



# On an integrated Krylov-ADI solver for large-scale Lyapunov equations

Peter Benner<sup>1</sup> · Davide Palitta<sup>2</sup> · Jens Saak<sup>1</sup>

Received: 1 April 2022 / Accepted: 2 September 2022  
© The Author(s) 2022

## Abstract

One of the most computationally expensive steps of the low-rank ADI method for large-scale Lyapunov equations is the solution of a shifted linear system at each iteration. We propose the use of the extended Krylov subspace method for this task. In particular, we illustrate how a single approximation space can be constructed to solve all the shifted linear systems needed to achieve a prescribed accuracy in terms of Lyapunov residual norm. Moreover, we show how to fully merge the two iterative procedures in order to obtain a novel, efficient implementation of the low-rank ADI method, for an important class of equations. Many state-of-the-art algorithms for the shift computation can be easily incorporated into our new scheme, as well. Several numerical results illustrate the potential of our novel procedure when compared to an implementation of the low-rank ADI method based on sparse direct solvers for the shifted linear systems.

**Keywords** Lyapunov equations · Low-rank ADI · Extended Krylov method · Shifted linear systems

---

**Novelty statement:** Extended Krylov subspace methods (EKSM) and the low-rank alternating directions implicit (LR-ADI) iteration have been competing methods for the solution of large-scale algebraic Lyapunov equations. In this paper, we make an important step towards a new method merging them into a combined procedure that inherits advantages from both worlds.

---

✉ Davide Palitta  
davide.palitta@unibo.it

Peter Benner  
benner@mpi-magdeburg.mpg.de

Jens Saak  
saak@mpi-magdeburg.mpg.de

<sup>1</sup> Research Group Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany

<sup>2</sup> Centro AM2, Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato 5, 40127 Bologna, Italy

## 1 Introduction

The low-rank alternating direction implicit (LR-ADI) [42, 54] method is one of the state-of-the-art methods for the numerical solution of large-scale Lyapunov equations [19, 65]. This linear matrix equation can be encountered in many applications: control and system theory [34, 66], especially some model reduction techniques for dynamical systems [3, 15], but also discretization of certain partial differential equations (PDEs) [71], and many more.

We consider Lyapunov equations of the form

$$AXE^T + EXA^T + BB^T \tag{1}$$

where  $A, E \in \mathbb{R}^{n \times n}$ , and  $B \in \mathbb{R}^{n \times q}$ ,  $q \ll n$ . Moreover,  $E$  is supposed to be symmetric positive definite (SPD) and the matrix pencil  $(A, E)$  to be asymptotically stable, i.e., its spectrum is contained in the open left half plane  $\mathbb{C}_-$ , which guarantees that a unique solution  $X$  exists, and is symmetric positive semidefinite [53].

A special case of (1) is attained whenever  $E = I$ , namely the equation of interest is

$$AX + XA^T + BB^T = 0. \tag{2}$$

Oftentimes the coefficient matrix  $E$  possesses a structured sparsity pattern. For instance, it is (block) diagonal when the matrices stem from a finite element discretization that uses mass-lumping. In this case, we can easily transform (1) and obtain an equation of the form (2). This can, for example, be achieved by simply pre- and post-multiplying (1) by  $E^{-\frac{1}{2}}$  to potentially preserve symmetry of  $A$ . For the sake of simplicity, we thus focus on (2) in the following.

In case of very large problem dimensions, the solution  $X$  cannot be stored since this matrix is, in general, dense. However, it is well known that its singular values quickly decay to zero under suitable assumptions, see, e.g., [5, 13, 33, 55], so that accurate low-rank approximations  $ZZ^T \approx X$ ,  $Z \in \mathbb{R}^{n \times t}$ ,  $t \ll n$ , can be constructed. The efficient computation of the low-rank factor  $Z$  is the task of LR-ADI and of all other *low-rank methods* (see, e.g., the survey papers [19, 65] for further details on different low-rank methods for linear matrix equations).

It is well known that the convergence rate of the LR-ADI method is strictly connected to the selection of some parameters  $\{p_i\}_{i=1, \dots, j} \subset \mathbb{C}_-$  called *shifts*<sup>1</sup>. The computation of effective shifts is a highly non-trivial task and it has been a rather active research topic in the last decades. Many strategies are available in the literature and these can be divided into two categories: *Offline* routines [54, 60, 73], where the shifts are computed a priori, before LR-ADI starts and then, potentially, cyclically reused, and *online* schemes [12, 37], where the shifts are

<sup>1</sup> We only consider *proper* sets of shifts, namely  $\{p_i\}_{i=1, \dots, j}$  is closed with respect to complex conjugation.

---

```

input :  $A \in \mathbb{R}^{n \times n}$  stable,  $B \in \mathbb{R}^{n \times q}$ , max. iteration count  $j_{\max}$ ,
         normalized residual bound  $\varepsilon > 0$ .
output:  $Z_j \in \mathbb{R}^{n \times jq}$ ,  $Z_j Z_j^T = X_j \approx X$  approximate solution to (2).

1 Set  $W_0 = B$ ,  $Z_0 = []$ ,  $j = 1$ , and select  $p_1 \in \mathbb{C}_-$ 
2 while  $\|W_{j-1}^* W_{j-1}\|_F \geq \varepsilon \|B^* B\|_F$  and  $j \leq j_{\max}$  do
3   Solve  $(A + p_j I)S_j = W_{j-1}$ 
4   Set  $W_j = W_{j-1} - 2\text{Re}(p_j)S_j$ 
5   if  $\text{Im}(p_j) \neq 0$  then
6     Set  $\beta = \text{Re}(p_j)/\text{Im}(p_j)$  and  $p_{j+1} = \bar{p}_j$ 
7     Set  $S_{j+1} = \sqrt{2(\beta^2 + 1)}\text{Im}(S_j)$ 
8     Set  $W_{j+1} = W_{j-1} - 4\text{Re}(p_j)(\text{Re}(S_j) + \beta\text{Im}(S_j))$ 
9     Set  $S_j = \sqrt{2}(\text{Re}(S_j) + \beta\text{Im}(S_j))$ 
10    Set  $Z_{j+1} = [Z_{j-1}, \sqrt{-2\text{Re}(p_j)}S_j, \sqrt{-2\text{Re}(p_{j+1})}S_{j+1}]$ 
11    Set  $j = j + 1$ 
12  else
13    Set  $Z_j = [Z_{j-1}, \sqrt{-2\text{Re}(p_j)}S_j]$ 
14  Choose the next shift  $p_{j+1} \in \mathbb{C}_-$ 
15  Set  $j = j + 1$ 

```

---

**Algorithm 1** LR-ADI for Lyapunov equations

computed on the fly within the iterative procedure. The name shifts for the values  $p_j$  comes from the fact that in each LR-ADI iteration we need to solve a shifted linear system with a coefficient matrix of the form  $A + p_j E$ , or  $A + p_j I$ , in case of (1), or (2), respectively. Notice that since  $(A, E)$  (or  $A$  in case of (2)) is asymptotically stable and  $\{p_j\} \subset \mathbb{C}_-$ , all the linear systems involved in the LR-ADI scheme are well defined.

In Algorithm 1, we report an implementation of the LR-ADI scheme for the solution of (1). Notice that Algorithm 1 is designed to drastically reduce the amount of complex arithmetic calculations. Indeed, even though  $A$  and  $B$  in (2) are real, the shifts  $p_j$  are often complex if  $A$  is nonsymmetric, so that complex arithmetic may occur (see [11], [36, Chapter 4], and references therein for details and derivations).

One of the most computationally expensive steps of Algorithm 1 is the solution of the shifted linear systems with  $q$  right-hand sides in line 3. Such a job has to be carried out at each LR-ADI iteration. In this contribution, we propose to employ state-of-the-art block Krylov subspace methods for this task. In particular, for (2), we illustrate how to efficiently reuse the approximation space employed at the  $j$ th LR-ADI iteration and utilize it also in the next one. To this end, it is crucial that the right-hand side of the linear system we need to solve at the  $(j + 1)$ -st iteration can be represented in terms of the basis of the subspace employed in the previous iteration. This simple but critical observation lets us design a novel, efficient procedure that can lead to noticeable savings in the running time for the solution of (2). Indeed, all the LR-ADI steps can be completely merged into the Krylov routine so that the LR-ADI iteration is only implicitly

performed. Moreover, also the LR-ADI shift computation can be incorporated into the framework proposed in this paper.

The following is a synopsis of the paper. Section 2 is devoted to recalling the general (block) Krylov subspace framework for shifted linear systems. In particular, some details about the extended Krylov subspace method presented in [64] are given in Section 2.1. In Section 3, we present the main contribution of the paper and we show how to fully merge the LR-ADI iteration into the projection method adopted for the linear system solution. The selection of effective shifts is crucial for attaining a fast convergence in terms of number of LR-ADI iterations, and numerous strategies have been proposed in the literature to accomplish this task (see, e.g., [12, 16, 37, 54, 58, 60, 71, 73]). In Section 5, we illustrate how many of these routines can be integrated into our novel framework with no additional cost. The potential of our strategy is depicted in Section 6, where several numerical results are reported. We close the paper with our conclusions in Section 7.

Throughout the paper, we adopt the following notation. The matrix inner product is defined as  $\langle X, X \rangle_F := \text{trace}(Y^T X)$  so that the induced norm is  $\|X\|_F^2 = \langle X, X \rangle_F$ . The Kronecker product is denoted by  $\otimes$  whereas  $I_n$  and  $O_{n \times m}$  denote the identity matrix of order  $n$  and the  $n \times m$  zero matrix, respectively. Only one subscript is used for a square zero matrix, i.e.,  $O_{n \times n} = O_n$ , and the subscript is omitted whenever the dimension of  $I$  and  $O$  is clear from the context. Moreover,  $e_i$  is the  $i$ th basis vector of the canonical basis of  $\mathbb{R}^n$ . The brackets  $[\cdot]$  are used to concatenate matrices of conformal dimensions. In particular, a MATLAB-like notation is adopted and  $[M, N]$  denotes the matrix obtained by putting  $M$  on the left of  $N$  whereas  $[M; N]$  the one obtained by putting  $M$  on top of  $N$ , i.e.,  $[M; N] = [M^T, N^T]^T$ . If  $w \in \mathbb{R}^n$ ,  $\text{diag}(w)$  denotes the  $n \times n$  diagonal matrix whose  $i$ th diagonal entry corresponds to the  $i$ th component of  $w$ . Given  $X \in \mathbb{C}^{n \times m}$ , we write  $X = \text{Re}(X) + \iota \text{Im}(X)$ , where  $\text{Re}(X)$  and  $\text{Im}(X)$  are its real and imaginary parts, respectively, and  $\iota$  is the imaginary unit. The complex conjugate of  $X$  is denoted by  $\bar{X} = \text{Re}(X) - \iota \text{Im}(X)$ .

## 2 Block Krylov methods for shifted linear systems

The literature about the numerical solution of shifted linear systems by Krylov subspace methods is rather vast. Indeed, sequences of shifted linear systems arise in many applications belonging to different research areas like control theory [23, 41], wave propagation problems [8], mechanical systems [27], and quantum chromodynamics [32].

This algebraic problem is trickier than it looks and many researchers have contributed to its understanding providing important insights on its properties and designing efficient, robust algorithms for its solution. Here is an incomplete list of contributions on numerical schemes for sequences of shifted linear systems and their analysis [7, 28, 29, 48, 62, 67, 68, 70].

In this section, we consider sequences of shifted linear systems of the form

$$(A + p_j I)S_j = W, \quad W \in \mathbb{R}^{n \times q}, \tag{3}$$

where the right-hand side  $W$  does not depend on the index  $j$ , even though, in line 3 of Algorithm 1,  $W_{j-1}$  does change at every LR-ADI iteration. In Section 3, we show

how to adapt the machinery, presented here, to the case of linear systems of the form  $(A + p_j I)S_j = W_{j-1}$ , arising within the LR-ADI scheme.

Any Krylov routine for (3) computes a numerical solution of the form  $S_m^{(j)} = S_0 + V_m Y_m^{(j)} \approx S_j$ ,  $V_m = [V_1, \dots, V_m] \in \mathbb{R}^{n \times m \ell q}$ ,  $\ell \geq 1^2$ ,  $V_i \in \mathbb{R}^{n \times \ell q}$ ,  $i = 1, \dots, m$ ,  $Y_m^{(j)} \in \mathbb{C}^{m \ell q \times q}$ , where the orthonormal columns of  $V_m$  span a suitable subspace  $\mathcal{K}_m$ , namely,  $\text{Range}(V_m) = \mathcal{K}_m$ ,  $S_0$  is an initial guess, and the matrix  $Y_m^{(j)}$  can be computed by imposing different conditions. In particular,  $Y_m^{(j)}$  is often computed by either imposing a Galerkin condition on the residual or minimizing the residual norm. For the sake of simplicity, we consider  $S_0 = O$  in the following.

