the Ritz value-based shift routines led to a rapid residual reduction leaving hardly any room for further speed ups. Due to the need to set up and solve (compressed) optimization problems, the generation times of the residual-minimizing shifts was in some cases slightly higher compared to the other approaches. Therefore, for the examples `cd3d`, `chip` the ultimate choice of the shift selection routine depends on whether the gained slight acceleration of LR-ADI (and, thus, the minor storage savings for the low-rank factors Z) is worth the additional shift generation effort. The results in Table 5.5 also confirm the findings of Section 5.1 that the $\mathcal{Z}(h)$ subspace choice (Section 3.2.1) appears to be adequate (for the single step minimization) in most situations. Although the multistep shift selection approach yielded higher iteration numbers compared to the single step versions, they led to a substantial reduction in the computation times t_{total} because of the reuse of LU factorizations over several iteration steps. If small computations times are more important than acquiring the smallest possible low-rank factors, then the multistep approaches might be the method of choice. Moreover, they might be invaluable for situations where sparse factorizations are expensive but also iterative linear solvers are difficult to employ. As outlined in Section 5.1, approximating the objective function by means of extended Krylov subspaces seems to be the safer choice.

6. Summary.

This article discussed dynamically generated shift parameters for the LR-ADI iteration for large Lyapunov equations. The selection of shifts was based on a residual norm minimization principle, which could be formulated as a nonlinear least-squares or function minimization problem. Since the involved objective functions are too expensive to evaluate, a framework using approximated objective functions was developed. These approximations were built using projections onto low-dimensional subspaces, whose efficient construction from the data generated by the LR-ADI iteration was presented. The numerical experiments showed that the proposed shift generation approach resulted in the fastest convergence of LR-ADI, bringing it very close to the rational Krylov subspace method in terms of the iteration numbers. At the expense of higher iteration numbers, a substantial computation time reduction could be achieved by a multistep shift selection approach. Obvious future research direction might include similar shift selection strategies in LR-ADI type methods for other matrix equations, e.g., algebraic Sylvester and Riccati equations, where first investigations can be found in [6, 9, 25]. Deriving a similar multistep selection for RKSM is also an open topic. Improving the solution of the occurring optimization problems by, e.g., providing better constraints or initial guesses, would further increase the performance of the residual norm-minimizing shift selection.

Acknowledgments. I am grateful towards my former employer, the Max Planck Institute in Magdeburg, Germany, where the main part of this work was created. Thanks also go to Tim Mitchell for helpful discussions regarding numerical optimization as well as for assistance with his software GRANSO. Additional thanks go to Davide Palitta for proofreading a first draft and to the referees for their comments which helped improving the manuscript.

REFERENCES

[1] A. C. ANTOULAS, D. C. SORENSEN, AND S. GUGERCIN, *A survey of model reduction methods for large-scale systems*, in Structured Matrices in Mathematics, Computer Science, and Engineering I, V. Olshevsky, ed., vol. 280 of Contemp. Math., Amer. Math. Soc., Providence, 2001, pp. 193–219.

[2] A. C. ANTOULAS, D. C. SORENSEN, AND Y. ZHOU, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett., 46 (2002), pp. 323–342.

[3] J. BAKER, M. EMBREE, AND J. SABINO, *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 656–668.

[4] B. BECKERMANN AND A. TOWNSEND, *On the singular values of matrices with displacement structure*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1227–1248.

[5] P. BENNER AND Z. BUJANOVIĆ, *On the solution of large-scale algebraic Riccati equations by using low-dimensional invariant subspaces*, Linear Algebra Appl., 488 (2016), pp. 430–459.

[6] P. BENNER, Z. BUJANOVIĆ, P. KÜRSCHNER, AND J. SAAK, *RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations*, Numer. Math., 138 (2018), pp. 301–330.

[7] P. BENNER, P. KÜRSCHNER, AND J. SAAK, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numer. Algorithms, 62 (2013), pp. 225–251.

[8] ———, *An improved numerical method for balanced truncation for symmetric second-order systems*, Math. Comput. Model. Dyn. Syst., 19 (2013), pp. 593–615.

[9] ———, *Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations*, Electron. Trans. Numer. Anal., 43 (2014–2015), pp. 142–162.
http://etna.math.kent.edu/vol.43.2014-2015/pp142-162.dir

[10] P. BENNER AND J. SAAK, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM-Mitt., 36 (2013), pp. 32–52.

[11] A. CASTAGNOTTO, H. K. F. PANZER, AND B. LOHMANN, *Fast $\mathcal{H}_2$-optimal model order reduction exploiting the local nature of Krylov-subspace methods*, in European Control Conference 2016, Aalborg, Denmark, 2016, IEEE Conference Proceedings, Los Alamitos, 2016, pp. 1958–1963.

[12] T. F. COLEMAN AND Y. LI, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[13] M. CROUZEIX AND C. PALENCIA, *The numerical range is a $(1+\sqrt{2})$-spectral set*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 649–655.

[14] F. E. CURTIS, T. MITCHELL, AND M. L. OVERTON, *A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles*, Optim. Methods Softw., 32 (2017), pp. 148–181.

[15] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), Art. 1 (25 pages).

[16] V. DRUSKIN, L. KNIZHNERMAN, AND V. SIMONCINI, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.

