Electronic Transactions on Numerical Analysis.
Volume 43, pp. 142-162, 2014.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

SELF-GENERATING AND EFFICIENT SHIFT PARAMETERS IN ADI METHODS FOR LARGE LYAPUNOV AND SYLVESTER EQUATIONS*

PETER BENNER†, PATRICK KÜRSCHNER†, AND JENS SAAK†

Abstract. Low-rank versions of the alternating direction implicit (ADI) iteration are popular and well established methods for the numerical solution of large-scale Sylvester and Lyapunov equations. Probably the biggest disadvantage of these methods is their dependence on a set of shift parameters that are crucial for fast convergence. Here we firstly review existing shift generation strategies that compute a number of shifts before the actual iteration. These approaches come with several disadvantages such as, e.g., expensive numerical computations and the difficulty to obtain necessary spectral information or data needed to initially setup their generation. Secondly, we propose two novel shift selection strategies with the motivation to resolve these issues at least partially. Both approaches generate shifts automatically in the course of the ADI iterations. Extensive numerical tests show that one of these new approaches, based on a Galerkin projection onto the space spanned by the current ADI data, is superior to other approaches in the majority of cases both in terms of convergence speed and required execution time.

Key words. Lyapunov equation, Sylvester equation, alternating directions implicit, shift parameters

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

1. Introduction.

The approximate numerical solution of large-scale algebraic matrix equations has attracted enormous attention in the last two decades. In this work we consider large-scale Sylvester matrix equations of the form

$$AXG - EXF = R \tag{1.1}$$

with $A, E \in \mathbb{R}^{n\times n}$, $F, G \in \mathbb{R}^{r\times r}$, E, G nonsingular, and $R \in \mathbb{R}^{n\times r}$. In particular, this includes generalized Lyapunov equations, i.e., the case $G = E^T$, $F = A^T$, and $R = R^T$. It can be shown that when the rank of the right-hand side R of these equations is much lower than the dimension of the equations, i.e., $\operatorname{rank} R \ll \min(n, r)$, the solution often exhibits a low numerical rank [1, 19, 28, 29, 34]. Hence, it can be accurately approximated by a low-rank factorization. This is the backbone for several numerical algorithms of different kinds that try to find such low-rank factors; see [14, 33] for recent surveys. Here we focus on low-rank versions of methods based on the alternating directions implicit (ADI) iteration [9, 10, 25, 28, 30, 40, 43]. Probably the largest disadvantage of ADI methods is their dependence on shift parameters, which are crucial for fast convergence. Optimal or high-quality shifts are usually difficult to obtain for large-scale problems. Either, they rely on spectral data which are hard to get for large problems, or their generation involves inefficient and expensive computations. Thus, our emphases in this work are new and efficient strategies for computing shift parameters that also lead to fast convergence but without these drawbacks. We especially look for approaches that are automatic in the sense that they do not require any special a priori knowledge or setup data. The remainder of our article is divided into two main parts: Section 2 is devoted to generalized Lyapunov equations. There, after giving a concise derivation and overview of recent numerical enhancements of low-rank ADI methods for Lyapunov equations, we discuss some popular existing shift strategies and give two novel approaches. These new strategies are tested and compared to the existing ones in several numerical experiments. Then Section 3 is concerned with the low-rank ADI iteration for the more difficult generalized Sylvester equations. As before we review existing shift

*Received October 29, 2013. Accepted October 30, 2014. Published online on December 12, 2014. Recommended by K. Jbilou.

†Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany, ({benner,kuerschner,saak}@mpi-magdeburg.mpg.de).

ETNA
Kent State University
http://etna.math.kent.edu

strategies and propose new ones which solve some of the issues of the existing approaches. Numerical experiments illustrate their performance. Finally, we conclude and give possible future research perspectives in Section 4.

We use the following notation in this paper: $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex numbers, and $\mathbb{R}_-, \mathbb{C}_-$ refer to the set of strictly negative real numbers and the open left half plane. In the matrix case, $\mathbb{R}^{n\times m}, \mathbb{C}^{n\times m}$ denote $n \times m$ real and complex matrices, respectively. For any complex quantity $X = \operatorname{Re}(X) + \jmath \operatorname{Im}(X)$, $\operatorname{Re}(X)$, $\operatorname{Im}(X)$ are its real and imaginary parts, and $\jmath$ denotes the imaginary unit. The complex conjugate of X is denoted by $\overline{X} = \operatorname{Re}(X) - \jmath \operatorname{Im}(X)$. The absolute value of $\xi \in \mathbb{C}$ is denoted by $|\xi|$, and, if not stated otherwise, $\|\cdot\|$ is the Euclidean vector or subordinate matrix norm (spectral norm). The matrix A^T is the transpose of a real $n \times m$ matrix, and $A^H = \overline{A}^T$ is the complex conjugate transpose of a complex matrix. The identity matrix of dimension n is indicated by I_n. The inverse of a nonsingular matrix A is denoted by A^{-1}, and $A^{-H} = (A^H)^{-1}$. The vector $(1, \ldots, 1)^T$ of length m is expressed by $\mathbf{1}_m$. For symmetric positive (negative) definite matrices ($A = A^T \succ 0$ ($\prec 0$)), we use the abbreviation spd (snd). For a pair of two square matrices A, E, the spectrum is given by $\Lambda(A, E) := \{z \in \mathbb{C} : \det(A - zE) = 0\}$, where det is the determinant. Moreover, the spectral radius is given by $\rho(A, E) := \max\{|\lambda|, \lambda \in \Lambda(A, E)\}$. If $E = I$, the second argument is neglected.

2. Lyapunov equations. In this section we investigate, as an important special case of (1.1), generalized Lyapunov equations

$$AXE^T + EXA^T = -BB^T, \tag{2.1}$$

where $B \in \mathbb{R}^{n\times m}$ with $m \ll n$. We employ the usual assumption $\Lambda(A, E) \subset \mathbb{C}_-$ to ensure the existence of a unique solution. In the following subsection we give a concise derivation of the low-rank alternating directions implicit (ADI) method for computing low-rank solution factors of (2.1). There we also include recent developments regarding some efficiency improvements. After that, we review a number of existing strategies for generating shift parameters, which are a crucial factor for convergence of the ADI iteration. These approaches come with some issues in a large-scale setting, e.g., they are not numerically feasible, they depend on, e.g., spectral data of A, E which are hard to get, or they involve certain a priori setup parameters for which there are no known optimal selection strategies. We then investigate shift strategies which resolve all or at least some of these issues. This will lead to two new approaches where shifts are generated automatically during the ADI iteration. The treatment of special cases of (2.1) is also briefly discussed. Numerical tests using a range of different examples show the often superior performance of the new shift strategies compared to the existing ones.

2.1. Low-rank ADI methods for Lyapunov equations. The alternating directions implicit (ADI) iteration [40] for (2.1) is given by

$$\begin{aligned} EX_jE^T =& (A - \overline{\alpha_j}E)(A + \alpha_j E)^{-1} EX_{j-1}E^T (A + \alpha_j E)^{-H} (A - \overline{\alpha_j}E)^H \\ &- 2\operatorname{Re}(\alpha_j)\, E(A + \alpha_j E)^{-1} BB^T (A + \alpha_j E)^{-H} E^T \end{aligned} \tag{2.2}$$

for $j \geq 1$, some shift parameters $\{\alpha_1, \alpha_2, \ldots, \alpha_j\} \subset \mathbb{C}_-$, and an initial guess $X_0 = X_0^T \in \mathbb{R}^{n\times n}$. These shift parameters steer the convergence and are the main focus of this paper. The above iteration operates on dense $n \times n$ matrices and hence is not feasible for large-scale problems. There are several experimental [28] and theoretical results [1, 19, 29, 34, 37] showing that when $m \ll n$, the numerical rank of the solution X of (2.1) is small, e.g., in the sense that the singular values of X decay rapidly towards zero. This

ETNA
Kent State University
http://etna.math.kent.edu

serves as motivation to approximate X via $X \approx ZZ^T$, where $Z \in \mathbb{R}^{n\times t}$ is a low-rank factor with $\operatorname{rank}(Z) = t \ll n$. Introducing $X_j = Z_j Z_j^H$ into (2.2), setting $Z_0 = 0$, applying some basic algebraic manipulations, and reordering the shifts leads to the generalized low-rank ADI iteration (G-LR-ADI) [3, 9, 25, 28]

$$\begin{aligned} Z_1 = V_1 = (A+\alpha_1 E)^{-1}B, \quad Z_j &= \left[Z_{j-1}, \sqrt{-2\operatorname{Re}(\alpha_j)}V_j\right], \\ V_j = V_{j-1} - (\alpha_j + \overline{\alpha_{j-1}})(A+\alpha_j E)^{-1}(EV_{j-1}), \quad j &> 1. \end{aligned} \tag{2.3}$$

Now in each iteration step, m new columns are added to the previous low-rank solution factor. The main computational costs result from the solution of the shifted linear systems with m right-hand sides. We assume in the following that we are able to efficiently solve these linear systems. In [7] it is shown that it holds for the Lyapunov residual at step j that

$$\mathcal{L}(X_j) := \mathcal{L}_j = AZ_jZ_j^HE^T + EZ_jZ_j^HA^T + BB^T = W_jW_j^T,$$

where

$$W_j = W_{j-1} - 2\operatorname{Re}(\alpha_j)\,EV_j, \qquad W_0 := B, \tag{2.4}$$

such that $\|\mathcal{L}_j\| = \|W_j^HW_j\|$ can be cheaply evaluated in the spectral or Frobenius norm. Moreover, the iterates can be rewritten as

$$V_j = (A+\alpha_j E)^{-1}W_{j-1}, \tag{2.5}$$

which gives a reformulated version of G-LR-ADI [6], where the residual factors W_j are an integral part of the iteration. So far we have used complex low-rank factors since some of the shift parameters might be complex. To ensure that X_j is real, these complex shifts have to occur in pairs of complex conjugate shifts, i.e., if $\alpha_j \in \mathbb{C}_- \setminus \mathbb{R}$, then $\alpha_{j+1} = \overline{\alpha_j}$. Under this assumption it is possible to prove [6, 7, 8] that the iterates V_{j+1} and W_{j+1} associated to $\overline{\alpha_j}$ can be constructed from data available at step j via

$$V_{j+1} = \overline{V_j} + 2\tfrac{\operatorname{Re}(\alpha_j)}{\operatorname{Im}(\alpha_j)}\operatorname{Im}(V_j) \in \mathbb{C}^{n\times m}, \tag{2.6}$$

$$W_{j+1} = W_{j-1} - 4\operatorname{Re}(\alpha_j)\,E\left(\operatorname{Re}(V_j) + \tfrac{\operatorname{Re}(\alpha_j)}{\operatorname{Im}(\alpha_j)}\operatorname{Im}(V_j)\right) \in \mathbb{R}^{n\times m}. \tag{2.7}$$

Hence, only one complex shifted linear system has to be solved for each pair of complex conjugate shifts. Moreover, Z_{j+1} is obtained by augmenting Z_{j-1} by $2m$ real columns such that the low-rank factor is a real matrix after termination of G-LR-ADI. The complete reformulated G-LR-ADI iteration [6] including this handling of complex shifts is given in Algorithm 2.1. This is the algorithm we shall use from now on for solving Lyapunov equations. Note that this formulation is mathematically equivalent to the original low-rank iteration (2.3), although more efficient.