One of the most common choices for the approximation space  $\mathcal{K}_m$  is the block Krylov subspace

$$\mathbf{K}_m^\square(A, W) = \text{Range}([W, AW, \dots, A^{m-1}W]). \tag{4}$$

See, e.g., [30, 50, 61] and the references therein for further details on the block polynomial Krylov subspace  $\mathbf{K}_m^\square(A, W)$  and related methods.

However, Simoncini showed in [64] that the extended Krylov subspace [24]

$$\mathbf{EK}_m^\square(A, W) = \text{Range}([W, A^{-1}W, AW, A^{-2}W, \dots, A^{m-1}W, A^{-m}W]), \tag{5}$$

can be a powerful alternative for the solution of (3) in many cases. For instance, when  $A$  is large and real while the  $p_j$ 's are complex (see also Section 2.1).

The basis  $V_m$  of both the polynomial and extended Krylov subspace can be constructed by means of the (extended) Arnoldi process and the following Arnoldi relation is fulfilled

$$AV_m = V_m T_m + V_{m+1} E_{m+1}^\top T_m, \tag{6}$$

where  $T_m = V_{m+1}^\top AV_m \in \mathbb{R}^{(m+1)\ell q \times m \ell q}$ ,  $T_m$  is its principal square submatrix, and  $E_{m+1} = e_{m+1} \otimes I_{\ell q}$  (see, e.g., [56, 63]).

The Arnoldi relation (6) is one of the most crucial tools in the solution of (3) by Krylov methods. Indeed, it can be used to show the fundamental *shift-invariance* property of the Krylov subspaces (4) and (5), and the following relation holds true

$$(A + p_j I_n)V_m = V_m(T_m + p_j I_{m \ell q}) + V_{m+1} E_{m+1}^\top T_m. \tag{7}$$

See, e.g., [62, Equation (2.1)], [64, Equation (3.1)].

Equation (7) says that we can compute only one approximation space for solving (3). In particular, the space constructed using  $A$ , i.e.,  $\mathbf{K}_m^\square(A, W)$  or  $\mathbf{EK}_m^\square(A, W)$ , can be employed, by possibly being expanded, to solve all the shifted linear systems in the sequence (3).

Polynomial Krylov subspace methods often need many iterations to achieve the prescribed accuracy, so that a large subspace is constructed. This leads to an increment in both the storage demand and the computational efforts of the selected

<sup>2</sup> The value of  $\ell$  depends on the adopted approximation space. It holds,  $\ell = 1$  for the polynomial Krylov subspace in (4), whereas  $\ell = 2$  for the extended Krylov subspace in (5).

solution procedure. Different strategies have been developed to avoid the construction of a too large subspace.

With the goal of achieving a fast convergence in terms of number of iterations, the linear system (3) can be preconditioned, namely is transformed into an equivalent problem with better spectral properties. However, designing effective preconditioning operators for a sequence of shifted linear systems is a difficult task and often highly problem dependent. Very sophisticated schemes have been proposed in the literature (see, e.g., [4, 9, 20, 21, 46]).

Restarted routines are an alternative solution. In this framework, the approximation space  $\mathcal{K}_m$  is expanded until it reaches a prescribed maximum dimension. If the desired level of accuracy is not achieved, the last computed basis block  $\mathcal{V}_{m+1}$  is employed as initial block in the construction of a new subspace  $\mathcal{K}'_m$ . This procedure is iterated until a stopping criterion is fulfilled (see, e.g., [29, 62] and [30, Section 3.2.1]). However, in our framework, the LR-ADI shifts  $p_j$ 's are often computed on the fly and, thus, are not all available at the same time. Therefore, to fully take advantage of the computational efforts needed to solve the linear system  $(A + p_{j-1}I)S_{j-1} = W$ , we would have to store all the bases computed during the employed restarted Krylov procedure and use them to solve the  $j$ th linear system, as well. Unfortunately, this would destroy all the benefits in terms of storage complexity gained from the restart-paradigm.

In [64], Simoncini showed that the employment of the extended Krylov subspace (5), in place of (4), often leads to a faster convergence, in terms of iterations, to the point that the constructed subspace is usually smaller than the polynomial counterpart needed to reach the same level of accuracy. We, thus, decide to use such an approximation space for the solution of the shifted linear systems within the LR-ADI method and in the next section we recall some details of the extended Krylov subspace method.

Notice that the faster convergence of the extended Krylov subspace (5) comes with a toll. Indeed, at each iteration, a linear system with  $A$  has to be solved during the basis construction. Nevertheless, the increase in the overall workload of the solution process can be limited in general. Indeed, if we want to use a direct solver to invert  $A$ , for instance, the LU factors of  $A$  can be computed once and for all before the LR-ADI scheme starts. On the other hand, if an iterative procedure is employed, analogously a single preconditioner for  $A$  has to be designed once.

As already mentioned, in the formulation (3), the right-hand side  $W$  is fixed, namely it does not depend on the shift index  $j$ . However, in line 3 of Algorithm 1, the linear systems we need to solve are of the form

$$(A + p_j I)S_j = W_{j-1}.$$

At a first glance, having a nonconstant right-hand side does not allow for the employment of the shifted Krylov framework we briefly described above. A larger class of solvers, the so-called *recycling* Krylov methods, seems more appropriate (see, e.g., [31, 52, 67, 69, 70] for general sequences of shifted linear systems, and [1, 2, 26] for some recycling Krylov techniques applied in a model reduction context). However, in Section 3, we show that, in the LR-ADI context for  $j > 1$ , the residual factor  $W_{j-1}$  belongs to the subspace  $\mathcal{K}_m$  employed in the solution of the  $(j - 1)$ -st linear system  $(A + p_{j-1}I)S_{j-1} = W_{j-2}$ . Along with the shift-invariance property of the

Krylov subspace, this observation allows us to utilize only one subspace for the solution of all the shifted linear systems within the LR-ADI method. In turn, as shown in Section 6, we can notably reduce the computational effort of the overall procedure.

### 2.1 The extended Krylov subspace method for shifted linear systems

In this section, we recall the extended Krylov subspace method for shifted linear systems presented in [64].

Given the sequence of shifted linear systems (3), the extended Krylov subspace method computes a solution of the form  $S_m^{(j)} = V_m Y_m^{(j)}$ , where the  $2mq$  orthonormal columns of  $V_m$  span the extended Krylov subspace (5), whereas the  $2mq \times q$  matrix  $Y_m^{(j)}$  can be computed in different manners.

For instance,  $Y_m^{(j)}$  can be computed by imposing a Galerkin condition on the residual  $R_m^{(j)} = (A + p_j I)V_m Y_m^{(j)} - W$ , namely by imposing  $V_m^T R_m^{(j)} = 0$ . Thanks to the shifted Arnoldi relation (7), it is easy to show that such a Galerkin condition is equivalent to solving the projected linear systems

$$(T_m + p_j I)Y_m^{(j)} = E_1 \gamma, \tag{8}$$

where  $E_1 = e_1 \otimes I_{2q}$ , and  $\gamma \in \mathbb{R}^{2q \times q}$  is such that  $W = V_1 \gamma$ .

With  $Y_m^{(j)}$  at hand, the Frobenius norm of the residual  $\|R_m^{(j)}\|_F$  can be computed at low cost, as

$$\|R_m^{(j)}\|_F = \|E_{m+1}^T \underline{T}_m Y_m^{(j)}\|_F, \tag{9}$$

following [64, Equation (3.2)].

Alternatively, following the discussion in [68, Section 4.1], the matrix  $Y_m^{(j)}$  can be computed also by minimizing the residual norm, i.e.,

$$Y_m^{(j)} = \operatorname{argmin}_{Y \in \mathbb{R}^{2mq \times q}} \|(A + p_j I)V_m Y - W\|_F. \tag{10}$$

Once again, thanks to the shifted Arnoldi relation (7), the minimization problem in (10) simplifies, and we can compute  $Y_m^{(j)}$  as

$$Y_m^{(j)} = \operatorname{argmin}_{Y \in \mathbb{R}^{2mq \times q}} \|(T_m + p_j [I_{2mq}; O_{2q \times 2mq}])Y - E_1 \gamma\|_F. \tag{11}$$

Note the abuse of notation: in (11)  $E_1 \in \mathbb{R}^{2(m+1)q \times 2q}$  whereas  $E_1 \in \mathbb{R}^{2mq \times 2q}$  in (8).

If  $QP = \underline{T}_m + p_j [I_{2mq}; O_{2q \times 2mq}]$  denotes the QR factorization of  $\underline{T}_m + p_j [I_{2mq}; O_{2q \times 2mq}]$ , and we consider the following partition

$$Q = [Q_1, Q_2], Q_1 \in \mathbb{R}^{2(m+1)q \times 2mq}, Q_2 \in \mathbb{R}^{2(m+1)q \times 2q}, P = \begin{bmatrix} P_1 \\ O_{2q \times 2mq} \end{bmatrix}, P_1 \in \mathbb{R}^{2mq \times 2mq},$$

then the matrix  $Y_m^{(j)}$  in (11) can be computed as

$$Y_m^{(j)} = P_1^{-1} Q_1^T E_1 \gamma, \tag{12}$$

---

**input** :  $A \in \mathbb{R}^{n \times n}$ ,  $\{p_j\} \subset \mathbb{C}$ ,  $W \in \mathbb{R}^{n \times q}$ , and normalized residual bound  $\varepsilon > 0$   
**output**:  $S_m^{(j)} \in \mathbb{R}^{n \times q}$ ,  $S_m^{(j)} = V_m Y_m^{(j)} \approx S_j$ , where  $(A + p_j I)S_j = W_j$

- 1 Set  $\beta = \|W\|_F$ ,  $\Sigma = \{1, \dots, \max j\}$ ,  $\Sigma_C = \emptyset$
- 2 Perform economy-size QR,  $[W, A^{-1}W] = [\mathcal{V}_1^{(1)}, \mathcal{V}_1^{(2)}][\gamma, \theta]$ ,  $\gamma, \theta \in \mathbb{R}^{2q \times q}$
- 3 **for**  $m = 1, 2, \dots$ , *till convergence*, **do**
- 4     Compute next basis block  $\mathcal{V}_{m+1}$  as in [63] and set  $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$
- 5     Update  $T_m = V_m^T A V_m$  as in [63]
- 6     Compute  $Y_m^{(j)}$  for  $j \in \Sigma \setminus \Sigma_C$  as in (8) or (12)
- 7     Compute  $\|R_m^{(j)}\|_F$  accordingly as in (9) or (13)
- 8     **if**  $\|R_m^{(j)}\|_F / \beta < \varepsilon$  **then**
- 9         Set  $\Sigma_C = \Sigma_C \cup \{j\}$
- 10     **if**  $\Sigma \setminus \Sigma_C = \emptyset$  **then**
- 11         **Break** and go to 12
- 12 Set  $S_m^{(j)} = V_m Y_m^{(j)}$

---

**Algorithm 2** Extended Krylov subspace method for shifted linear systems

and the residual norm is given by

$$\|R_m^{(j)}\|_F = \|Q_2^T E_1 \gamma\|_F. \tag{13}$$

The overall procedure is summarized in Algorithm 2, where  $\Sigma$  contains the indices of all yet unsolved systems, whereas  $\Sigma_C$  contains the indices of all the systems that have already been solved. The basis block  $\mathcal{V}_{m+1}$  can be computed by following [63]. This operation involves both matrix-vector products and linear system solves with  $A$ . Moreover, the basis  $V_m$  is real whenever  $A$  and  $W$  are so. Complex arithmetic may occur in the computation of  $Y_m^{(j)}$ , if  $\text{Im}(p_j) \neq 0$ .

Notice that as soon as the  $j$ th linear system has converged, namely the related relative residual norm is sufficiently small, we stop solving the  $j$ th projected problem<sup>3</sup>. Once all the linear systems have converged, we terminate the iterative process.

To conclude, we would like to point out that, to the best of our knowledge, this is the first time the minimal residual condition (11) is proposed within the extended Krylov subspace method for shifted linear systems.

### 3 Merging the two iterative procedures

In this section, we show how the LR-ADI iteration and the extended Krylov subspace method for shifted linear systems can be merged together into a novel, efficient iterative procedure for the solution of (2).

---

<sup>3</sup> Either (8) or (11).



