

Residual minimizing shift parameters for the low-rank ADI iteration

Patrick Kürschner[†]

November 15, 2018

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 coefficients in the Lyapunov equation. This article represents a follow up of Benner et al. [ETNA, 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, compressed 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.

1 Introduction

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

$$AX + XA^* + BB^* = 0 \quad (1)$$

defined by matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times s}$, $s \ll n$, and $X \in \mathbb{R}^{n \times n}$ is the sought solution. For large sizes n of the problem, directly computing and storing X is infeasible. For dealing with (1), it has become common practice to approximate X by a low-rank factorization $X \approx ZZ^*$ with $Z \in \mathbb{R}^{n \times r}$, $\text{rank } Z = r \ll n$. Theoretical evidence for the existence of such low-rank approximations can be found, e.g., in [2, 4, 20, 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, belong to the most successful and often used representatives of iterative low-rank methods for (1), see, e.g., [8, 10, 15, 16, 29, 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 is 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.

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 $X = \text{Re}(X) + j\text{Im}(X)$, $\text{Re}(X)$, $\text{Im}(X)$ are its real and imaginary

[†]Max Planck Institute for Dynamics of Complex Technical Systems, Computational Methods in Systems and Control Theory, Magdeburg, Germany, kuerschner@mpi-magdeburg.mpg.de

parts, and j 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^{-1} is the inverse of a nonsingular matrix A , and $A^{-*} = (A^*)^{-1}$. The identity matrix of dimension n is indicated by I_n . The spectrum of a matrix A is given by $\Lambda(A)$ and the spectral radius is defined as $\rho(A) := \max\{|\lambda|, \lambda \in \Lambda(A)\}$.

For a multivariate function $f(x_1, \dots, x_d) : \mathbb{R}^d \mapsto \mathbb{R}$, we employ the typical shorthand notation $\psi_{x_i} = \frac{\partial f}{\partial x_i}$ and $\psi_{x_i x_j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ for the first and second order partial derivatives, accumulated in gradient $\text{grad } f = [\psi_{x_i}]$ and Hessian $\text{grad}^2 = [\psi_{x_i x_j}]$, respectively. For a vector valued function $F(x_1, \dots, x_d) = [\psi_1, \dots, \psi_v]^T$, the Jacobian is given by $[\frac{\partial \psi_i}{\partial x_j}]$. For complex functions $g(z_1, \dots, z_d, \overline{z_1}, \dots, \overline{z_d})$ depending on complex variables and their conjugates, Wirtinger calculus [37] is used to define complex and complex conjugate derivatives, i.e., derivatives with respect to z_i and $\overline{z_i}$, respectively.

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). 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 generalized Lyapunov equations

$$AX^*M^* + MXA^* + BB^* = 0 \quad (2)$$

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 is 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) 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

$$X_j = (A - \overline{\alpha_j}I)(A + \alpha_j I)^{-1} X_{j-1} (A + \alpha_j I)^{-H} (A - \overline{\alpha_j}I)^* - 2 \text{Re}(\alpha_j) (A + \alpha_j I)^{-1} BB^* (A + \alpha_j I)^{-H}, \quad j \geq 1, \quad X_0 \in \mathbb{R}^{n \times n} \quad (3)$$

for the CALE (1). There, $\alpha_i \in \mathbb{C}_-$, $1 \leq i \leq j$, are the previously mentioned shift parameters, discussed further in Section 2.1. By introducing low-rank approximations $X_j = Z_j Z_j^*$ in each step and assuming $Z_0 = 0$, the above iteration can be rearranged [8, 24, 29, 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 1 ([8, 52]). Assume j steps of the LR-ADI iteration with the shift parameters $\{\alpha_1, \dots, \alpha_j\} \subset \mathbb{C}_-$ have been applied to (1). Then the Lyapunov residual matrix can be factorized via

$$R_j = AZ_j Z_j^* + Z_j Z_j^* A^* + BB^* = W_j W_j^*, \quad (4)$$

where the *residual factor* $W_j \in \mathbb{C}^{n \times s}$ is given by

$$W_j := (A - \overline{\alpha_j}I)V_j = W_{j-1} - 2 \text{Re}(\alpha_j) V_j = W_0 + Z_j G_j, \quad (5)$$

with $W_0 := B$, $G_j := [\gamma_1, \dots, \gamma_j]^* \otimes I_s \in \mathbb{R}^{j s \times s}$, $\gamma_i := \sqrt{-2 \text{Re}(\alpha_i)}$ for $i = 1, \dots, j$.

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

Input : Matrices A, B defining (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 α_1 .
- 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 $j = j + 1$.

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 (4) 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 [28, 29].

Corollary 2 ([52, Lemma 3.1], [51, Lemma 5.12], [24, Corollary 3.9]). With same assumptions and notations as in Theorem 1, the low-rank factor Z_j after j steps of the LR-ADI iteration (Algorithm 1) satisfies the Sylvester equations

$$AZ_j - Z_j S_j = B G_j^*, \quad (6a)$$

$$AZ_j + Z_j \bar{S}_j^* = W_j G_j^*, \quad (6b)$$

where

$$S_j := \begin{bmatrix} \alpha_1 & \gamma_1 \gamma_2 & \cdots & \gamma_1 \gamma_j \\ & \ddots & \ddots & \vdots \\ & & \ddots & \gamma_{j-1} \gamma_j \\ & & & \alpha_j \end{bmatrix} \otimes I_s \in \mathbb{C}^{js \times js}. \quad (6c)$$

Remark 1. In practice, although (1) is defined by real A, B , complex shift parameters can occur. We assume that the set of shifts $\{\alpha_1, \dots, \alpha_j\}$ is closed under complex conjugation and that pairs of complex conjugated shifts occur subsequently, i.e., $\alpha_{i+1} = \bar{\alpha}_i$ if $\operatorname{Im}(\alpha_i) \neq 0$. These complex parameters pairs are in practice dealt within the LR-ADI iteration by a double step fashion [7, 9, 24] resulting in real low-rank factors Z_j and, important for this study, real low-rank residual factors W_j . Real version 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 residual R_j can be expressed as

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

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

is a Cayley transformation of A . Taking norms leads to

$$\frac{\|X - X_j\|}{\|X - X_0\|} \leq c \rho(\mathcal{M}_j)^2, \quad \frac{\|R_j\|}{\|R_0\|} \leq c \rho(\mathcal{M}_j)^2, \quad c \geq 1.$$

Because of $\Lambda(A) \subset \mathbb{C}_-$ as well as $\alpha_i \in \mathbb{C}_-$ it holds $\rho(\mathcal{C}(A, \alpha_i)) < 1$, $1 \leq i \leq j$ 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

$$\min_{\alpha_1, \dots, \alpha_j \in \mathbb{C}_-} \left(\max_{\lambda \in \Lambda(A)} \left| \prod_{i=1}^j \frac{\lambda - \bar{\alpha}_i}{\lambda + \alpha_i} \right| \right). \quad (7)$$