2.2. Existing strategies for precomputed shifts. The convergence speed of the ADI iteration (2.2) is strongly influenced by the spectral radii of

$$A_j := \prod_{k=1}^{j} A_{\alpha_k}, \quad A_{\alpha_k} := (A+\alpha_k E)^{-1}(A-\overline{\alpha_k}E)$$

(see [21, 31]), where A_{α_k} are the iteration matrices of (2.2). Good shifts should therefore make the radii $\rho(A_j)$ as small as possible to ensure fast convergence. A well-known result

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 2.1: Reformulated Real G-LR-ADI Iteration.

Input : Matrices A, E, B defining (2.1), shift parameters $\{\alpha_1, \ldots, \alpha_{j_{\max}}\} \subset \mathbb{C}_-$, and tolerance $0 < \tau \ll 1$.
Output: $Z \in \mathbb{R}^{n \times mj_{\max}}$ such that $ZZ^T \approx X$.

$W_0 = B, \quad Z_0 = [\,], \quad j = 1.$
while $\|W_{j-1}^T W_{j-1}\| \geq \tau \|B^T B\|$ **do**
  Solve $(A + \alpha_j E)V_j = W_{j-1}$ for V_j.
  if $\operatorname{Im}(\alpha_j) = 0$ **then**
    $W_j = W_{j-1} - 2\operatorname{Re}(\alpha_j) EV_j, \quad Z_j = [Z_{j-1}, \sqrt{-2\alpha_j} V_j].$
  else
    $\gamma_j = 2\sqrt{-\operatorname{Re}(\alpha_j)}, \quad \delta_j = \frac{\operatorname{Re}(\alpha_j)}{\operatorname{Im}(\alpha_j)}.$
    $W_{j+1} = W_{j-1} + \gamma_j^2 E(\operatorname{Re}(V_j) + \delta_j \operatorname{Im}(V_j)).$
    $Z_{j+1} = [Z_{j-1}, \ \gamma_j(\operatorname{Re}(V_j) + \delta_j \operatorname{Im}(V_j)), \ \gamma_j\sqrt{(\delta_j^2 + 1)} \cdot \operatorname{Im}(V_j)].$
    $j = j + 1$
  $j = j + 1$

for minimizing the spectral radii (see, e.g., [42, 43]) is that the optimal shifts $\{\alpha_1, \ldots, \alpha_J\}$ for J iteration steps of (2.2) (and of its low-rank version in Algorithm 2.1) are given by the solution of the rational min–max problem

$$\min_{\alpha_1,\ldots,\alpha_J \subset \mathbb{C}_-} \left(\max_{1 \leq \ell \leq n} \left| \prod_{i=1}^{J} \frac{\overline{\alpha_i} - \lambda_\ell}{\alpha_i + \lambda_\ell} \right| \right), \qquad \lambda_\ell \in \Lambda(A, E). \tag{2.8}$$

One conceptual issue of relating the above optimization problem to ADI shift parameters is that the derivation of (2.8) does not embrace the low-rank structure of the right-hand side BB^T of the Lyapunov equation. However, the low-rank property of the right-hand side is of tremendous significance for the existence of low-rank solutions. Apart from that, (2.8) has lead to a number of different shift strategies which are frequently and often also successfully applied in low-rank ADI methods. In the following we briefly describe two of these strategies, which we are also going to employ in our numerical tests.

2.2.1. Wachspress and approximate Wachspress shifts. In [43] an analytic solution for (2.8) is proposed which uses the values $a := \min_i \operatorname{Re}(\lambda_i)$, $b := \max_i \operatorname{Re}(\lambda_i)$ and $\phi := \max_i \arctan\left|\frac{\operatorname{Im}(\lambda_i)}{\operatorname{Re}(\lambda_i)}\right|$ for $\lambda_i \in \Lambda(A, E)$ to estimate the shape of the spectrum $\Lambda(A, E)$ via an elliptic functions domain. The computation of optimal shifts (to achieve that the absolute error of the approximate solution is smaller than a tolerance ϵ) is then based on elliptic integrals involving the tolerance ϵ and the above spectral data a, b, and ϕ. If the spectrum $\Lambda(A, E)$ is real or the imaginary parts of the complex eigenvalues are small compared to the real parts, this approach always provides real shift parameters. In the case of large imaginary parts, there exists a modification that produces complex shift parameters. We refer to these shifts as Wachspress shifts in the following. For large-scale matrices, the required spectral data, especially the angle ϕ for complex spectra, can be hard to obtain. An easy way to get approximate Wachspress shifts [11] (also called suboptimal shifts [30, Section 4.3.2.]) is to approximate $\Lambda(A, E)$ by a small number of k_+ Ritz and k_- harmonic Ritz values, i.e., Ritz values with respect to $E^{-1}A$ and $A^{-1}E$. These Ritz values can be computed using Arnoldi or Lanczos processes. One then computes a, b, ϕ on the basis of this typically small set of Ritz values and carries out the Wachspress computations as before. This approach will

ETNA
Kent State University
http://etna.math.kent.edu

be referred to as approximate Wachspress shifts for which an implementation can be found in [30, Algorithm 4.2]. The quality of these shifts depends on the quality of the approximation of a, b, and ϕ by the Ritz values. Hence, the prescribed numbers k_+, k_-, but also ϵ, have a certain influence. Moreover, the Arnoldi methods introduce additional computations which are dominated by the k_+ and k_- solves with E and A for generating the Ritz values. Note that for symmetric systems, i.e., A snd and E spd, only a, b need to be estimated, which can be done less costly in one run of a Lanczos process using the inner product induced by E. The computability of a, b, ϕ obtained from the Ritz values may be increased by using shifted matrices [11].

2.2.2. The heuristic Penzl strategy. Another frequently used heuristic approach to obtain ADI shifts was proposed by Penzl in [28]. There, $\Lambda(A, E)$ is again replaced by a much smaller set consisting of Ritz values and reciprocals of Ritz values with respect to $E^{-1}A$ and $A^{-1}E$, respectively, also using k_+ and k_- Arnoldi steps. The complete procedure for the generation of J shift parameters is given in [28, Algorithm 5.1]. Although this strategy has been used successfully in numerous cases, it comes with several drawbacks. As for the approximate Wachspress shifts, the procedure requires that the values k_+, k_-, and here additionally J, are provided by the user, but there is no known rule how to actually set these values. Numerical experiments show that even small changes in at least one of these parameters can lead to a significantly different performance of G-LR-ADI in the end. In some cases the values k_+, k_- need to be so large that the cost for the Arnoldi processes is non-negligible. The Arnoldi process requires a starting vector for which there is also no known result on how to choose a suitable one. The authors in [8] used $B\mathbf{1}_m$ in their numerical experiments, but whether there are better choices, remains unclear. Of course, the quality of the Ritz values influences the quality of the shifts in the end. If the Arnoldi convergence is slow and the Ritz values are poor approximations of eigenvalues, the shifts may be of poor quality. The computed Ritz values can have positive real parts if $AE^T + EA^T$ is indefinite. These must be neglected.

2.2.3. IRKA shifts. The Iterative Rational Krylov Algorithm (IRKA) [20] is a prominent method for computing reduced order models of large dynamical systems which are locally optimal in the $\mathcal{H}_2$-norm. In [4] it is shown, by drawing connections to a Riemannian optimization framework [39], that IRKA can also be used for the computation of low-rank solutions of large Lyapunov equations. If $A = A^T \prec 0$ and $E = E^T \succ 0$, the obtained approximate solution satisfies an optimality condition with respect to a certain energy norm. For the unsymmetric case, a similar optimality property holds with respect to the residual. Let Q, U be rectangular, orthonormal matrices which span J-dimensional rational Krylov subspaces computed by IRKA, and denote the eigenvalues $\mathcal{A} := \{\alpha_1, \ldots, \alpha_J\} = \Lambda(U^T AQ, U^T EQ)$. Then the approximations to the Lyapunov equation computed by IRKA and G-LR-ADI with $\mathcal{A}$ as shifts are identical [16, 17]. We refer to these shifts as IRKA shifts, which have attracted some attention recently. The main drawback of these shifts is that their computation, i.e., running IRKA until a certain stopping criterion is met, is very expensive. Assume IRKA requires h iterations until convergence. Thus, $2hJ$ shifted linear systems with A, E have to be solved, which makes these IRKA shifts a rather theoretical tool. Nevertheless, we are going to use this shift approach in G-LR-ADI for comparisons in some of our numerical examples. However, we point out that the IRKA shifts should not be considered a competitive alternative. On the other hand, their strong theoretical background may help to improve the strategies investigated later and serves as the initial motivation for the method introduced in Section 2.3.2.

ETNA
Kent State University
http://etna.math.kent.edu

2.2.4. Other shift strategies. There exist a number of other shift parameter approaches. For completeness we mention a few here. For $E = I_n$, an approach based on Leja points is given in [35], where the spectra of $I_n \otimes A^T$ and $A^T \otimes I_n$ are embedded into subsets $\mathcal{E}, \mathcal{F} \subset \mathbb{C}$. For arbitrary values from $\mathcal{E}, \mathcal{F}$, shift parameters are recursively obtained by maximizing the rational function in (2.8). A related potential theory-based approach can be found in [31]. For real spectra and shifts, an improvement of Penzl's heuristic selection strategy (Section 2.2.2), which introduces marginal additional costs, is also proposed in [31, Section 2.2.4]. In [38] a shift strategy is presented which uses the eigenvalues of a small subblock of A corresponding to the nonzero block of the right-hand side BB^T, which is present in certain applications. For the case where the considered Lyapunov equation is related to a linear, time-invariant control system, dominant pole-based shifts are investigated in [30, Section 4.3.3]. The investigation shows that these shifts can be beneficial for a subsequent model order reduction process. A number of related and further shift approaches can be found in [31].

2.3. Self-generating shifts. The previously mentioned shifts are computed before the actual G-LR-ADI iteration. Here we investigate two approaches to compute shift parameters automatically during the iteration. The first of those should, in the current state, be regarded as theoretically more sound but practically less relevant due to its rather high computational costs. The second currently lacks a proper theoretical backing but provides outstanding convergence of the ADI iteration for some examples as reported for the numerical experiments in Section 2.5.

