BIT Numerical Mathematics (2020) 60:1221–1259
https://doi.org/10.1007/s10543-020-00813-4

BIT

Inexact methods for the low rank solution to large scale Lyapunov equations

Patrick Kürschner[1,2] · Melina A. Freitag[3]

Received: 30 September 2019 / Accepted: 1 May 2020 / Published online: 18 June 2020

Abstract

The rational Krylov subspace method (RKSM) and the low-rank alternating directions implicit (LR-ADI) iteration are established numerical tools for computing low-rank solution factors of large-scale Lyapunov equations. In order to generate the basis vectors for the RKSM, or extend the low-rank factors within the LR-ADI method, the repeated solution to a shifted linear system of equations is necessary. For very large systems this solve is usually implemented using iterative methods, leading to inexact solves within this inner iteration (and therefore to "inexact methods"). We will show that one can terminate this inner iteration before full precision has been reached and still obtain very good accuracy in the final solution to the Lyapunov equation. In particular, for both the RKSM and the LR-ADI method we derive theory for a relaxation strategy (e.g. increasing the solve tolerance of the inner iteration, as the outer iteration proceeds) within the iterative methods for solving the large linear systems. These theoretical choices involve unknown quantities, therefore practical criteria for relaxing the solution tolerance within the inner linear system are then provided. The theory is supported by several numerical examples, which show that the total amount of work for solving Lyapunov equations can be reduced significantly.

Keywords Lyapunov equation · Alternating direction implicit · Rational Krylov subspaces · Low-rank approximations

Communicated by Daniel Kressner.

✉ Patrick Kürschner
patrick.kurschner@kuleuven.be

Melina A. Freitag
melina.freitag@uni-potsdam.de

[1] Group Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium

[2] Department of Electrical Engineering ESAT/STADIUS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium

[3] Institut für Mathematik, Universität Potsdam, Karl-Liebknecht-Str. 24-25, 14476 Potsdam, Germany

Mathematics Subject Classification 15A24 · 65F10 · 65F30

1 Introduction

We consider the numerical solution of large scale Lyapunov equations of the form

$$AX + XA^T = -BB^T, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times r}$. Lyapunov equations play a fundamental role in many areas of applications, such as control theory, model reduction and signal processing, see, e.g. [2,14]. Here we assume that the spectrum of A, $\Lambda(A)$, lies in the open left half plane, e.g. $\Lambda(A) \subset \mathbb{C}_-$ and that the right hand side is of low rank, e.g. $r \ll n$, which is often the case in practice. A large and growing amount of literature considers the solution to (1.1), see [45] and references therein for an overview of developments and methods.

The solution matrix X of (1.1) is, in general, dense, making it virtually impossible to store it for large dimensions n. For a low-rank right hand side, however, the solution X often has a very small numerical rank [3,5,25,36,40,51] and many algorithms have been developed that approximate X by a low-rank matrix $X \approx ZZ^T$, where $Z \in \mathbb{R}^{n\times s}$, $s \ll n$. Important low-rank algorithms are, for instance, projection type methods based on Krylov subspaces [20,21,28,43,44] and low-rank alternating directions implicit (ADI) methods [12–14,30,33,36].

This paper considers both the rational Krylov subspace method (from the family of projection type methods) and the low-rank ADI method for the low rank solution to (1.1). One of the computationally most expensive parts in both methods is that, in each iteration step, shifted linear systems of the form

$$(A - \sigma I)y = z, \quad z \in \mathbb{R}^{n\times r},$$

need to be solved, where the shift σ is usually variable and both the shift σ and the right hand side z depend on the particular algorithm used. Normally these linear systems are solved by sparse-direct or iterative methods. When iterative methods, such as preconditioned Krylov subspace methods, are used to solve the linear systems, then these solves are implemented inexactly and we obtain a so-called inner-outer iterative method (sometimes the term "inexact method" is used). The outer method is (in our case) a rational Krylov subspace method or a low-rank ADI iteration. The inner problem is the iterative solution to the linear systems. The inner solves are often carried out at least as accurately as the required solution accuracy for the Lyapunov equation (cf., e.g., the numerical experiments in [20]), usually in terms of the associated Lyapunov residual norms. It turns out that this is not necessary, and the underlying theory is the main contribution of this paper.

Inexact methods have been considered for the solution to linear systems and eigenvalue problems (see [17,23,42,47,53] and references therein). One focus has been on inexact inverse iteration and similar methods, where, in general, the accuracy of the inexact solve has to be increased to obtain convergence [22]. For subspace expansion methods, such as the Krylov methods considered in [17,24,42,47,53], it has been observed numerically and shown theoretically that it is possible to relax the solve tolerance as the outer iteration proceeds.

In this paper we show that for both the rational Krylov subspace method and the low-rank ADI algorithm we can terminate the inner iteration before full precision has been reached (in a way that depends on the outer iteration accuracy) and still obtain very good accuracy in the final solution to the Lyapunov equation. We develop a strategy which shows that we can relax (e.g. increase) the stopping tolerance within the inner solve as the outer iteration proceeds. The key approach is to consider the so-called "residual gap" between the exact and inexact methods for solving Lyapunov equations and show that it depends on the linear system accuracy. Using this connection between inner and outer solve accuracy we can derive relaxation criteria for inexact solves to Lyapunov equations, which is the main contribution of this article.

This is a similar behavior as observed for Krylov methods with inexact matrix-vector products applied to eigenvalue methods [23,42], linear systems [17,47,53], and matrix functions [24]. We provide practical relaxation strategies for both methods and give numerical examples which support the theory, and show that the amount of computation time can be significantly reduced by using these strategies.

The paper is organised as follows. In Sect. 2 we review results about rational Krylov subspace methods. Those are used to show important properties about Galerkin projections. A new inexact rational Arnoldi decomposition is derived, extending the theory of [37]. We show in Theorem 2.1, Corollaries 2.1 and 2.2 that the entries of the solution to the projected Lyapunov equation have a decreasing pattern, as seen in Fig. 1. This crucial novel result is then used to demonstrate that, in the inexact rational Krylov subspace method, the linear system solve can be relaxed, in a way inversely proportional to this pattern. Section 3 is devoted to low-rank ADI methods. We show that the low-rank factors arising within the inexact ADI iteration satisfy an Arnoldi like decomposition in Theorem 3.2. This theory is again new and significant for deriving a specially tailored relaxation strategy for inexact low-rank ADI methods in Theorem 3.3. Finally, in Sect. 4 we test several practical relaxation strategies and provide numerical evidence for our findings. Our examples show that, in particular for very large problems, we can save up to half the number of inner iterations within both methods for solving Lyapunov equations.

Notation Throughout the paper, if not stated otherwise, we use $\|\cdot\|$ to denote the Euclidean vector and associated induced matrix norm. $(\cdot)^*$ stands for the transpose and complex conjugate transpose for real and, respectively, complex matrices and vectors. The identity matrix of dimension k is denoted by I_k with the subscript omitted if the dimension is clear from the context. The jth column of the identity matrix is e_j. The spectrum of a matrix A and a matrix pair (A, M) is denoted by $\Lambda(A)$ and $\Lambda(A, M)$, respectively. The smallest (largest) eigenvalues and singular values of a matrix X are denoted by $\lambda_{\min}(X)$ $(\lambda_{\max}(X))$ and $\sigma_{\min}(X)$ $(\sigma_{\max}(X))$.

2 Rational Krylov subspace methods and inexact solves

2.1 Introduction and rational Arnoldi decompositions

Projection methods for (1.1) follow a Galerkin principle similar to, e.g., the Arnoldi method for eigenvalue computations or linear systems (in this case called full orthogonalization method). Let $\mathscr{Q} = \operatorname{range}(Q) \subset \mathbb{C}^n$ with a matrix $Q \in \mathbb{C}^{n\times k}$, $k \ll n$ whose columns form an orthonormal basis of $\mathscr{Q}$: $Q^*Q = I_k$. We look for low-rank approximate solutions to (1.1) in the form $Q\tilde{X}Q^*$ with $\tilde{X} = \tilde{X}^* \in \mathbb{C}^{k\times k}$, i.e.

$$X \in \mathscr{X}(\mathscr{Q}) := \{Q\tilde{X}Q^* \in \mathbb{C}^{n\times n},\ \tilde{X} = \tilde{X}^*,\ \operatorname{range}(Q) = \mathscr{Q}\}.$$

The Lyapunov residual matrix for approximate solutions $X \in \mathscr{X}(\mathscr{Q})$ is

$$R := R(Q\tilde{X}Q^*) = A(Q\tilde{X}Q^*) + (Q\tilde{X}Q^*)A^* + BB^*. \tag{2.1}$$

Imposing a Galerkin condition $R \perp \mathscr{X}$ [28,38] leads to the projected Lyapunov equation

$$T\tilde{X} + \tilde{X}T^* + Q^*BB^*Q = 0, \quad T := Q^*AQ, \tag{2.2}$$

i.e., $\tilde{X}$ is the solution of a small, k-dimensional Lyapunov equation which can be solved by algorithms employing dense numerical linear algebra, e.g., the Bartels–Stewart method [4]. Throughout this article we assume that the compressed problem (2.2) admits a unique solution which is ensured by $\Lambda(T) \cap -\Lambda(T) = \emptyset$, which holds, for example, when $\Lambda(T) \subset \mathbb{C}_-$. A sufficient condition for ensuring $\Lambda(T) \subset \mathbb{C}_-$ is $A + A^* < 0$. In practice, however, this condition is rather restrictive because, on the one hand, often $\Lambda(T) \subset \mathbb{C}_-$ holds even for indefinite $A + A^*$ and, one the other hand, the discussed projection methods will still work if the restriction T has eigenvalues in $\mathbb{C}_+$ provided that $\Lambda(T) \cap -\Lambda(T) = \emptyset$ holds.

Usually, one produces sequences of subspaces of increasing dimensions in an iterative manner, e.g. $\mathscr{Q}_1 \subseteq \mathscr{Q}_2 \subseteq \ldots \subseteq \mathscr{Q}_j$. For practical problems, using the standard (block) Krylov subspace

$$\mathscr{Q}_j = \mathscr{K}_j(A, B) = \operatorname{range}\left(\left[B, AB, \ldots, A^{j-1}B\right]\right) \tag{2.3}$$

is not sufficient and leads to a slowly converging process and, hence, large low-rank solution factors. A better performance can in most cases be achieved by using rational Krylov subspaces which we use here in the form

$$\mathscr{Q}_j = \mathscr{K}_j^{\mathrm{rat}}(A, B, \boldsymbol{\xi}) := \operatorname{range}\left(\left[B, (A - \xi_2 I)^{-1}B, \ldots, \prod_{i=2}^{j}(A - \xi_i I)^{-1}B\right]\right), \tag{2.4}$$

with shifts $\xi_i \in \mathbb{C}_+$ (hence $\xi_i \notin \Lambda(A)$), $i = 2, \ldots, j$. An orthonormal basis for $\mathscr{K}_j^{\text{rat}}$ can be computed by the (block) rational Arnoldi process [37] leading to $Q_j \in \mathbb{C}^{n\times jr}$ of full column rank. The resulting rational Krylov subspace method (RKSM) for computing approximate solutions to (1.1) is given in Algorithm 1. The shift parameters are crucial for a rapid reduction of the Lyapunov residual norm $\|R_j\| := \|R(Q_j Y_j Q_j^*)\|$ (cf. (2.1)) and can be generated a-priori or adaptively in the course of the iteration [21]. The extension of the orthonormal basis range (Q_j) by w in Line 6 of Algorithm 1 should be done by a robust orthogonalization process, e.g., using a repeated (block) Gram–Schmidt process. The orthogonalization coefficients are accumulated in the block Hessenberg matrix $H_j = [h_{i,k}] \in \mathbb{C}^{jr\times jr}$, $h_{i,k} \in \mathbb{C}^{r\times r}$, $i, k = 1, \ldots, j$, $i < k + 2$. If the new basis blocks are normalized using a thin QR factorization, then the $h_{i+1,i}$ matrices are upper triangular. In the following we summarize properties of the RKSM method, in particular known results about the rational Arnoldi decomposition and the Lyapunov residual, which will be crucial later on. We simplify this discussion of the rational Arnoldi process and Algorithm 1 to the case $r = 1$. From there, one can generalize to the situation $r > 1$.

Algorithm 1: Rational Krylov subspace method for solving (1.1)

```
   Input : $A$, $B$ as in (1.1), shifts $\{\xi_2, \ldots, \xi_{j_{\max}}\} \subset \mathbb{C}_+$, tolerance $0 < \tau \ll 1$.
   Output: $Q \in \mathbb{C}^{n\times jr}$, $\tilde{X} \in \mathbb{C}^{jr\times jr}$ such that $Q\tilde{X}Q^* \approx X$ with $jr \ll n$.
1  Compute thin QR decomposition of $B$: $Q_1\beta = B$, $j = 1$.
2  while $\|R(Q_j Y_j Q_j^*)\| > \tau\|BB^*\|$ do
3      Solve $(A - \xi_{j+1}I)w_{j+1} = q_j$ for $w_{j+1}$.
4      Orthogonally extend basis matrix $Q_j$:
5      $h_{1:j,j} = Q_j^* w_{j+1}$, $\hat{w}_{j+1} = w_{j+1} - Q_j h_{1:j,j}$,
6      $q_{j+1}h_{j+1,j} = \hat{w}_{j+1}$, $Q_{j+1} = [Q_j,\ q_{j+1}]$.
7      Projected matrices: $T_j = Q_j^*(AQ_j)$, $\tilde{B}_j = Q_j^* B = [\beta^*, 0]^*$.
8      Solve $T_j Y_j + Y_j T_j^* + \tilde{B}_j \tilde{B}_j^* = 0$ for $Y_j$.
9      Compute residual norm $\|R(Q_j Y_j Q_j^*)\|$.
10     $j = j + 1$.
11 $\tilde{X} = Y_j$, $Q = Q_j$
```

Assuming at first that the linear systems in Line 3 of Algorithm 1 are solved *exactly*, then, after j steps of Algorithm 1, the generated quantities satisfy a rational Arnoldi decomposition [21,27,37]

$$AQ_j = Q_j T_j + g_j h_{j+1,j} e_j^* H_j^{-1}, \quad g_j := q_{j+1}\xi_{j+1} - (I - Q_j Q_j^*)Aq_{j+1}, \tag{2.5}$$

which can also be expressed as

$$AQ_{j+1}H_{j+1,j} = Q_{j+1}M_{j+1,j}, \quad M_{j+1,j} := \begin{bmatrix} I + H_j D_j \\ \xi_{j+1}h_{j+1,j}e_j^* \end{bmatrix}, \tag{2.6}$$

where $H_{j+1,j}$ and $M_{j+1,j}$ are $(j+1)\times j$ matrices, $D_j = \operatorname{diag}\big(\xi_2,\ldots,\xi_{j+1}\big)$, H_j and D_j are square matrices of size j, see, e.g., [16,37]. Here we generally use $H_j = H_{j,j}$ for simplicity. From the relations (2.5) and (2.6) it follows that the restriction of A onto $\mathscr{K}_j^{\mathrm{rat}}$ can be written as

$$T_j = Q_j^*(AQ_j) = (I + H_jD_j)H_j^{-1} - Q_j^*Aq_{j+1}h_{j+1,j}e_j^*H_j^{-1}, \tag{2.7}$$

which enables to compute the restriction T_j efficiently in terms of memory usage. The restriction T_j can also be associated with the Hessenberg–Hessenberg pair $(M_{j,j}, H_{j,j})$ [16,37].