Several shift selection strategies have been developed based on (7), e.g., the often used Wachspress [42, 49] and Penzl [35] selection approaches, which precompute a number of shifts before the actual LR-ADI iteration. There, the spectrum $\Lambda(A)$ in (7) 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 [35]. Starting from (7) for selecting shifts has, however, some shortcomings. A disadvantage from the conceptual side is that the min-max problem (7) does only take (approximate) eigenvalues of A into account. No information regarding the inhomogeneity BB^* of the CALE (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, 20, 47]. Furthermore, no information regarding the eigenvectors of A enters (7). 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 provide in advance. Even tiny changes in these quantities can greatly alter the speed of the error or residual reduction in the ADI iteration. Because the strategies based on (7) 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 incorporate 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 $\mathcal{Q}_\ell = \text{range}(Q_\ell) \subseteq \text{range}(Z_j)$, 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}_ℓ :

$$\mathcal{Q}_\ell = \mathcal{Z}(h) := \text{range}([V_{j-h+1}, \dots, V_j]) \quad (8)$$

with $1 \leq h \leq j$ to keep the space dimension small. The Ritz values are given by $\Lambda(H_\ell)$ with $H_\ell := Q_\ell^* A Q_\ell$ and can, e.g., be plugged into the Penzl heuristic to select $g \leq \ell$ shift parameters. It can happen that $\Lambda(H_\ell) \cap \mathbb{C}_+ \neq \emptyset$ in which case we simply 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 significant speed ups of the LR-ADI iteration, in particular for nonsymmetric problems where the a priori computed shifts resulted in a very slow convergence. Due to this success, 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 [15, 17, 28, 51, 52] we can borrow the greedy shift selection strategy from [16] which was developed for the rational Krylov subspace method for (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. (7)) 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 to the full already computed rational Krylov subspace, while in LR-ADI we only use a smaller subspace (8).

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

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

then $X = PQ^{-1}$ solves (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 $A\hat{X} + \hat{X}A^T = -R_j$ [24, Corollary 3.8], where $R_j = W_jW_j^*$ is the residual associated to 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 $\tilde{\mathcal{H}}_{j,\ell} := \begin{bmatrix} H_\ell^* & 0 \\ Q_\ell^*W_jW_j^*Q_\ell & -H_\ell \end{bmatrix}$, 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 1). The goal is to find a shift α_{j+1} for the next iteration step. By Theorem 1 it holds for the next Lyapunov residual $\|R_{j+1}\| = \|W_{j+1}\|^2$ with

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

Obviously, for executing step $j+1$ everything except the shift α_{j+1} is known. The parameter $\alpha_{j+1} \in \mathbb{C}_-$ is to be determined which reduces the Lyapunov residual norm the most from step j to $j+1$. This minimization problem can be formulated as, e.g., a complex nonlinear least squares problem (NLS)

$$\begin{aligned} \alpha_{j+1} &= \operatorname{argmin}_{\alpha \in \mathbb{C}_-} \frac{1}{2} \|\Psi_j(\alpha, \bar{\alpha})\|^2, \\ \Psi_j(\alpha, \bar{\alpha}) &= \mathcal{C}(A, \alpha)W_j = (A - \bar{\alpha}I)(A + \alpha I)^{-1}W_j. \end{aligned} \tag{9}$$

The complex function $\Psi_j(\alpha, \bar{\alpha}) : \mathbb{C} \mapsto \mathbb{C}^{n \times s}$ is obviously not analytic in the complex variables $\alpha, \bar{\alpha}$ alone but in the full variable $(\alpha, \bar{\alpha})$, a property typically referred to as polyanalyticity. In their original appearance [9], the residual minimizing shifts were considered via the minimization of the real valued function $\psi_j = \|W_{j+1}\|^2$, which corresponds to the Lyapunov residual norm after using the shift parameter α starting from R_j . The complex minimization problem takes the form

$$\alpha_j = \operatorname{argmin}_{\alpha_j \in \mathbb{C}_-} \psi_j^{\mathbb{C}}(\alpha, \bar{\alpha}), \quad \psi_j^{\mathbb{C}}(\alpha, \bar{\alpha}) := \|\Psi_j(\alpha, \bar{\alpha})\|^2. \tag{10}$$

It is clear that (9) and (10) essentially encode the same optimization task but differences will occur in the numerical treatment of both formulations. Splitting the complex variable into two real ones by $\alpha = \nu + j\xi$ with $\nu < 0$ yields that real and imaginary parts of the next shift $\alpha_{j+1} = \nu_{j+1} + j\xi_{j+1}$ can be obtained from solving

$$\begin{aligned} [\nu_{j+1}, \xi_{j+1}] &= \operatorname{argmin}_{\nu \in \mathbb{R}_-, \xi \in \mathbb{R}} \psi_j(\nu, \xi), \\ \psi_j &= \psi_j(\nu, \xi) := \|W_j - 2\nu((A + (\nu + j\xi)I)^{-1}W_j)\|^2. \end{aligned} \tag{11}$$

Note that the objective function ψ in (11) always maps real variables to real values, whereas the function Ψ in the NLS formulation (9) is only real for $\alpha \in \mathbb{R}$. This would still be the case if α in (9) was decomposed in ν, ξ as we did for (11). Since $\|X\|_2 = \lambda_{\max}(X^*X)$, the minimization problems (10) and (11) can also be understood as eigenvalue optimization problems if $s > 1$.

Naturally, if one knows that real shift parameters are sufficient, e.g. when $A = A^*$, the above minimization problems simplify in the obvious manner by restricting the optimization to \mathbb{R}_- . For achieving reduction of the residual as well as avoiding the singularities at $-\Lambda(A) \subset \mathbb{C}_+$, the constraint $\nu < 0$ is mandatory so that tools from constrained optimization are required. Originally, an unconstrained version of (11) and derivative-free methods were used to find a minimum of ψ_j [9]. This strategy turned out to be unreliable, in particular, because unusable shifts ($\nu \geq 0$) were frequently generated. In this article, we employ constrained, derivative based optimization approaches using the complex nonlinear least squares (9) and the

real valued minimization problem (11). 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 subsection gives the structure of the required derivatives of $\Psi_j(\alpha, \bar{\alpha})$, $\psi_j^{\mathbb{R}}$. 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 (9), the Jacobian and conjugate Jacobian [45] of $\Psi_j(\alpha, \bar{\alpha})$ are

$$\begin{aligned}\frac{\partial \Psi_j(\alpha, \bar{\alpha})}{\alpha} &= -(A - \bar{\alpha}I)(A + \alpha I)^{-2}W_j = \mathcal{C}(A, \alpha)(A + \alpha I)^{-1}W_j, \\ \frac{\partial \Psi_j(\alpha, \bar{\alpha})}{\bar{\alpha}} &= -(A + \alpha I)^{-1}W_j.\end{aligned}\tag{12}$$

The structure of the derivatives for ψ_j in (11) is more complicated.