[17] V. DRUSKIN AND V. SIMONCINI, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems Control Lett., 60 (2011), pp. 546–560.

[18] G. M. FLAGG AND S. GUGERCIN, *On the ADI method for the Sylvester equation and the optimal-$\mathcal{H}_2$ points*, Appl. Numer. Math., 64 (2013), pp. 50–58.

[19] A. FROMMER, K. LUND, AND D. B. SZYLD, *Block Krylov subspace methods for computing functions of matrices applied to multiple vectors*, Electron. Trans. Num. Anal., 47 (2017), pp. 100–126.
http://etna.math.kent.edu/vol.47.2017/pp100-126.dir

[20] A. FROMMER AND V. SIMONCINI, *Matrix functions*, in Model Order Reduction: Theory, Research Aspects and Applications, W. Schilders, H. A. van der Vorst, and J. Rommes, eds., Math. Ind., 13, Eur. Consort. Math. Ind. (Berl.), Springer, Berlin, 2008, pp. 275–303.

[21] L. GRASEDYCK, *Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389.

[22] S. GUGERCIN, D. C. SORENSEN, AND A. C. ANTOULAS, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32 (2003), pp. 27–55.

[23] S. GÜTTEL, *Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.

[24] L. A. KNIZHNERMAN, *Calculation of functions of unsymmetric matrices using Arnoldi's method*, U.S.S.R. Comput. Math. and Math. Phys., 31 (1991), pp. 1–9.

[25] P. KÜRSCHNER, *Efficient Low-Rank Solution of Large-Scale Matrix Equations*, PhD. Thesis, Faculty of Mathematics, Otto-von-Guericke-University, Magdeburg, April 2016.

[26] P. LANCASTER, *On eigenvalues of matrices dependent on a parameter*, Numer. Math., 6 (1964), pp. 377–387.

[27] A. S. LEWIS, *The mathematics of eigenvalue optimization*, Math. Program., 97 (2003), pp. 155–176.

[28] A. S. LEWIS AND M. L. OVERTON, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163.

[29] J.-R. LI, *Model reduction of large linear systems via low rank system Gramians*, PhD. Thesis, Dept. of Math., MIT, Cambridge, 2000.

[30] J.-R. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[31] E. MENGI, *A support function based algorithm for optimization with eigenvalue constraints*, SIAM J. Optim., 27 (2017), pp. 246–268.

[32] E. MENGI, E. A. YILDIRIM, AND M. KILIÇ, *Numerical optimization of eigenvalues of Hermitian matrix functions*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 699–724.

[33] C. MOOSMANN, E. B. RUDNYI, A. GREINER, AND J. G. KORVINK, *Model order reduction for linear*

convective thermal flow, in Proceedings on 10th International Workshop on Thermal Investigations of ICs and Systems THERMINIC 2004, Sophia Antipolis, France, 2004, pp. 317–321.

[34] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 1999.

[35] M. L. OVERTON, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[36] T. PENZL, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (1999/00), pp. 1401–1418.

[37] ———, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.

[38] R. REMMERT, *Theory of Complex Functions*, Springer, New York, 1991.

[39] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, 1992.

[40] J. SAAK, *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, PhD. Thesis, Faculty of Mathematics, Chemnitz University of Technology, Chemnitz, July 2009.

[41] J. SAAK, M. KÖHLER, AND P. BENNER, *M-M.E.S.S.-1.0.1 – The Matrix Equations Sparse Solvers library*, DOI:10.5281/zenodo.50575, 2016. https://www.mpi-magdeburg.mpg.de/projects/mess

[42] J. SABINO, *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, PhD. Thesis, Rice University, Houston, 2006.

[43] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[44] ———, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441.

[45] L. SORBER, M. VAN BAREL, AND L. DE LATHAUWER, *Unconstrained optimization of real functions in complex variables*, SIAM J. Optim., 22 (2012), pp. 879–898.

[46] K. SUN, *Model Order Reduction and Domain Decomposition for Large-Scale Dynamical Systems*, PhD. Thesis, Rice University, Houston, 2008.

[47] N. TRUHAR AND K. VESELIĆ, *Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix*, Systems Control Lett., 56 (2007), pp. 493–503.

[48] N. VERVLIET, O. DEBALS, L. SORBER, M. VAN BAREL, AND L. DE LATHAUWER, *Tensorlab 3.0*, 2016. Available online at https://www.tensorlab.net/.

[49] E. WACHSPRESS, *The ADI Model Problem*, Springer, New York, 2013.

[50] R. A. WALTZ, J. L. MORALES, J. NOCEDAL, AND D. ORBAN, *An interior algorithm for nonlinear optimization that combines line search and trust region steps*, Math. Program., 107 (2006), pp. 391–408.

[51] T. WOLF, *$\mathcal{H}_2$ Pseudo-Optimal Model Order Reduction*, PhD. Thesis, Lehrstuhl für Regelungstechnik, Technische Universität München, Munich, 2015.

[52] T. WOLF AND H. K. F. PANZER, *The ADI iteration for Lyapunov equations implicitly performs $\mathcal{H}_2$ pseudo-optimal model order reduction*, Internat. J. Control, 89 (2016), pp. 481–493.