As already mentioned, in the sequence of shifted linear systems in line 3 of Algorithm 1, also the right-hand side  $W_{j-1}$  depends on the current LR-ADI iteration  $j$ . Therefore, at a first glance, we seemingly have to build a new subspace at each iteration  $j$ , by employing the current  $W_{j-1}$  as initial block. However, in the following theorem, we show that  $W_{j-1}$  belongs to the subspace constructed to solve the  $(j - 1)$ -st linear system so that such a space can be used, by being possibly expanded, also in the solution of the subsequent linear system.

**Theorem 3.1** Let  $S_j = V_{m_j} Y_{m_j}$ ,  $j \geq 1$ ,  $\text{Range}(V_{m_j}) = \mathbf{EK}_{m_j}^\square(A, B)$  for certain  $m_j \geq 0$ . Then

$$\text{Range}(W_j) \subseteq \mathbf{EK}_{m_j}^\square(A, B).$$

*Proof* We are going to show the statement by induction on  $j$ .

The first linear system to be solved within the LR-ADI method is  $(A + p_1 I)S_1 = B$  and the extended Krylov subspace  $\mathbf{EK}_{m_1}^\square(A, B)$  can be employed to this end. The computed solution is of the form  $S_1 = V_{m_1} Y_{m_1}$ ,  $m_1 > 0$ , where  $\text{Range}(V_{m_1}) = \mathbf{EK}_{m_1}^\square(A, B)$  and  $Y_{m_1} \in \mathbb{C}^{2m_1, q \times q}$ . It is thus easy to show that  $W_1 = B - 2\text{Re}(p_1)S_1 = V_{m_1}(E_1 \gamma - 2\text{Re}(p_1)Y_{m_1})$  is such that  $\text{Range}(W_1) \subseteq \mathbf{EK}_{m_1}^\square(A, B)$ .

We now assume the statement holds for a certain  $j - 1 \geq 1$ , and we show it holds for  $j$  as well. Since  $S_j = V_{m_j} Y_{m_j}$  by assumption and  $\text{Range}(W_{j-1}) \subseteq \mathbf{EK}_{m_{j-1}}^\square(A, B)$  by inductive hypothesis, namely we can write  $W_{j-1} = V_{m_{j-1}} Y_{j-1}$  for a certain  $Y_{j-1} \in \mathbb{R}^{2m_{j-1}, q \times q}$ , we have

$$\begin{aligned} W_j &= W_{j-1} - 2\text{Re}(p_j)S_j = V_{m_{j-1}} Y_{j-1} - 2\text{Re}(p_j)V_{m_j} Y_{m_j} \\ &= V_{m_j} \left( [Y_{j-1}; \mathcal{O}_{2(m_j-m_{j-1}), q \times q}] - 2\text{Re}(p_j)Y_{m_j} \right). \end{aligned}$$

Therefore,  $\text{Range}(W_j) \subseteq \mathbf{EK}_{m_j}^\square(A, B)$ .  $\square$

Theorem 3.1 shows that  $W_j$  is exactly represented in  $\mathbf{EK}_{m_j}^\square(A, B)$ . This means that the latter subspace can be still employed for the computation of  $S_{j+1}$  by being possibly expanded. Indeed, no components of  $W_j$  are annihilated when either the Galerkin or the minimal residual condition is imposed. In the following corollary, we show how to easily write down the projected problems (8) and (11) along with the corresponding residual norm computation.

**Corollary 3.1** Assume the prerequisites of Theorem 3.1 hold. If a Galerkin condition is imposed for the computation of  $S_j = V_{m_j} Y_{m_j}$ , then the matrix  $Y_{m_j}$  amounts to the solution of the projected linear system

$$(T_{m_j} + p_j I_{2m_j, q})Y_{m_j} = [Y_{j-1}; \mathcal{O}_{2(m_j-m_{j-1}), q \times q}], \tag{14}$$

where  $Y_{j-1} \in \mathbb{R}^{2m_{j-1}, q \times q}$  is such that  $W_{j-1} = V_{m_{j-1}} Y_{j-1}$ ,  $m_{j-1} \leq m_j$ . The related residual norm can be computed by

$$\|R_{m_j}\|_F = \|E_{m_j+1}^T T_{m_j} Y_{m_j}\|_F. \tag{15}$$

Similarly, if a minimal residual norm condition is imposed, we have

$$Y_{m_j} = \underset{Y \in \mathbb{C}^{2m_j q \times q}}{\operatorname{argmin}} \|(\underline{T}_{m_j} + p_j [I_{2m_j q}; O_{2q \times 2m_j q}])Y - [\Upsilon_{j-1}; O_{2(m_j-m_{j-1}+1)q \times q}]\|_F, \tag{16}$$

so that

$$\|R_{m_j}\|_F = \|Q_2^T [\Upsilon_{j-1}; O_{2(m_j-m_{j-1}+1)q \times q}]\|_F, \tag{17}$$

where the  $2q$  orthonormal columns of  $Q_2$  are a basis of the kernel of  $\underline{T}_{m_j} + p_j [I_{2m_j q}; O_{2q \times 2m_j q}]$ .

**Proof** Since  $W_{j-1} = V_{m_{j-1}} \Upsilon_{j-1}$  and we look for a solution  $S_j = V_{m_j} Y_{m_j}$  to  $(A + p_j I)S_j = W_{j-1}$ , we can write

$$\begin{aligned} R_{m_j} &= (A + p_j I)S_j - W_{j-1} = (A + p_j I)V_{m_j} Y_{m_j} - V_{m_{j-1}} \Upsilon_{j-1} \\ &= V_{m_j} \left( (T_{m_j} + p_j I_{2m_j q})Y_{m_j} - [\Upsilon_{j-1}; O_{2(m_j-m_{j-1})q \times q}] \right) + V_{m_j+1} E_{m_j+1}^T T_{m_j} \\ &= V_{m_j+1} \left( (\underline{T}_{m_j} + p_j [I_{2m_j q}; O_{2q \times 2m_j q}])Y_{m_j} - [\Upsilon_{j-1}; O_{2(m_j-m_{j-1}+1)q \times q}] \right). \end{aligned}$$

If a Galerkin condition is imposed, namely  $V_{m_j}^T R_{m_j} = 0$ , then  $Y_{m_j}$  is the solution of the linear system in (14) and the related residual norm  $\|R_{m_j}\|_F$  can be computed as in (15).

Similarly, if a minimal residual condition is imposed,  $Y_{m_j}$  solves the minimization problem (16) and the  $\|R_{m_j}\|_F$  fulfills (17).  $\square$

Once  $S_j = V_{m_j} Y_{m_j}$  is computed, namely the related residual norm  $\|R_{m_j}\|_F$  is sufficiently small, we proceed with the remaining LR-ADI operations.

We would like to point out that the expression of  $W_j$ , i.e.,  $W_j = V_{m_j} \Upsilon_j$ , can be exploited for the Lyapunov residual norm as well. Indeed,

$$\|W_j^* W_j\|_F = \|\Upsilon_j^* \Upsilon_j\|_F. \tag{18}$$

This means that also the computation of the Lyapunov residual norm can be carried out by manipulating small matrices of dimension  $2m_j q \times q$ . Similarly, the solution  $Z_j$  can be assembled at the very end of the LR-ADI procedure once the residual norm in (18) is sufficiently small. Indeed,

$$\begin{aligned} Z_j &= [Z_{j-1}, \sqrt{-2\operatorname{Re}(p_j)} S_j] = [\sqrt{-2\operatorname{Re}(p_1)} S_1, \sqrt{-2\operatorname{Re}(p_2)} S_2, \dots, \sqrt{-2\operatorname{Re}(p_j)} S_j] \\ &= [\sqrt{-2\operatorname{Re}(p_1)} V_{m_1} Y_{m_1}, \sqrt{-2\operatorname{Re}(p_2)} V_{m_2} Y_{m_2}, \dots, \sqrt{-2\operatorname{Re}(p_j)} V_{m_j} Y_{m_j}] \\ &= V_{m_j} [[Y_{m_1}; O_{2(m_j-m_1)q \times q}], [Y_{m_2}; O_{2(m_j-m_2)q \times q}], \dots, Y_{m_j}] \\ &\quad \cdot (\sqrt{-2\operatorname{diag}(\operatorname{Re}(p_1), \dots, \operatorname{Re}(p_j))} \otimes I_q). \end{aligned} \tag{19}$$

The overall procedure combining the LR-ADI iteration with the extended Krylov subspace method for shifted linear systems is depicted in Algorithm 3<sup>4</sup>.

As in Algorithm 1, if  $\text{Im}(p_j) \neq 0$ , in lines 15 to 19 we set  $p_{j+1} = \bar{p}_j$ , and we follow the implementation suggested in [11, 36] to reduce the amount of complex arithmetic. In particular,  $Y_{m_{j+1}}$  can be obtained from  $Y_{m_j}$  without solving (14) or (16). Moreover, the adopted scheme results in a real  $Z_j$  (see [11] and [36, Algorithm 4.3] for further details).

**Remark 4** Theorem 3.1 shows that  $\text{Range}(W_j) \subseteq \mathbf{EK}_{m_j}^\square(A, B)$  whenever  $W_j$  is updated as  $W_j = W_{j-1} - 2\text{Re}(p_j)S_j$ , namely whenever all the employed shifts are real. In case of shifts with nonzero imaginary part, the LR-ADI implementation we adopt sets

$$W_{j+1} = W_{j-1} - 4\text{Re}(p_j)(\text{Re}(S_j) + \beta\text{Im}(S_j)).$$

Therefore, we need to show that  $W_{j+1}$  defined as above is still such that  $\text{Range}(W_{j+1}) \subseteq \mathbf{EK}_{m_j}^\square(A, B)$ . This can be done by applying the same exact arguments used in the proof of Theorem 3.1. In particular, the result follows by noticing that the basis  $V_m$  is real, as we assumed  $A$  and  $B$  to be real matrices, and that we can write

$$\begin{aligned} W_{j+1} &= W_{j-1} - 4\text{Re}(p_j)(\text{Re}(S_j) + \beta\text{Im}(S_j)) \\ &= V_{m_j} \left( [Y_{j-1}; \mathcal{O}_{2(m_j-m_{j-1})q \times q}] - 4\text{Re}(p_j)(\text{Re}(Y_{m_j}) + \beta\text{Im}(Y_{m_j})) \right). \end{aligned}$$

Notice that two tolerances  $\epsilon_{\text{inn}}^{(j)}$ , and  $\epsilon$  are employed in Algorithm 3. In particular,  $\epsilon$  is used to assess the accuracy of the computed solution in terms of the Lyapunov residual norm, whereas  $\epsilon_{\text{inn}}^{(j)}$  is employed to determine whether the solution of the current linear system is sufficiently correct. In principle, the user can provide a fixed value for the inner tolerance, i.e.,  $\epsilon_{\text{inn}}^{(j)} \equiv \bar{\epsilon}_{\text{inn}}$  for all  $j$ . However, the theory developed in [38] can be used to adaptively compute  $\epsilon_{\text{inn}}^{(j)}$  as the LR-ADI iterations proceed. The *relaxation* strategy presented in [38, Section 3] allows us to increase  $\epsilon_{\text{inn}}^{(j)}$  as  $j$  grows. Therefore, especially when  $\epsilon_{\text{inn}}^{(j)}$  is rather large, there is no need to expand the current extended Krylov subspace in general. In all the results reported in Section 6, we employ such a strategy and  $\epsilon_{\text{inn}}^{(j)}$  is computed according to [38, Equation (3.18b)] (see also [44] for similar results in case of Sylvester equations).

We would like to point out that the lines 23 to 28 in Algorithm 3 and the use of the flag `flag_noexpand` are crucial to reduce the computational cost of the overall procedure. Indeed, those lines are devoted to check whether the current subspace already contains enough spectral information to solve the current linear system. If this is the case, we do not expand the current space avoiding unnecessary increments in the memory requirements and computational efforts.

<sup>4</sup> Many subscripts have been removed to make the algorithm more readable.