The final part of this section discusses how the Lyapunov residual in Line 9 of Algorithm 1 can be computed efficiently. If the projected Lyapunov equation in Line 8

$$T_jY_j + Y_jT_j^* + \tilde{B}_j\tilde{B}_j^* = 0 \tag{2.8}$$

is solved for Y_j, then for the Lyapunov residual (2.1) after j steps of the RKSM, we have

$$R_j = R(Q_jY_jQ_j^*) = F_j + F_j^*, \tag{2.9a}$$

with

$$F_j := L_jQ_j^* \in \mathbb{C}^{n\times n}, \quad L_j := g_jh_{j+1,j}e_j^*H_j^{-1}Y_j \in \mathbb{C}^{n\times j}, \tag{2.9b}$$

as was shown in [21,44]. The term L_j is sometimes referred to as "semi-residual". Since $g_j \in \mathbb{C}^n$, $h_{j+1,j}e_j^*H_j^{-1}Y_j \in \mathbb{C}^{1\times j}$, the matrix F_j is of rank one. In general for $B \in \mathbb{R}^{n\times r}$, $r > 1$ and a block-RKSM, F_j is of rank r.

Note that relation (2.9a) is common for projection methods for (1.1), but the special structure of L_j in (2.9b) arises from the rational Arnoldi process. Because $g_j^*Q_j = 0$ we have that $F_j^2 = 0$ and $\|R_j\| = \|F_j\| = \|L_j\|$ (using the relationship between the spectral norm and the largest eigenvalues/singular values). This enables an efficient way for computing the norm of the Lyapunov residual R_j via

$$\|R_j\| = \|\|g_j\|h_{j+1,j}e_j^*H_j^{-1}Y_j\|, \tag{2.10}$$

for $g_j \in \mathbb{C}^n$.

2.2 The inexact rational Arnoldi method

The solution of the linear systems for w_{j+1} in each step of RKSM (line 3 in Algorithm 1) is one of the computationally most expensive stages of the rational Arnoldi method. In this section we investigate inexact solves of this linear systems, e.g., by iterative Krylov subspace methods, but we assume that this is the only source of inaccuracy in the algorithm. Clearly, some of the above properties do not hold anymore if the linear systems are solved inaccurately. Let

$$s_j := q_j - (A - \xi_{j+1} I)\tilde{w}_{j+1}, \quad \text{with} \quad \|s_j\| \le \tau_j^{\mathrm{LS}}$$

be the residual vectors with respect to the linear systems and inexact solution vectors $\tilde{w}_{j+1}$ with $\tau_j^{\mathrm{LS}} < 1$ being the solve tolerance of the linear system at step j of the rational Arnoldi method.

The derivations in [37], [21, Proof of Prop. 4.2.] can be modified with

$$q_j = (A - \xi_{j+1} I)Q_{j+1} h_{1:j+1,j} + s_j,$$

in order to obtain

$$AQ_{j+1}H_{j+1,j} = Q_{j+1}M_{j+1,j} - S_j,\; S_j := [s_1, \ldots, s_j],$$

where $M_{j+1,j}$ is as in (2.6). Note that for $S_j = 0$ we recover (2.6), as expected. Hence $AQ_jH_j = [Q_j(I + H_jD_j) + (\xi_{j+1}q_{j+1} - Aq_{j+1})e_j^* h_{j+1,j}] - S_j$ which leads to the perturbed rational Arnoldi relation

$$AQ_j = Q_j T_j^{\text{impl.}} + g_j e_j^* h_{j+1,j} H_j^{-1} - S_j H_j^{-1}, \tag{2.11}$$

where $T_j^{\text{impl.}} := [(I + H_jD_j) - Q_j^* A q_{j+1})e_j^* h_{j+1,j}]H_j^{-1}$ marks the restriction of A by the implicit formula (2.7). However, the explicit restriction of A can be written as

$$\begin{aligned} T_j^{\text{expl.}} &:= Q_j^*(AQ_j) = T_j^{\text{impl.}} - Q_j^* S_j H_j^{-1}, \\ &= ((I + H_jD_j) - Q_j^* A q_{j+1})e_j^* h_{j+1,j} - Q_j^* S_j)H_j^{-1} \end{aligned} \tag{2.12}$$

which highlights the problem that the implicit (computed) restriction $T_j^{\text{impl.}}$ from (2.7) is not the true restriction of A onto range $\big(Q_j\big)$. In fact, the above derivations reveal that

$$T_j^{\text{impl.}} = Q_j^*(A + E_j)Q_j, \quad E_j := S_j H_j^{-1} Q_j^*,$$

i.e., the implicit restriction (2.7) is the exact restriction of a perturbation of A [26,31].

Similar to [26] we prefer to use $T_j^{\text{expl.}}$ for defining the projected problem as this keeps the whole process slightly closer to the original matrix A. Moreover, (2.12) reveals that unlike $T^{\text{impl.}}$, the explicit restriction $T_j^{\text{expl.}}$ is not connected to a Hessenberg–Hessenberg pair because the term $Q_j^* S_j$ has no Hessenberg structure. Computing $T_j^{\text{expl.}}$ by either (2.12) or explicitly generating $T_j^{\text{expl.}} = Q_j^*(AQ_j)$ by adding new columns and rows to $T_{j-1}^{\text{expl.}}$ will double the memory requirements because an additional $n \times j$ array, $S_j \in \mathbb{C}^{n \times j}$ or $W_j := AQ_j$, has to be stored. Hence, the per-step storage requirements are similar to the extended Krylov subspace method (see, e.g., [43] and the corresponding paragraph in Sect. 2.5). Alternatively, if matrix vector products with

A^* are available, we can also update the restriction

$$T_j^{\text{expl.}} = Q_j^*(AQ_j) = \begin{bmatrix} T_{j-1}^{\text{expl.}} & Q_{j-1}^*(Aq_j) \\ (z_j)^* Q_{j-1} & q_j^*(Aq_j) \end{bmatrix}, \quad z_j := A^* q_j,$$

thereby bypassing the increased storage requirements. We will use this approach in our numerical experiments. Trivially, we may also compute $T_j^{\text{expl.}}$ from scratch row/columnwise, requiring j matrix vector products with A but circumventing the storage increase. Using $T_j^{\text{expl.}}$ leads to the *inexact rational Arnoldi relation*

$$AQ_j = Q_j T_j^{\text{expl.}} + \tilde{g}_j H_j^{-1}, \quad \tilde{g}_j := g_j h_{j+1,j} e_j^* - (I - Q_j Q_j^*) S_j, \tag{2.13}$$

which we employ in the subsequent investigations.

Lemma 2.1 *Consider the approximate solution $Q_j Y_j Q_j^*$ of (1.1) after j iterations of inexact RKSM, where Y_j solves the projected Lyapunov equation (2.8) defined by either $T_j^{expl.}$ or $T_j^{impl.}$. Then the true Lyapunov residual matrix $R_j^{true} = R(Q_j Y_j Q_j^*)$ can be written in the following forms.*

(a) If $T_j^{expl.}$ is used it holds

$$\begin{aligned} R_j^{true} &= F_j^{expl.} + (F_j^{expl.})^*, \quad F_j^{expl.} := \tilde{g}_j H_j^{-1} Y_j Q_j^* \\ &= F_j - (I - Q_j Q_j^*) S_j H_j^{-1} Y_j Q_j^*, \end{aligned} \tag{2.14a}$$

(b) and, if otherwise $T_j^{impl.}$ is used, it holds

$$R_j^{true} = F_j^{impl.} + (F_j^{impl.})^*, \quad F_j^{impl.} := F_j - S_j H_j^{-1} Y_j Q_j^*. \tag{2.14b}$$

Proof For case (a), using (2.13) immediately yields

$$\begin{aligned} R_j^{\text{true}} =& R(Q_j Y_j Q_j^*) = AQ_j Y_j Q_j^* + Q_j Y_j Q_j^* A^* + BB^* \\ =& [Q_j T_j^{\text{expl.}} + \tilde{g}_j H_j^{-1}] Y_j Q_j^* + Q_j Y_j [Q_j T_j^{\text{expl.}} \\ & + \tilde{g}_j H_j^{-1}]^* + Q_j Q_j^* BB^* Q_j Q_j^* \\ =& Q_j \left[T_j^{\text{expl.}} Y_j + Y_j (T_j^{\text{expl.}})^* + Q_j^* BB^* Q_j \right] Q_j^* \\ & + \tilde{g}_j H_j^{-1} Y_j Q_j^* + Q_j Y_j H_j^{-*} \tilde{g}_j^* = F_j^{\text{expl.}} + (F_j^{\text{expl.}})^*. \end{aligned}$$

Case (b) follows similarly using (2.11). □

Hence in both cases the true residual R_j^{true} is a perturbation of the computed residual given by (2.9a)–(2.9b) which we denote in the remainder by $R_j^{\text{comp.}}$. In case $T_j^{\text{expl.}}$ is used, since $Q_j \perp \tilde{g}_j$ one can easily see that $\|R_j^{\text{true}}\| = \|F_j^{\text{expl.}}\| = \|\tilde{g}_j H_j^{-1} Y_j\|$, a property not shared when using $T_j^{\text{impl.}}$.

Our next step is to analyze the difference between the true and computed residual.

Definition 1 The *residual gap* after j steps of inexact RKSM for Lyapunov equations is given by

$$\begin{aligned} \Delta R_j &:= R_j^{\text{true}} - R_j^{\text{comp.}} = \eta_j^{\text{expl./impl.}} + (\eta_j^{\text{expl./impl.}})^*, \\ \text{where} \quad \eta_j^{\text{expl.}} &:= F_j - F_j^{\text{expl.}} = (I - Q_j Q_j^*) S_j H_j^{-1} Y_j Q_j^* \quad (\text{if } T_j^{\text{expl.}} \text{ is used}), \\ \eta_j^{\text{impl.}} &:= F_j - F_j^{\text{impl.}} = S_j H_j^{-1} Y_j Q_j^* \quad (\text{if } T^{\text{impl.}} \text{ is used}). \end{aligned}$$

We have

$$\|\Delta R_j\| = \begin{cases} \|\eta_j^{\text{expl.}} + (\eta_j^{\text{expl.}})^*\| & = \|\eta_j^{\text{expl.}}\|, \\ \|\eta_j^{\text{impl.}} + (\eta_j^{\text{impl.}})^*\| & \leq 2\|\eta_j^{\text{impl.}}\|. \end{cases}$$

The result for $\|\eta_j^{\text{expl.}}\|$ follows from the orthogonality of left and rightmost factors of $\eta_j^{\text{expl.}}$: with $\hat{Q} := [Q_j, Q_j^\perp] \in \mathbb{C}^{n \times n}$ unitary we have

$$\begin{aligned} \|\Delta R_j\|_2 &= \left|\lambda_{\max}(\eta_j^{\text{expl.}} + (\eta_j^{\text{expl.}})^*)\right| = \left|\lambda_{\max}(\hat{Q}^*(\eta_j^{\text{expl.}} + (\eta_j^{\text{expl.}})^*)\hat{Q})\right| \\ &= \left|\lambda_{\max}\left(\begin{bmatrix} 0 & Q_j^{\perp *} \eta_j^{\text{expl.}} Q_j \\ (Q_j^{\perp *} \eta_j^{\text{expl.}} Q_j)^* & 0 \end{bmatrix}\right)\right| = \sigma_{\max}(\eta_j^{\text{expl.}}) = \|\eta_j^{\text{expl.}}\|_2. \end{aligned}$$

In the following we use $T_j^{\text{expl.}}$ to define the projected problem and omit the superscripts $^{\text{expl.}}$, $^{\text{impl.}}$. Suppose the desired accuracy is so that $\|R_j^{\text{true}}\| \leq \varepsilon$, where $\varepsilon > 0$ is a given threshold. In practice the computed residual norms often show a decreasing behavior very similar to the exact method. However, the norm of the residual gap $\|\eta_j\|$ indicates the attainable accuracy of the inexact rational Arnoldi method because $\|R_j^{\text{true}}\| \leq \|R_j^{\text{comp.}}\| + \|\eta_j\|$ and the true residual norm is bounded by $\|\eta_j\|$ even if $\|R_j^{\text{comp.}}\| \leq \varepsilon$, which would indicate convergence of the computed residuals. This motivates to enforce $\|\eta_j\| < \varepsilon$, such that small true residual norms $\|R_j^{\text{true}}\| \leq 2\varepsilon$ are obtained overall. Since, at step j,

$$\|\eta_j\| \leq \|S_j H_j^{-1} Y_j\| = \left\|\sum_{k=1}^{j} s_k e_k^* H_j^{-1} Y_j\right\| \leq \sum_{k=1}^{j} \|s_k\| \|e_k^* H_j^{-1} Y_j\|, \tag{2.15}$$

it is sufficient that only one of the factors in each addend in the sum is small and the other one is bounded by, say, $\mathcal{O}(1)$ in order to achieve $\|\eta_j\| \leq \varepsilon$. In particular, if the $\|e_k^* H_j^{-1} Y_j\|$ terms decrease with k, the linear residual norms $\|s_k\|$ are allowed to increase to some extent, and still achieve a small residual gap $\|\eta_j\|$. Hence, the solve tolerance τ_k^{LS} of the linear solves can be relaxed in the course of the outer iteration which has coined the term *relaxation*. For this to happen, however, we first need to investigate if there is a decay of $\|e_k^* H_j^{-1} Y_j\|$ as k increases, which we will do next.

2.3 Properties of the solution of the Galerkin system

By (2.15), the norm of the residual gap η_j strongly depends on the solution Y_j of the projected Lyapunov equation (2.8). We will see in Theorem 2.1 and Corollary 2.1 that the entries of Y_j decrease away from the diagonal in a manner proportional to the Lyapunov residual norm.

In the second part of this section we consider the rows of $H_j^{-1} Y_j$, since the residual formula (2.9a–2.9b) and the residual gap (2.15) depend on this quantity. It turns out that the norm of those rows also decay with the Lyapunov residual norms. Both decay bounds will later be used to develop practical relaxation criteria for achieving $\|\eta_j\| \leq \varepsilon$.

Consider the solution to the projected Lyapunov equation (2.8). We are interested in the transition from step k to j, where $k < j$. At first, we investigate this transition for a general Galerkin method including RKSM as a special case.

Theorem 2.1 *Assume $A + A^* < 0$ and consider a Galerkin projection method for (1.1) with $T_j = Q_j^*(AQ_j)$, $Q_j^* Q_j = I_j$, and the first basis vector given by $B = q_1\beta$. Let the $k \times k$ matrix Y_k and the $j \times j$ matrix Y_j be the solution to the projected Lyapunov equation (2.8) after k and j steps of this method (e.g., Algorithm 1 with $r = 1$), respectively, where $k < j$. Consider the $j \times j$ difference matrix $\Delta Y_{j,k} := Y_j - \begin{bmatrix} Y_k & 0 \\ 0 & 0 \end{bmatrix}$, where the zero blocks are of appropriate size. Then*

$$\|\Delta Y_{j,k}\| \leq c_A \|R_k^{true}\|, \quad c_A := \frac{(1+\sqrt{2})^2}{2\alpha_A}, \tag{2.16}$$

where $\alpha_A := \frac{1}{2}|\lambda_{\max}(A + A^*)|$, and R_k^{true} is the Lyapunov residual matrix after k steps of Algorithm 1.