Theorem 3 (Gradient and Hessian of the objective function (11)). Let $\alpha = \nu + j\xi \in \mathbb{C}_-$, $W = W_j \in \mathbb{R}^{n \times s}$, and define $L(\nu, \xi) := A + \alpha I$,

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

for $i = 0, \dots, 3$. Assume $(W_\alpha^{(0)})^* W_\alpha^{(0)}$ has s distinct eigenvalues $\theta_1 > \dots > \theta_s > 0$ and let $(\theta_\ell, u_\ell) = (\theta_\ell(\nu, \xi), u_\ell(\nu, \xi))$ with $\|u_\ell\| = 1$, $\ell = 1, \dots, s$ be its eigenpairs. Then, gradient and Hessian of (11) are given by

$$\text{grad } \psi_j = 4 \begin{bmatrix} \text{Re}(u_1^* ((W_\alpha^{(0)})^* \hat{W}^{(1)})_{u_1}) \\ -\nu \text{Im}(u_1^* ((W_\alpha^{(0)})^* S^{(2)})_{u_1}) \end{bmatrix} = 4 \begin{bmatrix} \text{Re}(u_1^* \tilde{R}_\nu u_1) \\ -\nu \text{Im}(u_1^* \tilde{R}_\xi u_1) \end{bmatrix}\tag{13}$$

and

$$\text{grad } \psi_j = 8 \begin{bmatrix} \text{Re}(u_1^* ((\hat{W}^{(2)})^* W_\alpha^{(0)} + (\hat{W}^{(1)})^* \hat{W}^{(1)})_{u_1}) & h_{12} \\ h_{12} & \nu \text{Re}(u_1^* ((S^{(3)})^* W_\alpha^{(0)} + \nu (S^{(2)})^* S^{(2)})_{u_1}) \end{bmatrix}\tag{14a}$$

$$+ \sum_{k=2}^s \frac{8}{\theta_1 - \theta_k} \begin{bmatrix} |u_1^* (\tilde{R}_\nu^* + \tilde{R}_\nu) u_k|^2 & \tilde{h}_{12}^{(k)} \\ \tilde{h}_{12}^{(k)} & |u_1^* (\tilde{R}_\xi^* - \tilde{R}_\xi) u_k|^2 \end{bmatrix}\tag{14b}$$

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

Proof. The results are obtained by building the partial derivatives of $\hat{\mathcal{C}}(A, \alpha = \nu + j\xi)$ and of $\psi_j(\nu, \xi) = \sigma_{\max}^2(\mathcal{C}(A, \nu + j\xi)W) = \lambda_{\max}(\Psi^* \Psi) = \lambda_{\max}((W_\alpha^{(0)})^* W_\alpha^{(0)})$ using results on derivatives of eigenvalues of parameter dependent matrices, e.g., [25, 30]. For detailed proof the reader is referred to [24, Section 5], where also the formulas adapted to (2) are given. \square

3.2 Approximating the objective functions

The main issue arising when solving the optimization problems (9), (11) is that each evaluation of the objective functions Ψ_j , ψ_j at a value α requires one linear solve with $A + \alpha I$. Moreover, the derivative formula reveal that evaluating derivatives requires, depending on the order of the derivatives and on the used optimization problem, at least one additional linear solve. Thus, each evaluation within a derivate-based optimization method will be more expensive than a single LR-ADI iteration step, making the numerical solution of (9), (11) very costly, regardless of the employed optimization algorithm, and, consequently, the shift generation would be prohibitively expensive.

As main contribution of this paper, this section proposes strategies to work with approximations of the objective functions $\tilde{\Psi}_j \approx \Psi_j$, $\tilde{\psi}_j \approx \psi_j$ which can be much cheaper evaluated and whose construction is numerically efficient as well. The resulting shifts are generated by using $\tilde{\psi}_j$ in (11) and $\tilde{\Psi}_j$ in (9). 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

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

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 $\hat{\Psi}_j(\alpha, \bar{\alpha}) := \mathcal{C}(H_\ell, \alpha) \tilde{W}_{\ell,j}$. 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 (9), (11) 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 [24] it is suggested to augment the Ritz value based shifts (Section 2.1.1) by the optimization problem (11) using $\tilde{\psi}_j$, i.e., after step j , the space $\mathcal{Q} = \mathcal{Z}(h)$ spanned by the last $1 \leq h \leq j$ block columns of the already generated low-rank solution factor $Z_j = [V_1, \dots, V_j]$ is selected as in (8). The reduced objective function $\tilde{\psi}_j$ is then built by H_ℓ and $\tilde{W}_{\ell,j}$. The restriction H_j of A can be built without additional multiplications with A because of (6). Let $R_j \in \mathbb{C}^{hs \times hs}$ so that $Q_j = [V_{j-h+1}, \dots, V_j] R_j$ has orthonormal columns. Then

$$H_j := Q_j^* A Q_j = Q_j^* W_j G_{j,h}^* R_j - R_j^{-1} S_{j,h}^* R_j, \quad (15)$$

where $G_{j,h}$, $S_{j,h}$ indicate the last h block rows (and columns) of G_j , S_j from (6). Even though this space selection is rather intuitive, it led to impressive results often outperforming existing shift selecting strategies in [24]. The obtained rate of the residual norm reduction in the LR-ADI iteration was very close to the case when the the exact objection function was used in (11), 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 for other problems, 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:

$$\mathcal{Q} = \mathcal{K}_p(A, W_j) := \text{span}\{W_j, AW_j, \dots, A^{p-1}W_j\}.$$

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., [18, 19, 23, 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.,

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

and, moreover, $\text{grad}^2 \psi_j$ 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

$$\begin{aligned}\mathcal{Q} &= \mathcal{E}_{p,m}(A, W_j) := \mathcal{K}_p(A, W_j) \cup \mathcal{K}_m(A^{-1}, A^{-1}W_j) \\ &= \text{span}\{W_j, AW_j, \dots, A^{p-1}W_j, A^{-1}W_j, \dots, A^{-m}W_j\}\end{aligned}$$

as projection subspace. 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, for approximating Ψ_j , ψ_j , the right hand side W_j changes throughout the ADI iteration, which would necessitate to construct a new orthonormal basis associated to $\mathcal{E}_{p,m}(A, W_j)$ in each LR-ADI iteration step. As an auxiliary contribution, the next theorem shows that this is not needed for

- (b) 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 $\text{Re}(\alpha) < -\phi < 0$. 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.
- (c) A intuitive extension would be the use of rational Krylov subspaces