---

**input** :  $A \in \mathbb{R}^{n \times n}$ ,  $A$  stable,  $B \in \mathbb{R}^{n \times q}$ , max. inner and outer iteration count  $m_{\max}, j_{\max}$ , normalized residual bound  $\varepsilon_{\text{out}} > 0$

**output**:  $Z_j \in \mathbb{R}^{n \times jq}$ ,  $Z_j Z_j^T \approx X$  approximate solution to (2).

- 1 Set  $v = \|B^* B\|_F$ ,  $m_0 = 0$ ,  $Y_0 = O$ , and select  $p_1 \in \mathbb{C}_-$
- 2 Perform economy-size QR,  $[B, A^{-1} B] = [\mathcal{V}_1^{(1)}, \mathcal{V}_1^{(2)}][\gamma, \theta]$ ,  $\gamma, \theta \in \mathbb{R}^{2q \times q}$
- 3 Set  $\Upsilon_0 = E_1 \gamma$ ,  $m = j = 1$ , and select  $\varepsilon_{\text{inn}}^{(1)}$
- 4 **while**  $m \leq m_{\max}$  **and**  $j \leq j_{\max}$  **do**
- 5     Compute next basis block  $\mathcal{V}_{m+1}$  as in [63] and set  $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$
- 6     Update  $T_m = V_m^T A V_m$  as in [63]
- 7     Compute  $Y_m$  as in (14) or (16)
- 8     Compute  $\|R_m\|_F$  accordingly as in (15) or (17)
- 9     **if**  $\|R_m\|_F \leq \varepsilon_{\text{inn}}^{(j)}$  **then**
- 10         flag\_noexpand = 1
- 11         **while** flag\_noexpand **do**
- 12             Set  $m_j = m$  and  $Y_{m_j} = Y_m$
- 13             Set  $\Upsilon_j = \Upsilon_{j-1} - 2\text{Re}(p_j)Y_{m_j}$
- 14             **if**  $\text{Im}(p_j) \neq 0$  **then**
- 15                 Set  $m_{j+1} = m_j$ ,  $\beta = \text{Re}(p_j)/\text{Im}(p_j)$ , and  $p_{j+1} = \bar{p}_j$
- 16                 Set  $Y_{m_{j+1}} = \sqrt{2(\beta^2 + 1)}\text{Im}(Y_{m_j})$
- 17                 Set  $\Upsilon_{j+1} = \Upsilon_{j-1} - 4\text{Re}(p_j)(\text{Re}(Y_{m_j}) + \beta\text{Im}(Y_{m_j}))$
- 18                 Set  $Y_{m_j} = \sqrt{2}(\text{Re}(Y_{m_j}) + \beta\text{Im}(Y_{m_j}))$
- 19                 Set  $j = j + 1$
- 20             **if**  $\|\Upsilon_j^* \Upsilon_j\|_F \leq v \cdot \varepsilon_{\text{out}}$  **then**
- 21                 **Break** and go to 30
- 22             Choose the next shift  $p_{j+1} \in \mathbb{C}_-$
- 23             Set  $j = j + 1$
- 24             Compute  $Y_m$  as in (14) or (16)
- 25             Compute  $\|R_m\|_F$  accordingly as in (15) or (17)
- 26             Select  $\varepsilon_{\text{inn}}^{(j)}$
- 27             **if**  $\|R_m\|_F \geq \varepsilon_{\text{inn}}^{(j)}$  **then**
- 28                 flag\_noexpand = 0
- 29     Set  $m = m + 1$
- 30  $Z_j = V_m[[Y_{m_1}; O_{2(m_j-m_1)q \times q}], [Y_{m_2}; O_{2(m_2-m_1)q \times q}], \dots, Y_{m_j}]$   
 $(\sqrt{-2\text{diag}(\text{Re}(p_1), \dots, \text{Re}(p_j))} \otimes I_q)$

---

**Algorithm 3** LR-ADI-EKSM for Lyapunov equations

If  $\mathbf{Y} := [[Y_{m_1}; O_{2(m_j-m_1)q \times q}], [Y_{m_2}; O_{2(m_2-m_1)q \times q}], \dots, Y_{m_j}]$ , (19) shows that the numerical solution computed by the proposed LR-ADI implementation is of the form

$$Z_j Z_j^T = -2V_{m_j}(\mathbf{Y}(\text{diag}(\text{Re}(p_1), \dots, \text{Re}(p_j)) \otimes I_q)\mathbf{Y}^T)V_{m_j}^T. \tag{20}$$

The right-hand side in (20) has the typical form of an approximate solution computed by a projection method applied to (2). In particular, if the extended Krylov subspace method (K-PIK) presented in [63] is applied to solve (2), the computed approximation is of the form  $X_m = V_m L_m V_m^T$ , where the orthonormal columns of  $V_m$  are a basis of  $\mathbf{EK}_m^\square(A, B)$  and  $L_m$  is computed by imposing a Galerkin condition on the residual matrix  $AV_m L_m V_m^T + V_m L_m V_m^T A^T + BB^T$ . Therefore, the proposed LR-ADI implementation can be seen as a novel projection method where the coefficients of the linear combination in terms of the basis vectors that provides the approximate solution, namely the matrix  $\mathbf{Y}(\text{diag}(\text{Re}(p_1), \dots, \text{Re}(p_j)) \otimes I_q) \mathbf{Y}^T$ , is computed as outlined above and not by imposing a Galerkin condition on the residual matrix. This perspective may provide new insights on the relation between LR-ADI and K-PIK. However, this is beyond the scope of this paper. Similar investigations, relating LR-ADI and rational Krylov subspace methods, have been reported in [25, 74, 75].

The expression (20) resembles the  $LDL^T$ -form of the LR-ADI solution. This formulation, while being more natural for projection-based solvers, also turned out to be advantageous when LR-ADI is employed as linear solver for differential matrix equations (see [39]).

### 4 Shift computation

Many of the procedures, available in the literature, for the ADI shift computation need the explicit construction of a basis of  $\text{Range}(Z_j)$  or a subspace thereof. For instance, in [12], the authors suggest to use, as shifts  $p_j$ , a subset of the Ritz values of  $A$  with respect to  $Z_j = \text{Range}(\tilde{Z}_j)$ , where  $\tilde{Z}_j \in \mathbb{R}^{n \times h}$  consists of the last  $h > 0$  columns of  $Z_j$  that have been orthogonalized with respect to each other. However, (19) shows that Algorithm 3 provides us with a matrix  $Z_j$  such that  $\text{Range}(Z_j) \subseteq \mathbf{EK}_{m_j}^\square(A, B)$  so that the Ritz values of  $A$  with respect to  $\mathbf{EK}_{m_j}^\square(A, B)$  can be employed as shifts. Moreover, in standard LR-ADI implementations, one has to explicitly compute the projection of  $A$  onto  $Z_j$  increasing the computational efforts of the overall procedure. In our approach, the projection of  $A$  onto  $\mathbf{EK}_{m_j}^\square(A, B)$  is given for free as this amounts to  $T_{m_j}$  and no additional operations are required.

The observation above can be applied to many schemes for the shift computation. In the following, we give some details for the residual-Hamiltonian-based shifts and the residual norm-minimizing shifts presented in [37].

In [37, Section 2.1.3], at the  $j$ th LR-ADI iteration, the Hamiltonian matrix  $\mathcal{H}_j = \begin{bmatrix} A^T & O \\ W_j W_j^T & -A \end{bmatrix}$  is considered and its projection onto  $Z_j$ , namely  $\tilde{\mathcal{H}}_j = \begin{bmatrix} (\tilde{Z}_j^T A \tilde{Z}_j)^T & O \\ \tilde{Z}_j^T W_j W_j^T \tilde{Z}_j & -\tilde{Z}_j^T A \tilde{Z}_j \end{bmatrix}$  is constructed. In our case, we can easily construct the projection of  $\mathcal{H}_j$  onto  $\mathbf{EK}_{m_j}^\square(A, B)$  and this is given by

$$\tilde{\mathcal{H}}_j = \begin{bmatrix} T_{m_j}^T & O \\ Y_j^T Y_j & -T_{m_j} \end{bmatrix} \in \mathbb{R}^{4m_j q \times 4m_j q}. \tag{21}$$

With (21) at hand, we compute its stable eigenpairs  $\left( \lambda_k, \begin{bmatrix} s_k \\ t_k \end{bmatrix} \right)$ ,  $\text{Re}(\lambda_k) < 0$ ,  $s_k, t_k \in \mathbb{R}^{2m_j q}$ , and the  $(j + 1)$ -st residual-Hamiltonian-based shift  $p_{j+1}$  is selected as the eigenvalue  $\lambda_{\hat{k}}$  such that  $t_{\hat{k}} = \text{argmax}\{\|t_k\|\}$ .

For the computation of residual-norm-minimizing shifts, in [37, Section 3], a rather involved optimization procedure is presented. In particular, the real and imaginary parts of  $p_{j+1} = \theta_{j+1} + i\xi_{j+1}$  are computed by solving the following minimization problem

$$[\theta_{j+1}, \xi_{j+1}] = \underset{\theta \in \mathbb{R}, \xi \in \mathbb{R}}{\text{argmin}} \left\| W_j - 2\theta((A + (\theta + i\xi)I)^{-1}W_j) \right\|^2. \tag{22}$$

The objective function in (22) is expensive to evaluate, making the shift computation often more expensive than a single LR-ADI iteration. To overcome this issue, Kürschner proposes to employ smaller matrices  $\tilde{A}$  and  $\tilde{W}_j$  in place of  $A$  and  $W_j$ . Once again,  $\tilde{A}$  and  $\tilde{W}_j$  are the projection of  $A$  and  $W_j$  onto a suitable subspace. This subspace is chosen to be  $\mathbf{EK}_\ell^\square(A, B) \cup \text{Range}(Z_j)$  for a certain, usually small,  $\ell > 0$ . In our implementation,  $\mathbf{EK}_\ell^\square(A, B) \cup \text{Range}(Z_j) \subseteq \mathbf{EK}_{m_j}^\square(A, B)$  if  $\ell \leq m_j$ . Therefore, we can set  $\tilde{A} = T_{m_j}$  and  $\tilde{W}_j = Y_j$  for the approximation of  $[\theta_{j+1}, \xi_{j+1}]$  (22).

### 5 Numerical examples

In this section, we illustrate the potential of the scheme we propose in this paper. The two variants of the LR-ADI-EKSM method, we have illustrated in Section 3, will be denoted by LR-ADI-EKSM(G) and LR-ADI-EKSM(MR). In particular, in LR-ADI-EKSM(G), we solve the linear systems by imposing a Galerkin condition, i.e., the matrix  $Y$  is computed by solving the reduced problem (14). In LR-ADI-EKSM(MR),  $Y$  solves the least squares problem (16).

We test Algorithm 3 on different instances of (2) coming from the discretization of certain PDEs, and we study how the computational cost of the main steps of Algorithm 3 depends on the problem dimension  $n$  and rank of the right hand side  $q$ .

The results achieved by Algorithm 3 are also compared to the ones obtained by running a standard implementation of the LR-ADI method. In particular, we employed the MATLAB function `mess_lradi` available in the M-M.E.S.S. package [59]. Notice that `mess_lradi` is intended to be a *black-box* routine so that many checks and inspections are performed before the actual solution process starts. This may increase the overall running time of `mess_lradi`. Therefore, to have fair comparisons, we also report the results obtained by running a standard implementation of LR-ADI where the overhead cost mentioned above is not present. Such a routine is simply denoted by `lradi` in the tables that follow.

**Table 1** SOLVER: solver employed for solving the linear systems with  $A$  in LR-ADI-EKSM and K-PIK and with  $A + p_j I$  in `lradi` and `mess_lradi`. In the column RELAXATION, we indicate whether a certain scheme is equipped with the relaxation strategy proposed in [38]

		Solver	Relaxation
Experiment 1	LR-ADI-EKSM(G)	backslash	✓
	<code>lradi</code>	backslash	✗
	<code>mess_lradi</code>	backslash	✗
Experiment 2	LR-ADI-EKSM(MR)	PGMRES	✓
	<code>lradi</code>	PGMRES	✓
	<code>mess_lradi</code>	PGMRES	✗
Experiment 3	LR-ADI-EKSM(G)	backslash	✓
	<code>mess_lradi</code>	backslash	✗
	K-PIK	backslash	—
Experiment 4	LR-ADI-EKSM(G)	backslash	✓
	<code>mess_lradi</code>	backslash	✗

For a better understanding, in Table 1, we summarize the adopted linear system solver included in the tested routines for each of the numerical experiments that follow. Similarly, in Table 1, we indicate whether a given scheme is equipped with the relaxation strategy coming from [38] for the selection of  $\epsilon_{\text{inn}}^{(j)}$ .

For all experiments, the tolerance  $\epsilon$  for the relative residual norm is set to  $10^{-8}$ . Moreover, except for Experiment 3, we always employ the residual-Hamiltonian-based shifts presented in [37] and computed as illustrated in Section 5.

All results were obtained by running MATLAB R2020b [47] on a standard node<sup>5</sup> of the Linux cluster `mechthild` hosted at the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany.

**Experiment 1** In the first experiment, we consider a Lyapunov equation where

$$A = I_h \otimes D_h + D_h \otimes I_h, \quad D_h = \text{tridiag}(1, -2, 1) \in \mathbb{R}^{h \times h}.$$

Therefore,  $A \in \mathbb{R}^{n \times n}$ ,  $n = h^2$ , is symmetric and stable. We first consider a matrix  $B \in \mathbb{R}^{n \times q}$  with random entries and unit norm, and in Table 2, we depict how the overall solution time distributes among the main steps of our algorithm for different values of  $n$  and  $q$ .

