SIAM J. SCI. COMPUT.
Vol. 21, No. 4, pp. 1401–1418

A CYCLIC LOW-RANK SMITH METHOD FOR LARGE SPARSE LYAPUNOV EQUATIONS*

THILO PENZL†

Abstract. In this paper we present the cyclic low-rank Smith method, which is an iterative method for the computation of low-rank approximations to the solution of large, sparse, stable Lyapunov equations. It is based on a generalization of the classical Smith method and profits by the usual low-rank property of the right-hand side matrix. The requirements of the method are moderate with respect to both computational cost and memory. Furthermore, we propose a heuristic for determining a set of suboptimal alternating direction implicit (ADI) shift parameters. This heuristic, which is based on a pair of Arnoldi processes, does not require any a priori knowledge on the spectrum of the coefficient matrix of the Lyapunov equation. Numerical experiments show the efficiency of the iterative scheme combined with the heuristic for the ADI parameters.

Key words. ADI iteration, Smith method, iterative methods, Lyapunov equation, matrix equations

AMS subject classifications. 65F30, 65F10, 15A24, 93C05

PII. S1064827598347666

1. Introduction.

The Lyapunov matrix equation plays an important role in control theory. For example, it arises in stability analysis [22], the solution of Riccati matrix equations [18], model reduction [25, 33, 39], and H_∞ optimal control [10]. In this paper, we consider the Lyapunov equation

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

where the matrix $A \in \mathbb{R}^{n,n}$ is stable, i.e., its eigenvalues are contained in $\mathbb{C}_-$, which denotes the set of the complex numbers with negative real parts. Under this assumption a unique solution $X \in \mathbb{R}^{n,n}$ exists, which is symmetric and positive semidefinite, e.g., [21]. There are a number of direct methods for solving the Lyapunov equation (1.1) numerically, the most important of which are the Bartels–Stewart method [4] and the Hammarling method [13]. Unfortunately, these methods, which are based on the QR algorithm, ignore any sparsity in the equation and are not very attractive for parallelization. A rough estimation of the complexity of the Bartels–Stewart method and the Hammarling method gives about $25n^3$ flops and $3n^2$ words of memory. Note that we count one flop as a single floating point operation according to [11]. Although these methods should be considered as standard methods for small, dense Lyapunov equations, their use is very limited if large, sparse equations have to be solved. For example, dynamical systems arising from the discretization of parabolic differential equations lead to large, sparse Lyapunov equations, e.g., [29]. It is important to note that the number of columns in the matrix $B \in \mathbb{R}^{n,m}$, which is related to the number of inputs and outputs of the underlying dynamical system, is usually very small and does not depend on the fineness of the discretization. This fact is of importance for the method presented in section 4.

*Received by the editors November 19, 1998; accepted for publication (in revised form) February 25, 1999; published electronically February 2, 2000. This work was supported by the Deutsche Forschungsgemeinschaft within the Sonderforschungsbereich 393 "Numerische Simulation auf massiv parallelen Rechnern."

http://www.siam.org/journals/sisc/21-4/34766.html

†The author is deceased. Former address: Fakultät für Mathematik, Technische Universität Chemnitz, 09107 Chemnitz, Germany.

If large, sparse problems have to be solved, iterative schemes are often the method of choice because they do not destroy sparsity. Mostly, they are much more suitable for parallelization than direct methods. We briefly review two popular iterative methods for Lyapunov equations which do not benefit from the low-rank property of the right-hand side.

The iterates X_i^A of the *alternating direction implicit* (ADI) iteration [26, 43] are usually generated by the alternating solution of two linear systems with multiple right-hand sides

$$\begin{aligned}(A^T + p_i I)X_{i-1/2}^A &= -BB^T - X_{i-1}^A(A - p_i I),\\ (A^T + p_i I){X_i^A}^T &= -BB^T - {X_{i-1/2}^A}^T(A - p_i I),\end{aligned} \tag{1.2}$$

where $X_0^A = 0$ and the shift parameters $p_1, p_2, p_3, \ldots$ are elements of $\mathbb{C}_-$. This pair of equations is mathematically equivalent to the iteration step

$$\begin{aligned}X_i^A = (A^T - p_i I)(A^T + p_i I)^{-1}X_{i-1}^A(A - p_i I)(A + p_i I)^{-1}\\ -2p_i(A^T + p_i I)^{-1}BB^T(A + p_i I)^{-1}.\end{aligned} \tag{1.3}$$

The error matrices $D_i = X - X_i^A$ obey the recursion

$$D_l = \left(r_l(A)r_l(-A)^{-1}\right)^T D_0 r_l(A)r_l(-A)^{-1}, \tag{1.4}$$

where r_l is the polynomial $r_l(t) = (t - p_1) \times \cdots \times (t - p_l)$. The rate of convergence is dominated by the spectral radius of the error transfer operator given by (1.4). The minimization of this spectral radius with respect to the shift parameters $p_1, \ldots, p_l$ leads to the ADI minimax problem

$$\{p_1, \ldots, p_l\} = \operatorname*{argmin}_{\{p_1,\ldots,p_l\}\subset\mathbb{C}_-} \max_{t\in\sigma(A)} \frac{|r_l(t)|}{|r_l(-t)|}, \tag{1.5}$$

which delivers criteria for the optimal and suboptimal choice of these parameters. Here, $\sigma(A)$ denotes the spectrum of A. The minimax problem (1.5) is solved for equations with symmetric matrices A, e.g., [40]. Unfortunately, there is still a lack in theory for the general case, where the eigenvalues of A are possibly not real. Contributions to the solution of the complex ADI minimax problem can be found in [3, 6, 9, 23, 36, 37, 38, 42, 44], for example.