2.3.1. Residual norm-minimizing shifts. As shown in Section 2.1, the residual in the spectral or Frobenius norm is, combining (2.4) and (2.5), given by

$$\|\mathcal{L}_j\| = \|W_j\|^2 \qquad \text{with} \quad W_j = W_{j-1} - 2\,\mathrm{Re}\,(\alpha_j)\, E\left((A + \alpha_j E)^{-1} W_{j-1}\right).$$

Assume that iteration step $j-1$ is completed and we look for the next shift α_j. Since apart from that shift, every quantity in the above formula is known after iteration $j-1$, an intuitive idea is to find a shift α_j that minimizes $\|W_j\|$ because this will also minimize $\|\mathcal{L}_j\|$. Let $\alpha_j = \nu_j + \jmath\mu_j$ with $\nu_j < 0$, and define the bivariate function

$$f_j(\nu, \mu) := \|W_{j-1} - 2\nu E\left((A + (\nu + \jmath\mu)E)^{-1} W_{j-1}\right)\|. \tag{2.9}$$

Then the real and imaginary parts of α_j can be obtained as

$$[\nu_j, \mu_j] = \operatorname*{argmin}_{\nu \in \mathbb{R}_-, \mu \in \mathbb{R}} f_j(\nu, \mu), \tag{2.10}$$

i.e., by solving a minimization problem. Complex shifts can alternatively be produced by using the relations (2.6), (2.7) and minimizing the function

$$g_j(\nu, \mu) := \|W_{j+1}\| = \left\| W_{j-1} - 4\nu E\left[\mathrm{Re}\,(V_j) + \tfrac{\nu}{\mu}\,\mathrm{Im}\,(V_j)\right]\right\|, \tag{2.11}$$

where $V_j = (A + (\nu + \jmath\mu)E)^{-1} W_{j-1}$. In that case the residual norm is minimized with respect to two iteration steps associated with a pair of complex conjugate shifts. Numerical tests did not reveal a significant difference between using (2.9) or (2.11). The minimization problems can in any case be solved by standard routines from optimization software packages such as the MATLAB commands `fminsearch`, `fminunc`, `fminbnd`, or `fmincon`. The latter one can incorporate the constraint that $\nu_j = \mathrm{Re}\,(\alpha_j) < 0$. Such optimization algorithms usually also require initial guesses, which might have a strong influence on their performance. One possibility is to set these initial guesses to the shift found in the previous iterations.

ETNA
Kent State University
http://etna.math.kent.edu

These norm-minimizing shifts are obviously a rather theoretical concept because they are computationally not feasible. Running the optimization methods for their detection requires solving several linear systems for (2.10). Hence, the computation of the shift itself will easily become more expensive than carrying out the current iteration of G-LR-ADI. Moreover, f_j and g_j might have several local minima, and it is difficult to ensure that the global one is found. In the form given above, both approaches will most likely produce a complex shift every time. Real shifts can be obtained, e.g., by neglecting the imaginary parts which are too small in magnitude although it is not clear how to define 'too small'. If it is known that the spectrum of A, E is real, the shifts should also be real, and (2.9) can be simplified by setting $\mu = 0$.

2.3.2. Shifts obtained from a Galerkin projection on spaces spanned by LR-ADI iterates. The heuristic shifts in Section 2.2.2 are essentially Ritz values with respect to A, E. Here we propose a novel idea that also uses Ritz values which are generated from different spaces where the possibly expensive Krylov subspace construction is not needed. Before G-LR-ADI is started, initial shifts are created as follows: let the columns of $\hat{B} \in \mathbb{R}^{n\times m}$ form an orthonormal basis for $\operatorname{span}\{B\}$. Then the first shifts are taken as the eigenvalues of the projected matrices with respect to a Galerkin projection of A, E onto $\operatorname{span}\{\hat{B}\}$, i.e., $\{\alpha_1, \ldots, \alpha_{\hat{m}}\} = \Lambda(\hat{B}^T A\hat{B}, \hat{B}^T E\hat{B}) \cap \mathbb{C}_-$. The intersection with $\mathbb{C}_-$ ensures that possible unstable eigenvalues of $(\hat{B}^T A\hat{B}, \hat{B}^T E\hat{B})$ are neglected such that $\hat{m} \leq m$. Alternatively, unstable eigenvalues might just be reflected at the imaginary axis. In some cases this is not required, e.g., when $E = I_n$ and A is dissipative (i.e., its symmetric part is negative definite). After LR-ADI has processed all of these initial shifts, there are two similar variants to get the next set of shift parameters:

1. Let $V_{\hat{m}}$ be the G-LR-ADI iterate associated to the last processed shift parameter. Compute an orthonormal matrix $\hat{V}_{\hat{m}}$ whose columns are an orthonormal basis for $\operatorname{span}\{V_{\hat{m}}\}$ or $\operatorname{span}\{\operatorname{Re}(V_{\hat{m}}), \operatorname{Im}(V_{\hat{m}})\}$ if the last shift was real or complex, respectively. The next set of shifts is
$$\{\alpha_{\hat{m}+1}, \ldots, \alpha_{\hat{m}+\operatorname{card}(\mathcal{A})}\} = \mathcal{A} := \Lambda(\hat{V}_{\hat{m}}^T A\hat{V}_{\hat{m}}, \hat{V}_{\hat{m}}^T E\hat{V}_{\hat{m}}) \cap \mathbb{C}_-,$$
where $\operatorname{card}(\mathcal{A})$ is at most either m or $2m$ depending on $V_{\hat{m}}$ being a real or complex iterate. In the following we call the shifts obtained in that way V-shifts.
2. Let $W_{\hat{m}}$ be the LR-ADI residual factor associated to the last shift parameter. Compute an orthonormal matrix $\hat{W}_{\hat{m}}$ that spans an orthonormal basis for $\operatorname{span}\{W_{\hat{m}}\}$. The next set of shifts is
$$\{\alpha_{\hat{m}+1}, \ldots, \alpha_{\hat{m}+\operatorname{card}(\mathcal{A})}\} = \mathcal{A} := \Lambda(\hat{W}_{\hat{m}}^T A\hat{W}_{\hat{m}}, \hat{W}_{\hat{m}}^T E\hat{W}_{\hat{m}}) \cap \mathbb{C}_-.$$
Note that $W_{\hat{m}}$ is, according to Algorithm 2.1 and (2.7), always a real $n \times m$ matrix. The so constructed shifts will be referred to as W-shifts in the remainder.

LR-ADI is then continued with these new shifts, and the above procedure is repeated each time the set of shifts has been fully processed. If it happens that all eigenvalues of the projected matrices are unstable, LR-ADI is continued with the previous set of shifts. The main computational cost for this shift generation is the orthogonalization of an $n \times m$ or $n \times 2m$ matrix whenever new shifts are required. This is not expensive since $m \ll n$. It can occur that the columns of $V_{\hat{m}}$ or $W_{\hat{m}}$ have linear dependencies, which should be taken care of by a clever orthogonalization routine. For instance, $\hat{W}_{\hat{m}}$ can have less than m columns. The solution of the at most $2m$-dimensional eigenvalue problem introduces only negligible extra costs. The big advantage of both proposed variants is, compared to the heuristic approach in Section 2.2.2, that no setup parameters such as J, k_+, k_- are required, which makes this

ETNA
Kent State University
http://etna.math.kent.edu

approach completely automatic and hence user-friendly. Additionally, for several numerical tests, these shifts even seem to outperform the heuristic shifts. One disadvantage occurs for problems with a rank-one right-hand side, i.e., when $m = 1$. Then the single shift computed in both variants is actually a generalized Rayleigh quotient. In that case the W-shift is given by

$$\alpha = \frac{\hat{W}_{\hat{m}}^T A W_{\hat{m}}}{\hat{W}_{\hat{m}}^H E W_{\hat{m}}},$$

and hence it will always be a real number, which can be disadvantageous for problems with a complex spectrum. Another drawback of the V- and W-shifts is the lack of a deeper theoretical foundation. It is also not clear which of the two variants is better although in most of our numerical tests the V-shifts seem to be superior.

To complete this section we mention a third approach which uses $\operatorname{span}\{Z_J\}$ as projection basis. There, after J shifts have been processed, Jm Ritz values are computed with respect to the reduced matrix pair generated by an Galerkin projection onto $\operatorname{span}\{Z\}$. These may be taken as new shifts, or, optionally, $h \leq Jm$ of them are selected. A number of possible choices can be used in this case. The simplest would be the h Ritz values largest or smallest in magnitude. Alternatively, one might exploit the increasingly better approximation of the entire spectrum of A and use the computed Ritz values as inputs for the Penzl or Wachspress shift strategies to perform a more educated selection.

Obviously, this third variant is significantly more expensive than the V- and W-shifts since computing an orthogonal space for $\operatorname{span}\{Z_J\}$ requires the orthogonalization of the span of V_j for each $j = 1, \ldots, J$ against the previous Z_{j-1}. Also, the eigenvalue problem is now of dimension Jm and the cost for its solution might not be negligible anymore. Which of the h values of $\hat{A}$ to select for optimal results is also not clear. We do not pursue this approach further but note that in [12, 30], $\operatorname{span}\{Z_J\}$ is used to perform a Galerkin projection on the Lyapunov equation (2.1) to gain a convergence boost in G-LR-ADI.

2.4. Special cases. In this section we discuss the application of the self-generating shift strategies in some selected structure-exploiting variants of G-LR-ADI.

2.4.1. Second-order ADI. Lyapunov equations such as (2.1) are often related to linear, time-invariant dynamical systems of the form

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad A, E \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, \tag{2.12}$$

with $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$. Now consider the second-order, linear, time-invariant dynamical system

$$M\ddot{q}(t) + D\dot{q}(t) + Kq(t) = B_1 u(t), \quad M, D, K \in \mathbb{R}^{n_1\times n_1}, B_1 \in \mathbb{R}^{n_1\times m},$$

with $q(t) \in \mathbb{R}^{n_1}$ and $u(t) \in \mathbb{R}^m$, which can equivalently be written as a system of first differential order (2.12), e.g., with

$$E = \begin{bmatrix} D & M \\ M & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} \in \mathbb{R}^{2n_1\times 2n_1}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \in \mathbb{R}^{2n_1\times m}, \tag{2.13}$$

and $x(t) = [q(t)^T, \dot{q}(t)^T]^T$; see [36]. There exist structure-exploiting variants of G-LR-ADI called second-order LR-ADI (SO-LR-ADI) [7, 13, 27, 30] which do not explicitly form the augmented matrices E, A, B in (2.13) and work with the original data M, D, K, B_1 instead. Of course, such a structure exploitation should also be used in the shift strategies of the previous sections. See, for instance, [7] for details on how to solve the linear systems which

ETNA
Kent State University
http://etna.math.kent.edu

arise also in the computation of the norm-minimizing shifts. The Galerkin projections of Section 2.3.2 are implicitly carried out with the augmented matrices (2.13), i.e., only matrix vector products with M, D, K, and $n_1 \times m$ matrices are required. The resulting small eigenvalue problem does not inherit the block structure given in (2.13).