Proof The residual matrix $N_{j,k}$ of (2.8) w.r.t. T_j and $\begin{bmatrix} Y_k & \\ & 0 \end{bmatrix}$ is given by

$$N_{j,k} := T_j \begin{bmatrix} Y_k & \\ & 0 \end{bmatrix} + \begin{bmatrix} Y_k & \\ & 0 \end{bmatrix} T_j^* + \begin{bmatrix} \begin{bmatrix} \beta\beta^* & 0 \\ 0 & 0 \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}$$

as T_j is built cumulatively. Since $Q_j [\begin{smallmatrix} Y_k & \\ & 0 \end{smallmatrix}] Q_j^* = Q_k Y_k Q_k^*$, it follows that $\|N_{j,k}\| \leq \|R_k^{\mathrm{true}}\|$. The difference matrix $\Delta Y_{j,k}$ satisfies the Lyapunov equation $T_j \Delta Y_{j,k} + \Delta Y_{j,k} T_j^* = -N_{j,k}$, and since $\Lambda(T_j) \subset \mathbb{C}_-$ (using $A + A^* < 0$) it can be expressed via the integral

$$\Delta Y_{j,k} = \int\limits_0^\infty \mathrm{e}^{T_j t}\, N_{j,k}\, \mathrm{e}^{T_j^* t}\, dt. \tag{2.17}$$

Moreover, using results from [19], we have

$$\|\, \mathrm{e}^{T_j t}\, \| \leq (1+\sqrt{2}) \max_{z \in \mathscr{W}(T_j)} |\, \mathrm{e}^{zt}\, | = (1+\sqrt{2})\, \mathrm{e}^{\max\limits_{z \in \mathscr{W}(T_j)} \mathrm{Re}\,(z) t},$$

where $\mathscr{W}(\cdot)$ denotes the field of values. Using the assumption $A + A^* < 0$ it holds that $\mathscr{W}(A) \subset \mathbb{C}_-$ and, consequently, $\mathscr{W}(T_j) \subseteq \mathscr{W}(A),\ \forall j > 0$. Hence,

$$\mathrm{e}^{\max\limits_{z \in \mathscr{W}(T_j)} \mathrm{Re}\,(z) t} \leq \mathrm{e}^{-\alpha_A t} \quad \text{with} \quad \alpha_A := \frac{1}{2} |\lambda_{\min}(A + A^*)|.$$

Finally, from (2.17) we obtain $\|\Delta Y_{j,k}\| \leq \int\limits_0^\infty \|\, \mathrm{e}^{T_j t}\, \|^2 dt \|N_{j,k}\| \leq \dfrac{(1+\sqrt{2})^2 \|R_k^{\mathrm{true}}\|}{2\alpha_A}$.

□

Theorem 2.1 shows that the difference matrix $\Delta Y_{j,k}$ decays at a similar rate as the Lyapunov residual norms.

Remark 2.1 Note that the constant c_A in Theorem 2.1 is a bound on $\int\limits_0^\infty \|\, \mathrm{e}^{T_j t}\, \|^2 dt$ and could be much smaller than the given value. In the following considerations we assume that c_A is small or of moderate size. If c_A is large (e.g., the field of values is close to the imaginary axis) then the ability to approximate X by a low-rank factorization might be reduced, see, e.g. [3,25]. Hence, this situation could be difficult for the low-rank solvers, regardless whether exact or inexact linear solves are used, and it is therefore appropriate to assume that c_A is small (or of moderate size).

The above theorem can be used to obtain results about the entries of the solution Y_j of the projected Lyapunov equation (2.8).

Corollary 2.1 *Let the assumptions of Theorem 2.1 hold. For the (ℓ, i)th entry of Y_j we have*

$$|e_\ell^* Y_j e_i| \leq c_A \|R_k^{true}\|, \quad \ell, i = 1, \ldots, j, \tag{2.18}$$

where $k < \max(\ell, i)$ and c_A is as in Theorem 2.1.

Proof We have

$$e_\ell^* Y_j e_i = e_\ell^* \begin{bmatrix} Y_k & 0 \\ 0 & 0 \end{bmatrix} e_i + e_\ell^* \Delta Y_{j,k} e_i. \tag{2.19}$$

For $k < \ell$ the first summand vanishes and hence, using Theorem 2.1

$$|e_\ell^* Y_j e_i| = |e_\ell^* \varDelta Y_{j,k} e_i| \leq \frac{(1+\sqrt{2})^2}{2\alpha_A} \|R_k^{\text{true}}\|, \quad k = 1, \ldots, \ell - 1. \tag{2.20}$$

Since, $Y_j = Y_j^*$ the indices ℓ, i can be interchanged, s.t. $k < \max(\ell, i)$ and we end up with the final bound (2.18). □

Remark 2.2 Note that the sequence $\{\|R_k^{\text{true}}\|\}$ is not necessarily monotonically decreasing and we can further extend the bound in (2.18) and obtain

$$|e_\ell^* Y_j e_i| \leq c_A \min_{p=1,\ldots,\max(\ell,i)} \|R_{p-1}^{\text{true}}\|, \quad \ell, i = 1, \ldots, j.$$

Corollary 2.1 shows that the (ℓ, i)-th entry of Y_j can be bounded from above using the Lyapunov residual norm at step $k < \max(\ell, i)$. This indicates that the further away from the diagonal, the smaller the entries of Y_j will become in the course of the iteration, provided the true residual norms exhibit a sufficiently decreasing behavior. We can observe this characteristic in Fig. 1 in Sect. 4. The decay of Lyapunov solutions has also been investigated by different approaches. Especially when T_j is banded, which is, e.g., the case when a Lanczos process is applied to $A = A^*$, more refined decay bounds for the entries of Y_j can be established, see, e.g., [18,35].

In exact and inexact RKSM, the formula for residuals (2.9), (2.14) and residual gaps (2.15) depend not only on Y_j but rather on $H_j^{-1} Y_j$. In particular, the rows of $H_j^{-1} Y_j$ appear in (2.15) and will later be substantial for defining relaxation strategies. Hence, we shall investigate if the norms $\|e_k^* H_j^{-1} Y_j\|$ also exhibit a decay for $1 \leq k \leq j$. For the last row, i.e. $k = j$, using (2.10) readily reveals

$$\|e_j^* H_j^{-1} Y_j\| = \frac{\|R_j^{\text{comp.}}\|}{|h_{j+1,j}|\|g_j\|}, \quad \text{assuming} \quad g_j \neq 0,\ h_{j+1,j} \neq 0.$$

In the spirit of Theorem 2.1, Corollary 2.1, we would like to bound the ℓth row of $H_j^{-1} Y_j$ by the $(\ell - 1)$th computed Lyapunov residual norm $\|R_{\ell-1}^{\text{comp.}}\|$. For this we require the following lemma showing that, motivated by similar results in [53], the first k $(1 < k \leq j)$ entries of $e_k^* H_j^{-1}$ are essentially determined by the left null space of $\underline{H}_j := H_{j+1,j}$ and $e_{k-1}^* H_{k-1}^{-1}$ modulo scaling.

Lemma 2.2 *Let* $\underline{H}_j = \begin{bmatrix} H_j \\ h_{j+1,j} e_j^* \end{bmatrix} \in \mathbb{C}^{(j+1)\times j}$ *with* H_j *an unreduced upper Hessenberg matrix, with* $\text{rank}(\underline{H}_j) = j$, $H_k := H_{1:k,1:k}$ *nonsingular* $\forall 1 \leq k \leq j$, *and let* $\omega \in \mathbb{C}^{1\times(j+1)}$, $\omega \neq 0$ *satisfy* $\omega \underline{H}_j = 0$. *Define the vectors* $f_m^{(k)} := e_k^* H_m^{-1}$, $1 \leq k \leq m \leq j$. *Then, for* $1 \leq k \leq j$, *we have*

$$f_j^{(j)} = -\frac{\omega_{1:j}}{\omega_{j+1}h_{j+1,j}}, \quad f_j^{(k)} = \frac{v_j^{(k)}}{\phi_j^{(k)}}, \quad \textit{where} \tag{2.21a}$$

$$v_j^{(k)} := \omega_{1:j} + [0_{1,k}, [0_{1,j-k-1}, h_{j+1,j}\omega_{j+1}]H^{-1}_{k+1:j,k+1:j}], \quad \phi_j^{(k)} := v_j^{(k)} H_j e_k. \tag{2.21b}$$

Moreover, for $k > 1$, the first k entries of $v_j^{(k)}$ can be expressed by

$$(v_j^{(k)})_{1:k} = [-h_{k,k-1} f_{k-1}^{(k-1)}, 1]\omega_k. \tag{2.21c}$$

Proof See "Appendix A". □

Corollary 2.2 *Let the assumptions of Theorem 2.1, Corollary 2.1, and Lemma 2.2 hold and consider the RKSM as a special Galerkin projection method. Then*

$$\|e_\ell^* H_j^{-1} Y_j\| \leq \begin{cases} c_A \|e_1^* H_j^{-1}\| \|R_0^{true}\|, & \ell = 1, \\ \frac{1}{\phi_j^{(\ell)} \|g_{\ell-1}\|} \|R_{\ell-1}^{comp.}\| + c_A \|e_\ell^* H_j^{-1}\| \|R_{\ell-1}^{true}\|, & \ell = 2, \ldots, j \end{cases} \tag{2.22}$$

with g_ℓ from (2.5) and $\phi_j^{(\ell)}$ from Lemma 2.2.

Proof Using Lemma 2.2 for $f_j^{(\ell)} := e_\ell^* H_j^{-1}$, $1 < \ell \leq j$, and the structure (2.9b) of the computed Lyapunov residuals yields

$$\begin{aligned} e_\ell^* H_j^{-1} Y_j &= f_j^{(\ell)} \left(\begin{bmatrix} Y_{\ell-1} & 0 \\ 0 & 0 \end{bmatrix} + \Delta Y_{j,\ell-1} \right) \\ &= [-h_{\ell,\ell-1} f_{\ell-1}^{(\ell-1)} \omega_\ell]/\phi_j^{(\ell)} Y_{\ell-1} + f_j^{(\ell)} \Delta Y_{j,\ell-1} \\ &= -\frac{g_{\ell-1}^* L_{\ell-1}}{\|g_{\ell-1}\|^2 \phi_j^{(\ell)}} \omega_\ell + f_j^{(\ell)} \Delta Y_{j,\ell-1}. \end{aligned}$$

Taking norms, using that ω is a unit norm vector, noticing for $\ell = 1$ only the second term exists, and applying Theorem 2.1 gives the result. □

Corollary 2.2 shows that, similar to the entries of Y_j, the rows of $H_j^{-1} Y_j$ can be bounded using the previous Lyapunov residual norm. However, due to the influence of H_j^{-1}, the occurring constants in front of the Lyapunov residual norms can potentially be very large.

2.4 Relaxation strategies and stopping criteria for the inner iteration

In order to achieve accurate results, the difference between the true and computed residual, the residual gap, needs to be small.

Corollary 2.2 indicates that $\|e_k^* H_j^{-1} Y_j\|$ decreases with the computed Lyapunov residual, and hence $\|s_k\|$ may be relaxed during the RKSM iteration in a manner inverse proportional to the norm of the Lyapunov residual (assuming the Lyapunov residual norm decreases).

Theorem 2.2 (Theoretical relaxation strategy in RKSM) *Let the assumptions of Theorem 2.1 and Corollary 2.2 hold. Assume we carry out j steps of Algorithm 1 always using the explicit projection $T_j^{expl.}$. If we choose the solve tolerances $\|s_k\|$, $1 \leq k \leq j$ for the inexact solves within RKSM such that*

$$\|s_k\| \leq \tau_k^{LS} = \begin{cases} \frac{\varepsilon}{j c_A \|e_1^* H_j^{-1}\| \|R_0^{true}\|}, & k = 1, \\ \frac{\varepsilon}{\frac{j}{\phi_j^{(k)} \|g_{k-1}\|} \|R_{k-1}^{comp.}\| + j c_A \|e_k^* H_j^{-1}\| \|R_{k-1}^{true}\|}, & 1 < k \leq j, \end{cases} \tag{2.23}$$

with the same notation as before, then, for the residual gap $\|\eta_j\| \leq \varepsilon$ holds.

Proof Consider a single addend in the sum expression (2.15) for the norm of the residual gap

$$\|s_k\| \|e_k^* H_j^{-1} Y_j\| \leq \|s_k\| \left(\frac{1}{\phi_j^{(k)} \|g_{k-1}\|} \|R_{k-1}^{\text{comp.}}\| + c_A \|e_k^* H_j^{-1}\| \|R_{k-1}^{\text{true}}\| \right), k > 1.$$

Choosing s_k such that (2.23) is satisfied for $1 \leq k \leq j$ then gives $\|\eta_j\| \leq \sum_{k=1}^{j} \frac{\varepsilon}{j} = \varepsilon$ where we have used (2.15), Theorem 2.1, Corollaries 2.1 and 2.2. □

The true norms $\|R_{k-1}^{\text{true}}\|$ can be estimated by $\|R_{k-1}^{\text{true}}\| \leq \|R_{k-1}^{\text{comp.}}\| + \|\eta_{k-1}\|$. For this, we might either use some estimation for $\|\eta_{k-1}\|$ or simply assume that all previous residual gaps were sufficiently small, i.e., $\|\eta_{k-1}\| \leq \varepsilon$.

Practical relaxation strategies for inexact RKSM The relaxation strategy in Theorem 2.2 is far from practical. First, the established bounds for the entries and rows of Y_j and $H_j^{-1} Y_j$, respectively, can be a vast overestimation of the true norms. Hence, the potentially large denominators in (2.23) result in very small solve tolerances τ_k^{LS} and, therefore, prevent a substantial relaxation of the inner solve accuracies. Second, several quantities in the used bounds are unknown at step $k < j$, e.g. H_j^{-1}, Y_j, and the constants $\phi_j^{(k)}$. If we knew H_j^{-1}, Y_j, we could employ a relaxation strategy of the form $\|s_k\| \leq \frac{\varepsilon}{j \|e_k^* H_j^{-1} Y_j\|}$ and only use Corollary 2.2 as theoretical indication that $\|e_k^* H_j^{-1} Y_j\|$ decreases as the outer method converges.

In the following we therefore aim to develop a more practical relaxation strategy by trying to estimate the relevant quantity $\|e_k^* H_j^{-1} Y_j\|$ differently using the most recent available data. Suppose an approximate solution with residual norm $\|R_k^{\text{true}}\| \leq \varepsilon$, $0 < \varepsilon \ll 1$ is desired which should be found after at most $j_{\max}$ rational Arnoldi steps. This goal is achieved if $\|\eta_{j_{\max}}\| \leq \frac{\varepsilon}{2}$ and if $\|R_{j_{\max}}^{\text{comp.}}\| \leq \frac{\varepsilon}{2}$ is obtained by the inexact RKSM.

Consider the left null vectors of the augmented Hessenberg matrices, $\omega_k \underline{H}_k = 0$, $k \leq j_{\max}$. It is easy to show that ω_k can be updated recursively, in particular it is possible to compute $\omega_m = (\omega_k)_{1:m+1}$, $m \leq k \leq j_{\max}$. Consequently, at the beginning of step k we already have $\underline{H}_{k-1}$ and hence, also the first k entries of $\omega_{j_{\max}}$ without knowing the full matrix $\underline{H}_{j_{\max}}$. Using Lemma 2.2 and proceeding similar as in the proof of Corollary 2.2 results in

$$\begin{aligned}
\|e_k^* H_{j_{\max}}^{-1} Y_{j_{\max}}\| &= \left\| e_k^* H_{j_{\max}}^{-1} \left(\begin{bmatrix} Y_{k-1} & 0 \\ 0 & 0 \end{bmatrix} + \Delta Y_{j_{\max},\ell-1} \right) \right\| \\
&= \left\| -\frac{\omega_{j_{\max}}(k)}{\phi_{j_{\max}}^{(k)}} [h_{k,k-1} e_{k-1}^* H_{k-1}^{-1}, *] \begin{bmatrix} Y_{k-1} & 0 \\ 0 & 0 \end{bmatrix} + e_k^* H_{j_{\max}}^{-1} \Delta Y_{j_{\max},\ell-1}) \right\| \\
&= \left\| -\frac{h_{k,k-1}\omega_k(k)}{\phi_{j_{\max}}^{(k)}} e_{k-1}^* H_{k-1}^{-1} Y_{k-1} + e_k^* H_{j_{\max}}^{-1} \Delta Y_{j_{\max},k-1} \right\| \\
&\leq \left| \frac{h_{k,k-1}\omega_k(k)}{\phi_{j_{\max}}^{(k)}} \right| \|e_{k-1}^* H_{k-1}^{-1} Y_{k-1}\| + \|e_k^* H_{j_{\max}}^{-1}\| c_A \| R_{k-1}^{\text{true}}\| \\
&\approx \left| \frac{h_{k,k-1}\omega_k(k)}{\phi_{j_{\max}}^{(k)}} \right| \|e_{k-1}^* H_{k-1}^{-1} Y_{k-1}\|,
\end{aligned}$$

if $\|e_k^* H_{j_{\max}}^{-1}\| c_A \| R_{k-1}^{\text{true}}\|$ is small. Only the scaling parameter $\phi_{j_{\max}}^{(k)}$ contains missing data at the beginning of step k, because H_{k-1}, Y_{k-1}, ω_k are known from the previous step. We suggest to omit the unknown data and use the following *practical relaxation strategy*

$$\|s_k\| \leq \tau_k^{\text{LS}} = \begin{cases} \delta \frac{\varepsilon}{j_{\max}}, & k = 1, \\ \delta \frac{\varepsilon}{j_{\max} \|h_{k,k-1} e_{k-1}^* H_{k-1}^{-1} Y_{k-1}\|}, & k > 1, \end{cases} \tag{2.24a}$$

where $0 < \delta \leq 1$ is a safeguard constant to accommodate for the estimation error resulting from approximating $\|e_k^* H_{j_{\max}}^{-1} Y_{j_{\max}}\|$ and omitting unknown quantities (e.g., $\phi_{j_{\max}}^{(k)}$).