The *Smith method* [34] is derived from the Stein equation

$$X - S^T X S = T \tag{1.6}$$

with

$$S = (A - pI)(A + pI)^{-1} \qquad \text{and} \qquad T = -2p(A + pI)^{-T}BB^T(A + pI)^{-1}, \tag{1.7}$$

which is equivalent to (1.1) for any real $p < 0$. Under this assumption, the sequence $\{X_i^S\}_{i=0}^\infty$ generated by

$$X_0^S = 0, \qquad X_{i+1}^S = T + S^T X_i^S S \tag{1.8}$$

converges to the solution X and the iterates can be written as

$$X_i^S = \sum_{j=1}^{i} \left(S^{j-1}\right)^T T S^{j-1}. \tag{1.9}$$

The Smith method can be shown to be mathematically equivalent to ADI if $p = p_1 = p_2 = \ldots$, but in general it converges much slower than ADI with nonconstant shift parameters. Nevertheless, it has become quite popular since there exists an accelerated version—the so-called *squared Smith method*. This version is based on the recursion

$$X_0^S = 0, \qquad X_{2^0}^S = T, \qquad X_{2^{j+1}}^S = X_{2^j}^S + \left(S^{2^j}\right)^T X_{2^j}^S S^{2^j},$$

which is derived from (1.9). Thus, the subsequence $\{X_{2^j}^S\}_{j=0}^{\infty}$ can be obtained with linear computational cost. Despite the quadratic rate of convergence, one should be reluctant to apply the squared method to large, sparse equations. The matrices S^{2^j}, which have to be squared explicitly in each step of the iteration, are dense even if A is sparse.

If the structure of the matrix A enables the efficient solution of linear systems $(A^T + pI)x = y$, e.g., if A is a banded matrix, both ADI and the standard Smith method should be considered as sparse matrix techniques. However, since the iterates are formed explicitly in both methods, neither method can be applied if the dimension n becomes so large that dense $n \times n$ matrices cannot be stored in memory. There are only a few iterative methods which can really be used to solve very large, sparse Lyapunov equations, e.g., [12, 14, 15, 16, 17, 31]. These methods produce low-rank approximations to the solution X. The iterates are stored implicitly in factored form, which decreases the memory requirement significantly. However, these methods often fail to determine approximate solutions of high accuracy. Moreover, the rank of their approximations tends to be relatively large. This is indicated by the numerical experiments reported in the original references as well as in our own experiments with one of the classical low-rank methods introduced below.

The method we refer to as the *full orthogonalization method* for Lyapunov equations (FOM-L) [16, 17, 31] could be considered as an extension of FOM for systems of linear equations [30] to matrix equations. Note that this method is frequently called the Arnoldi method or the Galerkin method. FOM-L is based on the Arnoldi process (if $m = 1$) or the block Arnoldi process (if $m > 1$) applied to the matrices A^T and B. The purpose of this process is to establish an orthonormal basis, $V_k \in \mathbb{R}^{n,w}$ $(w \leq mk)$, in the Krylov subspace

$$\mathcal{K}_k(A^T, B) = \text{range}\left[\begin{array}{cccc} B & A^T B & \left(A^T\right)^2 B & \ldots \quad \left(A^T\right)^{k-1} B \end{array}\right].$$

For details of the Arnoldi process or its block version see, e.g., [2, 11, 45]. The FOM-L iterates X_k^F defined by

$$X_k^F = V_k \tilde{X}_k V_k^T$$

are required to fulfill the Galerkin condition

$$V_k^T \left(A^T X_k^F + X_k^F A + BB^T\right) V_k = 0.$$

Hence, $\tilde{X}_k \in \mathbb{R}^{w,w}$ is given by the solution of the Lyapunov equation

$$V_k^T A^T V_k \tilde{X}_k + \tilde{X}_k V_k^T A V_k = -V_k^T B B^T V_k. \tag{1.10}$$

If the symmetric part of A is negative definite, it can be shown by Bendixon's theorem (e.g., [24]) that $V_k^T A V_k$ is stable. Under this assumption the Lyapunov equation

(1.10) has a unique solution and the matrices $\tilde{X}_k$ and X_k^F are symmetric, positive semidefinite. If w is much smaller than n, this equation can be solved by direct standard methods. The problem with FOM-L is that it converges rather slowly in many cases, i.e., relatively large values of k and w are necessary to attain a quite accurate approximate solution. This in turn may cause problems because the dense $n \times w$ matrix V_k has to be stored in memory.