In both LR-ADI-EKSM(G) and LR-ADI-EKSM(MR), the linear systems with  $A$  required for the basis construction are solved by means of the MATLAB sparse direct solver “backslash.” In particular,  $A$  is factorized once and for all before the iterative procedures start so that only triangular systems are actually solved during the basis construction. The computational time for the factorization of  $A$  is always included in the results that follow.

In this experiment, LR-ADI-EKSM(G) and LR-ADI-EKSM(MR) perform very similarly. We thus report only the results achieved by the former.

<sup>5</sup> CPU: 2x Intel Xeon Skylake Silver 4110 @ 2.1 GHz, 8 cores per CPU. RAM: 192 GB DDR4 ECC.

**Table 2** Experiment 1. Computational timings devoted to the different main steps of LR-ADI-EKSM(G) for different values of the problem size  $n$  and rank of the right-hand side  $q$

$n$	$q$	BASIS	PROJECTED PR.	SHIFT	ETC	TOTAL	Time (s)
		It.	Time (s)	Time (s)	Time (s)	Time (s)	
360 000	1	28	8.00	0.07	1.84	2.33	12.24
	3	27	21.70	0.26	10.35	3.66	35.97
	5	27	32.79	0.50	22.38	4.60	60.27
640 000	1	29	16.87	0.09	2.59	4.74	24.29
	3	31	42.74	0.31	14.91	7.45	65.41
	5	31	101.15	0.81	34.70	9.26	145.92
1 000 000	1	26	28.99	0.09	2.64	7.52	39.24
	3	31	112.60	0.45	19.04	11.67	143.76
	5	29	183.80	1.12	40.56	14.52	240.00

BASIS, basis construction (Algorithm 3 — lines 2, 5, and 6); PROJECTED PR., computation of  $Y$  (Algorithm 3 — lines 7 and 24); SHIFT, shifts computation (Algorithm 3 — line 22); ETC, remainder of the algorithm (e.g., factorization of  $A$ ). It. indicates the number of ADI iterations that have been implicitly performed

As expected, the time devoted to the basis construction represents the majority of the overall computational efforts. This is the usual case in Krylov projection algorithms. This cost increases as  $q$  grows. Indeed, a larger subspace is computed making the basis construction, and in particular the orthogonalization step, rather demanding. Having a large dimensional approximation space leads to a more expensive shift computation, as well.

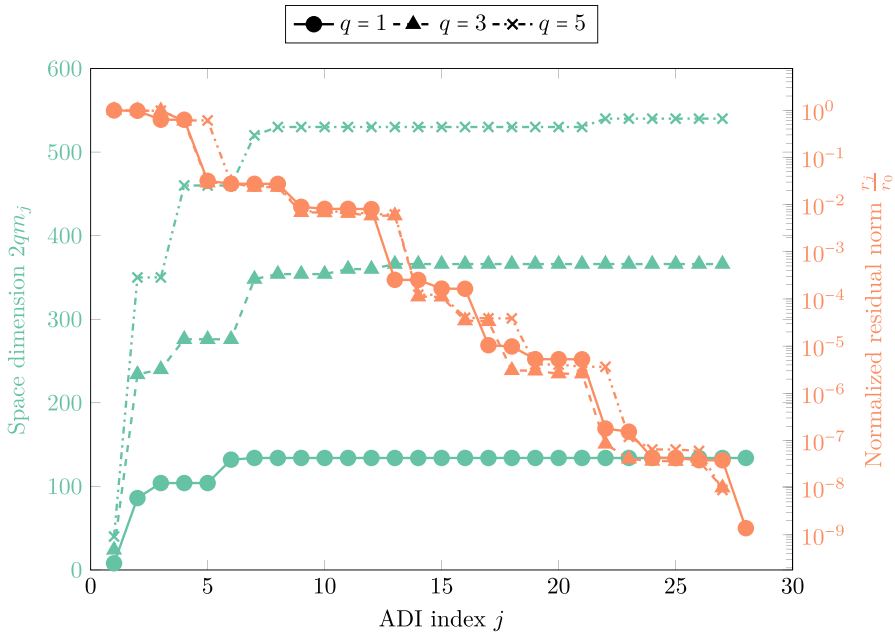
In Fig. 1 (left y-axis), we illustrate how the dimension of the computed extended Krylov subspace grows in terms of  $j$  for  $n=360\,000$  and different values of  $q$ .

In this experiment, we can notice that the subspace constructed to solve the second shifted linear system, namely  $(A + p_2I)S_2 = W_1$ , is a very rich approximation space in terms of spectral information. Indeed, we need to only slightly expand it to solve the subsequent linear systems without compromising the decrease in the Lyapunov residual norm (see Fig. 1 (right y-axis)). This means that the majority of the computational efforts are dedicated to solve the second linear system, and we can capitalize on them for  $j > 2$  reducing the overall workload of the solution process. We would like to mention that such a phenomenon is partially due to the adaptive selection of the inner tolerance  $\epsilon_{inn}^{(j)}$  coming from [38].

We now compare LR-ADI-EKSM(G) with the function `mess_lradi` of the M-M.E.S.S. package [59], an abstract function handle-based implementation of the LR-ADI, and `lradi`, a plain matrix-based implementation of the same algorithm.

To this end, we make  $B \in \mathbb{R}^n$  the normalized vector of all ones. For having fair comparisons, we employ the shifts computed by the LR-ADI-EKSM(G) in all the different implementations. This leads to a very similar trend in the relative residual norm achieved by the routines even though the shifted linear systems in `mess_lradi` and





**Fig. 1** Experiment 1. Dimension of the constructed extended Krylov subspace and computed normalized residual norms as  $j$  grows, i.e., the ADI progresses, for problem size  $n=360\,000$

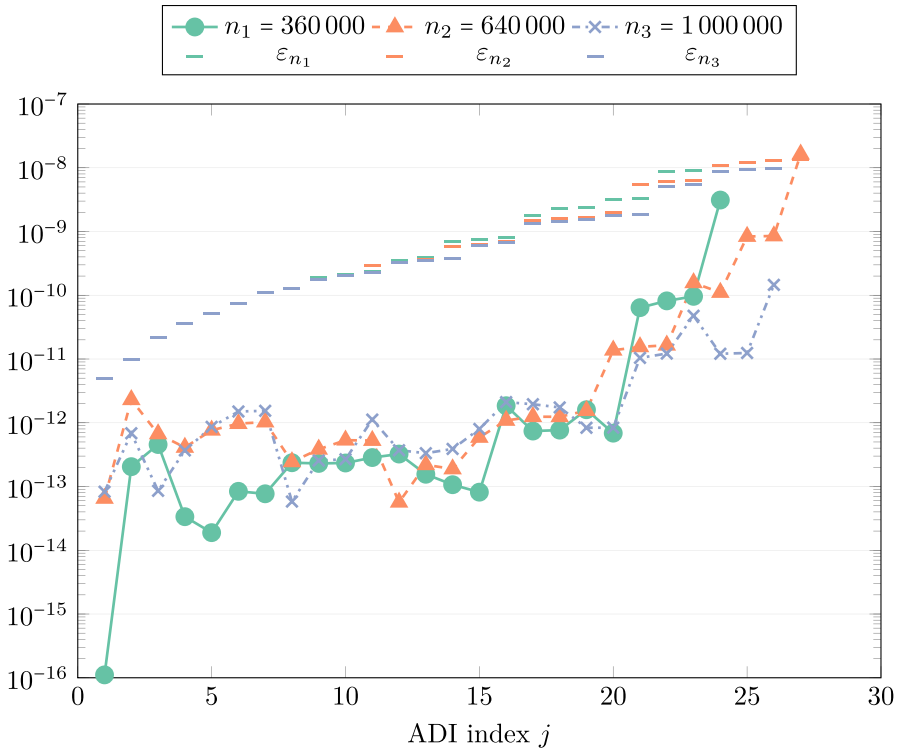
`lradi` are solved at very high accuracy<sup>6</sup>, whereas the relaxation strategy of [38] is implemented in LR-ADI-EKSM(G). In Fig. 2, we report the relative difference between the relative residual norms computed by LR-ADI-EKSM(G) and `mess_lradi` throughout all the necessary iterations  $j$  for different problem dimension  $n$  along with the values of  $\epsilon_{\text{inn}}^{(j)}$  we employed. In agreement with the results presented in [38], we can notice that the distance between the computed relative residual norms is always rather moderate and smaller than  $\epsilon_{\text{inn}}^{(j)}$ <sup>7</sup>. Very similar results are obtained by comparing the residual norms attained by `lradi` in place of `mess_lradi`.

We also compare the routines in terms of computation time. The results are collected in Table 3. Since we employ the shifts computed within LR-ADI-EKSM(G) also for `mess_lradi` and `lradi`, we do not consider the time devoted to the shift computation when reporting the performances of LR-ADI-EKSM(G) in Table 3.

The results in Table 3 show that, for this experiment, our proposed scheme combined with the relaxation strategy presented in [38] leads to a remarkable speed-up of the solution process — up to 50% — when compared to a standard implementation of the LR-ADI method.

<sup>6</sup> The MATLAB sparse direct solver “backslash” is employed for solving  $(A + p_j I)S_j = W_{j-1}$  for all  $j$ .

<sup>7</sup> This is true for all the experiments we ran except for  $n=640\,000$ , at the very last iteration where  $r_{27}^{\text{LR-ADI-EKSM}} \approx 1.3 \times 10^{-8}$  whereas  $\epsilon_{\text{inn}}^{(27)} \approx 1.6 \times 10^{-8}$



**Fig. 2** Experiment 1. Relative gap  $\frac{|r_j^{LR-ADI-EKSM} - r_j^{mess\_lradi}|}{r_j^{mess\_lradi}}$  between the residual norms  $r_j^{LR-ADI-EKSM}$  and  $r_j^{mess\_lradi}$  computed by LR-ADI-EKSM(G) and `mess_lradi`, respectively, as  $j$  grows, i.e., ADI converges, and different problem sizes  $n$ , together with the corresponding inner inexact solver tolerance  $\epsilon^{(j)}$ , denoted  $\epsilon_n$  to relate the problem sizes

**Table 3** Experiment 1. Computational timings achieved by LR-ADI-EKSM(G), `lradi`, and `mess_lradi` for different problem sizes  $n$

$n$	It.	LR-ADI-EKSM(G)	<code>lradi</code>	<code>mess_lradi</code>
		Time (s)	Time (s)	Time (s)
360 000	24	10.08	31.84	30.94
640 000	27	19.99	65.84	66.22
1 000 000	36	39.18	101.18	101.61

It. indicates the number of ADI iterations that have been (implicitly) performed

**Experiment 2** In the second experiment, we consider a problem similar to [51, Example 6]. In particular, the matrix  $A$  comes from the centered finite difference discretization of the 3-dimensional convection-diffusion operator

$$\mathcal{L}(u) = -\zeta \Delta u + \mathbf{w} \cdot \nabla u,$$

on the unit cube with zero Dirichlet boundary conditions. The convection vector  $\mathbf{w}$  is given by  $\mathbf{w} = (\phi_1(x)\psi_1(y)\pi_1(z), 0, \pi_3(z)) = ((1 - x^2)yz, 0, e^z)$  whereas  $\zeta > 0$ . By employing  $h$  nodes in each direction, the discretization phase leads to a matrix  $A$  that can be written as

$$A = (D_h + \Pi_3 N^T) \otimes I_h \otimes I_h + I_h \otimes D_h \otimes I_h + I_h \otimes I_h \otimes D_h + \Pi_1 \otimes \Psi_1 \otimes \Phi_1 N,$$

where  $D_h = \zeta(h - 1)^2 \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{h \times h}$ ,  $N = -\frac{(h-1)}{2} \cdot \text{tridiag}(-1, 0, 1) \in \mathbb{R}^{h \times h}$ , and  $\Phi_i$ ,  $\Psi_i$ , and  $\pi_i$  are diagonal matrices whose diagonal entries correspond to the nodal values of the corresponding functions  $\phi_i$ ,  $\psi_i$ , and  $\pi_i$  (see [51] for further details).  $B \in \mathbb{R}^n$ ,  $n = h^3$ , is a vector with random entries.

Due to the 3D nature of the problem, the nonsymmetric linear systems with  $A$  involved in the basis construction in LR-ADI-EKSM are solved by GMRES [57]. In particular, we employ the GMRES implementation written by Lund et al [35], namely the function `bgmres` in [45]. GMRES is stopped whenever the computed relative residual norm gets smaller than  $10^{-10}$ .