Remark 2.3 The parameter δ is a safeguard constant. Choosing δ too large may lead to divergence of the inexact method, choosing it too small could lead to unnecessary extra work. Investigating the choice of this constant is future work, however we would like to remark, that similar constants have been used in other works on inexact iterative solves, e.g. [46].

The reader might notice by following Algorithm 1 closely, that the built up subspace at the beginning of iteration step $k \leq j_{\max}$ is already k-dimensional and since we are using the explicit projection $T_k^{\text{expl.}}$ to define the Galerkin systems, we can already compute Y_k directly after building Q_k. This amounts to a simple rearrangement of Algorithm 1 by moving Lines 7, 8 before Line 3. Hence, the slight variation

$$\|e_k^* H_{j_{\max}}^{-1} Y_{j_{\max}}\| \approx \left|\frac{\omega_k(k)}{\phi_{j_{\max}}^{(k)}}\right| \|[-h_{k,k-1}e_{k-1}^* H_{k-1}^{-1}, 1]Y_k\|$$

of the above estimate is obtained, which suggests the use of the (slightly different) *practical relaxation strategy*

$$\|s_k\| \leq \tau_k^{\mathrm{LS}} = \begin{cases} \delta \frac{\varepsilon}{j_{\max}}, & k = 1, \\ \delta \frac{\varepsilon}{j_{\max}\|[-h_{k,k-1}e_{k-1}^* H_{k-1}^{-1}, 1]Y_k\|}, & k > 1. \end{cases} \tag{2.24b}$$

For both dynamic stopping criteria, in order to prevent too inaccurate and too accurate linear solves, it is reasonable to enforce $\tau_k^{\mathrm{LS}} \in [\tau_{\min}^{\mathrm{LS}}, \tau_{\max}^{\mathrm{LS}}]$, where $0 < \tau_{\min}^{\mathrm{LS}} < \tau_{\max}^{\mathrm{LS}} \leq 1$ indicate minimal and maximal linear solve thresholds.

The numerical examples in Sect. 4 show that these practical relaxation strategies are effective and can reduce the number of inner iterations for RKSM by up to 50%.

2.5 Implementation issues and generalizations

This section contains several remarks on the implementation of the inexact RKSM, in particular the case when the right hand side of the Lyapunov equation has rank greater than one, as well as considerations of the inner iterative solver and preconditioning. We also briefly comment on extensions to generalized Lyapunov equations and algebraic Riccati equations.

The case $r > 1$ The previous analysis was restricted to the case $r = 1$ but the block situation, $r > 1$, can be handled similarly by using appropriate block versions of the relevant quantities, e.g., $q_k,\ w_k,\ s_k \in \mathbb{C}^{n\times r}$, $h_{ij} \in \mathbb{R}^r$, $\omega \in \mathbb{C}^{r\times(j+1)r}$, and e_k by $e_k \otimes I_r$, as well as replacing absolute values by spectral norms in the right places. When solving the linear system with r right hand sides q_k, such that, $\|s_k\| \leq \tau_k^{\mathrm{LS}}$ is achieved, one can either used block variants of the iterative methods (see, e.g., [49]), or simply apply a single vector method and sequentially consider every column $q_k(:, \ell)$, $\ell = 1, \ldots, r$ and demand that $s_k(:, \ell) \leq \tau_k^{\mathrm{LS}}/r$.

Choice of inner solver One purpose of low-rank solvers for large matrix equations is to work in a memory efficient way. Using a long-recurrence method such as GMRES to solve unsymmetric inner linear systems defies this purpose in some sense because it requires storing the full Krylov basis. Unless a very good preconditioner is available, this can lead to significant additional storage requirements within the inexact low-rank method, where the Krylov method consumes more memory than the actual low-rank Lyapunov solution factor of interest. Therefore we exclusively used short-recurrence Krylov methods (e.g., BiCGstab) for the numerical examples defined by an unsymmetric matrix A.

Preconditioning The preceding relaxation strategies relate to the residuals s_k of the underlying linear systems. For enhancing the performance of the Krylov subspace methods, using preconditioners is typically inevitable. Then the inner iteration itself inherently only works with the preconditioned residuals $s_k^{\text{Prec.}}$ which, if left or two-sided preconditioning is used, are different from the residuals s_k of the original linear systems. Since $\|s_k^{\text{Prec.}}\| \leq \tau_k^{\text{LS}}$ does not imply $\|s_k\| \leq \tau_k^{\text{LS}}$ this can result in linear systems solved not accurately enough to ensure small enough Lyapunov residual gaps. Hence, some additional care is needed to respond to these effects from preconditioning. The obvious approach is to use right preconditioning which gives $\|s_k^{\text{Prec.}}\| = \|s_k\|$.

Complex shifts In practice A, B are usually real but some of the shift parameters can occur in complex conjugate pairs. Then it is advised to reduce the amount of complex operations by working with a real rational Arnoldi method [37] that constructs a real rational Arnoldi decomposition and slightly modified formulae for the computed Lyapunov residuals, in particular for F_j. The actual derivations are tedious and are omitted here for the sake of brevity, but our implementation for the numerical experiments works exclusively with the real Arnoldi process.

Generalized Lyapunov equations In many practical relevant situations, generalized Lyapunov equations of the form

$$AXM^* + MXA^* = -BB^*, \tag{2.25}$$

with an additional, nonsingular matrix $M \in \mathbb{R}^{n\times n}$ have to be solved. Projection methods tackle (2.25) by implicitly running on equivalent Lyapunov equations defined by $A_M := L_M^{-1}AU_M^{-1}$, $B_M := L_M^{-1}B$ using a factorization $M = L_M U_M$, which could be a LU-factorization or, if M is positive definite, a Cholesky factorization ($U_M = L_M^*$). Other possibilities are $L_M = M$, $U_M = I$ and $L_M = I$, $U_M = M$. Basis vectors for the projection subspace are obtained by

$$Q_1\beta = L_M^{-1}B, \quad (A - \xi_{j+1}M)\hat{w}_{j+1} = L_M q_j, \quad w_{j+1} = U_M\hat{w}_{j+1}. \tag{2.26}$$

After convergence, the low-rank approximate solution of the original problem (2.25) is given by $X \approx (U_M^{-1}Q_j)Y_j(U_M^{-1}Q_j)^*)$, where Y_j solves the reduced Lyapunov equation defined by the restrictions of A_M and B_M. This requires solving extra linear systems defined by (factors of) M in certain stages of Algorithm 1: setting up the first basis vector, building the restriction T_j of A_M (either explicitly or implicitly using (2.7)), and recovering the approximate solution after termination. Since the coefficients of these linear system do not change throughout the iteration, often sparse factorizations of M are computed once at the beginning and reused every time they are needed. In this case the above analysis can be applied with minor changes of the form

$$s_j := L_M q_j - (A - \xi_{j+1}M)\hat{\tilde{w}}_{j+1}, \quad \|s_j\| \leq \tau_j^{\text{LS}}, \quad w_{j+1} = U_M\hat{\tilde{w}}_{j+1},$$

for the inexact linear solves. We obtain an inexact rational Arnoldi decomposition with respect to A_M of the form

$$\begin{aligned} A_M Q_j &= Q_j \hat{T}_j^{\text{expl.}} + \hat{g}_j e_j^* h_{j+1,j} H_j^{-1} - (I - Q_j Q_j^*) L_M^{-1} S_j H_j^{-1}, \\ \text{with} \quad \hat{T}_j^{\text{expl.}} &= Q_j^* A_M Q_j, \quad \hat{g}_j = q_{j+1}\xi_{j+1} - (I - Q_j Q_j^*) A_M q_{j+1}. \end{aligned}$$

If $Q_j Y_j Q_j^*$ is an approximate solution of the equivalent Lyapunov equation defined by $A_M,\ B_M$, and Y_j solves the reduced Lyapunov equation defined by $\hat{T}_j^{\text{expl.}}$ and $Q_j^* B_M$, then the associated residual is

$$\begin{aligned} &A_M Q_j Y_j Q_j^* + Q_j Y_j Q_j^* A_M^* + B_M B_M^* = \hat{F}_j + \Delta\hat{F}_j + (\hat{F}_j + \Delta\hat{F}_j)^*, \\ &\hat{F}_j := \hat{g}_j^* h_{j+1,j} H_j^{-1} Y_j Q_j^*, \quad \Delta\hat{F}_j := (I - Q_j Q_j^*) L_M^{-1} S_j H_j^{-1} Y_j Q_j^*. \end{aligned}$$

Hence, the generalized residual gap is $\hat{\eta}_j = \Delta\hat{F}_j$. If $L_M = I, U_M = M$, bounding $\|\hat{\eta}_j\|$ works in the same way as in the case $M = I$, otherwise an additional constant $1/\sigma_{\min}(L_M)$ (or an estimation thereof) has to be multiplied to the established bounds. Allowing inexact solves of the linear systems defined by (factors of) M substantially complicates the analysis. In particular, the transformation to a standard Lyapunov equation is essentially not given exactly since, in that case, only a perturbed version of A_M and its restriction are available. This situation is, hence, similar to the case when no exact matrix vector products with A are available. If $L_M \neq I$, also B_M is not available exactly leading to further perturbations in the basis generation. For these reasons, we will not further pursue inexact solves with M or its factors. This is also motivated from practical situations, where solving with M, or computing a sparse factorization thereof, is usually much less demanding compared to factorising $A - \xi_{j+1} M$.

Extended Krylov subspace methods A special case of the rational Krylov subspace appears when only the shifts zero and infinity are used, leading to the extended Krylov subspace $\mathscr{E}\mathscr{K}_k(A_M, B_M) = \mathscr{K}_k(A_M, B_M) \cup \mathscr{K}_k(A_M^{-1}, A_M^{-1} B_M)$ (using the notation from the previous subsection). Usually, in the resulting extended Arnoldi process the basis is expanded by vectors from $\mathscr{K}_k(A_M, B_M)$ and $\mathscr{K}_k(A_M^{-1}, A_M^{-1} B_M)$ in an alternating fashion, starting with $\mathscr{K}_k(A_M, B_M)$. The extended Krylov subspace method (EKSM) [43] for (1.1) and (2.25) uses a Galerkin projection onto $\mathscr{E}\mathscr{K}_k(A_M, B_M)$. In each step the basis is orthogonally expanded by $w_{j+1} = [A_M q_j(:, 1 : r), A_M^{-1} q_j(: , r + 1 : 2r)]$, where q_j contains the last $2r$ basis vectors. This translates to the following linear systems and matrix vector products

$$\begin{aligned} U_M z = q_j(:, 1 : r), &\quad L_M w_{j+1}(:, 1 : r) = Az, \\ \text{and} \quad Az = L_M q_j(:, r + 1 : 2r), &\quad w_{j+1}(:, r + 1 : 2r) = U_M z, \end{aligned}$$

that have to be dealt with. Similar formula for the implicit restriction of A and the Lyapunov residual as in RKSM can be found in [43]. Since these coefficient matrices do not change during the iteration, a very efficient strategy is to compute, if possible, sparse factorizations of A, M once before the algorithm and reuse them in every step. Incorporating inexact linear solves by using inexact sparse factorizations $A \approx \tilde{L}_A \tilde{U}_A$, $M \approx \tilde{L}_M \tilde{U}_M$ would make it difficult to incorporate relaxed solve tolerances, since there is little reason to compute a less accurate factorization once a highly accurate one has been constructed. For the same reasons stated in the paragraph above, we restrict ourselves to the iterative solution of the linear systems defined by A. These linear systems affect only the columns in $w_{j+1}(:, r+1 : 2r)$. In particular, by proceeding as for inexact RKSM, one can show that

$$A_M Q_j = Q_{j+1} T^{\text{expl.}}_{j+1,j} - (I - Q_j Q_j^*) S_j^{\text{EK}} \quad \text{with}$$
$$T^{\text{expl.}}_{j+1,j} = Q_{j+1}^* A_M Q_j, \quad S_j^{\text{EK}} := [s_1^{\text{EK}}, \dots, s_j^{\text{EK}}], \quad s_i^{\text{EK}} := [0, U_M^{-1} s_i] \in \mathbb{C}^{n \times 2r},$$

where $s_i := L_M q_i(:, r+1 : 2r) - Az$, $1 \leq i \leq j$. Note that, allowing inexact solves w.r.t. M or even inexact matrix vector products with A, M, would destroy the zero block columns in s_i^{EK}.

Algebraic Riccati equations The RKSM in Algorithm 1 can be generalized to compute low-rank approximate solutions of generalized algebraic Riccati equations (AREs)

$$AXM^* + MXA^* - MXCC^*XM + BB^* = 0, \quad C \in \mathbb{C}^{n \times p}, \quad p \ll n, \tag{2.27}$$

see, e.g., [44,48]. The majority of steps in Algorithm 1 remain unchanged, the main difference is that the Galerkin system is now a reduced ARE

$$T_j Y_j + Y_j T_j^* - Y_j (Q_j^* C_M)(C_M^* Q_j) Y_j + Q_j^* B_M B_M^* Q_j = 0, \quad C_M := U_M^{-*} C,$$

which has to be solved. Since we do not alter the underlying rational Arnoldi process, most of the properties of the RKSM hold again, especially the residual formulas in both the exact and inexact case, and a residual gap is defined again as in the Lyapunov case. Differences occur in the bounds for the entries of Y_j and rows of $H_j^{-1} Y_j$, since Theorem 2.1 cannot be formulated in the same way. Under some additional assumptions, a bound of the form $\|\varDelta Y_{j,j-1}\| \leq c_{\text{ARE}} \|R_{j-1}\|$ can be established [44], where the constant c_{ARE} is different from c_A. We leave concrete generalizations of Theorem 2.1, Corollaries 2.1, 2.2 for future research and only show in some experiments that relaxation strategies of the form (2.24a), (2.24b) also work for the inexact RKSM for AREs.

3 The inexact low-rank ADI iteration

3.1 Derivation, basic properties, and introduction of inexact solves

Using the Cayley transformation $C(A, \alpha) := (A + \alpha I)^{-1}(A - \overline{\alpha} I)$, for $\alpha \in \mathbb{C}_-$, (1.1) can be reformulated as discrete-time Lyapunov equation (symmetric Stein equation)

$$X = C(A, \alpha) X C(A, \alpha)^* - 2 \operatorname{Re}(\alpha) B(\alpha) B(\alpha)^*, \; B(\alpha) := (A + \alpha I)^{-1} B.$$

For suitably chosen α_j this motivates the iteration

$$X_j = C(A,\alpha_j)X_{j-1}C(A,\alpha_j)^* - 2\,\mathrm{Re}\,\left(\alpha_j\right) B(\alpha_j)B(\alpha_j)^*, \tag{3.1}$$

which is known as alternating directions implicit (ADI) iteration [54] for Lyapunov equations. It converges for shift parameters $\alpha_j \in \mathbb{C}_-$ because $\rho(C(A,\alpha_j)) < 1$ and it holds [10,33,40,54]

$$X_j - X = \mathfrak{C}_j(X_0 - X)\mathfrak{C}_j^*, \tag{3.2}$$

$$R_j = AX_j + X_jA^* + BB^* = \mathfrak{C}_j R_0 \mathfrak{C}_j^*, \tag{3.3}$$

where $\mathfrak{C}_j := \prod_{i=1}^{j} C(A,\alpha_i)$.