2.4.2. SLRCF-ADI for index-1 DAEs. Another class of dynamical systems (2.12) are differential algebraic equations (DAE) of index 1 with

$$E = \begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \in \mathbb{R}^{n\times m}, \tag{2.14}$$

where $E_{11} \in \mathbb{R}^{n_f\times n_f}$, $A_{22} \in \mathbb{R}^{n-n_f\times n-n_f}$ are nonsingular and all the other blocks are of appropriate sizes. Here, n_f denotes the number of finite eigenvalues in $\Lambda(A, E)$. Such DAEs can be equivalently rewritten in state space form

$$E_{11}\dot{x_1}(t) = \tilde{A}x_1(t) + \tilde{B}u(t), \quad \tilde{A} \in \mathbb{R}^{n_f\times n_f}, \tilde{B} \in \mathbb{R}^{n_f\times m},$$

with

$$\tilde{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}, \quad \tilde{B} = B_1 - A_{12}A_{22}^{-1}B_2.$$

In [18] a specially tailored G-LR-ADI (SLRCF-ADI) is proposed which solves the Lyapunov equation $\tilde{A}XE_{11}^T + E_{11}X\tilde{A}^T = -\tilde{B}\tilde{B}^T$ without forming the matrices $\tilde{A}, \tilde{B}$ explicitly. The key ingredient is the observation that the solution of the dense linear system $(\tilde{A} + \alpha_j E_{11})V_j = W_{j-1}$ of size n_f can be equivalently and more efficiently obtained from the sparse linear system

$$\begin{bmatrix} A_{11} + \alpha_j E_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V_j \\ \Gamma \end{bmatrix} = \begin{bmatrix} W_{j-1} \\ 0 \end{bmatrix} \tag{2.15}$$

of size n, where the right-hand side in the first iteration is $[B_1^T, B_2^T]^T$ and $\Gamma \in \mathbb{C}^{n-n_f\times m}$ is an auxiliary variable. The same trick can be employed within the minimization algorithms for the residual norm-minimizing shifts described in Section 2.3.1. It also holds that $W_j = W_{j-1} - 2\,\mathrm{Re}\,(\alpha_j)\,E_{11}V_j$. A straightforward application of the projection-based shifts of Section 2.3.2 requires the computation of the matrices

$$\hat{V}^T\tilde{A}\hat{V} = \hat{V}^TA_{11}\hat{V} - \hat{V}^TA_{12}\left(A_{22}^{-1}\left(A_{21}\hat{V}\right)\right), \quad \hat{V}^TE_{11}\hat{V}$$

for the V-shifts and similarly with $\hat{W}$ for the W-shifts. The initial shifts are obtained using an orthonormal basis for $\tilde{B}$. This requires the solution of m linear systems of size $n - n_f$ with A_{22} at each time when new shifts are required, possibly leading to a significant increase in the computational cost.

As a modification of the V-shifts, we propose to carry out the Galerkin projection with the original matrices (2.14) and the augmented iterates $V_j^{\text{aug}} := [V_j^T, \Gamma^T]^T$ from (2.15). Let $\check{V}_j$ be an orthonormal basis for V_j^{aug}, and choose the shifts from $\Lambda(\check{V}_j^TA\check{V}_j, \check{V}_j^TE\check{V}_j) \cap \mathbb{C}_-$. Additionally, possible infinite eigenvalues should also be neglected. We refer to this modification as V^{aug}-shifts. Similarly, we can work with the augmented residual factors for the W-shifts

$$W_j^{\text{aug}} = W_{j-1}^{\text{aug}} - 2\,\mathrm{Re}\,(\alpha_j)\,EV_j^{\text{aug}} = \begin{bmatrix} W_j \\ \Upsilon \end{bmatrix}, \qquad W_0^{\text{aug}} = B,$$

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 2.1
Dimensions n and m, desired residual norm $\epsilon_{\mathcal{L}}$, maximum number of allowed ADI iterations $j^{\max}$, structural properties, and sources for the used Lyapunov test examples. Here, OC and IFISS refer to the Oberwolfach Model Reduction Benchmark Collection and the IFISS [32] FEM package.

Example	n	m	$\epsilon_{\mathcal{L}}$	$j^{\max}$	Properties	Source
FDM1	3 600	5	10^{-10}	250	$E = I$, B random	[22, B in Example 2]
rail5k	5 177	7	10^{-10}	150	A snd, E spd	OC[1], ID=38881
rail79k	79 188	7	10^{-10}	100	A snd, E spd	OC[1], ID=38881
ifiss1	16 641	4	10^{-10}	150	E spd, $B = A \cdot \texttt{rand}(n, m)$	IFISS [32] T-CD3
chain	9 002	5	10^{-8}	400	structure (2.13), B random	[38]
bips	21 128	4	10^{-8}	400	structure (2.14), $n_f = 3078$	[18], bips07_3078[2]

with an auxiliary matrix $\Upsilon \in \mathbb{C}^{n-n_f \times m}$. A simple calculation using the structure of E shows that $\Upsilon = B_2$. This yields the W^{aug}-shifts. For both the V^{aug}- and W^{aug}-shifts, the initial shifts can be obtained by using an orthonormal basis of B. Note that there are also LR-ADI approaches for handling DAE systems of higher indices [26], e.g., the recent work [2] regarding the case of index 2 arising in optimal control of the (Navier)-Stokes equation. The proposed shift approaches can be adapted to these cases in a straightforward manner.

2.5. Numerical experiments. We are now going to evaluate and compare the performance of the presented shift generation strategies. To this end, G-LR-ADI (Algorithm 2.1) is run until $\|\mathcal{L}\|/\|B\|^2 \leq \epsilon_{\mathcal{L}}$ with $0 < \epsilon_{\mathcal{L}} \ll 1$ is achieved or a maximum allowed number $j^{\max}$ of iterations is reached. All experiments have been carried out in MATLAB 7.11.0 on an Intel®Xeon®W3503 execution with 2.40 GHz and 6 GB RAM. We use a collection of test examples whose dimensions n, m, the required residual tolerance $\epsilon_{\mathcal{L}}$, the maximum allowed number of G-LR-ADI iterations $j^{\max}$, as well as selected information regarding symmetry properties, sources, and references of the examples are given in Table 2.1. There, OC stands for Oberwolfach Model Reduction Benchmark Collection[1], and the ID gives a unique identifier for obtaining the example. IFISS refers to the MATLAB finite-element package [32]. The examples *chain* and *bips*[2] belong to the special cases mentioned in Section 2.4 and are handled by SO-LR-ADI and SLRCF-ADI, respectively. For *bips* we used the shifted matrix $A - 0.05E$ as in [18, Section V.A]. The complete identifier for this example is given in the last column.

The results for these examples and different shift strategies are summarized in Table 2.2. There, the heuristic strategy and its settings are denoted by "heur(J, k_+, k_-)". Likewise, "wachs(ϵ, k_+, k_-)" stands for approximate Wachspress shifts obtained from k_+, k_- Ritz values and a tolerance ϵ. The number of shifts J is also given. For these two approaches, the initial vector for the Arnoldi processes is $B\mathbf{1}_m$. Moreover, IRKA(J) refers to J shifts obtained after IRKA, initialized with random data, converged using a tolerance of 10^{-3} and the stopping criterion in [20]. All of these precomputed shifts are used in a cyclic manner if it occurs that the required number of G-LR-ADI iterations is higher than the number of the available shifts. The computation of the orthonormal bases of B, V_j, or W_j for the V- and W-shifts was carried out using the MATLAB routine `orth`. The residual-minimizing shifts were obtained using the MATLAB routine `fminsearch` since the constrained optimization routine `fmincon` did not converge for our examples. The initial guess for `fminsearch` was always set to the previously computed shift. Due to the expensive nature of the IRKA- and residual norm-minimizing shifts, both strategies are only applied to the moderately sized

[1] http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark.
[2] Available at http://sites.google.com/site/rommes/software.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 2.2

Results for the examples using different shift strategies: t_{shift} and t_{ADI} denote the times (in seconds) spent for computing the shifts and executing G-LR-ADI, respectively, and the total consumed time is t_{total}. The required iterations j^{iter} and the final obtained residual norm $\|\mathcal{L}_{j^{iter}}\|$ are also given. The smallest values of t_{total} and j^{iter} for each example are emphasized using bold letters.

Ex.	Shift strategy	t_{shift}	t_{ADI}	t_{total}	j^{iter}	$\|\mathcal{L}_{j^{\text{iter}}}\|$
FDM1	heur(10, 20, 20)	0.53	0.74	1.27	26	$9.91{\cdot}10^{-11}$
	wachs(10^{-10}, 10, 10), $J = 13$	0.23	0.75	0.98	26	$1.76{\cdot}10^{-12}$
	IRKA(30)	15.35	0.81	16.16	29	$5.40{\cdot}10^{-12}$
	res.min	82.87	0.81	83.67	**24**	$2.67{\cdot}10^{-11}$
	V-shifts	0.03	0.98	1.00	30	$3.93{\cdot}10^{-11}$
	W-shift	0.03	0.87	**0.89**	31	$6.23{\cdot}10^{-13}$
rail5k	heur(10, 20, 10)	0.50	3.31	3.80	59	$3.03{\cdot}10^{-11}$
	wachs(10^{-10}, 20, 10), $J = 40$	0.47	2.79	**3.26**	**40**	$5.82{\cdot}10^{-11}$
	IRKA(60)	28.98	7.80	36.78	122	$6.94{\cdot}10^{-11}$
	res.min	76.40	12.07	88.47	150	$7.00{\cdot}10^{-9}$
	V-shifts	0.03	3.44	3.46	57	$9.83{\cdot}10^{-12}$
	W-shifts	0.06	10.46	10.52	150	$6.01{\cdot}10^{-2}$
rail79k	heur(20, 40, 40)	44.87	85.96	130.84	54	$7.00{\cdot}10^{-11}$
	wachs(10^{-10}, 20, 10), $J = 47$	13.14	111.10	**124.24**	**47**	$6.36{\cdot}10^{-11}$
	V-shifts	0.78	158.58	159.35	85	$2.80{\cdot}10^{-12}$
	W-shifts	0.91	229.60	230.51	100	$3.19{\cdot}10^{-2}$
ifiss1	heur(20, 30, 20)	6.48	12.14	18.62	**48**	$5.09{\cdot}10^{-11}$
	wachs(10^{-10}, 20, 10), $J = 33$	2.57	22.66	25.23	97	$8.97{\cdot}10^{-11}$
	V-shifts	0.11	15.11	**15.22**	62	$2.64{\cdot}10^{-11}$
	W-shifts	0.21	34.37	34.58	150	$3.70{\cdot}10^{-10}$
chain	heur(40, 50, 50)	1.85	4.82	6.67	383	$6.60{\cdot}10^{-9}$
	wachs(10^{-10}, 20, 10), $J = 130$	1.63	2.73	4.37	309	$8.29{\cdot}10^{-9}$
	V-shifts	0.21	1.96	**2.16**	**147**	$6.04{\cdot}10^{-9}$
	W-shifts	1.35	3.85	5.20	400	$5.06{\cdot}10^{0}$
bips	heur(40, 50, 70)	11.79	35.87	47.66	378	$6.55{\cdot}10^{-9}$
	heur(60, 80, 80)	12.01	21.30	33.32	226	$5.95{\cdot}10^{-9}$
	wachs(10^{-8}, 20, 20), $J = 35$	2.46	30.11	32.57	268	$7.56{\cdot}10^{-9}$
	V-shifts	1.71	7.86	9.56	**83**	$8.88{\cdot}10^{-9}$
	W-shifts	1.73	9.79	11.52	104	$8.23{\cdot}10^{-9}$
	V^{aug}-shifts	0.16	7.95	**8.11**	84	$4.45{\cdot}10^{-9}$
	W^{aug}-shifts	0.46	37.11	37.57	400	$2.84{\cdot}10^{-8}$

examples *FDM1* and *rail5k*. Because of the symmetry properties and $\Lambda(A, E) \subset \mathbb{R}_-$ in *rail5k*, both approaches are further simplified such that only real shifts are considered. In addition to the data collected in Table 2.2, Figure 3.1 displays the scaled residual norm against the ADI iteration number in the top plots and in the bottom plots the scaled residual norm against the cumulative execution time, i.e., the total consumed time so far, for the examples *FDM1* and *rail5k*.