It is well known that (polynomial) Krylov methods for linear systems need to be preconditioned to achieve a fast convergence in terms of number of iterations. To this end, as suggested in [51], we employ the following preconditioning operator when solving the linear systems with  $A$ ,

$$\mathcal{P} = (D_h + \Pi_3 N^T) \otimes I_h \otimes I_h + I_h \otimes D_h \otimes I_h + I_h \otimes I_h \otimes D_h + \bar{\pi}_1 I_h \otimes \Psi_1 \otimes \Phi_1 N,$$

where  $\bar{\pi}_1$  is the mean value of the function  $\pi_1$  in  $[0,1]$ . At each GMRES iteration, we thus have to invert  $\mathcal{P}$ , namely we have to compute  $\bar{v} = \mathcal{P}^{-1}v$  for  $v \in \mathbb{R}^n$ . This operation is performed by solving the Sylvester equation

$$(D_h \otimes I_h + I_h \otimes D_h + \bar{\pi}_1 \Psi_1 \otimes \Phi_1 N) \bar{\mathbf{V}} + \bar{\mathbf{V}}(D_h + \Pi_3 N^T)^T = \mathbf{V},$$

where  $\bar{\mathbf{V}}, \mathbf{V} \in \mathbb{R}^{h^2 \times h}$  are such that  $\text{vec}(\bar{\mathbf{V}}) = \bar{v}$  and  $\text{vec}(\mathbf{V}) = v$ . Since the coefficient matrices in the equation above have moderate dimensions, the Bartels-Stewart method [6] is employed for its solution and the Schur decompositions of the coefficient matrices are computed once and for all before the iterative procedure starts. We always employ a right preconditioning scheme in order to easily have access to the actual residual norm.

Also, for the shifted linear systems with  $A + p_j I$ , within `mess_lradi` and `lradi`, we employ preconditioned GMRES equipped with the preconditioning operator  $\mathcal{P} + p_j I$ . Once again, this preconditioner is applied by solving the Sylvester equation

$$(D_h \otimes I_h + I_h \otimes D_h + \bar{\pi}_1 \Psi_1 \otimes \Phi_1 N) \bar{\mathbf{V}} + \bar{\mathbf{V}}(D_h + \Pi_3 N^T + p_j I_h)^T = \mathbf{V}.$$

Even though this is in general a better preconditioner for  $A + p_j I$  compared to  $\mathcal{P}$ , its application involves complex arithmetic whenever  $\text{Im}(p_j) \neq 0$  with a consequent increment in the computational efforts devoted to the preconditioning step.

For this experiment, `lradi` is equipped with the relaxation strategy presented in [38].

**Table 4** Experiment 2. Computational timings achieved by LR-ADI-EKSM(MR), `lradi`, and `mess_lradi` for different problem sizes  $n$  and diffusivities  $\zeta$

$\zeta$	$n$	It.	$\#\{p_j \notin \mathbb{R}\}$	LR-ADI-EKSM(MR) Time (s)	<code>lradi</code> Time (s)	<code>mess_lradi</code> Time (s)
0.05	125 000	20	12	80.61	153.28	187.52
	512 000	20	12	812.74	1 342.27	2 161.99
	1 000 000	22	10	3 183.54	5 764.21	6 133.54
0.005	125 000	45	44	65.23	361.58	382.11
	512 000	61	60	419.45	2 156.94	3 497.51
	1 000 000	67	62	1 194.37	5 802.41	10 517.13

It. indicates the number of ADI iterations that have been (implicitly) performed

Also for this experiment, LR-ADI-EKSM(G) and LR-ADI-EKSM(MR) perform very similarly, with LR-ADI-EKSM(MR) achieving slightly better results in terms of computational time. We thus report only the performance of LR-ADI-EKSM(MR).

The results are collected in Table 4 for different values of  $n$  and  $\zeta$ . In Table 4, we also report the number of shifts with nonzero imaginary part.

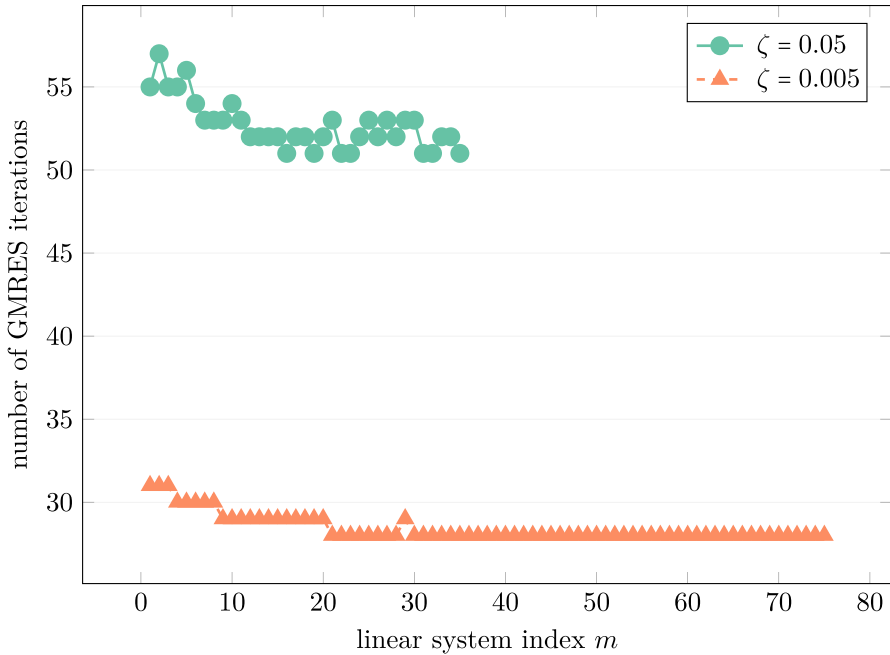
We would like to mention that we ran some experiments with `mess_lradi` where the shifted linear systems were solved by means of the MATLAB sparse direct solver “backslash” in place of preconditioned GMRES. However, for this example, the potentially higher accuracy of the direct solves did not benefit the computation and the execution times we achieved with “backslash” could not keep up with the ones reported for GMRES in Table 4. We, thus, decided to omit them here.

From the results in Table 4, we can see that LR-ADI-EKSM(MR) is very competitive and always achieves computational timings that are significantly smaller than the ones required by `mess_lradi`. Thanks to the relaxation procedure coming from [38], `lradi` performs better than `mess_lradi`.

The performance of all the tested routines is strictly related to the number of complex shifts needed to converge. When this is sizable with respect to the total number of iterations, many of the  $n \times n$  linear systems  $A + p_j I$  within `mess_lradi` and `lradi` involve complex arithmetic, whereas this is needed only in the solution of the small dimensional least squares problem for the computation of  $Y$  in LR-ADI-EKSM(MR).

We notice that, for a fixed  $n$ , the computational time of LR-ADI-EKSM(MR) decreases, in general, by reducing  $\zeta$ , even though the number of LR-ADI iterations that are implicitly performed increases. This is due to the computational efforts required by the solution of the linear systems with  $A$  during the basis construction. Indeed, for  $\zeta = 0.05$ , many more GMRES iterations are required than what is necessary for  $\zeta = 0.005$ . In Fig. 3, we report the number of GMRES iterations needed to solve the linear system with  $A$  at each  $m$ , namely every time a new basis vector of the adopted extended Krylov subspace needs to be computed.

A rather large number of GMRES iterations are required for solving the linear systems with  $A$  in case of  $\zeta = 0.05$  making the construction of the basis of  $\mathbf{EK}_m^\square(A, B)$  more



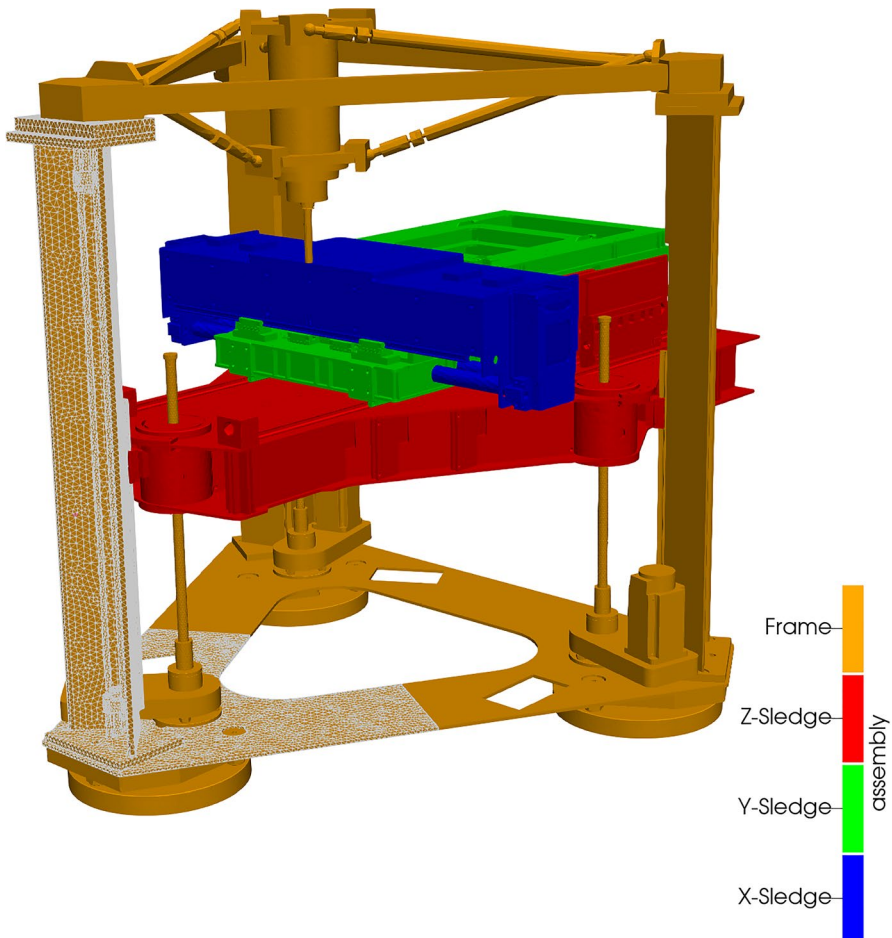
**Fig. 3** Experiment 2. Number of GMRES iterations needed to solve the linear systems with  $A$  during the basis construction in LR-ADI-EKSM(MR) for different values of the diffusivity  $\zeta$  and  $n = 125\,000$

demanding. On the other hand, few GMRES iterations are sufficient to meet the prescribed accuracy for  $\zeta = 0.005$  and the overall solution procedure turns out to be very successful.

**Experiment 3** In this experiment, we compare LR-ADI-EKSM also with K-PIK [63], since the two routines construct the same subspace<sup>8</sup>. We consider the thermal part of the thermo-elastic modeling of a building-block of an experimental machine tool given by the following heat equation

$$\left\{ \begin{array}{l} c_p \rho \frac{\partial T}{\partial t} = \lambda \Delta T, \quad \text{in } \Omega, \\ \lambda \frac{\partial T}{\partial \mathbf{n}} = f, \quad \text{on } \Gamma_c \subset \partial \Omega, \\ \lambda \frac{\partial T}{\partial \mathbf{n}} = \alpha(T_{ext} - T), \quad \text{on } \Gamma_{ext} \subset \partial \Omega, \\ T(0) = 0. \end{array} \right. \quad (23)$$

<sup>8</sup> The implementation of K-PIK we employed will be available in the next M-M.E.S.S. release, along with other projection methods for matrix equations. Such implementation is equivalent to the one that can be found on Simoncini’s webpage, <http://www.dm.unibo.it/~simoncin/software.html>.