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

with poles $\beta = \{\beta_1, \dots, \beta_r\}$. The motivation for this choice is to approximate Ψ_j, ψ_j at values β_i , $1 \geq i \geq r$ that may lie in the interior of the optimization region. However, this requires knowledge of adequate shifts β_i such that the relevant behavior of Ψ_j, ψ_j is captured. Exactly this makes the rational Krylov approximation problematic, since it is currently not known where in \mathbb{C}_- the local minima of Ψ_j, ψ_j are located and where suitable β_i should be placed. Therefore, we will restrict in the remainder to the standard and extended Krylov subspace approaches. The basis construction for $\mathcal{K}_r^{\text{rat}}(A, W_j, \beta)$ can be build similarly from an a priori generated basis of $\mathcal{K}_r^{\text{rat}}(A, B, \beta)$ obtained by a rational (block) Arnoldi process [22, 38]. If $\beta_i = \alpha_i$, $1 \leq i \leq r \leq j$, it is well known that $\mathcal{K}_r^{\text{rat}}(A, B, \beta) \subseteq \text{range}(Z_j)$ [17, 29, 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

Before the approaches for solving the optimization problems are investigated, the numerical implementation of the proposed strategy for approximating the objective function Ψ_j, ψ_j within the LR-ADI iteration is discussed. Assume that before the LR-ADI iteration is started, a block extended Arnoldi process [43] with orders p, m is applied to A, B which provides

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

$Q_B(1 : s, :) \eta = B$, $\eta \in \mathbb{R}^{s \times s}$, 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 w.r.t. $\mathcal{E}\mathcal{K}_{p,m}(A, B)$. For later use, Q_B, P_B , and H_B are stored. If no shift parameters for the first LR-ADI steps are provided, a residual-norm minimizing shift α_1 is computed by solving a reduced optimization problem (11) defined by H_B and $\tilde{W}_0 := Q_B^* B = [\eta^*, 0, \dots, 0]^* \in \mathbb{R}^{(p+m)s \times s}$. Suppose j steps of the LR-ADI iterations have been carried out and we look for a shift α_{j+1} for the next step by using a reduced objective function constructed from the approximation space $\mathcal{K}_p(A, W_j)$. Motivated by Theorem 4, the augmented basis matrix (17) w.r.t. the augmented space $\mathcal{E}\mathcal{K}_{p,m}(A, B) \cup \text{span}\{\omega_j\}$, 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 $\mathcal{E}\mathcal{K}_{p,m}(S_j, G_j)$. 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. (5), (6a)). The matrix ω_j gives the relevant part of $\text{range}(Z_j)$ needed for $\mathcal{E}\mathcal{K}_{p,m}(A, W_j)$. If $j \leq p + m$, Z_j has less than or exactly $(p + m)s$ columns, and $w_j := Z_j$ is used as simplification. To orthogonally extend the basis of $\mathcal{E}\mathcal{K}_{p,m}(A, B)$ by $\text{span}\{\omega_j\}$, we may employ any stable orthogonalization routine, e.g., an iterative block Gram Schmidt process. Consider for illustration one sweep of block Gram Schmidt

$$h_j := Q_B^* \omega_j \in \mathbb{C}^{(p+m)s \times (p+m)s}, \quad \hat{\omega}_j := \omega_j - Q_B h_j, \quad Q_{Z_j} := \hat{\omega}_j \hat{h}_j,$$

where \hat{h}_j orthonormalizes the columns of $\hat{\omega}_j$ and is obtained by a thin QR decomposition of $\hat{\omega}_j$. Hence, $Q_j := [Q_B, Q_{Z_j}] \in \mathbb{C}^{n \times 2(p+m)s}$ is the sought orthogonal basis matrix, $\tilde{W}_j := Q_j^* W_j$, and

$$H_j := Q_j^* A Q_j = \begin{bmatrix} H_B & \\ & (Q_{Z_j})^* P_B \end{bmatrix} Q_j^* P_{Z_j} \in \mathbb{C}^{2ps \times 2ps}, \quad P_{Z_j} := A Q_{Z_j}. \quad (19a)$$

The additional $p + m$ matrix vector products with A can be avoided by constructing P_{Z_j} in conjunction with the Gram-Schmidt orthogonalization of ω_j against Q_B and using (6a):

$$\hat{P} := A w_j = A Z_j Q_{S_j} = B G_j^* Q_{S_j} + Z_j S_j Q_{S_j}, \quad (19b)$$

$$P_{Z_j} = A Q_{Z_j} = A \hat{\omega}_j \hat{h}_j = (A w_j - A Q_B h_j) \hat{h}_j = (\hat{P} - P_B h_j) \hat{h}_j. \quad (19c)$$

Unless $A + A^* \prec 0$, it can happen that the restriction H_j has unstable eigenvalues which would be problematic for the usage of the compressed objective functions. As a basic counter measure, we replace H_j

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 α_{j+1} for LR-ADI iteration

```

1 if  $j > 1$  then
2   if  $p > 0$  and  $m > 0$  then
3     if  $j \leq p + m$  then
4       | Set  $Q_{S_j} = 1$ .
5     else
6       | Generate orthonormal basis  $Q_{S_j} \in \mathbb{C}^{s_j \times (p+m)s}$  for  $\mathcal{E}_{p,m}(S_j, G_j)$  with  $S_j, G_j$  from (5), (6).
7       |  $Q_j = \text{orth}[Q_B, Z_j Q_{S_j}]$ .
8     else
9       |  $Q_j = \text{orth}[Z_j(:, (j - \min(j, h))s + 1 : js)]$ .
10       $H_j = Q_j^*(AQ_j)$  (using (15) or (19)),  $\tilde{W}_j := Q_j^*W_j$ .
11 else
12    $H_j = H_B, \tilde{W}_j = Q_B^*W_0 (= Q_B^*B)$ .
13 Compute Schur form  $H_j \leftarrow Q_{j,H}^*H_jQ_{j,H}$  (negate unstable eigenvalues on demand),  $\tilde{W}_j \leftarrow Q_{j,H}^*\tilde{W}_j$ 
14 Find local minimizer  $\alpha_{j+1} = \nu + \gamma\xi \in \mathbb{C}_-$  by solving compressed optimization problems (9), (11)
    defined by  $H_j, \tilde{W}_j$ .
```

by its Schur form $H_j \leftarrow Q_{j,H}^*H_jQ_{j,H}$ and simply negate any arising unstable eigenvalues that appear on the diagonal of H_j . Transforming the compressed objective function 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.

The generation of the shift α_{j+1} for the next LR-ADI step $j + 1$ is summarized in Algorithm 2 including both projection subspace choices from Sections 3.2.1 and 3.2.2.