A low-rank version of the ADI iteration is obtained by setting $X_0 = 0$ in (3.1), exploiting that $(A+\alpha_j I)^{-1}$ and $(A - \overline{\alpha_i} I)$ commute for $i, j \geq 1$, and realising that the iterates are given in low-rank factored form $X_j = Z_j Z_j^*$ with low-rank factors Z_j constructed by

$$\begin{aligned} Z_j &= \left[\gamma_1 v_1,\ \gamma_2 v_2,\ \ldots, \gamma_j v_j\right] = \left[Z_{j-1}, \gamma_j v_j\right], \quad \gamma_j := \sqrt{-2\,\mathrm{Re}\,\left(\alpha_j\right)},\\ v_j &= (A - \overline{\alpha_{j-1}} I)(A+\alpha_j I)^{-1} v_{j-1},\ j \geq 1, \quad v_1 := (A+\alpha_1 I)^{-1}B, \end{aligned} \tag{3.4}$$

see [33,39] for more detailed derivations. Thus, in each step Z_{j-1} is augmented by r new columns v_j. Moreover, from (3.3) with $X_0 = 0$ and (3.4) it is evident that

$$R_j = w_j w_j^*, \quad w_j := \mathfrak{C}_j B = w_{j-1} - \gamma_j^2 (A+\alpha_j I)^{-1} w_{j-1}, \quad w_0 := B, \tag{3.5}$$

see also [10,55]. Hence, the residual matrix has at most rank r and its norm can be cheaply computed as $\|R_j\|_2 = \|w_j^* w_j\|_2$ which coined the name *residual factors* for the w_j. The low-rank ADI (LR-ADI) iteration using these residual factors is

$$v_j = (A+\alpha_j I)^{-1} w_{j-1}, \quad w_j = w_{j-1} + \gamma_j^2 v_j, \quad w_0 := B. \tag{3.6a}$$

For generalized Lyapunov equations (2.25), this iteration can be formally applied to an equivalent Lyapunov equations defined, e.g. by $M^{-1}A$, $M^{-1}B$. Basic manipulations (see, e.g. [10,30]) resolving the inverses of M yield the generalized LR-ADI iteration

$$v_j = (A+\alpha_j M)^{-1} w_{j-1}, \quad w_j = w_{j-1} + \gamma_j^2 M v_j, \quad w_0 := B. \tag{3.6b}$$

As for RKSM, the choice of shift parameters α_j is crucial for a rapid convergence. Many approaches have been developed for this problem, see, e.g., [40,52,54], where shifts are typically obtained from (approximate) eigenvalues of A, M. Among those exist asymptotically optimal shifts for certain situations, e.g., $A = A^*$, $M = M^*$ and $\Lambda(A, M) \subset \mathbb{R}_-$. Adaptive shift generation approaches, where shifts are computed during the running iteration, were proposed in [12,30] and often yield better results,

Algorithm 2: Inexact LR-ADI iteration.

```
Input  : Matrices $A,\ M,\ B$ defining (2.25), set of shift parameters $\{\alpha_1, \dots, \alpha_{j_{\max}}\} \subset \mathbb{C}_-$,
         tolerance $0 < \varepsilon \ll 1$.
Output: $Z_j \in \mathbb{C}^{n \times rj}$, such that $ZZ^* \approx X$.
$w_0 = B, \quad Z_0 = [\,], \quad R_0 = \|w_0^* w_0\|, \quad j = 1.$
while $\|R_{j-1}\| \geq \varepsilon$ do
    Get $v_j$ s.t. $s_j = w_{j-1} - (A + \alpha_j M)v_j$, $\|s_j\| \leq \delta_j$.
    $w_j = w_{j-1} - 2\,\mathrm{Re}\,(\alpha_j)\,Mv_j$.
    $Z_j = [Z_{j-1},\ \sqrt{-2\,\mathrm{Re}\,(\alpha_j)}v_j]$.
    $R_j = \|w_j^* w_j\|_2$.
    $j = j + 1$.
```

especially for nonsymmetric coefficients with complex spectra. In this work we mainly work with these dynamic shift selection techniques. The main computational effort in each step is obviously the solution of the shifted linear system with $(A + \alpha_j M)$ and r right hand sides for v_j in (3.6). Allowing inexact linear solves but keeping the other steps in (3.6) unchanged results in the *inexact low-rank ADI iteration* illustrated in Algorithm 2. We point out that a different notion of an inexact ADI iteration can be found in [34] in the context of operator Lyapunov equations, where inexactness refers to the approximation of infinite dimensional operators by finite dimensional ones.

Of course, allowing $s_j \neq 0$ will violate some of the properties that were used to derive the LR-ADI iteration. Inexact solves within the closely related Smith methods have been investigated in, e.g., [40,50], from the viewpoint of an inexact nonstationary iteration (3.1) that led to rather conservative results on the allowed magnitude of the norm of the linear system residual s_j. The analysis we present here follows a different path by exploiting the well-known connection of the LR-ADI iteration to rational Krylov subspaces [20,32,33].

Theorem 3.1 [32,33,55] *The low-rank solution factors Z_j after j steps of the exact LR-ADI iteration ($\|s_j\| = 0,\ \forall j \geq 1$) span a (block) rational Krylov subspace:*

$$\mathrm{range}\left(Z_j\right) \subseteq \mathrm{range}\left(\left[(A + \alpha_1 M)^{-1}B, \dots, \left[\prod_{i=2}^{j}(A + \alpha_i M)^{-1}\right](A + \alpha_1 M)^{-1}B\right]\right). \quad (3.7)$$

Although for LR-ADI, the Z_j do not have orthonormal columns and there is no rational Arnoldi process in Algorithm 2, it is still possible to find decompositions similar to (2.5) and (2.11).

Theorem 3.2 *The low-rank solution factors Z_j after j steps of the inexact LR-ADI iteration (Algorithm 2) satisfy a rational Arnoldi like decomposition*

$$AZ_j = MZ_jT_j + w_jg_j^* - S_j\varGamma_j, \quad S_j := [s_1,\dots,s_j], \quad \textit{where} \tag{3.8}$$

$$T_j = -\begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ \gamma_1\gamma_2 & \alpha_2 & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \gamma_1\gamma_j & \dots & & \alpha_j \end{bmatrix} \otimes I_r, \quad g_j := \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_j \end{bmatrix} \otimes I_r, \quad \varGamma_j := \operatorname{diag}(g_j). \tag{3.9}$$

Moreover, the Lyapunov residual matrix associated with the approximation $X_j = Z_jZ_j^* \approx X$ is given by

$$\begin{aligned} R_j^{true} &= AZ_jZ_j^*M^* + MZ_jZ_j^*A^* + BB^* \\ &= -S_j\varGamma_jZ_j^*M^* - MZ_j\varGamma_jS_j^* + w_jw_j^*. \end{aligned} \tag{3.10}$$

Note that here, T_j and g_j denote different quantities than in Sect. 2.

Proof For $S_j = 0$ the decomposition (3.8) was established in [30,55] and can be entirely derived from the relations (3.6). Inexact solves in the sense $w_{j-1} - (A + \alpha_jM)v_j = s_j$ can be inserted in a straightforward way leading to (3.8). By construction, it holds $B = w_0 = w_j - MZ_jg_j$ which, together with (3.8), gives

$$R_j^{\text{true}} = MZ_j(T_j + T_j^* + g_jg_j^*)Z_j^*M^* + w_jw_j^* - S_j\varGamma_jZ_j^*M^* - MZ_j\varGamma_jS_j^*$$

and (3.10) follows by verifying that $T_j + T_j^* = -g_jg_j^*$.

Remark 3.1 The LR-ADI iteration is in general not a typical Galerkin projection method using orthonormal bases of the search spaces and solutions of reduced matrix equations. For instance, in [55] it is shown that the exact LR-ADI iteration can be seen as an implicit Petrov-Galerkin type method with a hidden left projection subspace. In particular, as we exploited in the proof above, the relation $T_j + T_j^* + g_jg_j^* = 0$ can be interpreted as reduced matrix equation solved by the identity matrix I_j. It is also possible to state a decomposition similar to (3.8) which incorporates $w_0 = B$ instead of w_j [20,55]. This can be used to state conditions which indicate when the LR-ADI approximation satisfies a Galerkin condition [20, Theorem 3.4]. Similar to inexact RKSM, if $s_j \neq 0$ these result do in general not hold any more.

3.2 Computed Lyapunov residuals, residual gap, and relaxation strategies in inexact LR-ADI

Similar to inexact RKSM when inexact solves are allowed in LR-ADI, the computed Lyapunov residuals $R_j^{\text{comp.}} = w_jw_j^*$ are different from the true residuals R_j^{true} and, thus, $\|w_j\|^2$ ceases to be a safe way to assess the quality of the current approximation $Z_jZ_j^*$. In the exact case, $s_j = 0$, it follows from (3.5) and $\rho_j := \rho(C(A, \alpha_j)) < 1$

that the Lyapunov residual norms decrease in the form $\|R_j\| \leq c\rho_j^2\|R_{j-1}\|$ for some $c \geq 1$. For the factors w_j of the computed Lyapunov residuals $R_j^{\text{comp.}}$ in the inexact method we have the following result.

Lemma 3.1 *Let w_j^{exact} be the factors of the Lyapunov residuals of the exact LR-ADI method, i.e., $s_i = 0$, and $C_i := C(A, \alpha_i)$, $1 \leq i \leq j$. The factors w_j of the computed Lyapunov residuals of the inexact LR-ADI are given by*

$$w_j = w_j^{exact} + \sum_{i=1}^{j} \left[\prod_{k=i+1}^{j} C_k \right] (C_i - I)s_i.$$

Proof For simplicity, let $r = 1$, $M = I_n$, and exploit $C_j = A_{-\overline{\alpha_j}} A_{\alpha_j}^{-1} = I + \gamma_j^2 A_{\alpha_j}^{-1}$, $A_\alpha := A + \alpha I$. Denoting the errors by $\delta v_j = A_{\alpha_j}^{-1} w_{j-1} - v_j = A_{\alpha_j}^{-1} s_j$ for $j \geq 1$, we have

$$\begin{aligned} w_j &= w_{j-1} + \gamma_j^2 v_j = w_{j-1} + \gamma_j^2 A_{\alpha_j}^{-1} w_{j-1} - \gamma_j^2 \delta v_j = C_j w_{j-1} - \gamma_j^2 \delta v_j \\ &= \ldots = \prod_{k=1}^{j} C_k w_0 - \sum_{i=1}^{j} \gamma_i^2 \left[\prod_{k=i+1}^{j} C_k \right] \delta v_i \\ &= \prod_{k=1}^{j} C_k w_0 - \sum_{i=1}^{j} \left[\prod_{k=i+1}^{j} C_k \right] (C_i - I)s_i \end{aligned}$$

and we notice that the first term is exactly the exact Lyapunov residual factor from (3.5). □

Constructing $w_j w_j^*$ from the above formula and taking norms indicates that, by the contraction property of C_i, the linear system residuals s_i get damped. In fact, similar to the inexact projection methods, in practice we often observe that the computed Lyapunov residual norms $\|R_j^{\text{comp.}}\| = \|w_j w_j^*\|$ also show a decreasing behavior.

In the next step we analyze the difference between the computed and true residuals, in a similar manner as we did for the RKSM in Sect. 2. Theorem 3.2 motivates the definition of a *residual gap* analogue to the inexact RKSM.

Definition 2 The *residual gap* after j steps of the inexact LR-ADI iteration is given by

$$\begin{aligned} \Delta R_j^{\text{ADI}} &:= R_j^{\text{true}} - R_j^{\text{comp.}} = R_j^{\text{true}} - w_j w_j^* = \eta_j^{\text{ADI}} + (\eta_j^{\text{ADI}})^*, \\ \eta_j^{\text{ADI}} &:= -S_j \Gamma_j Z_j^* M^*. \end{aligned} \tag{3.11}$$

Assuming we have $\|R_j^{\text{comp.}}\| = \|w_j w_j^*\| \leq \varepsilon$ and are able to bound the residual gap, e.g., $\|\Delta R_j^{\text{ADI}}\| \leq \varepsilon$, then we achieve small true residual norms $\|R_j^{\text{true}}\| \leq 2\varepsilon$. A theoretical approach for bounding $\|\Delta R_j^{\text{ADI}}\|$ is given in the next theorem.

Theorem 3.3 (Theoretical relaxation for inexact LR-ADI) *Let the residual gap be given by Definition 2 with w_j, γ_j as in (3.4)–(3.6b). Let $j_{\max}$ be the maximum number of steps of Algorithm 2, $\sigma_k := \|M(A+\alpha_k M)^{-1}\|$, and $0 < \varepsilon < 1$ the desired residual tolerance.*

(a) If, for $1 \leq k \leq j_{\max}$, the linear system residual satisfies

$$\|s_k\| \leq \frac{1}{2}\left(\sqrt{\|w_{k-1}\|^2 + \frac{2\varepsilon}{\sigma_k \gamma_k^2 j_{\max}}} - \|w_{k-1}\|\right), \tag{3.12a}$$

then $\|\Delta R_{j_{\max}}^{ADI}\| \leq \varepsilon$.

(b) If, for $1 \leq k \leq j_{\max}$, the linear system residual satisfies

$$\|s_k\| \leq \frac{1}{2}\left(\sqrt{\|w_{k-1}\|^2 + \left(\frac{k\varepsilon}{j_{\max}} - 2\|\eta_{k-1}^{ADI}\|\right)\frac{2}{\sigma_k \gamma_k^2}} - \|w_{k-1}\|\right), \tag{3.12b}$$

then $\|\Delta R_{j_{\max}}^{ADI}\| \leq \varepsilon$.

Proof Consider the following estimate

$$\begin{aligned}\|\Delta R_{j_{\max}}^{\text{ADI}}\| &\leq 2\|\eta_{j_{\max}}^{\text{ADI}}\| = 2\|S_{j_{\max}}\Gamma_{j_{\max}}Z_{j_{\max}}^* M^*\| \\ &\leq 2\|S_{j_{\max}-1}\Gamma_{j_{\max}-1}Z_{j_{\max}-1}^* M^*\| + 2\gamma_{j_{\max}}^2\|s_{j_{\max}}\|\|Mv_{j_{\max}}\| \qquad (3.13)\\ &\leq 2\sum_{k=1}^{j_{\max}} \gamma_k^2 \|s_k\|\|Mv_k\|. \qquad (3.14)\end{aligned}$$

Moreover,

$$\|Mv_k\| = \|M(A+\alpha_k M)^{-1}(w_{k-1} - s_k)\| \leq \sigma_k(\|w_{k-1}\| + \|s_k\|). \tag{3.15}$$

If the linear residual norms $\|s_k\|$ are so that each addend in the sum (3.14) is smaller than $\frac{\varepsilon}{j_{\max}}$ we achieve $\|\eta_{j_{\max}}^{\text{ADI}}\| \leq \varepsilon/2$. With $\phi_k := 2\gamma_k^2\sigma_k$, $\omega_{k-1} := \|w_{k-1}\| = \sqrt{\|R_{k-1}\|}$ this means we require $\phi_k(\omega_{k-1}\|s_k\| + \|s_k\|^2) \leq \dfrac{\varepsilon}{j_{\max}}$. Hence, the desired largest allowed value $\|s_k\|$ is given by the positive root of the inherent quadratic equation $\phi_k(\omega_k\varsigma + \varsigma^2) - \frac{\varepsilon}{j_{\max}} = 0$ such that

$$\|s_k\| \leq \frac{1}{2}\left(\sqrt{\omega_{k-1}^2 + \frac{4\varepsilon}{\phi_k j_{\max}}} - \omega_{k-1}\right) \tag{3.16}$$

leading to the desired result (a). The second strategy can be similarly shown by using (3.13) and

$$\|\eta_k^{\text{ADI}}\| \leq 2\|\eta_{k-1}^{\text{ADI}}\| + \phi_k(\|w_{k-1}\|\|s_k\| + \|s_k\|^2),$$

and then finding $\|s_k\|$ such that the right hand side in the above inequality is less than $\dfrac{k\varepsilon}{j_{\max}}$. □

The motivation behind the second stopping strategy (3.12b) is that we can take the previous $\|\eta_{k-1}^{\mathrm{ADI}}\|$ into account. This can be helpful if at steps $i \leq k-1$ the used iterative method produced smaller linear residual norms than demanded, such that the linear residual norms $\|s_i\|$ are allowed to grow slightly larger at later iteration steps $i > k-1$.