For the heuristic shifts it is apparent that, compared to the plain ADI computation time t_{ADI}, a signification portion t_{shift} of the total execution time t_{total} is spent for the in-

ETNA
Kent State University
http://etna.math.kent.edu

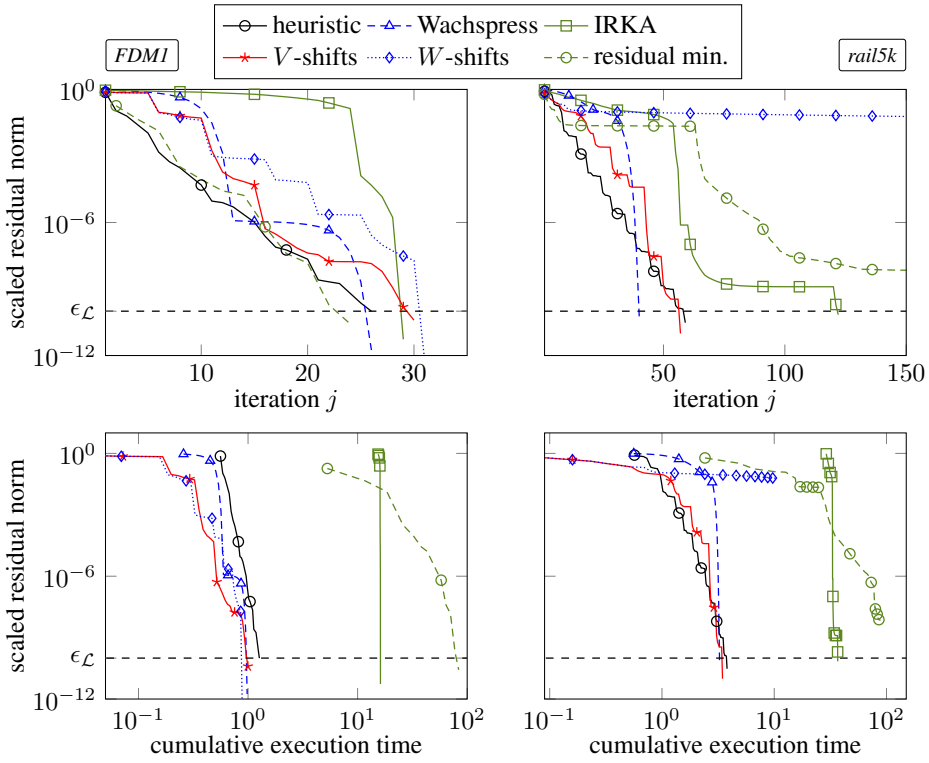

FIG. 2.1. *Scaled residual norm against iteration index j (top plots) and cumulative execution time at iteration j (bottom plots) of G-LR-ADI using different shift strategies for the FDM1 (left plots) and rail5k (right plots) examples.*

volved Arnoldi processes. They lead to the desired accuracy although for each example there was at least one other shift strategy which required less ADI iterations. The number of used Arnoldi steps, k_+, k_-, influences the quality of the heuristic shifts as it is seen in the *bips* example, where we used two settings: the first one uses exactly the values J, k_+, k_- as in the original SLRCF-ADI paper [18, Section V, Table V], while the second one was chosen through extensive trial and error optimization. The difference in both execution time (47.3 against 33.3 seconds) as well as the ADI iteration numbers (378 against 226) is significant. The approximate Wachspress shifts also rely on Arnoldi processes, but there usually smaller numbers k_+, k_- were sufficient to get accurate estimates of the required spectral data. Hence, t_{shift} is smaller for heuristic shifts. As expected, these shifts lead to the best performance both in terms of execution time and required iterations for the symmetric examples *rail5k*, *rail79k*. Their typical residual curves can be seen in Figure 2.1 (top right plot). They loose this superiority for those examples where complex spectra with large imaginary parts are encountered. Especially for *ifiss1*, they can not compete with the heuristic shifts. In additional tests, the Wachspress shifts seemed to be less sensitive with respect to the values k_+, k_- than the heuristic shifts. For the IRKA shifts, the computation times t_{shift} exceed t_{ADI} by far, and hence the total execution time is also very large (also see the bottom plots of Figure 2.1). They lead to a fast convergence for *FDM1* but to a comparably slow convergence for *rail5k*. We observed that the settings for J and the initial data for IRKA have a large influence on its convergence. In other similar tests, different starting data lead to completely distinctive IRKA shifts and thus to a different ADI convergence. Anyway, their expensive computation makes this approach impractical as a source for good ADI shifts.

ETNA
Kent State University
http://etna.math.kent.edu

Now we move on to the novel self-generating shifts proposed in Sections 2.3.1–2.3.2. It is no surprise that the generation time t_{shift} for the residual-minimizing shifts is extremely high, i.e., even higher than those of the IRKA shifts, which makes this approach the most expensive and time consuming one. They lead, however, to the fastest convergence for *FMD1* (24 iterations), which is also nicely monotonic as it can be seen in the top left plot of Figure 2.1. For *rail5k*, these shifts do not lead to a convergence before $j^{\max}$ iterations. There are two possible reasons for this: on the one hand, the computed minimum of (2.9) was not the global one, and on the other hand, the computed shift was an unstable one. Both situations were also observed in other experiments. The computation of unstable shifts is a more severe problem but could be prevented if a constrained optimization method was employed. Shifts associated with non-global minima still lead to a reduction of the residual norm but delayed the convergence. This can be observed in the top right plot for *rail5k* in Figure 2.1. In further experiments not reported here, if the employed optimization routine managed to find the global minimizer, the residual-minimizing shifts lead to the smallest number of required iterations j^{iter} compared to the other strategies. Because of the large construction time of these shifts, this approach is at the current stage only of theoretical interest. Finding an analytic solution of the minimization problem or a cheap approximation thereof is an interesting future research topic.

The V- and W-shifts required in all examples a very small construction time t_{shift}, which is in most cases a negligible fraction of t_{total}. However, except for *FDM1* and *bips*, only the V-shifts lead to fast convergence. In all other examples, the W-shifts did not achieve the required accuracy before $j^{\max}$ ADI iterations. For the example *rail5k* (see, e.g., the top right plot in Figure 2.1), a stagnation phase is encountered in the later iterations because the computed W-shifts almost did not change anymore. We plan to investigate why this is the case in the future. One promising tool for this appears to be the recently established novel relations of low-rank ADI and rational Krylov subspace methods [4, 44, 45]. The V-shifts lead to the smallest timings t_{total} in all examples with nonsymmetric coefficient matrices. This can also be observed in the plot of the residual norm versus the consumed execution time in Figure 2.1. For *rail5k/79k*, the heuristic and Wachspress shifts are superior. Note that in the nonsymmetric examples, the number of required ADI iterations j^{iter} for the V-shifts is not always smaller than that of the heuristic shifts (see, e.g., example *ifiss1*), but due to the exceptionally cheap generation of the V-shifts, their overall execution time t_{total} is nonetheless smaller. They significantly outperform all other shift approaches in the *chain* and *bips* examples, where they lead to a drastically reduced number of required iterations. In fact, we never experienced a faster ADI convergence for the *bips* system. There, the V^{aug}-shifts are slightly better than the V-shifts, but the difference is negligible. Note that the W-shifts converged, while the W^{aug}-shifts did not. To conclude, the V-shifts appear to be a very promising approach especially for Lyapunov equations with nonsymmetric coefficient matrices where the spectrum contains complex eigenvalues. We plan to investigate their behavior deeper in subsequent work. Another big advantage of theirs, although not reflected in the timings and iteration counts, is that they can be applied completely automatically in the sense that they can be implemented without the user having to take care of selecting ADI shifts at all.

3. Sylvester equations. Now we consider generalized Sylvester equations of the form

$$AXG - EXF = BC^T \tag{3.1}$$

with $A, E \in \mathbb{R}^{n\times n}$, $F, G \in \mathbb{R}^{r\times r}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{r\times m}$, and the sought solution $X \in \mathbb{R}^{n\times r}$. We assume that E and G are nonsingular and, in order to allow a unique solution X to exist, $\Lambda(A, E) \cap \Lambda(F, G) = \emptyset$.

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 3.1: Generalized factored ADI iteration (G-fADI) [5] for (3.1).

Input : A, B, E, C, F, G as in (3.1), shift parameters $\{\alpha_1, \ldots, \alpha_{j_{\max}}\}$, $\{\beta_1, \ldots, \beta_{j_{\max}}\}$, and tolerance $0 < \tau \ll 1$.
Output: $Z_{j_{\max}} \in \mathbb{C}^{n \times rj_{\max}}$, $Y_{j_{\max}} \in \mathbb{C}^{m \times rj_{\max}}$, $D_{j_{\max}} \in \mathbb{C}^{rj_{\max} \times rj_{\max}}$ such that $Z_{j_{\max}} D_{j_{\max}} (Y_{j_{\max}})^H \approx X$.

$W_0 = B, T_0 = C$, $Z_0 = D_0 = Y_0 = [\,]$, $j = 1$.
while $\|W_{j-1} T_{j-1}^H\| \geq \tau \|BC^T\|$ **do**
  $\gamma_j = \beta_j - \alpha_j$.
  $V_j = (A - \beta_j E)^{-1} W_{j-1}$, $W_j = W_{j-1} + \gamma_j E V_j$.
  $S_j = (F - \alpha_j G)^{-H} T_{j-1}$, $T_j = T_{j-1} - \overline{\gamma_j} G^T S_j$.
  Update the low-rank solution factors

$$Z_j = [Z_{j-1}, V_j], Y_j = [Y_{j-1}, S_j],\ D_j = \operatorname{diag}(D_{j-1}, \gamma_j I_r).$$

  $j = j + 1$.