Dealing with Generalized Lyapunov Equations In practice often generalized Lyapunov equations (2) arise with an additional, invertible matrix $M \in \mathbb{R}^{n \times n}$. The LR-ADI iteration for (2) is given by

$$V_j = (A + \alpha_j M)^{-1}W_{j-1}, \quad W_j = W_{j-1} + \gamma_j^2 M V_j, \quad W_0 := B, \quad (20)$$

(see, e.g. [8, 24]) leading to generalizations of the objective functions

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

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

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

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, minor complications arise because these subspaces are only defined by a single $n \times n$ matrix. This can be dealt with by defining, e.g., $A_M := M^{-1}A, W_{M,j} := M^{-1}W_j$ and using

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

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 (20) the relations (6) 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

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

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 standard literature on numerical optimization [33].

The constraints in (9)–(11) can for practical purposes be given by

$$\nu_- \leq \nu \leq \nu_+, \quad 0 \leq \xi \leq \xi_+, \quad -\infty < \nu_- < \nu_+ < 0, \quad \xi \in \mathbb{R}_+, \quad (21)$$

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 ν_{\pm}, ξ_+ by approximate spectral data of A using the extremal eigenvalues of H_j . Often, optimization algorithms require an initial guess to start, and the first value returned by any of the projection based shift selection approaches (Section 2.1.1) can be employed for this. 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 square problems (9) with constraints (21) we use the routine `nlsb_gnd1` from the `Tensorlab` software package [48]. The routine `nlsb_gnd1` is based on a projected Gauss-Newton type method. In principle, the functionality of `Tensorlab` would also allow to solve the complex minimization problem (10). Only if (9) is restricted to real variables $\alpha \in \mathbb{R}_-$, the routine `lsqnonlin` of the `MATLAB`[®] optimization toolbox[™] is an alternative. For solving the real valued constrained minimization problem (11) in real variables, a large variety of software solutions is available. The `MATLAB` optimization toolbox[™] provides with the `fmincon` routine a general purpose optimizer, which comes with different choices for the internal optimization algorithms, e.g., interior-point [50] and trust-region-reflective algorithms [12], which both allow to specify hard-coded Hessians via the explicit formulas (14).

However, when $s > 1$ the assumption in Theorem 3 that the parameter dependent matrix $(W_\alpha^{(0)})^* W_\alpha^{(0)}$ has s distinct eigenvalues $\theta_i(\nu, \xi)$, $1 \leq i \leq s$ 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 do not exist, see, e.g., [13, 26, 27, 30, 31, 34]. Practical observations show that especially minima of ψ_j are often attained at those points. The same problems are also present for the reduced optimization problem with $\tilde{\psi}_j$. Obviously, if $s = 1$, such issues are not present which motivated the simple approach in [24] to prevent these instances by simply transforming the eigenvalue optimization to a scalar optimization problem. This can be achieved by multiplying the compressed residual factors \tilde{W}_j with an appropriate tangential vector $\hat{W}_j = \tilde{W}_j t$, where 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

$$\hat{W}_j = \tilde{W}_j t, \quad \hat{\psi}_j := \|\hat{W}_j - 2\nu(H_j + (\nu + j\xi)I)^{-1}\hat{W}_j\|^2. \quad (22)$$

Although this transformation is an additional approximation step regarding the original function ψ_j , numerical experiments in, e.g., [24] 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 (22). Methods based on the BFGS framework are capable of solving non-smooth optimization problems [13, 27] provided a careful implementation is used that allows to deal with points where the objective function does not have derivatives. The `GRANSO` package [13] provides `MATLAB` implementations of these BFGS type methods and will also be tested in the numerical experiments for (11) without the modification (22).

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. Theoretically, this may be generalized towards the generation of shifts to be used in multiple, say $g > 1$, future LR-ADI steps to reduce $\|W_{j+g}\|^2$ the most starting from $\|W_j\|^2$. The NLS formulation for $g \geq 1$ takes the form

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

Since we always assumed that if $\alpha_i \in \mathbb{C}_-$ also its complex conjugate is used (Remark 1), the above approach could yield parameters for up to $2g$ future LR-ADI steps. Obviously, solve this multistep optimization

problem is harder than the single-step one. For instance, since the order in which the shifts are applied is not important we have $\Psi_{j,j+g}(\alpha, \bar{\alpha}) = \Psi_{j,j+g}(\Pi_g \alpha, \Pi_g \bar{\alpha})$ 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 g times in succession. A similar generalization of (11) can be found in [24], where no substantial improvements over the single shift approach are reported.

A particular interesting special situation is when the $g > 1$ future shift parameters are restricted to be equal, $\alpha_{j+i} = \alpha_{j+1}$, $1 \leq i \leq g$. We point out that similar multi-step approaches were investigated for Smith-type methods in, e.g., [1, 21, 36, 46]. Although this restriction will likely slow down the convergence compared to different shifts in each step, it can be practical for reducing the computation time for solving the large-scale linear systems in the LR-ADI iteration. 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: $V_{j+i} = U^{-1}L^{-1}W_{j+i-1}$, $i \leq 1 \leq g$. 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. Hence, smaller overall computation times of LR-ADI can be achieved at the price of a slower Lyapunov residual reduction and larger generated low-rank factors. 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 supposed to be used in $g \geq 1$ iteration steps in residual norm minimization approach. The associated multi-shift NLS formulation is

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

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

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

By the same reasoning as in Section 3.2.2, these formula 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. The function minimization approach is extended in the same way by defining the scalar function $\psi_{j,j+g}(\nu, \xi) := \|\mathcal{C}(A, \nu + j\xi)^g W_j\|^2$. Theorem 3 for the derivatives of $\psi_{j,j+g}$ can easily be reformulated by using

$$W_\alpha^{(r)} := g \mathcal{C}(A, \alpha)^{g-1} W_j = g (I - 2\nu L(\nu, \xi)^{-1})^{g-1} W_j \quad (23)$$

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 @ 2.7GHz with 16 GB RAM. We wish to obtain an approximate solution such that the scaled Lyapunov residual norm satisfies