**Fig. 4** Experiment 2. Finite element grid of the machine frame indicated on the CAD model of the full machine. (Source: DFG CRC/TR-96 (<https://transregio96.de>))

The discretization in space using the finite element method (here applying the proprietary tool ANSYS<sup>9</sup>) on the three-dimensional domain, given by the machine frame indicated in Fig. 4, leads to the LTI system

$$E\dot{T} = \left( A - \sum_{i=1}^t \alpha_i F_i \right) T + Bu(t). \quad (24)$$

where  $A$  represents the discretized Laplacian together with the Robin boundary contributions from  $\Gamma_{ext}$  and represented by  $F_i$ , while  $B$  results from the external control inputs (heats fluxes, e.g., induced by the drive motors) on  $\Gamma_c$ . Note that the elastic

<sup>9</sup> <https://www.ansys.com/>

**Table 5** Experiment 3. Computational timings achieved by LR-ADI-EKSM(G), K-PIK, and `mess_lradi` for different values of problem size  $n$ , number of Robin boundary conditions  $t$ , and rank of the right-hand side  $q$

$n$	$t$	$q$	LR-ADI-EKSM(G)			<code>mess_lradi</code>		K-PIK		
			It.	$\dim(\mathbf{EK}^\square)$	Time (s)	It.	Time (s)	It.	$\dim(\mathbf{EK}^\square)$	Time (s)
4 813	1	23	54	644	2.73	54	7.93	15	736	8.13
13 551	2	5	53	430	8.89	53	18.09	43	440	24.93
25 872	1	10	63	1060	34.32	63	63.37	53	1080	97.74

It. indicates the number of ADI/K-PIK iterations that have been (implicitly) performed

part of the thermo-elastic model can be encoded entirely in the output equation of the corresponding dynamical system and is, thus, not relevant here [40]. The algebraic problem resulting from this system amounts to a Lyapunov equation of the form (1). However, due to mass lumping in ANSYS, the mass matrix  $E$  is diagonal and SPD. We can, thus, easily invert its square root and consider the Lyapunov equation

$$E^{-\frac{1}{2}} \left( A - \sum_{i=1}^t \alpha_i F_i \right) E^{-\frac{1}{2}} \tilde{X} + \tilde{X} E^{-\frac{1}{2}} \left( A - \sum_{i=1}^t \alpha_i F_i \right)^T E^{-\frac{1}{2}} + E^{-\frac{1}{2}} B B^T E^{-\frac{1}{2}} = 0, \quad \tilde{X} = (E^{\frac{1}{2}} X E^{\frac{1}{2}}).$$

So, again, we can efficiently retract to a problem of the form (2). Once a low-rank approximation  $\tilde{Z}\tilde{Z}^T$  to  $\tilde{X}$  is computed, the low-rank factor  $Z$  such that  $ZZ^T \approx X$  can be retrieved by performing  $Z = E^{-\frac{1}{2}} \tilde{Z}$ .

The actual machine frame in Fig. 4 consists of several parts itself, which are discretized separately. This leads to differently sized models of the structure in (24). These are reflected by the rows of Table 5. Accordingly, we solve the Lyapunov equation considering different configurations of the PDE (23), respectively the LTI system in (24). In particular, this allows us to vary the number of degrees of freedom employed in the discretization phase, leading to different problem dimensions  $n$ , modify the Neumann boundary conditions obtaining diverse matrices  $F_i$ , and consider different values for the rank  $q$  of  $B$ . Moreover, we set  $\alpha_i = 10$  for all  $i = 1, \dots, t$ .

The results are collected in Table 5. It turns out that the Wachspress ADI shifts [42, 73] are particularly effective for this experiment, since  $A$  as well as all the  $F_i$  and thus  $E^{-\frac{1}{2}} (A - \sum_{i=1}^t \alpha_i F_i) E^{-\frac{1}{2}}$  are symmetric, i.e., the spectrum is real. These are the ideal circumstances for Wachspress shifts. We, thus, employ those shifts in LR-ADI-EKSM(G) and `mess_lradi`.

For this experiment, the LR-ADI method, either based on our new formulation or on a standard scheme as the one in `mess_lradi`, turns out to be more efficient in terms of computational time than K-PIK. Indeed, in spite of the smaller number of iterations needed to converge, the large dimension of the extended Krylov subspace constructed by K-PIK leads to a rather costly solution of the projected equations. Also LR-ADI-EKSM(G) requires the construction of an extended Krylov subspace whose dimension is similar to the one computed by K-PIK. However, if  $\dim(\mathbf{EK}_m^\square(E^{-\frac{1}{2}}(A - \sum_{i=1}^t \alpha_i F_i)E^{-\frac{1}{2}}, E^{-\frac{1}{2}}B)) = 2mq$ , the computational cost of solv-

**Table 6** Experiment 4. Computational timings achieved by LR-ADI-EKSM(G) and `mess_lradi` for different values of the problem size  $n$

$n$	LR-ADI-EKSM(G)				<code>mess_lradi</code>		
	It.	$\dim(\mathbf{EK}^\square)$	$\text{rank}(X)$	Time (s)	It.	$\text{rank}(X)$	Time (s)
20 209	30	564	180	7.07	30	180	0.54
79 841	31	816	186	34.09	31	186	2.99

The running time devoted to the shift computation is not included. It. indicates the number of ADI iterations that have been (implicitly) performed

ing the inner problems within LR-ADI-EKSM(G) is  $\mathcal{O}(4m^2q^2)$  floating-point operations (FLOPs) whereas it amounts to  $\mathcal{O}(8m^3q^3)$  FLOPs for K-PIK.

We conclude by mentioning that in this experiment, we relied on the ease of computing  $E^{-\frac{1}{2}}$ . However, it may happen that the mass matrix  $E$  cannot be easily manipulated, e.g., it can be possibly singular, so that the routine presented in this paper cannot be readily applied as we have done in this experiment. We plan to extend the LR-ADI-EKSM framework to this more challenging class of equations in the near future.

**Experiment 4** In the last experiment, we show that the proposed framework still needs some further improvements to efficiently deal with generalized Lyapunov equations of the form (1) where the mass matrix  $E$  is not diagonal. To this end, we consider the *Steel Profile* data set [18, 49] from the MORwiki repository [72].

We compute the observability Gramian of the system, namely the solution  $X$  to the equation

$$A^T X E + E^T X A + C^T C = 0, \tag{25}$$

where  $A \in \mathbb{R}^{n \times n}$  is symmetric negative definite,  $C \in \mathbb{R}^{q \times n}$ ,  $q = 6$ , and  $E \in \mathbb{R}^{n \times n}$  is SPD but not diagonal (see [17] for further details on the model).

If  $E = LL^T$  denotes the Cholesky factorization of  $E$ , we consider the transformed equation

$$(L^{-1} A^T L^{-T}) \tilde{X} + \tilde{X} (L^{-1} A L^{-T}) + L^{-1} C^T C L^{-T} = 0, \quad \tilde{X} = L^T X L, \tag{26}$$

and, due to symmetry of  $A$ , employ the extended Krylov subspace  $\mathbf{EK}_m^\square(L^{-1} A L^{-T}, L^{-1} C^T)$  as approximation space. Notice that the matrix  $L^{-1} A L^{-T}$  does not need to be explicitly constructed (see, e.g., [63, Example 5.4]). As before, once  $\tilde{Z}\tilde{Z}^T \approx \tilde{X}$  is computed, we obtain a low-rank approximation to the original  $X$  by performing  $Z = L^{-T} \tilde{Z}$ .

In Table 6, we report the results achieved by LR-ADI-EKSM(G) and `mess_lradi` for different values of  $n$ .

From the results in Table 6, we can readily see that the standard scheme of the LR-ADI method implemented in `mess_lradi` is much faster than LR-ADI-EKSM(G). This is due to the fact that the latter algorithm needs to construct a quite



large subspace to achieve the prescribed accuracy with a consequent increment in the computational efforts of the overall procedure.

We also mention that the rank of the approximate solution computed by LR-ADI-EKSM(G) is much lower than the dimension of the constructed subspace. We believe that the transformation we performed in (26), and thus the employment of  $\mathbf{EK}_m^\square(L^{-1}AL^{-T}, L^{-1}C^T)$ , may lead to some spectral redundancy in the adopted approximation subspace and a slower convergence of the method. On the other hand, `mess_lradi` is able to deal with the original formulation (25) of the problem.

To address generalized equations of the form (1), we plan to study the employment of different techniques within the Krylov LR-ADI framework we presented in this paper. In particular, the use of nonstandard inner products and (extended) generalized Krylov subspaces [43] will be explored.

## 6 Conclusions

A new formulation of the LR-ADI algorithm for large-scale standard Lyapunov equations has been proposed. The computational core of the LR-ADI scheme consists in the solution of a shifted linear system at each iteration. We showed that the extended Krylov subspace method can be a valid candidate for this task. In particular, we described how only one extended Krylov subspace needs to be constructed to solve all the necessary linear systems required by the LR-ADI method. The LR-ADI iteration has been completely merged into the extended Krylov subspace method for shifted linear systems resulting in a novel, efficient solution procedure. We also showed that many state-of-the-art algorithms for the shift computation can be easily integrated into our new scheme. Numerical results demonstrate the potential of our novel algorithm, especially when this is equipped with the relaxation strategy proposed in [38], and many complex shifts are needed to converge.

In future work, we will consider more involved Lyapunov equations of the form (1) that cannot be easily transformed into (2). While standard implementations of the LR-ADI method naturally address such a scenario by solving linear systems of the form  $A + p_j E$ , further care has to be taken to employ the scheme we presented in this paper. Indeed, the shifted Arnoldi relation (7) can no longer be exploited. The use of non-standard inner products and generalized Krylov subspace methods [43] will be investigated.

The framework presented in this paper can be generalized to enhance other LR-ADI-like algorithms for matrix equations. For instance, the LR-ADI method for Sylvester equations [14] or LR-RADI schemes for Riccati equations [10, 22] can be equipped with a procedure similar to the one we proposed here.

**Acknowledgements** The second author is member of the Italian INdAM Research group GNCS. The work presented in this paper has been carried out when the second author was affiliated with the Research Group Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany.

**Funding** Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

**Data availability** The datasets generated during and/or analysed during the current study are available in the Zenodo repository, <https://zenodo.org/record/7101150#.YzLZzNJBWxQ>.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ahuja, K., Benner, P., de Sturler, E., Feng, L.: Recycling BiCGSTAB with an application to parametric model order reduction. *SIAM J. Sci. Comput* **37**, S429–S446 (2015). <https://doi.org/10.1137/140972433>
2. Ahuja, K., de Sturler, E., Gugercin, S., Chang, E.R.: Recycling BiCG with an application to model reduction. *SIAM J. Sci. Comput* **34**, A1925–A1949 (2012). <https://doi.org/10.1137/100801500>
3. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems vol. 6 of Adv. Des. Control. SIAM Publications, Philadelphia (2005). <https://doi.org/10.1137/1.9780898718713>
4. Anzt, H., Chow, E., Saak, J., Dongarra, J.: Updating incomplete factorization preconditioners for model order reduction. *Numer. Algorithms* **73**, 611–630 (2016). <https://doi.org/10.1007/s11075-016-0110-2>
5. Baker, J., Embree, M., Sabino, J.: Fast singular value decay for Lyapunov solutions with nonnormal coefficients. *SIAM J. Matrix Anal. Appl* **36**, 656–668 (2015). <https://doi.org/10.1137/140993867>
6. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation  $AX + XB = C$ : Algorithm 432. *Comm. ACM* **15**, 820–826 (1972). <https://doi.org/10.1145/361573.361582>
7. Baumann, M., van Gijzen, M.B.: Nested Krylov methods for shifted linear systems. *SIAM J. Sci. Comput* **37**, S90–S112 (2015). <https://doi.org/10.1137/140979927>
8. Baumann, M., van Gijzen, M.B.: Efficient Iterative Methods for Multi-Frequency Wave Propagation Problems: A Comparison Study. *Procedia Computer Science*, vol. 108, pp. 645–654. In: International Conference on Computational Science, ICCS 2017, 12–14 June 2017, Zurich, Switzerland. <https://doi.org/10.1016/j.procs.2017.05.088> (2017)
9. Bellavia, S., De Simone, V., di Serafino, D., Morini, B.: Efficient preconditioner updates for shifted linear systems. *SIAM J. Sci. Comput* **33**, 1785–1809 (2011). <https://doi.org/10.1137/100803419>
10. Benner, P., Bujanović, Z., Kürschner, P., Saak, J.: RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations. *Numer. Math* **138**, 301–330 (2018). <https://doi.org/10.1007/s00211-017-0907-5>
11. Benner, P., Kürschner, P., Saak, J.: Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method. *Numer. Algorithms* **62**, 225–251 (2013). <https://doi.org/10.1007/s11075-012-9569-7>
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). <http://etna.mcs.kent.edu/volumes/2011-2020/vol43/abstract.php?vol=43&pages=142-162>
13. Benner, P., Kürschner, P., Saak, J.: Frequency-limited balanced truncation with low-rank approximations. *SIAM J. Sci. Comput* **38**, A471–A499 (2016). <https://doi.org/10.1137/15M1030911>