3.1. The factored ADI for Sylvester equations. The ADI iteration for (3.1) (see [40] for $E = I_n$, $G = I_r$) is given by

$$\begin{aligned} EX_jG =& (A - \alpha_j E)(A - \beta_j E)^{-1} EX_{j-1} G (F - \alpha_j G)^{-1} (F - \beta_j G) \\ &+ (\beta_j - \alpha_j) E (A - \beta_j E)^{-1} BC^T (F - \alpha_j G)^{-1} G. \end{aligned} \tag{3.2}$$

Here $\{\alpha_1, \ldots, \alpha_J\}$, $\{\beta_1, \ldots, \beta_J\}$ are two sets of shift parameters with $\alpha_i \notin \Lambda(F, G)$, $\beta_i \notin \Lambda(A, E)$, and $\alpha_i \neq \beta_i$, for all i. Setting $X_0 = 0$ and using similar manipulations as in the Lyapunov case leads to the low-rank Sylvester ADI (or factored ADI (fADI)), cf. [10, Algorithm 1], [24, Algorithm 2.1], for computing low-rank solution factors $Z \in \mathbb{C}^{n \times f}$, $Y \in \mathbb{C}^{r \times f}$, $D \in \mathbb{C}^{f \times f}$, $f \ll \min(n, r)$ of (3.1) such that $ZDY^H \approx X$. Using generalizations of the techniques for Lyapunov equations in [7], the method can equivalently be rewritten [5, Algorithm 1] in the form illustrated in Algorithm 3.1, where, in addition to the iterates V_j, S_j with respect to the matrix pairs (A, E), (F, G), the low-rank residual factors W_j, T_j are included. The modified Algorithm 3.1 allows for a cheap computation of the residual norm

$$\|\mathcal{S}(X_j)\| := \|\mathcal{S}_j\| = \|AZ_j D_j Y_j^H G - EZ_j D_j Y_j^H F - BC^T\| = \|W_j T_j^H\|;$$

see [5, Theorem 4]. As in Algorithm 2.1 for Lyapunov equations, it is possible to take care of complex shift parameters by a suitable reformulation of Algorithm 3.1 [5, Algorithm 2]. For applying this real version of G-fADI, both sets of shifts have to be in a certain pairwise order, which can be achieved by a simple permutation. For the ease of presentation, we stick to the given complex formulation in the following but use the real version in our numerical examples.

3.2. Existing shift strategies. Similar to the Lyapunov case (Section 2.2), the convergence behavior of (3.2) and Algorithm 3.1 depends critically on the spectral radii of

$$\begin{aligned} A_j &:= \prod_{k=1}^{j} A_{\alpha_k, \beta_k}, \quad A_{\alpha_k, \beta_k} := (A - \beta_k E)^{-1} (A - \alpha_k E), \\ F_j &:= \prod_{k=1}^{j} F_{\alpha_k, \beta_k}, \quad F_{\alpha_k, \beta_k} := (F - \alpha_k G)^{-1} (F - \beta_k G). \end{aligned} \tag{3.3}$$

ETNA
Kent State University
http://etna.math.kent.edu

For normal matrix pairs (i.e., the left and right eigenvectors coincide) (A, E), (F, G) in (3.1), it can be shown [10, 24, 31, 41] that optimal shifts for J iterations of Algorithm 3.1 have to satisfy the optimization problem

$$\min_{\alpha_j,\beta_j\in\mathbb{C}} \left(\max_{\substack{1\le\ell\le n\\ 1\le k\le r}} \prod_{j=1}^{J} \left| \frac{(\lambda_\ell-\alpha_j)(\mu_k-\beta_j)}{(\lambda_\ell-\beta_j)(\mu_k-\alpha_j)} \right| \right), \quad \lambda_\ell \in \Lambda(A,E), \mu_k \in \Lambda(F,G). \tag{3.4}$$

The above rational optimization problem is also referred to as two-variable ADI parameter problem [41, 43] and is harder to solve than the optimization problem (2.8) for Lyapunov equations. In the following we review generalizations of the Wachspress, heuristic, and IRKA shifts for the Sylvester ADI. After that we propose two strategies for self-generating shifts.

3.2.1. Optimal Sylvester ADI shifts. Analytic solutions for solving (3.4) are proposed in [41], [43, Chapter 2 & 4] and are based on spectral alignment and, as in the Lyapunov case, elliptic integrals. They require the following knowledge of the smallest and largest real parts $a := \min_i \operatorname{Re}(\lambda_i)$, $b := \max_i \operatorname{Re}(\lambda_i)$, $c := \min_i \operatorname{Re}(\mu_i)$, $d := \max_i \operatorname{Re}(\mu_i)$, and the angles $\phi := \max_i \arctan|\frac{\operatorname{Im}(\lambda_i)}{\operatorname{Re}(\lambda_i)}|$, $\psi := \max_i \arctan|\frac{\operatorname{Im}(\mu_i)}{\operatorname{Re}(\mu_i)}|$ for $\lambda_i \in \Lambda(A,E)$ and $\mu_i \in \Lambda(F,G)$. An implementation of this shift generation strategy is given in the `parsyl`[3] routine provided in [43]. If the spectra $\Lambda(A,E)$, $\Lambda(F,G)$ are contained in real, disjoint intervals $[a,b]$, $[c,d]$, another similar approach for generating an equal number J of α- and β-shifts is given in [31, Algorithm 2.1].

As in the Lyapunov case, one might use Arnoldi or Lanczos processes to obtain approximations to a, b, c, d, ϕ, ψ in the large-scale case for both approaches. We propose to approximate $\Lambda(A,E)$ by a set consisting of k_+^A Ritz and k_-^A inverse Ritz values with respect to $E^{-1}A$ and $A^{-1}E$. Likewise, $\Lambda(F,G)$ is approximated by k_+^F Ritz and k_-^F inverse Ritz values with respect to $G^{-1}F$ and $F^{-1}G$. Approximations to the extremal eigenvalues and the spectral angles of $\Lambda(A,E)$ and $\Lambda(F,G)$ can then be read of easily. However, as for the approximate Wachspress shifts, the so obtained shifts can be sensitive with respect to the quality of the approximations of the extremal eigenvalues. This was numerically investigated in [31, Section 2.2.2] for the optimal real shift parameters.

3.2.2. Heuristic shifts. In [10, 24], a heuristic approach is proposed which generalizes the Penzl shifts (Section 2.2.2) to the solution of Sylvester equations. The spectra $\Lambda(A,E)$, $\Lambda(F,G)$ are approximated in the same way as for the optimal shifts above. With these sets of Ritz values, one solves (3.4) in an approximate sense to get J (with $J \le k_+^A + k_-^A$) α-shifts and L (with $L \le k_+^F + k_-^F$) β-shifts. A detailed implementation can be found in [10, Algorithm 2], [24, Algorithm 3.1]. Note that in [5] only the $k_+^A + k_-^A$ and $k_+^F + k_-^F$ Ritz values are used as shifts, which worked sufficiently well. This heuristic approach suffers from the same disadvantages as the heuristic approach for the Lyapunov equation in Section 2.2.2: there is no known rule how to select the predefined numbers J, L, k_+^A, k_-^A, k_+^F, and k_-^F, and the quality of the Ritz values (and hence of the shifts) depends on the performance of the Arnoldi processes, which also introduces additional costs due to the required linear solves. Moreover, there is no known strategy for choosing their initial vectors suitably.

3.2.3. IRKA shifts. For symmetric Sylvester equations with E, $-G$, $-A$, $-F$ spd, a generalization of IRKA (symmetric Sylvester IRKA (Sy)^{2}IRKA) is given in [4, Algorithm 3]. The obtained approximate solutions again satisfy an optimality condition with respect to their residual in a certain norm. The shifts obtained from (Sy)^{2}IRKA can also be used within the G-fADI leading to equivalent approximate solutions as discussed in [17].

[3] Available at `http://extras.springer.com/2013/978-1-4614-5121-1`.

ETNA
Kent State University
http://etna.math.kent.edu

(Sy)^{2}IRKA can be easily modified to handle general nonsymmetric Sylvester equations. Let Q, U and H, N be rectangular, orthonormal matrices which span J-dimensional rational Krylov subspaces computed by a Sylvester IRKA method (SyIRKA) with respect to A, E, B and F, G, C, respectively. For the symmetric Sylvester equation mentioned before, it holds that $Q = U$ and $H = N$. Then the Sylvester IRKA shifts are given by $\mathcal{A} := \{\alpha_1, \ldots, \alpha_J\} = \Lambda(Q^H AU, Q^H EU)$ and $\mathcal{B} := \{\beta_1, \ldots, \beta_J\} = \Lambda(H^H FN, H^H GN)$. This strategy has the same drawbacks as the similar one in the Lyapunov case; especially the high computational cost of SyIRKA makes it computationally less feasible. It is rather theoretically motivated due to the interesting properties [4, 17] of the IRKA shifts, which we use merely for reason of comparison in the numerical examples.

3.3. Self-generating shifts.

3.3.1. Residual norm-minimizing shifts. Motivated by the Lyapunov residual norm-minimizing shifts in Section 2.3.1, one can derive a similar framework for Sylvester equations. For simplicity we consider here only the case of real α- and β-shifts. The (spectral or Frobenius) norm of the Sylvester residual matrix $\mathcal{S}_j$ can be efficiently computed via

$$\|\mathcal{S}_j\| = \|W_j T_j^T\| = \sqrt{\|T_j W_j^T W_j T_j^T\|} = \|L_j\| \qquad \text{with} \quad L_j = W_j R_j^T$$

and a QR decomposition $T_j = \hat{T}_j R_j$; see [5]. According to Algorithm 3.1, we have

$$W_j = W_{j-1} + (\beta_j - \alpha_j) E V_j = W_{j-1} + (\beta_j - \alpha_j) E \left((A - \alpha_j E)^{-1} W_{j-1} \right),$$
$$T_j = T_{j-1} - (\beta_j - \alpha_j) G^T S_j = T_{j-1} - (\beta_j - \alpha_j) G^T \left((F - \beta_j G)^{-T} T_{j-1} \right).$$

Since, W_{j-1}, T_{j-1} are given at the beginning of iteration j, the only unknowns above are the shifts α_j, β_j, and we may regard $\|\mathcal{S}_j\|$ as bivariate function. The next shifts can be obtained by solving the optimization problem

$$[\alpha_j, \beta_j] = \operatorname*{argmin}_{\alpha \in \mathbb{R}, \beta \in \mathbb{R}} h_j(\alpha, \beta), \qquad h_j(\alpha, \beta) := \|\mathcal{S}_j\| = \|W_j(\alpha, \beta) T_j^T(\alpha, \beta)\|. \tag{3.5}$$

The incorporation of complex shifts is straightforward although one has to take care of the case when one computed shift is a complex and the other a real one [5]. Of course, this approach is again very expensive since each function evaluation in an optimization routine alone requires to solve two shifted linear systems with multiple right-hand sides. Also, it is difficult to guarantee that a global minimum is found. Local minima might lead to a slower convergence. Because of these severe drawbacks, these norm-minimizing shifts are at the current stage only of theoretical interest.