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

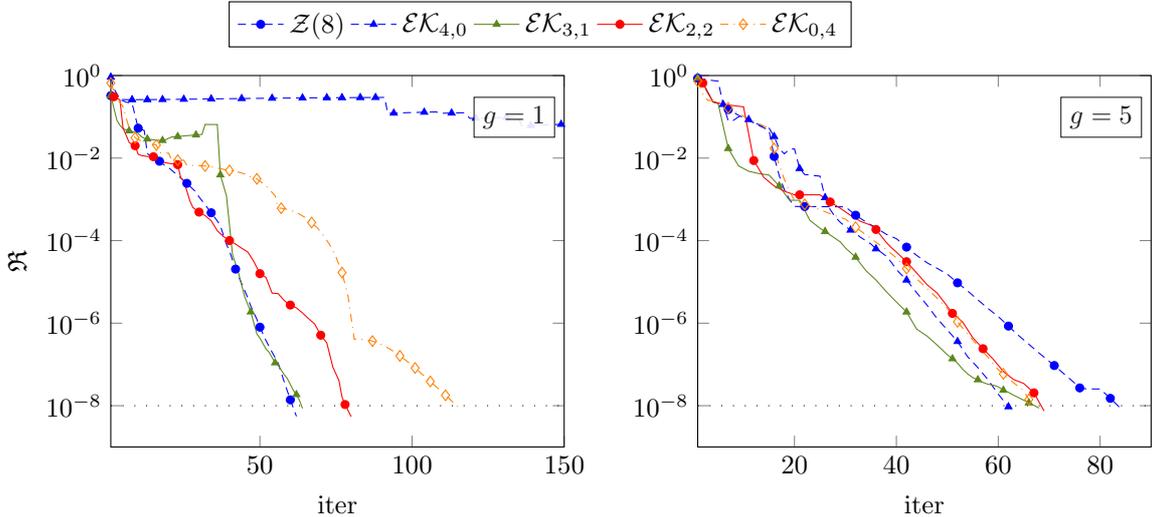
Table 1 summarizes the used test examples. The right hand side factors B for all examples except `chip` are generated randomly with uniformly distributed entries, where the random number generator is initialized by `randn('state', 0)` before each test. The maximal allowed number of LR-ADI steps is restricted 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

Table 1: Overview of examples

Example	n	s	details	ε
<i>cd2d</i>	40000	1, 5	finite difference discretization of 2d operator $\mathcal{L}(u) = \Delta u - 100x \frac{\partial u}{\partial x} - 1000y \frac{\partial u}{\partial y}$ on $[0, 1]^2$, homogeneous Dirichlet b.c.	1e-8
<i>cd3d</i>	27000	10	finite difference discretization of 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 $[0, 1]^3$, homogeneous Dirichlet b.c.	1e-8
<i>lung</i>	109460	10	model of temperature and water vapor transport in the human lung from suitesparse collection [14]	1e-8
<i>chip</i>	20082	5	finite element model of chip cooling process [32], $M = M^* \neq I$	1e-10

Figure 1: Residual norm history of LR-ADI iteration using different objective function approximations for *cd2d* example. Left: single step shift selection, right: multi step shift selection with $g = 5$.

LR-ADI iteration. This experiment is carried out on the *cd2d* example with a single vector in B and $\|B\| = 1$. The NLS formulation (9) is employed and dealt with by the Tensorlab routine `nlsb_gnd1`. 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. By the proposed subspace construction, the dimension of the basis is in all cases at most 8. Moreover, the experiment is carried out in the single-step as well as multi-step fashion with $g = 5$ from Section 4. Figure 1 and Table 2 summarize the obtained results. Apparently, for the single step optimization approach, using $\mathcal{Z}(h)$ as approximation space seems to yield the best shifts 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 over $g = 5$ steps, where the extended Krylov approximations of the objective function yield better shifts, i.e., less iteration steps compared to using $\text{range}(Z_j)$. 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 (only in every 5th step or even less if the generated shift is complex), the savings in the computation times reported in Table 2 are substantial. To conclude, while the standard objective function approximation using $\text{range}(Z_j)$ seems to work satisfactory in most cases for the single step shift selection, for $g > 1$ better results might be obtained by the extended Krylov approximations proposed in this work.

5.2 Choice of the optimization routine

Now we test different optimization problem formulations (9), (11) as well as different optimization routines for the *cd3d* example having $s = 10$ columns in B . As in the previous experiment, the Tensorlab routine `nlsb_gnd1` is used for the NLS problem, but for the function minimization problem (11) 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 (22) to avoid the potential nonsmoothness of ψ_j in (11) 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 3 summarizes the results. Judging from the number of required LR-ADI steps, the usage of different optimization

Table 2: Results with different projection subspaces for objective function approximation using the `cd2d` example. For single and multi step approaches ($g = 5$), listed are the executed iteration numbers (iters), total and shift computation times (t_{total} , t_{shift}) in seconds, and final residual norm \mathfrak{R} (res).

projection space	single step ($g = 1$)				multi step ($g = 5$)			
	iters	t_{total}	t_{shift}	res	iters	t_{total}	t_{shift}	res
$\mathcal{Z}(8)$	62	19.0	3.1	5.5e-09	84	8.3	1.7	8.7e-09
$\mathcal{EK}_{4,0}$	151	53.3	8.2	6.7e-02	62	6.9	1.3	9.4e-09
$\mathcal{EK}_{3,1}$	64	22.4	4.8	8.6e-09	68	7.5	1.7	8.8e-09
$\mathcal{EK}_{2,2}$	80	26.2	4.9	5.4e-09	69	7.5	1.7	7.4e-09
$\mathcal{EK}_{0,4}$	115	38.3	7.8	9.5e-09	67	8.3	1.9	9.5e-09

Table 3: Results with different optimization routines for the `cd3d` example.

opt. problem	opt. routine	iters	t_{total}	t_{shift}	res
NLS (9)	<code>nlsb_gnd1</code>	52	98.3	2.4	2.9e-09
NLS (22)	<code>nlsb_gnd1</code>	47	73.0	1.6	7.8e-09
fun.min. (11)	<code>fmincon</code> with int. point method	48	70.8	2.5	5.6e-09
fun.min. (22)	<code>fmincon</code> with int. point method	50	77.2	2.1	4.4e-09
fun.min. (11)	<code>fmincon</code> with thrust region reflective method	49	78.4	2.1	4.7e-09
fun.min. (22)	<code>fmincon</code> with thrust region reflective method	51	76.9	1.7	2.0e-09
fun.min. (11)	GRANSO	49	89.2	14.5	7.3e-09

routines appears to have less impact than working with different objective function approximations. The additional tangential approximation (22) seems to slow down the LR-ADI iteration only marginally. The exception for this is when the NLS formulation (9) and `nlsb_gnd1` is used, where 5 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 (22) 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 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 (21) 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 5.3.

We also run a few tests with the multishift approach with $g = 5$. For each example, we also compare the LR-ADI to the rational Krylov subspace method [15] (RKSM) equipped with the convex-hull based shift selection [16]. The reduced Lyapunov equation in RKSM is solved in every 5th step.