Convergence analysis of inexact ADI with relaxed solve tolerances Using the relaxation strategies in Theorem 3.3 reveals, under some conditions, further insight into the behaviour of the computed LR-ADI residual norms.

Theorem 3.4 (Decay of computed LR-ADI residuals) *Assume* $\|C_j\| < 1$, $j = 1, \ldots, j_{\max}$ *such that the residuals* $\|R_j^{exact}\|$ *of the exact LR-ADI iteration decrease. Let the assumptions of Theorem 3.3 hold. If the relaxation (3.12a) is used in the inexact LR-ADI iteration, then*

$$\|R_{j_{\max}}^{comp.}\| \leq \|R_{j_{\max}}^{exact}\| + \varepsilon.$$

Proof For simplicity, let $r = 1$, $M = I_n$ and $\{\alpha_j\}_{j=1}^{j_{\max}} \subset \mathbb{R}_-$. The assumption $\|C_j\| < 1$ gives $\xi_j := \gamma_j^2 \sigma_j = \|C_j - I\| < 2$ for $1 \leq j \leq j_{\max}$. Consider the identity $w_j = C_j w_{j-1} + (C_j - I)s_j$ established in the proof of Lemma 3.1. Then

$$\begin{aligned}\|R_j^{\mathrm{comp.}}\| = \|w_j\|^2 &= \|C_j w_{j-1}\|^2 + \|(C_j - I)s_j\|^2 + 2w_{j-1}^* C_j^*(C_j - I)s_j \\ &\leq \|C_j w_{j-1}\|^2 + \xi_j^2\|s_j\|^2 + 2\omega_{j-1}\xi_j\|s_j\|,\end{aligned}$$

where we have used $\omega_{j-1} = \|w_{j-1}\|$. Inserting (3.12a) yields

$$\begin{aligned}\xi_j^2\|s_j\|^2 + 2\omega_{j-1}\xi_j\|s_j\| \leq &\frac{\xi_j^2}{4}\left(\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}} - \omega_{j-1}\right)^2 \\ &+ \omega_{j-1}\xi_j\left(\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}} - \omega_{j-1}\right) \\ = &\frac{\xi_j^2}{4}\left(\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}} - 2\omega_{j-1}\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}} + \omega_{j-1}^2\right) \\ &+ \omega_{j-1}\xi_j\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}} - \omega_{j-1}^2\xi_j.\end{aligned}$$

Now, using $\xi_j < 2$ we obtain

$$\begin{aligned}\xi_j^2\|s_j\|^2 + 2\omega_{j-1}\xi_j\|s_j\| \leq &\frac{\xi_j}{2}\left(2\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}} - 2\omega_{j-1}\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}}\right) \\ &+ \omega_{j-1}\xi_j\sqrt{\omega_{j-1}^2 + \tfrac{2\varepsilon}{\xi_j j_{\max}}} - \omega_{j-1}^2\xi_j \\ = &\tfrac{\varepsilon}{j_{\max}}.\end{aligned}$$

resulting in $\|R_j^{\text{comp.}}\| \leq \|C_j w_{j-1}\|^2 + \frac{\varepsilon}{j_{\max}}$. Using Lemma 3.1 again for w_{j-1} and repeating the above steps for $1 \leq j \leq j_{\max}$ gives

$$\|R_{j_{\max}}^{\text{comp.}}\| = \|w_{j_{\max}}\|^2 = \left\| \prod_{j=1}^{j_{\max}} C_j w_0 \right\|^2 + \sum_{j=1}^{j_{\max}} \frac{\varepsilon}{j_{\max}} = \|R_{j_{\max}}^{\text{exact}}\| + \varepsilon.$$

□

Using Theorem 3.4 together with (3.11) yields the following conclusion.

Corollary 3.1 *Under the same conditions as Theorem 3.4 we have* $\|R_{j_{\max}}^{true}\| \leq \|R_{j_{\max}}^{exact}\| + 2\varepsilon$.

Hence, if (3.12a) is used, then the (true) Lyapunov residual norms in inexact LR-ADI are a small perturbation of the residuals of the exact method.

Remark 3.2 Another way to enforce decreasing computed Lyapunov residuals norms $\|R_j^{\text{comp.}}\|$ is using the stronger condition $\|C_j\| < 1$ together with a proportional inner accuracy $\|s_j\| < \mu \|R_{j-1}^{\text{comp.}}\|$, $0 < \mu < 1$, similar to [41] in the context of inexact stationary iterations. For the LR-ADI iteration this does, however, not ensure small residual gaps, $\|\varDelta R_j^{\text{comp.}}\| < \varepsilon$, and, thus, no small true residuals.

Practical relaxation strategies for inexact LR-ADI The proposed stopping criteria (3.12) are not very practical, since the employed bound (3.15) will often give substantial overestimation of $\|M v_k\|$ by several orders of magnitude which, in turn, will result in smaller inner tolerances τ^{LS} than actually needed. Furthermore, computing or estimating the norm of the large matrix $M(A + \alpha_k M)^{-1}$ is possible, e.g. by inexact Lanczos-type approaches, but the additional computational effort for this does not pay off because it can be easily more expensive compared to the remaining parts of a single iteration step. Here we propose some variations of the above approaches that are better applicable in an actual implementation. From the algorithmic description (Algorithm 2) of the LR-ADI iteration it holds

$$\|M v_k\| = \frac{1}{\gamma_k^2} \|w_k - w_{k-1}\| \leq \frac{1}{\gamma_k^2} (\|w_k\| + \|w_{k-1}\|). \tag{3.17}$$

In practice, the sequence $\{\|w_k\|\} = \{\sqrt{\|R_k^{\text{comp.}}\|}\}$ is often monotonically decreasing. Assuming $\|w_k\| \leq \|w_{k-1}\|$ suggests to use $\|M v_k\| \leq \frac{2\|w_{k-1}\|}{\gamma_k^2}$ in (3.14) leading to the relaxation criterion

$$\|s_k\| \leq \tau_k^{\text{LS}} = \frac{\varepsilon}{4 j_{\max} \sqrt{\|R_{k-1}^{\text{comp.}}\|}}. \tag{3.18a}$$

Starting from (3.13), assuming $\|\eta_{k-1}^{\mathrm{ADI}}\| \leq \frac{(k-1)\varepsilon}{2j_{\max}}$, and enforcing $\|\Delta R_k^{\mathrm{ADI}}\| < \frac{k\varepsilon}{j_{\max}}$ we obtain

$$\|s_k\| \leq \tau_k^{\mathrm{LS}} = \frac{\frac{k\varepsilon}{j_{\max}} - 2\|\eta_{k-1}^{\mathrm{ADI}}\|}{4\sqrt{\|R_{k-1}^{\mathrm{comp.}}\|}}. \tag{3.18b}$$

The second relaxation strategy (3.18b) requires $\|\eta_{k-1}^{\mathrm{ADI}}\| = \|S_{k-1}\Gamma_{k-1}Z_{k-1}^{*}M^{*}\|$ or an approximation thereof. From $\|\eta_{k-1}^{\mathrm{ADI}}\| \leq \|\eta_{k-2}^{\mathrm{ADI}}\| + \gamma_{k-1}^{2}\|Mv_{k-1}\|\|s_{k-1}\|$ basic bounds $\|\eta_i^{\mathrm{ADI}}\| \leq u_i$, $i = 0, \ldots, k-1$ can be computed in each step via

$$\|\eta_{k-1}^{\mathrm{ADI}}\| \leq u_{k-1} := u_{k-2} + \gamma_{k-1}^{2}\|Mv_{k-1}\|\|s_{k-1}\|, \quad u_0 := 0. \tag{3.19}$$

Note that Mv_{k-1} is required anyway to continue Algorithm 2. The linear residual norms $\|s_i\|$, $1 \leq i \leq k-1$ are sometimes available as byproducts of Krylov subspace solvers for linear systems if right preconditioning is used. In case of other forms of preconditioning, the s_i might need to be computed explicitly, which requires extra matrix vector products with A (and M), or the norms $\|s_i\|$ have to be estimated in some other way. Similar to RKSM, for problems defined by real but unsymmetric coefficients, pairs of complex conjugated shifts can occur in LR-ADI. These can be dealt with efficiently using the machinery developed in [10,11,30] to ensure that the majority of operations remains in real arithmetic. By following these results it is easily shown that if steps $k-2$, $k-1$ used a complex conjugated pairs of shifts, then in the formula (3.19) the real and imaginary parts of both v_{k-2}, s_{k-2} enter the update.

At the first look, (3.18) appears to allow somewhat less relaxation compared to RKSM since only the square roots of the computed Lyapunov residual norms appear in the denominator. However, the numerical examples in the next section show that with these relaxation strategies we can reduce the amount of work for inexact LR-ADI by up to 50%.

4 Numerical examples

In this section we consider several numerical examples and apply inexact RKSM and inexact LR-ADI with our practical relaxation strategies. The experiments were carried out in MATLAB ® 2016a on 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 true Lyapunov residual norm satisfies

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

i.e., $\varepsilon = \hat{\varepsilon}\|B\|^2$. In all tests we desire to achieve this accuracy with $\hat{\varepsilon} = 10^{-8}$ within at most $j_{\max} = 50$ iteration steps. In all but one case we employ dynamic shift generation strategies using the approach in [21] for RKSM and shifts based on a projection idea for LR-ADI, see [12,30] for details. For the latter one, the projection basis is chosen as the last $\min(\mathrm{coldim}(Z_r), 10r)$ columns of the low-rank factor Z_j. For one case, the symmetric problem, optimal shifts following [54] for the LR-ADI iteration are used. For comparative purposes, the exact (direct) EKSM is also tested for some examples.

We employ different Krylov subspace solvers and stopping criteria for the arising linear systems of equations, in particular, we use fixed as well as dynamically chosen inner solver tolerances τ^{LS} with the proposed relaxation strategies. The fixed solve tolerances were determined a-priori via trial-and-error such that the residual behavior of inexact method mimicked the residuals of the method using sparse-direct solution approaches for the linear systems. If not stated otherwise, the values $\tau^{\mathrm{LS}}_{\min} = 10^{-12}$, $\tau^{\mathrm{LS}}_{\max} = 0.1$ are taken as minimal and maximal linear solve tolerances. As Krylov subspace solvers we use BiCGstab and MINRES for problems with unsymmetric and symmetric ($A = A^*$, $M = M^*$) coefficients, respectively. Sparse-direct solves were carried out by the MATLAB *backslash*-routine, or, in case of EKSM, by precomputed sparse LU (or Cholesky) factorizations of A. EKSM and RKSM handled the generalized problems by means of the equivalent problem using sparse Cholesky factors of the matrix M as described in Sect. 2.5. We consider five examples, two standard Lyapunov equations (`cd2d` and `heat3d`, where `heat3d` is an example for which $r > 1$), two generalized Lyapunov equation (`fem3d`, `msd`) and an algebraic Riccati equation (`fem3d-are`), in order to illustrate the theoretical results in this paper. Details and setup on the examples we used are summarized in Table 1. The matrices B for examples `cd2d` and `heat3d` were generated randomly with a standard Gaussian distribution and the initialization `randn('state', 0)`. Examples `fem3d`, `fem3d-are` provide vectors B, C [8]. Because of the symmetric coefficient in `heat3d`, we use this example to experiment with precomputed shifts from [54] instead of the dynamic shifts in all other experiments. The example `msd` represents coefficients of a first order linearization of a second order dynamical system and is a modified version of an example used in [52]. Further details on the setup of this example can be found in “Appendix B”. For efficient iterative solves, this example required updating the preconditioner in every iteration step.

At first we briefly investigate the results in Sect. 2.3 on the decay of Galerkin solution Y_j using example `cd2d`. We run the exact RKSM and plot the row norms of Y_j, $H_j^{-1}Y_j$ and the corresponding bounds obtained in Corollaries 2.1 and 2.2 against the iteration number j, as well as the absolute values of the entries of the final Galerkin solution Y_j in the left plot of Fig. 1. Real shift parameters were used for this experiment. The figures are an example to show that our theoretical bounds can indeed be verified, but that they significantly overestimate the true norms. The right plot shows the decay of the entries Y_j as predicted by Corollary 2.1. Similar results are obtained for the other examples.

We now experiment with the different practical relaxation strategies (2.24) and (3.18) from Sects. 2 and 3 for the inner iteration. In Table 2 we report the results for all examples. There, we give, among other relevant information on the performance of the outer method under inexact inner solves, also the final obtained scaled computed residual norms $\mathfrak{R}_j^{\mathrm{comp}}$ (using the formula (2.9) for RKSM and (3.5) for LR-ADI). For assessing the reliability of the value of $\mathfrak{R}_j^{\mathrm{comp}}$, the distance $\delta\mathfrak{R}_j := |\mathfrak{R}_j^{\mathrm{comp}} - \mathfrak{R}_j^{\mathrm{true}}|$

Table 1 Properties and setting of used test equations including save guard constant δ in (2.24) and type of preconditioner used

Example	n	r	Matrices	Description	δ	Prec.
`cd2d`	40,000	1	$A \neq A^*$, $M = I$	Finite difference discretization of $\Delta u - 100x\frac{\partial u}{\partial x} - 200y\frac{\partial u}{\partial y}$ on $[0, 1]^2$	0.01	iLU(A,10^{-3})
`heat3d`	125,000	4	$A = A^*$, $M = I$	Finite difference discretization of heat equation on $[0, 1]^3$	1	iC(-A,10^{-2})
`fem3d`	24,389	1	$A \neq A^*$, $M = M^* \neq I$	Finite element model of 3d convection-diffusion problem from [8]	$\frac{1}{2}$	iLU(A,10^{-2})
`msd`	24,002	4	$A \neq A^*$, $M = M^* \neq I$	Modified version of second order dynamical system from [52]	0.1	iLU(A_j,10^{-2})
`fem3d-are`	24,389	1	$A \neq A^*$, $M = M^* \neq I$	Extension of `fem3d` to ARE (2.27) from [8] including $C \in \mathbb{R}^n$	$\frac{1}{2}$	iLU(A,10^{-2})

Here, iLU(X, ν) and iC(X, ν) refer to incomplete LU and, respectively, Cholesky factorization of the matrix X with drop tolerance ν

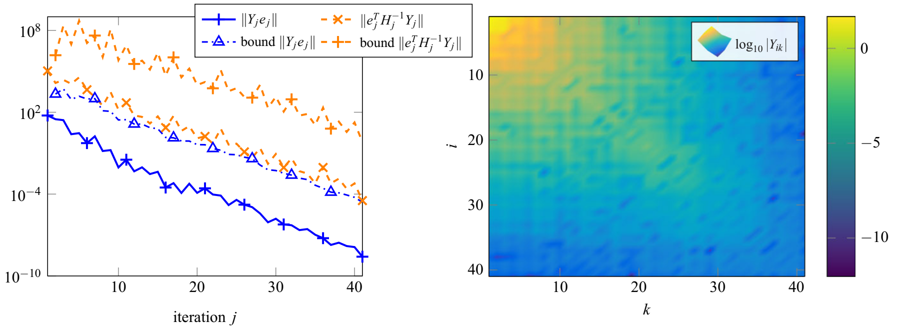

Fig. 1 Illustration of the decay of the Galerkin solution for Example `cd2d`. Left: Row norms of Y_j, $H_j^{-1}Y_j$ and the corresponding bounds (Corollary 2.1, Corollary 2.2) against the iteration number j. Right: Absolute values of the entries of the final Y_j

to the true scaled residual norms is also listed, where $\mathfrak{R}_j^{\text{true}}$ was computed using the Lanczos process on $\mathscr{R}_j$.