3.3.2. Shifts obtained via projections with ADI iterates. It is easy to generalize the V- and W-shifts for Lyapunov equations in Section 2.3.2 to Sylvester equations. Assume we have at iteration j of Algorithm 3.1 the iterates V_j, S_j and W_j, T_j available. Then the next α- and β-shifts can be obtained via the following two approaches:

1. $\mathcal{A} = \Lambda(\hat{V}^T A \hat{V}, \hat{V}^T E \hat{V})$ and $\mathcal{B} = \Lambda(\hat{S}^T A \hat{S}, \hat{S}^T E \hat{S})$, where $\hat{V}, \hat{S}$ span orthonormal bases of V_j, S_j. As in the Lyapunov case, one can work with orthonormal bases of $[\mathrm{Re}\,(V_j), \mathrm{Im}\,(V_j)]$, $[\mathrm{Re}\,(S_j), \mathrm{Im}\,(S_j)]$ when V_j, S_j are complex iterates. We refer to this strategy as V-S-shifts.
2. $\mathcal{A} = \Lambda(\hat{W}^T A \hat{W}, \hat{W}^T E \hat{W})$ and $\mathcal{B} = \Lambda(\hat{T}^T A \hat{T}, \hat{T}^T E \hat{T})$, where $\hat{W}, \hat{T}$ span orthonormal bases of W_j, T_j. These quantities are always real matrices in the real formulation [5, Algorithm 2] of Algorithm 3.1. This strategy is called W-T-shifts from now on.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 3.1
Dimensions n, f, and m, maximum number of allowed ADI iterations $j^{\max}$, structural properties, and sources for the used Sylvester test examples. The desired tolerance $\epsilon_{\mathcal{S}}$ for the normalized residual norm is 10^{-10}.

Example	n	f	m	$j^{\max}$	Properties	Source
FDM2	6 400	3 600	5	50	$E = I_n$, $G = I_f$, B, C random	[22, Example 2]
rail5k/1k	5 177	1 377	6	150	A, F snd, E, G spd	OC[1], ID=38881
ifiss2	16 641	4 225	4	100	$E, -G$ spd, B, C random	IFISS [32] T-CD3

For both variants, the initial α- and β-shifts can be obtained similarly by using orthonormal bases of B and C, respectively. Due to the orthogonalization process it can happen that nearly linearly dependent columns in V_j, S_j or W_j, T_j are discarded and hence $\text{card}(\mathcal{A}) \leq m$ and $\text{card}(\mathcal{B}) \leq m$. Note that one should ensure that the new shifts satisfy $\alpha \neq \beta$. Also note that, because the numbers of initial α- and β-shifts does not have to be equal, new α- and β-shifts do not need to be calculated at the same time, but we restrict ourselves to this situation here for simplicity.

3.4. Other shifts. An overview over several other approaches for generating shifts for the Sylvester ADI, for instance, generalizations of the Leja point-based shifts, can be found in [31]. For a generalized version of the iteration (3.2), specialized shift strategies can be found in [23]. Shifts for Sylvester equations occurring in image restoration are proposed in [15].

3.5. Related matrix equations. Several other linear matrix equations where the unknown X appears twice are special classes of the just discussed generalized Sylvester equation (3.1). Prominent examples are cross-Gramian Sylvester equations ($G = E$, $F = -A$), discrete-time Sylvester equations (interchange F and G), and generalized discrete-time Lyapunov equations ($G = A^T$, $F = E^T$, $B = C$), which are also known as Stein equations. Of course, the generalized Lyapunov equations (2.1) discussed in Section 2 also belong to this class. Exploiting the structure of these special cases, specially tailored low-rank ADI methods can be formulated [5, Section 4], and consequently the shift strategies discussed so far can be adapted accordingly.

3.6. Numerical examples. In this section we test some of the proposed shift strategies for the Sylvester ADI with the same hard- and software setting as for the Lyapunov experiments. The examples are given in Table 3.1, where we use similar notations and abbreviations as for the Lyapunov examples (Table 2.1). The factors of the right-hand side for the example *rail5k/1k* were taken as the output matrices C^T provided in the associated OC example. In all examples, G-fADI was terminated when $\|\mathcal{S}\| < \epsilon_{\mathcal{S}} \|BC^T\|$ with $\epsilon_{\mathcal{S}} = 10^{-10}$ (or after $j^{\max}$ iterations).

The results are summarized in Table 3.2. There, the entries "optimal(k_+^A, k_-^A, k_+^F, k_-^F)" and "heur(J, L, k_+^A, k_-^A, k_+^F, k_-^F)" refer to the optimal and heuristic shift approaches as considered in Sections 3.2.1–3.2.2, respectively. For the optimal shifts, the obtained number J is also given. The `parsyl` routine is used to compute optimal shifts for the examples *FDM2* and *ifiss2*, where we modified `parsyl` such that (inverse) Arnoldi processed are used to obtain the approximate spectral data. This was more efficient than using `eigs` as it is done in the original `parsyl` implementation. For the example *rail5k/1k*, the approach given in [31, Algorithm 2.1] is employed since `parsyl` did not lead to good shifts for this examples. SyIRKA(J) and (Sy)^{2}IRKA(J) stands for J shifts generated with the (symmetric) Sylvester IRKA. These IRKA shifts and the similarly expensive residual norm-minimizing shifts were only applied for the smaller examples *FDM2* and *rail5k/1k*, where it was in both

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 3.2
Results for the Sylvester examples using different shift strategies: t_{shift} and t_{ADI} denote the times spend for computing the shifts and executing G-fADI, respectively, and the total consumed time is t_{total}. The required iterations j^{iter} and the final obtained residual norm $\|\mathcal{S}_{j^{iter}}\|$ are also given. All timings are given in seconds. The smallest values of t_{total} and j^{iter} are emphasized by bold letters.

Ex.	Shift strategy	t_{shift}	t_{ADI}	t_{total}	j^{iter}	$\|\mathcal{S}_{j^{\text{iter}}}\|$
FDM2	heur(20, 10, 10, 20, 10, 10)	0.73	2.80	3.53	34	$9.87{\cdot}10^{-11}$
	optimal(10, 10, 10, 10), $J=10$	0.97	3.44	4.41	40	$3.08{\cdot}10^{-11}$
	SyIRKA(20)	186.87	2.60	189.47	31	$3.45{\cdot}10^{-11}$
	res.min	343.03	2.52	345.55	**28**	$5.93{\cdot}10^{-11}$
	V-S-shifts	0.42	2.61	**3.03**	31	$8.97{\cdot}10^{-11}$
	W-T-shifts	0.42	2.95	3.37	34	$1.75{\cdot}10^{-11}$
rail5k/1k	heur(40, 40, 20, 20, 20, 20)	1.27	5.20	6.47	74	$6.52{\cdot}10^{-11}$
	optimal(10, 5, 10, 5), $J=70$	0.31	4.72	5.03	63	$1.43{\cdot}10^{-11}$
	SyIRKA(60)	37.24	6.48	43.72	101	$2.54{\cdot}10^{-12}$
	res.min	183.62	8.89	192.51	111	$8.06{\cdot}10^{-11}$
	V-S-shifts	0.04	3.45	**3.50**	**50**	$6.46{\cdot}10^{-12}$
	W-T-shifts	0.10	10.41	10.51	150	$9.32{\cdot}10^{-5}$
ifiss2	heur(30, 30, 10, 20, 10, 20)	5.49	29.68	35.18	89	$8.57{\cdot}10^{-11}$
	optimal(10, 10, 10, 10), $J=15$	3.65	29.78	33.43	98	$7.65{\cdot}10^{-11}$
	V-S-shifts	0.16	25.41	**25.57**	**76**	$5.78{\cdot}10^{-12}$
	W-T-shifts	0.14	31.46	31.60	99	$4.46{\cdot}10^{-11}$

examples sufficient to restrict the computation to real residual norm-minimizing shifts. The construction of the V-S- and W-T-shifts was carried out using the `orth` command. Figure 3.1 shows the curves of the residual norm against the iteration number (top plots) as well as the consumed iteration time (bottom plots) for these two examples.

To some extent, similar observations can be made as in the Lyapunov examples. For the heuristic shifts, the time t_{shift} needed for their generation is a significant portion of the overall computational time t_{total}. They, however, manage to achieve the desired accuracy within $j^{\max}$ iterations for all examples. Compared to the heuristic shifts, the optimal shifts required smaller values of k_+^A, k_-^A, k_+^F, k_-^F to get the necessary spectral data. The top right plot of Figure 3.1 corresponding to example *rail5k/1k* reveals that they converge similarly to the Wachspress shifts for Lyapunov equations with real spectra. However, the required setup numbers seems to be highly influential for their performance. Different values than the ones used here lead to a different, often slower, convergence especially for the examples *FMD2* and *ifiss2*, which involve complex spectra. In terms of the required iterations j^{iter}, the IRKA shifts only work well for example *FDM2*. In example *rail5k/1k* they lead to a much higher value of j^{iter} as shown in the top right plot of Figure 3.1. Since their generation time is much larger than the actual ADI iteration time t_{ADI}, they are not a reasonable choice, which is also visible from the bottom plots in Figure 3.1. Similar to the corresponding Lyapunov examples we observed in further tests a strong dependence on the initial data for SyIRKA and $(\text{Sy})^2$IRKA. The residual-minimizing shifts require the longest generation time but lead to the smallest number j^{iter} for *FDM2*, where they also show a monotonically decreasing residual norm in Figure 3.1 (top left plot). For *rail5k/1k* this is not the case for similar reasons as in the Lyapunov example *rail5k*: the detection of minima of (3.5) which are not global minima. In other tests, the number of iterations j^{iter} was always smaller than for the other shifts provided

ETNA
Kent State University
http://etna.math.kent.edu

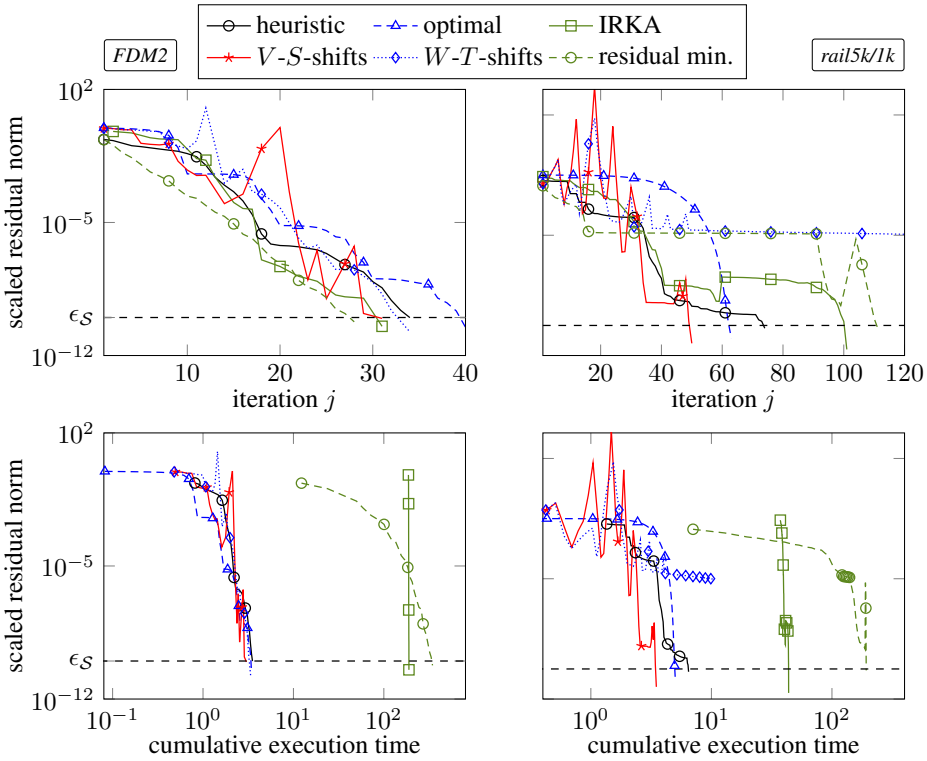

FIG. 3.1. *Scaled residual norm against iteration index j (top plots) and cumulative execution time at iteration j (bottom plots) of G-fADI using different shift strategies for FDM2 (left plots) and rail5k/1k (right plots) example.*

that global minima of (3.5) were used. Therefore, improving their computation and ensuring that global minima are found is current research.