Table 5 summarizes the results and Figure 2 shows the history of the scaled Lyapunov residual norms for some selection approaches and the `cd3d`, `chip` examples. 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. 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

Table 4: Overview of employed shift selection strategies.

type	abbreviation	description of strategy	info
precomputed	$\text{heur}(J, p, m)$	heuristic selection of $J \in \mathbb{N}$ shifts from Ritz values associated to $\mathcal{EK}_{p,m}(A, B\mathbf{1}_s)$, cyclic usage	[35, 40]
	$\text{Wachs}(\epsilon, p, m)$	Wachspres selection using Ritz values associated to $\mathcal{EK}_{p,m}(A, B\mathbf{1}_s)$ and tolerance $0 < \epsilon \ll 1$, cyclic usage	[40, 42, 49]
adaptive	$\mathcal{Z}(h)+\text{heur}$	projection based shifts using newest h block columns of Z_j and selection via heuristic	[9, 24], Section 2.1.1
	$\mathcal{Z}(h)+\text{conv}$	projection based shifts as above, but convex hull based selection	[16], Section 2.1.2
	$\mathcal{Z}(h)+\text{Hres}$	projection based shifts as above, but residual Hamiltonian based selection	[6], Section 2.1.3
	$\text{resmin}+Q+\text{OS}$	residual norm minimizing shifts with Q as approximation space and OS as optimization routine	[9, 24], Section 3

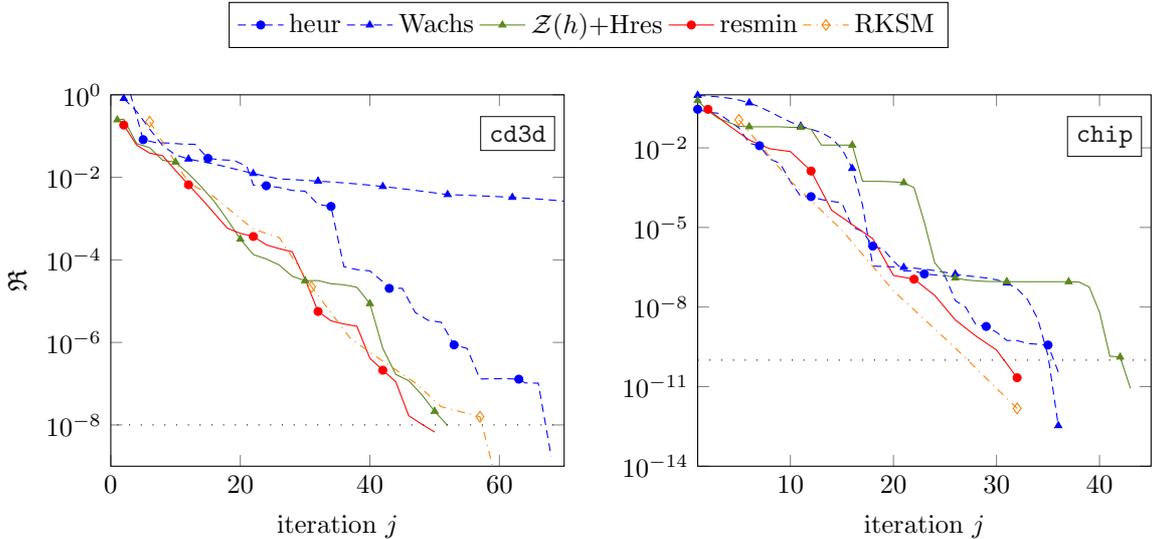


Figure 2: Residual norm history of LR-ADI iteration and RKSM using different shift selection strategies.

converge faster than LR-ADI [15]. Among the Ritz value based shift selection techniques (Section 2.1.1) for LR-ADI, the Residual-Hamiltonian selection (Section 2.1.3, $\mathcal{Z}(h)+\text{Hres}$) appears to perform best, leading to iteration numbers close to the ones obtained with the residual minimizing shifts. The precomputed shift approaches ($\text{heur}(J, p, m)$, $\text{Wachs}(\epsilon, p, m)$) 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} . Due to the need to solve (compressed) optimization problems, the generation times of the residual minimizing shifts was in some cases slightly higher compared to the other approaches. Although the multishift selection approach yielded higher iteration numbers, they led to a substantial reduction in the computation times t_{total} because of the reuse of LU factorizations over several iteration steps.

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 multishift

Table 5: Comparison of different shift routines and comparison against RKSM

ex.	method	shift selection strategy	iters	t_{total}	t_{shift}	res
cd2d	LR-ADI	heur(20, 30, 20)	137	35.3	0.8	9.0e-09
		wachs(10^{-8} , 30, 20)	93	24.0	0.8	7.2e-09
		$\mathcal{Z}(4)$ +heur	74	20.7	0.7	9.2e-09
		$\mathcal{Z}(4)$ +conv	80	21.5	2.2	6.1e-09
		$\mathcal{Z}(4)$ +Hres	74	19.4	1.6	2.8e-09
		resmin+ $\mathcal{Z}(4)$ +fmincon	60	19.8	4.0	3.0e-09
	RKSM	convex hull	61	39.0	4.5	4.3e-09
	cd3d	LR-ADI	heur(20, 40, 30)	68	85.8	1.9
wachs($1e-08$, 30, 20)			150	188.2	1.4	5.2e-04
$\mathcal{Z}(4)$ +heur			71	89.1	0.3	1.9e-09
$\mathcal{Z}(4)$ +conv			57	70.5	1.3	2.3e-09
$\mathcal{Z}(4)$ +Hres			52	64.9	1.1	9.7e-09
resmin+ $\mathcal{Z}(4)$ +fmincon			50	65.7	1.9	6.7e-09
RKSM		resmin+ $\mathcal{EK}_{1,1}$ +nlsb_gnd1, $g = 5$	59	23.4	0.9	6.2e-09
convex hull		61	115.8	7.7	5.6e-11	
lung	LR-ADI	heur(20, 30, 20)	150	60.5	4.0	1.7e-08
		wachs($1e-8$, 30, 20)	150	56.8	4.2	2.8e-02
		$\mathcal{Z}(2)$ +heur	94	37.5	0.9	4.6e-10
		$\mathcal{Z}(2)$ +conv	80	36.2	5.7	3.6e-09
		$\mathcal{Z}(2)$ +Hres	71	31.7	4.1	8.5e-09
		resmin+ $\mathcal{Z}(2)$ +fmincon	65	29.5	3.5	8.6e-09
	RKSM	resmin+ $\mathcal{Z}(2)$ +GRANSO, $g = 5$	69	8.4	1.7	9.5e-09
	convex hull	67	126.9	7.4	1.9e-09	
chip	LR-ADI	heur(10, 20, 10)	33	24.5	1.2	7.1e-11
		wachs($1e-12$, 20, 10)	34	24.4	1.2	5.0e-13
		$\mathcal{Z}(4)$ +heur	70	49.8	0.3	5.6e-11
		$\mathcal{Z}(4)$ +conv	55	39.1	0.3	5.6e-11
		$\mathcal{Z}(4)$ +Hres	43	30.5	0.2	8.6e-12
		resmin+ $\mathcal{Z}(4)$ +fmincon	32	27.0	0.9	2.2e-11
	RKSM	resmin+ $\mathcal{EK}_{1,1}$ +nlsb_gnd1, $g = 5$	39	9.5	0.1	7.8e-11
	convex hull	32	25.7	1.7	1.5e-12	

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, 24] and should be further refined in future research. Deriving a similar multishift 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

Thanks 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.