14. Benner, P., Li, R.-C., Truhar, N.: On the ADI method for Sylvester equations. *J. Comput. Appl. Math* **233**, 1035–1045 (2009). <https://doi.org/10.1016/j.cam.2009.08.108>
15. Benner, P., Mehrmann, V., Sorensen, D. C.: *Dimension Reduction of Large-Scale Systems*, vol. 45 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, Berlin (2005). <https://doi.org/10.1007/3-540-27909-1>
16. Benner, P., Mena, H., Saak, J.: On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations. *Electron. Trans. Numer. Anal* **29**, 136–149 (2008). <https://etna.math.kent.edu/volumes/2001-2010/vol29/abstract.php?vol=29&pages=136-149>
17. Benner, P., Saak, J.: Linear-quadratic regulator design for optimal cooling of steel profiles, Tech. Report SFB393/05-05, Sonderforschungsbereich 393 Parallele Numerische Simulation für Physik und Kontinuumsmechanik, TU Chemnitz, D-09107 Chemnitz (Germany). <http://nbn-resolving.de/urn:nbn:de:swb:ch1-200601597> (2005)
18. Benner, P., Saak, J.: A semi-discretized heat transfer model for optimal cooling of steel profiles. In: Benner, P., Mehrmann, V., Sorensen, D. (eds.) *Dimension Reduction of Large-Scale Systems* vol. 45 of *Lect. Notes Comput. Sci. Eng.*, Springer-Verlag, Berlin/Heidelberg, Germany, pp. 353–356. [https://doi.org/10.1007/3-540-27909-1\\_19](https://doi.org/10.1007/3-540-27909-1_19) (2005)
19. 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**, 32–52 (2013). <https://doi.org/10.1002/gamm.201310003>
20. Benzi, M., Bertaccini, D.: Approximate inverse preconditioning for shifted linear systems. *BIT* **43**, 231–244 (2003). <https://doi.org/10.1023/A:1026089811044>
21. Bertaccini, D.: Efficient preconditioning for sequences of parametric complex symmetric linear systems. *Electron. Trans. Numer. Anal* **18**, 49–64 (2004). <https://etna.math.kent.edu/volumes/2001-2010/vol18/abstract.php?vol=18&pages=49-64>
22. Bertram, C., Faßbender, H.: Riccati ADI: existence, uniqueness and new iterative methods, math. NA. arXiv:2004.11212 (2020)
23. Datta, B.N., Saad, Y.: Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment. *Linear Algebra. Appl.* **154–156**, 225–244 (1991). [https://doi.org/10.1016/0024-3795\(91\)90378-A](https://doi.org/10.1016/0024-3795(91)90378-A)
24. Druskin, V., Knizhnerman, L.: Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl* **19**, 755–771 (1998). <https://doi.org/10.1137/S0895479895292400>
25. Druskin, V., Knizhnerman, L., Simoncini, V.: Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM J. Numer. Anal* **49**, 1875–1898 (2011). <https://doi.org/10.1137/100813257>
26. Feng, L., Benner, P., Korvink, J.G.: Subspace recycling accelerates the parametric macromodeling of MEMS. *Internat. J. Numer. Methods Engrg* **94**, 84–110 (2013). <https://doi.org/10.1002/nme.4449>
27. Feriani, A., Perotti, F., Simoncini, V.: Iterative system solvers for the frequency analysis of linear mechanical systems. *Comp. Meth. Appl. Mech. Eng* **190**, 1719–1739 (2000). [https://doi.org/10.1016/S0045-7825\(00\)00187-0](https://doi.org/10.1016/S0045-7825(00)00187-0)
28. Freund, R.W.: Solution of shifted linear systems by quasi-minimal residual iterations. In: *Numerical Linear Algebra*, De Gruyter, pp. 101–122. (1993)
29. Frommer, A., Glässner, U.: Restarted GMRES for shifted linear systems. *SIAM J. Sci. Comput* **19**, 15–26 (1998). <https://doi.org/10.1137/S1064827596304563>
30. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices. *Electron. Trans. Numer. Anal* **47**, 100–126 (2017). <https://etna.math.kent.edu/volumes/2011-2020/vol47/abstract.php?vol=47&pages=100-126>
31. Gaul, A.: *Recycling Krylov Subspace Methods for Sequences of Linear Systems – Analysis and Applications*. PhD thesis Technische Universität, Berlin (2014). <https://doi.org/10.13140/2.1.4015.3284>
32. Glässner, U., Gösken, S., Lippert, T., Ritzenhöfer, G., Schilling, K., Frommer, A.: How to compute Green’s functions for entire mass trajectories within Krylov solvers. *Int. J. Mod. Phys. C* **7**, 635–644 (1996). <https://doi.org/10.1142/S0129183196000533>
33. Grasedyck, L.: Existence of a low rank or  $H$ -matrix approximant to the solution of a Sylvester equation. *Numer. Lin. Alg. Appl* **11**, 371–389 (2004). <https://doi.org/10.1002/nla.366>
34. Kalman, R.E., Falb, P.L., Arbib, M.A.: *Topics in Mathematical System Theory*. McGraw-Hill, New York (1969)

35. Kressner, D, Lund, K, Massei, S, Palitta, D: Compress-and-restart block Krylov subspace methods for Sylvester matrix equations, *Numer. Lin. Alg. Appl.*, vol. **28**. (2021) <https://doi.org/10.1002/nla.2339>
36. Kürschner, P.: Efficient Low-Rank Solution of Large-Scale Matrix Equations, PhD thesis, Otto-von-Guericke-Universität, Magdeburg. Shaker Verlag, ISBN, Germany (2016). 978-3-8440-4385-3. <http://hdl.handle.net/11858/00-001M-0000-0029-CE18-2>
37. Kürschner, P.: Approximate residual-minimizing shift parameters for the low-rank ADI iteration. *Electron. Trans. Numer. Anal* **51**, 240–261 (2019). [https://doi.org/10.1553/etna\\_vol51s240](https://doi.org/10.1553/etna_vol51s240)
38. Kürschner, P., Freitag, M.: Inexact methods for the low rank solution to large scale Lyapunov equations. *BIT* **60**, 1221–1259 (2020). <https://doi.org/10.1007/s10543-020-00813-4>
39. Lang, N., Mena, H., Saak, J.: On the benefits of the  $LDL^T$  factorization for large-scale differential matrix equation solvers. *Linear Algebra Appl.* **480**, 44–71 (2015). <https://doi.org/10.1016/j.laa.2015.04.006>
40. Lang, N., Saak, J., Benner, P.: Model order reduction for systems with moving loads. *at-Automatisierungstechnik* **62**, 512–522 (2014). <https://doi.org/10.1515/auto-2014-1095>
41. Laub, A.J.: Numerical linear algebra aspects of control design computations. *IEEE Trans. Autom. Control* **30**, 97–108 (1985). <https://doi.org/10.1109/TAC.1985.1103900>
42. Li, J.-R., Rebecca, J.: White, low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.* **24**, 260–280 (2002). <https://doi.org/10.1137/S0895479801384937>
43. Li, R.-C., Ye, Q.: A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM J. Matrix Anal. Appl.* **25**, 405–428 (2003). <https://doi.org/10.1137/S0895479802409390>
44. Liu, Z., Zhou, Y., Zhang, Y.: On inexact alternating direction implicit iteration for continuous Sylvester equations. *Numer. Lin. Alg. Appl.* **27**, e2320 (2020). <https://doi.org/10.1002/nla.2320>
45. Lund, K., Massei, S., Palitta, D.: CRKSM\_MatEQ (2020). <https://gitlab.com/katlund/compress-and-restart-KSM>
46. Luo, W.-H., Huang, T.-Z., Li, L., Zhang, Y., Gu, X.-M.: Efficient preconditioner updates for unsymmetric shifted linear systems. *Comput. Math. Appl.* **67**, 1643–1655 (2014). <https://doi.org/10.1016/j.camwa.2014.03.005>
47. The MathWorks, Inc.: MATLAB (2022). <https://www.matlab.com>
48. Meerbergen, K.: The solution of parametrized symmetric linear systems. *SIAM J. Matrix Anal. Appl.* **24**, 1038–1059 (2003). <https://doi.org/10.1137/S0895479800380386>
49. Oberwolfach benchmark collection: Steel profile. hosted at MORwiki – Model Order Reduction Wiki (2005), [http://modelreduction.org/index.php/Steel\\_Profile](http://modelreduction.org/index.php/Steel_Profile)
50. O’Leary, D.P.: The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.* **29**, 293–322 (1980). [https://doi.org/10.1016/0024-3795\(80\)90247-5](https://doi.org/10.1016/0024-3795(80)90247-5)
51. Palitta, D., Simoncini, V.: Matrix-equation-based strategies for convection-diffusion equations. *BIT* **56**, 751–776 (2016). <https://doi.org/10.1007/s10543-015-0575-8>
52. Parks, M.L., De Sturler, E., Mackey, G., Johnson, D.D., Maiti, S.: Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Sci. Comput.* **28**, 1651–1674 (2006). <https://doi.org/10.1137/040607277>
53. Penzl, T.: Numerical solution of generalized Lyapunov equations. *Adv. Comp. Math.* **8**, 33–48 (1997). <https://doi.org/10.1023/A:1018979826766>
54. Penzl, T.: A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.* **21**, 1401–1418 (2000). <https://doi.org/10.1137/S1064827598347666>
55. Penzl, T.: Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Syst. Control Lett.* **40**, 139–144 (2000). [https://doi.org/10.1016/S0167-6911\(00\)00010-4](https://doi.org/10.1016/S0167-6911(00)00010-4)
56. Saad, Y.: Iterative Methods for Sparse Linear Systems. USA, Second edn., SIAM Philadelphia (2003). <https://doi.org/10.1137/1.9780898718003>
57. Saad, Y., Schultz, M.H.: GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* **7**, 856–869 (1986). <https://doi.org/10.1137/0907058>
58. Saak, J.: Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction (2009), <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642>. PhD thesis, Technische Universität Chemnitz, Germany
59. Saak, J., Köhler, M., Benner, P.: M-M.E.S.S.-2.2 – the Matrix Equations Sparse Solvers library (2022). <https://doi.org/10.5281/zenodo.5938237>. See also: <https://www.mpi-magdeburg.mpg.de/projects/mess>

60. Sabino, J.: Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method (2007). [http://www.caam.rice.edu/tech\\_reports/2006/TR06-08.pdf](http://www.caam.rice.edu/tech_reports/2006/TR06-08.pdf). PhD thesis, Rice University, Texas
61. Schmelzer, T.: Block Krylov Methods for Hermitian Linear Systems. University of Kaiserslautern, PhD thesis (2004)
62. Simoncini, V.: Restarted full orthogonalization method for shifted linear systems. *BIT* **43**, 459–466 (2003). <https://doi.org/10.1023/A:1026000105893>
63. Simoncini, V.: A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.* **29**, 1268–1288 (2007). <https://doi.org/10.1137/06066120X>
64. Simoncini, V.: Extended Krylov subspace for parameter dependent systems. *Appl. Numer. Math.* **60**, 550–560 (2010). <https://doi.org/10.1016/j.apnum.2010.03.001>
65. Simoncini, V.: Computational methods for linear matrix equations. *SIAM Rev.* **38**, 377–441 (2016). <https://doi.org/10.1137/130912839>
66. Sontag, E.D.: *Mathematical Control Theory*, Texts in Applied Mathematics. 2nd edn., (1998) <https://doi.org/10.1007/978-1-4612-0577-7>. Springer, New York
67. Soodhalter, K.M.: Block Krylov subspace recycling for shifted systems with unrelated right-hand sides. *SIAM J. Sci. Comput.* **38**, A302–A324 (2016). <https://doi.org/10.1137/140998214>
68. Soodhalter, K.M.: Two recursive GMRES-type methods for shifted linear systems with general preconditioning. *Electron. Trans. Numer. Anal.* **45**, 499–523 (2016). <https://etna.math.kent.edu/volumes/2011-2020/vol45/abstract.php?vol=45&pages=499-523>
69. Soodhalter, K.M., De Sturler, E., Kilmer, M.: A survey of subspace recycling iterative methods, *GAMM Mitteilungen*, vol. **43** (2020). <https://doi.org/10.1002/gamm.202000016>
70. Soodhalter, K.M., Szyld, D.B., Xue, F.: Krylov subspace recycling for sequences of shifted linear systems. *Appl. Numer. Math.* **81**, 105–118 (2014). <https://doi.org/10.1016/j.apnum.2014.02.006>
71. Starke, G.: Optimal alternating directions implicit parameters for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.* **28**, 1431–1445 (1991). <https://doi.org/10.1137/0728074>
72. The MORwiki Community: MORwiki - model order reduction wiki (2022). <http://modelreduction.org>
73. Wachspress, E.L.: *The ADI Model Problem*. Springer, New York, (2013). <https://doi.org/10.1007/978-1-4614-5122-8>
74. Wolf, T.:  $H_2$  Pseudo-optimal model order reduction, PhD thesis, (2015). <https://d-nb.info/1064075568/34>. Technische Universität München, Munich, Germany
75. Wolf, T., Panzer, H.: The ADI iteration for Lyapunov equations implicitly performs  $H_2$  pseudo-optimal model order reduction. *Int. J. Control* **89**, 481–493 (2016). <https://doi.org/10.1080/00207179.2015.1081985>

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