As before, the shifts obtained from projections to spaces spanned by G-fADI iterates or residual factors require only a very small generation time t_{shift}. However, the W-T-shifts do not achieve convergence for example *rail5k/1k*, which is somehow similar to the Lyapunov case. The V-S-shifts lead to the smallest times t_{total} for *FMD2*, *rail5k/1k*; see also the bottom plots in Figure 3.1. The residual history of both the V-S- and W-T-shifts seems to be highly oscillatory as it is clearly visible in the residual plot for *rail5k/1k* in Figure 3.1 (top right plot). There are very high spikes in $\|\mathcal{S}_j\|$ which appear to unnecessarily prolong the iteration. A closer investigation of this phenomenon revealed that, in terms of (3.3), these peaks are the result of shift α_k, β_k with $\rho(A_{\alpha_k,\beta_k})\rho(F_{\alpha_k,\beta_k}) \gg 1$. This indicates that the corresponding computed shifts α_k, β_k are of no good quality. Avoiding these infeasible shifts is currently investigated and might lead to a further performance improvement. Due to the small execution and generation times as well as the advantage that they are computed in an entirely automatic way, the V-S-shift are nevertheless competitive to the other approaches.

4. Summary. We discussed shift parameter strategies for low-rank ADI methods for solving large-scale Lyapunov and Sylvester equations. After reviewing some prominent approaches to compute shifts a priori, two novel strategies have been proposed which generate shifts automatically during the ADI iteration without requiring any setup data. The first one is intrinsically designed to compute the new shift such that the residual norm is minimized at each step, and the second one uses orthonormal spaces spanned by the current ADI iter-

ETNA
Kent State University
http://etna.math.kent.edu

ates to obtain a small number of Ritz values as next shifts. Especially the latter one showed impressive numerical results that outperformed the existing shift strategies with respect to the required execution time but in most cases also in terms of the required ADI iterations. To conclude, the proposed projection-based V- and V-S-shifts are definitely competitive to existing shift parameter approaches especially for problems with complex spectra. However, a sound theoretical explanation for their often outstanding performance is not known yet. For Sylvester equations, the proposed dynamically updated shifts can also lead to a very oscillatory residual behavior which deteriorates the convergence. The (approximate) optimal shifts appear to be the method of choice for real spectra. At the current stage, the newly proposed residual norm-minimizing shifts are not competitive regarding their computational performance. Currently, we are investigating efficient ways to solve the occurring optimization problems in an approximate and efficient way. We also plan to adapt the proposed approaches to low-rank Newton-ADI methods [9, 11, 12, 14] for solving algebraic Riccati equations.

Acknowledgements. We would like to express our appreciation of the nice preliminary results on the projection based shifts that Manuela Hund found in her Bachelor thesis in our group. Her findings motivated the detailed investigation of the V- and $V - S$-shifts. We thank Tobias Breiten for providing his implementations of IRKA and (Sy)^{2}IRKA. We are also grateful towards Eugene Wachspress as well as Ninoslav Truhar for helpful discussions regarding the optimal and heuristic shift parameters, respectively, for the Sylvester ADI. We further thank Serkan Gugercin who openly signed his referee report regarding our manuscript. His valuable suggestions helped improving the presentation and sharpening our main contribution.

REFERENCES

[1] A. C. ANTOULAS, D. C. SORENSEN, AND Y. ZHOU, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett., 46 (2002), pp. 323–342.
[2] E. BÄNSCH, P. BENNER, J. SAAK, AND H. K. WEICHELT, *Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flow*, Preprint SPP1253-154, DFG-SPP1253, University Erlangen, September 2013.
[3] P. BENNER, *Solving large-scale control problems*, IEEE Control Systems Magazine, 14 (2004), pp. 44–59.
[4] P. BENNER AND T. BREITEN, *On optimality of approximate low rank solutions of large-scale matrix equations*, Systems Control Lett., 67 (2014), pp. 55–64.
[5] P. BENNER AND P. KÜRSCHNER, *Computing real low-rank solutions of Sylvester equations by the factored ADI method*, Comput. Math. Appl., 67 (2014), pp. 1656–1672.
[6] P. BENNER, P. KÜRSCHNER, AND J. SAAK, *A reformulated low-rank ADI iteration with explicit residual factors*, PAMM. Proc. Appl. Math. Mech., 13 (2013), pp. 585–586.
[7] ———, *An improved numerical method for balanced truncation for symmetric second-order systems*, Math. Comput. Model. Dyn. Syst., 19 (2013), pp. 593–615.
[8] ———, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numer. Algorithms, 62 (2013), pp. 225–251.
[9] P. BENNER, J.-R. LI, AND T. PENZL, *Numerical solution of large-scale Lyapunov equations, Riccati Equations, and linear-quadratic control problems*, Numer. Linear Algebra Appl., 15 (2008), pp. 755–777.
[10] P. BENNER, R.-C. LI, AND N. TRUHAR, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.
[11] P. BENNER, H. MENA, AND J. SAAK, *On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations*, Electron. Trans. Numer. Anal., 29 (2007/08), pp. 136–149. http://etna.mcs.kent.edu/vol.29.2007-2008/pp136-149.dir/pp136-149.html
[12] P. BENNER AND J. SAAK, *A Galerkin-Newton-ADI method for solving large-scale algebraic Riccati equations*, Preprint SPP1253-090, DFG-SPP1253, University Erlangen, January 2010.
[13] ———, *Efficient balancing-based MOR for large-scale second-order systems*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 123–143.
[14] ———, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM-Mitt., 36 (2013), pp. 32–52.

ETNA
Kent State University
http://etna.math.kent.edu

[15] D. CALVETTI AND L. REICHEL, *Application of ADI iterative methods to the restoration of noisy images*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 165–186.
[16] V. DRUSKIN, L. KNIZHNERMAN, AND V. SIMONCINI, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.
[17] G. M. FLAGG AND S. GUGERCIN, *On the ADI method for the Sylvester equation and the optimal-H_2 points*, Appl. Numer. Math., 64 (2013), pp. 50–58.
[18] F. FREITAS, J. ROMMES, AND N. MARTINS, *Gramian-based reduction method applied to large sparse power system descriptor models*, IEEE Trans. Power Systems, 23 (2008), pp. 1258–1270.
[19] L. GRASEDYCK, *Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389.
[20] S. GUGERCIN, A. C. ANTOULAS, AND C. BEATTIE, *$\mathcal{H}_2$ model reduction for large-scale dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638.
[21] S. GUGERCIN, D. C. SORENSEN, AND A. C. ANTOULAS, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32 (2003), pp. 27–55.
[22] K. JBILOU, *Low rank approximate solutions to large Sylvester matrix equations*, Appl. Math. Comput., 177 (2006), pp. 365–376.
[23] N. LEVENBERG AND L. REICHEL, *A generalized ADI iterative method*, Numer. Math., 66 (1993), pp. 215–233.
[24] R.-C. LI AND N. TRUHAR, *On the ADI method for Sylvester equations*, Tech. Report 2008-02, Department of Mathematics, University of Texas at Arlington, Arlington, 2008.
[25] J.-R. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.
[26] V. MEHRMANN AND T. STYKEL, *Balanced truncation model reduction for large-scale systems in descriptor form*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2005, pp. 83–115.
[27] C. NOWAKOWSKI, P. KÜRSCHNER, P. EBERHARD, AND P. BENNER, *Model reduction of an elastic crankshaft for elastic multibody simulations*, Z. Angew. Math. Mech., 93 (2013), pp. 198–216.
[28] T. PENZL, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (1999/00), pp. 1401–1418.
[29] ———, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.
[30] J. SAAK, *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, PhD Thesis, Fakultät für Mathematik, TU Chemnitz, July 2009. http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642
[31] J. SABINO, *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, PhD Thesis, Computational and Applied Mathematics, Rice University, Houston, June 2007. http://www.caam.rice.edu/tech_reports/2006/TR06-08.pdf
[32] D. SILVESTER, H. ELMAN, AND A. RAMAGE, *Incompressible Flow and Iterative Solver Software (IFISS) version 3.2*, May 2012. http://www.maths.manchester.ac.uk/~djs/ifiss/
[33] V. SIMONCINI, *Computational methods for linear matrix equations*, Survey article, Dept. of Mathematics, University of Bologna, March 2013. http://www.dm.unibo.it/~simoncin/matrixeq.pdf
[34] D. C. SORENSEN AND Y. ZHOU, *Bounds on eigenvalue decay rates and sensitivity of solutions to Lyapunov equations*, Tech. Report TR02-07, Computational and Applied Mathematics, Rice University, Houston, June 2002.
[35] G. STARKE, *Optimal alternating direction implicit parameters for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1431–1445.
[36] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[37] N. TRUHAR AND K. VESELIĆ, *Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix*, Systems Control Lett., 56 (2007), pp. 493–503.
[38] ———, *An efficient method for estimating the optimal dampers' viscosity for linear vibrating systems using Lyapunov equation*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 18–39.
[39] B. VANDEREYCKEN AND S. VANDEWALLE, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2553–2579.
[40] E. L. WACHSPRESS, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Lett., 1 (1988), pp. 87–90.
[41] ———, *Optimum parameters for two-variable ADI iteration*, Ann. Nuclear Energy, 19 (1992), pp. 765–778.
[42] ———, *ADI iteration parameters for the Sylvester equation*, preprint available from the author, 2000.
[43] ———, *The ADI Model Problem*, Springer, New York, 2013.
[44] T. WOLF AND H. K. F. PANZER, *The ADI iteration for Lyapunov equations implicitly performs H2 pseudo-optimal model order reduction*, Preprint on ArXiV, 2013. http://arxiv.org/abs/1309.3985
[45] T. WOLF, H. K. F. PANZER, AND B. LOHMANN, *Model order reduction by approximate balanced truncation: a unifying framework*, at-Automatisierungstechnik, 61 (2013), pp. 545–556.