References

- [1] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemp. Math.*, 280:193–219, 2001.
- [2] A. C. Antoulas, D. C. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *Syst. Cont. Lett.*, 46(5):323–342, 2002.
- [3] J. Baker, M. Embree, and J. Sabino. Fast singular value decay for Lyapunov solutions with nonnormal coefficients. *SIAM J. Matrix Anal. Appl.*, 36(2):656–668, 2015.

- [4] B. Beckermann and A. Townsend. On the Singular Values of Matrices with Displacement Structure. *SIAM J. Matrix Anal. Appl.*, 38(4):1227–1248, 2017.
- [5] P. Benner and Z. Bujanović. On the solution of large-scale algebraic Riccati equations by using low-dimensional invariant subspaces. *Linear Algebra Appl.*, 488:430–459, 2016.
- [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(2):301–330, Feb. 2018.
- [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(2):225–251, 2013.
- [8] P. Benner, P. Kürschner, and J. Saak. An improved numerical method for balanced truncation for symmetric second order systems. *Math. Comput. Model. Dyn. Syst.*, 19(6):593–615, 2013.
- [9] P. Benner, P. Kürschner, and J. Saak. Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations. *Electron. Trans. Numer. Anal.*, 43:142–162, 2014.
- [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 Mitteilungen*, 36(1):32–52, August 2013.
- [11] A. Castagnotto, H. K. F. Panzer, and L. B. Fast \mathcal{H}_2 -optimal model order reduction exploiting the local nature of krylov-subspace methods. In *European Control Conference 2016*, pages 1958–1963, Aalborg, Denmark, 2016.
- [12] T. Coleman and Y. Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM J. Optim.*, 6(2):418–445, 1996.
- [13] 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(1):148–181, 2017.
- [14] T. A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, Dec. 2011.
- [15] 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(5):1875–1898, 2011.
- [16] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Cont. Lett.*, 60(8):546–560, 2011.
- [17] 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(0):50–58, 2013.
- [18] A. Frommer, K. Lund, and D. B. Szyld. Block Krylov subspace methods for computing functions of matrices applied to multiple vectors. *Electr. Trans. Num. Anal.*, 47:100–126, 2017.
- [19] A. Frommer and V. Simoncini. Matrix Functions. In W. Schilders, H. A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 275–303. Springer Berlin Heidelberg, 2008.
- [20] L. Grasedyck. Existence of a low rank or H -matrix approximant to the solution of a Sylvester equation. *Numer. Lin. Alg. Appl.*, 11(4):371–389, 2004.
- [21] S. Gugercin, D. C. Sorensen, and A. C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms*, 32(1):27–55, 2003.
- [22] S. Güttel. Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen*, 36(1):8–31, 2013.
- [23] L. A. Knizhnerman. Calculation of functions of unsymmetric matrices using Arnoldi’s method. *Computational Mathematics and Mathematical Physics*, 31(1):1–9, 1992.
- [24] P. Kürschner. *Efficient Low-Rank Solution of Large-Scale Matrix Equations*. Dissertation, Otto-von-Guericke-Universität, Magdeburg, Germany, Apr. 2016. Shaker Verlag, ISBN 978-3-8440-4385-3.
- [25] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numer. Math.*, 6:377–387, 1964.

- [26] A. Lewis. The mathematics of eigenvalue optimization. *Mathem. Prog.*, 97(1-2):155–176, 2003.
- [27] A. S. Lewis and M. L. Overton. Nonsmooth optimization via quasi-Newton methods. *Math. Program.*, 141(1–2, Ser. A):135–163, 2013.
- [28] J.-R. Li. *Model Reduction of Large Linear Systems via Low Rank System Gramians*. Ph.D. Thesis, Massachusetts Institute of Technology, Sept. 2000.
- [29] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.
- [30] E. Mengi. A Support Function Based Algorithm for Optimization with Eigenvalue Constraints. *SIAM J. Optim.*, 27(1):246–268, 2017.
- [31] E. Mengi, E. A. Yildirim, and M. Kiliç. Numerical Optimization of Eigenvalues of Hermitian Matrix Functions. *SIAM J. Matrix Anal. Appl.*, 35(2):699–724, 2014.
- [32] C. Moosmann, E. B. Rudnyi, A. Greiner, and J. G. Korvink. Model order reduction for linear convective thermal flow. In *THERMINIC 2004*, pages 317–321, Sophia Antipolis, France, 2004.
- [33] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.
- [34] M. L. Overton. Large-Scale Optimization of Eigenvalues. *SIAM J. Optimiz.*, 2(1):88–120, 1992.
- [35] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.
- [36] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Syst. Cont. Lett.*, 40:139–144, 2000.
- [37] R. Remmert. *Theory of Complex Functions*. Springer-Verlag New York, 1991.
- [38] A. Ruhe. The Rational Krylov algorithm for nonsymmetric Eigenvalue problems. III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.
- [39] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, Manchester, UK, 1992.
- [40] J. Saak. *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*. Dissertation, Technische Universität Chemnitz, Chemnitz, Germany, July 2009.
- [41] J. Saak, M. Köhler, and P. Benner. M-M.E.S.S.-1.0.1 – the matrix equations sparse solvers library, Apr. 2016. see also: <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, Texas, June 2007.
- [43] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29(3):1268–1288, 2007.
- [44] V. Simoncini. Computational methods for linear matrix equations. *SIAM Rev.*, 38(3):377–441, 2016.
- [45] L. Sorber, M. Van Barel, and L. De Lathauwer. Unconstrained Optimization of Real Functions in Complex Variables. *SIAM J. Optim.*, 22(3):879–898, 2012.
- [46] K. Sun. *Model order reduction and domain decomposition for large-scale dynamical systems*. ProQuest LLC, Ann Arbor, MI, 2008. Thesis (Ph.D.)–Rice University.
- [47] N. Truhar and K. Veselić. Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix. *Syst. Cont. Lett.*, 56(7–8):493–503, 2007.
- [48] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. Tensorlab 3.0, 2016. Available online at <http://www.tensorlab.net/>.
- [49] E. L. Wachspress. *The ADI Model Problem*. Springer New York, 2013.

- [50] R. Waltz, J. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, Jul 2006.
- [51] T. Wolf. *H₂ Pseudo-Optimal Model Order Reduction*. Dissertation, Technische Universität München, Munich, Germany, 2015.
- [52] T. Wolf and H. K.-F. Panzer. The ADI iteration for Lyapunov equations implicitly performs H₂ pseudo-optimal model order reduction. *Internat. J. Control*, 89(3):481–493, 2016.