First, we observe that in all examples the difference between the true and the computed Lyapunov residual norm, $\delta\mathfrak{R}_j$, is of the order $\mathscr{O}(10^{-9})$, or smaller.

The second observation we make is that both practical relaxation criteria for RKSM, namely (2.24a) and (2.24b) are effective and lead to a reduction in overall inner iteration numbers. They both lead to nearly the same results for the number of inner iterations, the gain in using (2.24b) over (2.24a) is very minor. For our examples, by using the relaxed stopping criterion up to 50 per cent of inner iterations can by saved compared to a fixed inner tolerance.

For the LR-ADI method we consider the two relaxation criteria (3.18a) and (3.18b). Again, we observe a reduction in the total number of iterations for both relaxation strategies, but we also see that the second relaxation criterion (3.18b) reduces the iteration numbers even further compared to (3.18a), so the use of (3.18b) is generally recommended. Here it pays off that the second strategy takes the previous iterate into account. The total savings in inner iterations between fixed and relaxed tolerance solves for LR-ADI is between 14 and 42% in our examples.

The fairer comparison between the fixed and relaxed tolerance solve is the computation time; in Table 2 we report the computation time for direct solves as well as iterative solves with fixed and relaxed solve tolerances. For all examples we see that the relaxed solve tolerance also leads to (sometimes significant) savings in computation time. For Example `fem3d`, the generalized Lyapunov equation, the absolute time saving is not so significant (both for RKSM and LR-ADI), however, for the other three examples the savings are quite large, in particular for inexact RKSM. The results for the Riccati example `fem3d-are` indicate that the proposed relaxation criteria also work for inexact RKSM for Riccati equations. Note that for Example `cd2d`, the direct solver outperforms the iterative methods with fixed small solve tolerance (both for RKSM and LR-ADI), however, the relaxed tolerance versions perform similar to the direct methods in terms of computation time. This is to be expected as example `cd2d` represents a two-dimensional problem, where sparse-direct solvers are usually very

Table 2 Experimental results

Ex.	Outer	Inner	Stop	min τ^{LS}	max τ^{LS}	$\mathrm{it}^{\mathrm{out}}$	Dim	$\mathfrak{R}_j^{\mathrm{comp}}$	$\delta\mathfrak{R}_j$	$\mathrm{it}^{\mathrm{in}}$	Save	Time
cd2d	RKSM	Direct	–	–	–	37	37	2.7e−09		–	–	5.5
	RKSM	BICGSTAB	Fixed	1.00e−10		37	37	3.2e−09	2.0e−10	832		5.3
	RKSM	BICGSTAB	Relax (2.24a)	2.0e−12	2.8e−02	37	37	4.2e−09	2.2e−09	618	25.7%	3.9
	RKSM	BICGSTAB	Relax (2.24b)	2.0e−12	9.1e−03	37	37	3.0e−09	7.6e−09	614	26.2%	3.7
	EKSM	Direct	–	–	–	40	80	3.6e−09		–	–	1.5
	LR-ADI	Direct	–	–	–	49	49	7.0e−09		–	–	6.1
	LR-ADI	BICGSTAB	Fixed	1.0e−10		49	49	7.0e−09	1.4e−12	1382		6.8
	LR-ADI	BICGSTAB	Relax (3.18a)	5.0e−11	4.9e−03	49	49	7.0e−09	2.4e−11	1004	27.4%	4.7
	LR-ADI	BICGSTAB	Relax (3.18b)	5.0e−11	2.1e−02	49	49	6.9e−09	1.7e−10	971	29.7%	4.5
heat3d	RKSM	Direct	–	–	–	15	60	7.9e−09		–	–	69.0
	RKSM	MINRES	Fixed	1.00e−09		15	60	7.9e−09	5.3e−14	492		20.1
	RKSM	MINRES	Relax (2.24a)	2.0e−10	1.0e−01	15	60	7.9e−09	2.6e−10	265	46.1%	13.2
	RKSM	MINRES	Relax (2.24b)	2.0e−10	1.0e−01	15	60	7.9e−09	2.9e−10	252	48.8%	12.2
	EKSM	Direct	–	–	–	20	160	3.9e−11		–	–	23.9
	LR-ADI	Direct	–	–	–	16	64	4.4e−09		–	–	77.4
	LR-ADI	MINRES	Fixed	1.0e−09		16	64	4.4e−09	9.0e−14	439		17.2
	LR-ADI	MINRES	Relax (3.18a)	5.0e−11	1.1e−03	16	64	4.4e−09	8.7e−14	301	31.4%	13.5
	LR-ADI	MINRES	Relax (3.18b)	5.0e−11	7.7e−03	16	64	4.4e−09	1.3e−11	270	38.5%	12.2
fem3d	RKSM	Direct	–	–	–	23	23	3.5e−09		–	–	44.8
	RKSM	BICGSTAB	Fixed	2.00e−10		23	23	4.0e−09	1.1e−10	474		12.4
	RKSM	BICGSTAB	Relax (2.24a)	3.7e−11	4.5e−02	23	23	6.2e−09	1.2e−09	309	34.8%	9.8
	RKSM	BICGSTAB	Relax (2.24b)	3.9e−11	1.0e−01	23	23	4.7e−09	1.4e−10	299	36.9%	9.9

Table 2 continued

Ex.	Outer	Inner	Stop	min τ^{LS}	max τ^{LS}	$\mathrm{it}^{\mathrm{out}}$	Dim	$\mathfrak{R}_j^{\mathrm{comp}}$	$\delta\mathfrak{R}_j$	$\mathrm{it}^{\mathrm{in}}$	Save	Time
	EKSM	Direct	–	–	–	25	50	1.9e−09		–	–	5.5
	LR-ADI	Direct	–	–	–	27	27	4.2e−09		–	–	52.2
	LR-ADI	BICGSTAB	Fixed	2.0e−10		27	27	4.2e−09	1.8e−14	359		2.2
	LR-ADI	BICGSTAB	Relax (3.18a)	5.0e−11	2.7e−04	27	27	4.2e−09	1.8e−11	246	31.5%	1.6
	LR-ADI	BICGSTAB	Relax (3.18b)	5.0e−11	2.8e−03	27	27	4.4e−09	2.5e−10	209	41.8%	1.5
`msd`	RKSM	Direct	–	–	–	86	344	9.5e−09		–	–	120.9
	RKSM	BICGSTAB	Fixed	6.67e−11		86	344	9.6e−09	2.9e−08	738		53.6
	RKSM	BICGSTAB	Relax (2.24a)	2.7e−12	1.4e−02	86	344	8.7e−09	5.8e−12	668	9.5%	51.0
	RKSM	BICGSTAB	Relax (2.24b)	1.0e−12	3.3e−03	88	352	7.9e−09	2.3e−12	696	5.7%	53.0
	EKSM	Direct	–	–	–	70	560	1.5e−09		–	–	17.5
	LR-ADI	Direct	–	–	–	66	264	1.0e−08		–	–	164.4
	LR-ADI	BICGSTAB	Fixed	6.7e−11		59	236	3.0e−09	1.9e−11	1683		69.3
	LR-ADI	BICGSTAB	Relax (3.18a)	2.5e−11	1.5e−03	62	248	9.9e−09	1.1e−13	1451	13.8%	55.4
	LR-ADI	BICGSTAB	Relax (3.18b)	2.5e−11	1.5e−02	65	260	4.7e−10	1.7e−10	1065	36.7%	43.0
`fem3d-are`	RKSM	Direct	–	–	–	22	22	5.8e−09		–	–	43.6
	RKSM	BICGSTAB	Fixed	2.00e−10		22	22	5.8e−09	4.8e−14	436		34.8
	RKSM	BICGSTAB	Relax (2.24a)	1.1e−10	1.0e−01	22	22	6.7e−09	2.3e−09	222	49.1%	12.9
	RKSM	BICGSTAB	Relax (2.24b)	1.1e−10	1.0e−01	22	22	6.7e−09	2.3e−09	222	49.1%	12.9

The columns denote the outer (i.e., EKSM, RKSM or LR-ADI) and inner method (which Krylov subspace method), the used inner stopping criterion (fixed or relaxed), the smallest and largest generated inner solve tolerances min τ^{LS}, max τ^{LS}, the number of required outer iterations, column dimension of the low-rank solution factor, the final obtain scaled computed residual norm $\mathfrak{R}_j^{\mathrm{comp}}$, the difference $\delta\mathfrak{R}_j := |\mathfrak{R}_j^{\mathrm{comp}} - \mathfrak{R}_j^{\mathrm{true}}|$ between the true and the computed Lyapunov residual norm, the total number of inner iteration steps, the achieved relative savings regarding the amount of inner iteration steps compared to the run with a fixed inner tolerance, and the computing times in seconds

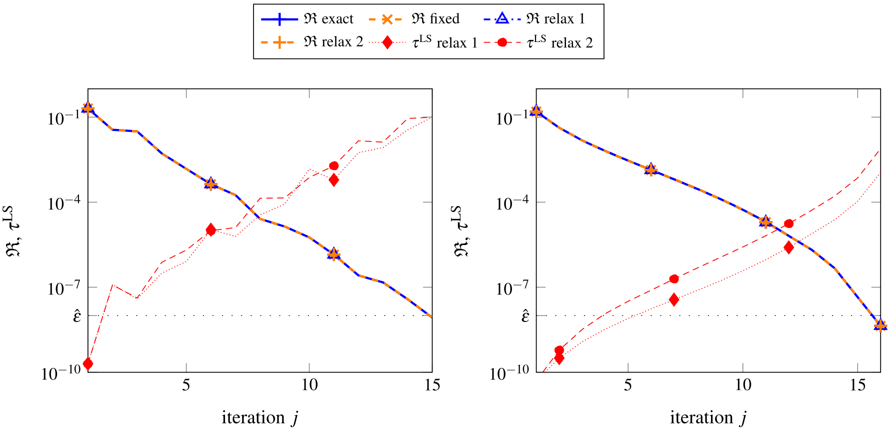

Fig. 2 Results for Example `heat3d`: Scaled computed residual norms $\mathfrak{R}^{\mathrm{comp}}$ and inner tolerances τ^{LS} vs iteration numbers obtained by (in)exact RKSM (left) and LR-ADI (right). Here, LR-ADI uses precomputed shifts

efficient. For the three-dimensional examples `heat3d`, `fem3d` and `fem3d-are` the iterative solvers significantly outperform the direct method.

Regarding the comparison with EKSM, we observe that EKSM achieves sometimes the smallest computations times (e.g., examples `cd2d`, `msd`) but always generates the largest subspace dimensions among all tested methods. Using a block-MINRES for the `heat3d` example (with $r = 4$) largely led to very similar results in terms of the numbers of inner iteration steps, but the overall computing times were slightly larger.

We point out that, in every experiment, RKSM, LR-ADI generated new shift parameters on their own. Interestingly and perhaps surprisingly, this did not lead to substantial differences regarding the Lyapunov residual behaviour, although the employed adaptive shift generation techniques [12,21] are based on the eigenvalues of T_j (rational Ritz values of A) and, hence, depend on the generated subspace. One explanation might be that the eigenvalue generation itself can be seen as inexact rational Arnoldi process for the eigenvalue problem and if the inner solve tolerances are chosen intelligently, no large differences regarding the Ritz values should appear [23,31,42]. Some small differences in the required number of outer iteration steps are observed for example `msd` were, for instance, the inexact LR-ADI required less outer steps than the exact LR-ADI. One possible explanation could be that for this example slightly different shifts were generated in exact and inexact methods.

For the `heat3d` example precomputed shifts [54] were used within LR-ADI leading to almost indistinguishable residual curves (cf. right plot in Fig. 2) indicating that the optimal shift approaches for certain problem classes still work under inexact solves. Similar observations were made for the remaining examples when precomputed shifts were used for both exact and inexact methods.

Figure 2 shows the scaled computed residual norms $\mathfrak{R}^{\mathrm{comp}}$ and inner tolerances τ^{LS} vs iteration numbers obtained by (in)exact RKSM and LR-ADI for Example `heat3d`.

On the left we plot the convergence history for the exact RKSM (e.g. direct solves within the inner iteration), the inexact RKSM with fixed small solve tolerance within the iterative solve and the inexact RKSM with relaxation criterion (2.24a) and (2.24b) within the iterative solution of the inner linear system. All computed residual norms are decreasing and virtually indistinguishable. The solve tolerances when using relaxation criterion (2.24a) are shown in dotted lines with diamonds and the ones using criterion (2.24b) are shown in dashed lines with red circles. The relaxation criteria lead to increasing inner solve tolerances, but as already observed in Table 2, both criteria for inexact RKSM give nearly the same results.

The right plot in Fig. 2 shows the same results for LR-ADI. Again,the convergence history of the residual norms for inexact LR-ADI using the two relaxation strategies is not visibly distinguishable from the exact LR-ADI. However, we observe that the second relaxation criterion (3.18b), shown in dashed lines with red circles, gives better results, e.g. more relaxation and hence fewer inner iterations, than the first relaxation criterion (3.18a), shown in dotted lines with diamonds, a result also observed in Table 2. Similar plots as in Fig. 2 can be obtained for other examples.

5 Conclusions and future work

The numerical solution of large scale Lyapunov equations is a very important problem in many applications. The rational Krylov subspace method (RKSM) and the low-rank alternating directions implicit (LR-ADI) iteration are well-established methods for computing low-rank solution factors of large-scale Lyapunov equations. The main task in both those methods is to solve a linear system at each step, which is usually carried out iteratively and hence inexactly.

We observed empirically that, when solving the linear system at each iteration step, the accuracy of the solve may be relaxed while maintaining the convergence to the solution of the Lyapunov equation. In this paper we have presented a theoretical foundation for explaining this phenomenon, both for the inexact RKSM method and the inexact low-rank ADI iteration. For both methods we introduced a so-called residual gap, which depends on the accuracy of the linear system solve and on quantities arising from the solution methods for the large scale Lyapunov equation. We analyzed this gap for each method which provided theoretical relaxation criteria for both inexact RKSM and inexact ADI. These criteria are often not applicable in practice as they contain unknown and/or overestimated quantities. Hence, we gave practical relaxation criteria for both methods. Our numerical results indicate that using flexible accuracies gives very good results and can reduce the amount of work for solving large scale Lyapunov equations by up to 50%.

One numerical experiment with inexact RKSM indicated that relaxation strategies might also be fruitful for low-rank methods for algebraic Riccati equations [6–8, 12,14,44,48], making this an obvious future research topic, together with inexact linear solves in low-rank methods for Sylvester equations [9,13,30]. In this work, we restricted ourselves to standard preconditioning techniques. Improved concepts such as *tuned* preconditioners and similar ideas [15,23] might further enhance the performance of the inner iteration process. Preliminary tests, however, did not yield any performance gain from these techniques worth mentioning, further investigations are necessary in this direction. A further research direction worth pursuing is to reduce the computational effort for solving the sequences of shifted linear systems by storing the Krylov basis obtained from solving one (e.g., the first) linear system and employing subspace recycling techniques as, e.g., discussed for LR-ADI in [32] and for rational Krylov methods in the context of model order reduction in [1]. Allowing inexact matrix vector products, and in case of generalized equations also inexact solves with M, represents a further, more challenging research perspective.

Acknowledgements The authors are grateful to Cost Action EU-MORNET (TD1307) and the Department of Mathematical Sciences at Bath, that provided funding for research visits of PK to the University of Bath, where substantial parts of this work have been conducted. This work was primarily generated while PK was affiliated with the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany and MAF was affiliated with the University of Bath, United Kingdom. Furthermore, the authors thank Kirk Soodhalter (Trinity College Dublin) for insightful discussion regarding block Krylov subspace methods.

Appendix A: Proof of Lemma 2.2

Proof From $\omega \underline{H}_j = 0$ we have for $k \leq j$

$$0 = \omega \left[\begin{array}{c|c} H_k & \\ e_k^* h_{k+1,k} & H_{:,k+1:j} \\ 0_{j-k,k} & \end{array} \right],$$

which, for $k = j$, immediately leads to $f_j^{(j)} = -\omega_{1:j}/(h_{j+1,j}\omega_{j+1})$, the first equality in (2.21a). Similarly, it is easy to show that for $k < j$, a left null space vector $\hat{\omega} \in \mathbb{C}^{1\times k+1}$ of $\underline{H}_k$ is given by the first $k+1$ entries of the null vector ω of $\underline{H}_j$. Hence, $f_k^{(k)} = -\omega_{1:k}/(h_{k+1,k}\omega_{k+1})$ holds for all $k \leq j$.

For computing $f_j^{(k)} = e_k^* H_j^{-1}$, $k < j$, we use the following partition of H_j and consider the splitting $f_j^{(k)} = [u, y]$, $u \in \mathbb{C}^{1\times k}$, $y \in \mathbb{C}^{1\times k-j}$:

$$e_k^* = \left[0_{1,k-1} \middle| 1 \middle| \, 0_{1,j-k} \right] = f_j^{(k)} H_j = [u, y] \left[\begin{array}{c|c} H_{k-1} & \\ e_{k-1}^* h_{k,k-1} & H_{:,k:j} \\ 0_{j-k,k-1} & \end{array} \right]$$
$$= \left[u \begin{bmatrix} H_{k-1} \\ e_{k-1}^* h_{k,k-1} \end{bmatrix} \middle| [u, y] H_{:,k:j} \right].$$

This structure enforces conditions on $[u, y]$, which we now explore. First, u has to be a multiple of $\omega_{1:k}$. Here we exploited that due to the Hessenberg structure, $\omega_{1:k}\underline{H}_{k-1} = 0$. In particular,

$$u_{1:k-1} H_{k-1} + u_k e_{k-1}^* h_{k,k-1} = 0,$$

such that $u_k h_{k,k-1} e_{k-1}^* H_{k-1}^{-1} = -u_{1:k-1}$. Since $f_{k-1}^{(k-1)} = e_{k-1}^* H_{k-1}^{-1}$ we can infer $u_{1:k-1} = -u_k h_{k,k-1} f_{k-1}^{(k-1)}$ and, consequently, (2.21c) for $k > 1$. Similarly, $[\omega_{1:k}, y]$

has to satisfy

$$\begin{aligned}0_{1,j-k} &= [\omega_{1:k}, y]H_{:,k+1:j} = \omega_{1:k}H_{1:k,k+1:j} + yH_{k+1:j,k+1:j} = \omega H_{1:j+1,k+1:j}\\ &= \omega_{1:k}H_{1:k,k+1:j} + \omega_{k+1:j+1}H_{k+1:j+1,k+1:j},\end{aligned}$$

leading to

$$y = \omega_{k+1:j+1}H_{k+1:j+1,k+1:j}H^{-1}_{k+1:j,k+1:j} = \omega_{k+1:j} + \omega_{j+1}h_{j+1,j}e^*_{j-k}H^{-1}_{k+1:j,k+1:j},$$

where e_{j-k} is a canonical vector of length $j - k$. Hence,

$$v_j^{(k)} = [\omega_{1:k}, y] = \omega_{1:j} + [0_{1,k}, [0_{1,j-k-1}, h_{j+1,j}\omega_{j+1}]H^{-1}_{k+1:j,k+1:j}],$$

leading to (2.21b). Finally, the normalization constant $\phi_j^{(k)}$ is obtained by the requirement $[u, y]H_{1:j,k} = 1$. □

Appendix B: Setup of example msd

The example *msd* represents a variant of one realization of the series of examples in [52]. Set $K := \operatorname{diag}(I_3 \otimes K_1, k_0 + k_1 + k_2 + k_3) + k_+e_{3n}^T - e_{3n}k_+ \in \mathbb{R}^{n_2 \times n_2}$, $K_1 := \operatorname{tridiag}(-1, 2, -1) \in \mathbb{R}^{n_1 \times n_1}$ with $k_0 = 0.1$, $k_1 = 1$, $k_2 = 2$, $k_3 = 4$, $n_2 = 3n_1 + 1$, and $k_+ := [(k_1, k_2, k_3) \otimes e_{n_1}^T, 0]^T$. Further, $M_1 := \operatorname{diag}\big(m_1 I_{n_1}, m_2 I_{n_1}, m_3 I_{n_1} m_0\big) \in \mathbb{R}^{n_2 \times n_2}$ with $m_0 = 1000$, $m_1 = 10$, $m_2 = k_2$, $m_3 = k_3$ and $D := \alpha M + \beta(K + K(M^{-1}K) + K(M^{-1}K)^2 + K(M^{-1}K)^3) + \nu[e_1, e_{n_1}, e_{2n_1+1}][e_1, e_{n_1}, e_{2n_1+1}]^T$, $\alpha = 0.8$, $\beta = 0.1$, $\nu = 16$. In [52] a different matrix D was used involving a term $M^{\frac{1}{2}}\sqrt{M^{-\frac{1}{2}}KM^{-\frac{1}{2}}}M^{\frac{1}{2}}$ which was infeasible to set up in a large scale setting (the middle matrix square is a dense matrix). The version of D used here was similarly used in [10]. Construct the $n \times n$, $n := 2n_2$ block matrices $\hat{A} := \begin{bmatrix} 0 & I_{n_2} \\ -K & -D \end{bmatrix}$, $\hat{M} := \operatorname{diag}\big(I_{n_2}, M_1\big)$ representing a linearization of the quadratic matrix pencil $\lambda^2 M_1 + \lambda D + K$. The right hand side factor is set up as $\hat{B} = \left[I_{2m}\ 0_{2m,n_2-m} \begin{bmatrix} 0_{m,n_2-m} & I_m \\ I_m & 0_{m,n_2-m} \end{bmatrix} \right]^T \in \mathbb{R}^{n \times 2m}$. In Sect. 4 we use $n_1 = 4000$, $m = 2$ leading to $n = 24{,}002$, $r = 4$. Finally, $A := P^T \hat{A} P$, $M := P^T \hat{M} P$, $B := P^T \hat{B}$ as in [29, Section 5.6], where P is a perfect shuffle permutation: This leads to banded matrices A, M, except for some sole off-band entries from the low-rank update of D, resulting in noticable computational savings when applying sparse direct solvers or computing sparse (incomplete) factorizations. An alternative to exploit the structure in $\hat{A}$, $\hat{M}$ within RKSM, LR-ADI is described in, e.g., [10,30].

References

1. Ahmad, M.I., Szyld, D.B., van Gijzen, M.B.: Preconditioned multishift BiCG for H_2-optimal model reduction. SIAM J. Matrix Anal. Appl. **38**(2), 401–424 (2017)
2. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. Advances in Design and Control, vol. 6. SIAM, Philadelphia, PA (2005)
3. Baker, J., Embree, M., Sabino, J.: Fast singular value decay for Lyapunov solutions with nonnormal coefficients. SIAM J. Matrix Anal. Appl. **36**(2), 656–668 (2015)
4. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation $AX+XB=C$: algorithm 432. Commun. ACM **15**, 820–826 (1972)
5. Beckermann, B., Townsend, A.: On the singular values of matrices with displacement structure. SIAM J. Matrix Anal. Appl. **38**(4), 1227–1248 (2017)
6. Benner, P., Bujanović, Z., Kürschner, P., Saak, J.: A numerical comparison of solvers for large-scale, continuous-time algebraic Riccati equations and LQR problems. SIAM J. Sci. Comput. **42**(2), A957–A996 (2020)
7. Benner, P., Bujanović, Z., Kürschner, P., Saak, J.: RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations. Numer. Math. **138**(2), 301–330 (2018)
8. Benner, P., Heinkenschloss, M., Saak, J., Weichelt, H.K.: An inexact low-rank Newton-ADI method for large-scale algebraic Riccati equations. Appl. Numer. Math. **108**, 125–142 (2016)
9. Benner, P., Kürschner, P.: Computing real low-rank solutions of Sylvester equations by the factored ADI method. Comput. Math. Appl. **67**(9), 1656–1672 (2014)
10. Benner, P., Kürschner, P., Saak, J.: An improved numerical method for balanced truncation for symmetric second order systems. Math. Comput. Model. Dyn. Sys. **19**(6), 593–615 (2013)
11. Benner, P., Kürschner, P., Saak, J.: Efficient Handling of complex shift parameters in the low-rank Cholesky factor ADI method. Numer. Algorithms **62**(2), 225–251 (2013)
12. Benner, P., Kürschner, P., Saak, J.: Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations. Electron. Trans. Numer. Anal. **43**, 142–162 (2014)
13. Benner, P., Li, R.-C., Truhar, N.: On the ADI method for Sylvester equations. J. Comput. Appl. Math. **233**(4), 1035–1045 (2009)
14. Benner, P., Saak, J.: 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 (2013)
15. Bergamaschi, L., Martínez, Á.: Two-stage spectral preconditioners for iterative eigensolvers. Numer. Linear Algebra Appl. **24**(3), e2084 (2017)
16. Berljafa, M., Güttel, S.: Generalized rational Krylov decompositions with an application to rational approximation. SIAM J. Matrix Anal. Appl. **36**(2), 894–916 (2015)
17. Bouras, A., Frayssé, V.: Inexact matrix–vector products in Krylov methods for solving linear systems: a relaxation strategy. SIAM J. Matrix Anal. Appl. **26**(3), 660–678 (2005)
18. Canuto, C., Simoncini, V., Verani, M.: On the decay of the inverse of matrices that are sum of Kronecker products. Linear Algebra Appl. **452**, 21–39 (2014)
19. Crouzeix, M., Palencia, C.: The numerical range is a $(1+\sqrt{2})$-spectral set. SIAM J. Matrix Anal. Appl. **38**(2), 649–655 (2017)
20. Druskin, V., Knizhnerman, L.A., Simoncini, V.: Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. SIAM J. Numer. Anal. **49**, 1875–1898 (2011)
21. Druskin, V., Simoncini, V.: Adaptive rational Krylov subspaces for large-scale dynamical systems. Syst. Control Lett. **60**(8), 546–560 (2011)
22. Freitag, M.A., Spence, A.: Convergence theory for inexact inverse iteration applied to the generalised nonsymmetric eigenproblem. Electron. Trans. Numer. Anal. **28**, 40–64 (2007/08)
23. Freitag, M.A., Spence, A.: Shift-invert Arnoldi's method with preconditioned iterative solves. SIAM J. Matrix Anal. Appl. **31**(3), 942–969 (2009)
24. Gaaf, S.W., Simoncini, V.: Approximating the leading singular triplets of a large matrix function. Appl. Numer. Math. **113**, 26–43 (2017)
25. Grasedyck, L.: Existence of a low rank or $\mathscr{H}$-matrix approximant to the solution of a Sylvester equation. Numer. Linear Algebra Appl. **11**(4), 371–389 (2004)
26. Güttel, S.: Rational Krylov methods for operator functions. Ph.D. thesis, Technische Universität Bergakademie Freiberg, Germany. http://eprints.ma.man.ac.uk/2586/ (2010)
27. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. GAMM-Mitteilungen **36**(1), 8–31 (2013)

28. Jaimoukha, I.M., Kasenally, E.M.: Krylov subspace methods for solving large Lyapunov equations. SIAM J. Numer. Anal. **31**, 227–251 (1994)
29. Kressner, D., Massei, S., Robol, L.: Low-rank updates and a divide-and-conquer method for linear matrix equations. SIAM J. Sci. Comput. **41**(2), A848–A876 (2019)
30. Kürschner, P.: Efficient low-rank solution of large-scale matrix equations. Dissertation, Otto-von-Guericke-Universität, Magdeburg, Germany. http://hdl.handle.net/11858/00-001M-0000-0029-CE18-2 (2016)
31. Lehoucq, R.B., Meerbergen, K.: Using generalized Cayley transformations within an inexact rational Krylov sequence method. SIAM J. Matrix Anal. Appl. **20**(1), 131–148 (1998)
32. Li, J.-R.:Model reduction of large linear systems via low rank system Gramians. Ph.D. thesis, Massachusettes Institute of Technology (2000)
33. Li, J.-R., White, J.: Low rank solution of Lyapunov equations. SIAM J. Matrix Anal. Appl. **24**(1), 260–280 (2002)
34. Opmeer, M., Reis, T., Wollner, W.: Finite-rank ADI iteration for operator Lyapunov equations. SIAM J. Control Optim. **51**(5), 4084–4117 (2013)
35. Palitta, D., Simoncini, V.: Numerical methods for large-scale Lyapunov equations with symmetric banded data. SIAM J. Sci. Comput. **40**(5), A3581–A3608 (2018)
36. Penzl, T.: A cyclic low rank Smith method for large sparse Lyapunov equations. SIAM J. Sci. Comput. **21**(4), 1401–1418 (2000)
37. Ruhe, Axel: The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: complex shifts for real matrices. BIT **34**, 165–176 (1994)
38. Saad, Y.: Numerical solution of large Lyapunov equation. In: Kaashoek, M.A., van Schuppen, J.H., Ran, A.C.M. (eds.) Signal Processing. Scattering, Operator Theory and Numerical Methods, pp. 503–511. Birkhäuser, Basel (1990)
39. Saak, J.: Efficient numerical solution of large scale algebraic matrix equations in PDE control and model order reduction. Ph.D. thesis, TU Chemnitz. http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642 (2009)
40. Sabino, J.: Solution of large-scale Lyapunov equations via the block modified Smith method. Ph.D. thesis, Rice University, Houston, TX. http://www.caam.rice.edu/tech_reports/2006/TR06-08.pdf (2007)
41. Shank, S.D., Simoncini, V., Szyld, D.B.: Efficient low-rank solution of generalized Lyapunov equations. Numer. Math. **134**, 327–342 (2015)
42. Simoncini, V.: Variable accuracy of matrix–vector products in projection methods for eigencomputation. SIAM J. Numer. Anal. **43**(3), 1155–1174 (2005)
43. Simoncini, V.: A new iterative method for solving large-scale Lyapunov matrix equations. SIAM J. Sci. Comput. **29**(3), 1268–1288 (2007)
44. Simoncini, V.: Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations. SIAM J. Matrix Anal. Appl. **37**(4), 1655–1674 (2016)
45. Simoncini, V.: Computational methods for linear matrix equations. SIAM Rev. **58**(3), 377–441 (2016)
46. Simoncini, V., Eldén, L.: Inexact Rayleigh quotient-type methods for eigenvalue computations. BIT Numer. Math. **42**(1), 159–182 (2002)
47. Simoncini, V., Szyld, D.B.: Theory of inexact Krylov subspace methods and applications to scientific computing. SIAM J. Sci. Comput. **25**(2), 454–477 (2003)
48. Simoncini, V., Szyld, D.B., Monsalve, M.: On two numerical methods for the solution of large-scale algebraic Riccati equations. IMA J. Numer. Anal. **34**(3), 904–920 (2014)
49. Soodhalter, K.: A block MINRES algorithm based on the banded Lanczos method. Numer. Algorithms **69**, 473–494 (2015)
50. Sun, K.: Model order reduction and domain decomposition for large-scale dynamical systems. Ph.D. thesis, Rice University, Houston. http://search.proquest.com/docview/304507831 (2008)
51. Truhar, N., Veselić, K.: Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix. Syst. Control Lett. **56**(7–8), 493–503 (2007)
52. Truhar, N., Veselić, K.: An efficient method for estimating the optimal dampers' viscosity for linear vibrating systems using Lyapunov equation. SIAM J. Matrix Anal. Appl. **31**(1), 18–39 (2009)
53. van den Eshof, J., Sleijpen, G.: Inexact Krylov subspace methods for linear systems. SIAM J. Matrix Anal. Appl. **26**(1), 125–153 (2004)
54. Wachspress, E.L.: The ADI Model Problem. Springer, New York (2013)

55. Wolf, T., Panzer, H.K.-F.: The ADI iteration for Lyapunov equations implicitly performs $\mathscr{H}_2$ pseudo-optimal model order reduction. Int. J. Control **89**(3), 481–493 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.