The remainder of this paper is organized as follows. In section 2, we introduce the cyclic Smith method, which is a fast converging generalization of the Smith method. It is related to the ADI iteration with cyclic shift parameters and yields a sequence of full-rank iterates. In section 3, we describe a phenomenon which strongly motivates low-rank iteration methods for a class of Lyapunov equations. Low-rank versions of ADI and the cyclic Smith method are proposed in section 4. For each of these methods a set of suboptimal ADI parameters is needed. In section 5, we propose a heuristic procedure for determining such parameters, which does not require any a priori knowledge of the spectrum of A. Numerical tests in section 6 demonstrate the efficiency of the cyclic low-rank Smith method combined with the heuristic procedure for determining ADI shift parameters. Conclusions are provided in section 7.

2. Smith(l)—a generalization of the Smith method. In this section, we study the special case of ADI where l different shift parameters are applied in a cyclic manner. In other words, we require $p_{i+jl} = p_i$ for $j = 1, 2, \ldots$ in (1.2). The practical importance of this special case is illustrated by an experiment with the following medium scale example.

Example 2.1 (see [15]). This example describes the boundary control of the heat flow in a thin rod. The discretization of the underlying parabolic differential equation by finite differences results in a dynamical system

$$\begin{aligned} \dot{x}(\tau) &= Ax(\tau) + Bu(\tau), \\ x(0) &= x_0, \\ y(\tau) &= Cx(\tau). \end{aligned} \tag{2.1}$$

This in turn leads to a Lyapunov equation of order $n = 400$. The matrices $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,1}$ are defined as

$$A = \begin{bmatrix} -1/h & 1/h & 0 & \cdots & 0 \\ 1/h & -2/h & 1/h & \ddots & \vdots \\ 0 & 1/h & -2/h & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1/h \\ 0 & \cdots & 0 & 1/h & -2/h \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1/h \end{bmatrix}$$

with $h = 1/(n+1)$.

We investigate the dependence of the convergence speed of ADI on l. To this end, we determine the extremal eigenvalues of the symmetric matrix A. After that we compute sets of optimal shift parameters for several values of l by an algorithm due to Wachspress [40, section 2].

Table 2.1 shows the numbers of iterations required to attain different relative residual norms (i.e., $\left\|A^T X_i + X_i A + BB^T\right\|_F / \left\|BB^T\right\|_F \leq tol$ with $tol = 10^{-4}$, $10^{-6}, \ldots$). It reveals two facts. First, the convergence is very slow for $l = 1$, which corresponds to the Smith method, but a moderate increase of l (say $l = 4$) accelerates

TABLE 2.1

ADI applied to Example 1. *Numbers of iterations required to attain different relative residual norms depending on* l *(the number of different shift parameters).*

l	Relative residual norm				
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
1	851	1368	1903	2448	3001
2	53	85	119	153	187
4	17	29	41	53	65
8	15	23	31	39	49
16	13	21	29	37	41
32	13	21	29	33	41

it to a high degree. Second, the speed of convergence is hardly improved by a further increase of l. In fact, this experiment and a number of further tests not reported here indicate that a relatively small number of different ADI parameters is sufficient to achieve rapid convergence of ADI.

Although the speed of convergence of the Smith iteration is often unsatisfactory, this method has two algorithmic advantages over ADI. First, there exists a squared version of the Smith method. Second, the low-rank version of the Smith method, which is presented in a generalized form in section 4, is more efficient with respect to the amount of computation than that of ADI. This leads to the question of whether there exists a generalization of the Smith method that is mathematically equivalent to the fast converging ADI with cyclic parameters. Indeed, such a generalization is easily derived from (1.4). If we extend the original definition of the matrices S and T given in (1.7) to

$$S = \prod_{j=1}^{l} (A - p_j I)(A + p_j I)^{-1} \qquad \text{and} \qquad T = X_l^A, \tag{2.2}$$

then (1.4) is equivalent to the Stein equation $X - S^T X S = T$, which has the same structure as (1.6). This Stein equation is the base for the generalized version of the Smith iteration we refer to as *cyclic Smith method* (Smith(l)). The only essential difference between the standard and the cyclic versions is that the matrix T is given explicitly in the first case, whereas in the second it is the result of l steps of the ADI iteration with shift parameters $p_1, \ldots, p_l$ applied to (1.1). Analogous to (1.8), the Smith(l) iterates are generated by the recursion

$$X_0 = 0, \qquad X_{(i+1)l} = T + S^T X_{il} S. \tag{2.3}$$

For consistency, we label these iterates by multiples of l. Note that, in contrast to ADI, standard Smith method, and FOM-L, the iterates of Smith(l) are not provided with an extra superscript. Using (1.4) and (2.2), it is easy to prove that $X_{il} = X_{il}^A$ actually holds for $i = 1, 2, \ldots$, if the ADI iterates X_{il}^A are generated by use of l-cyclic parameters. As a consequence, Smith(1) is identical with the classical Smith method. In fact, the implementation (2.3) of Smith(l) is not more favorable than that of the ADI iteration if sparse Lyapunov equations are to be solved. It should rather be considered as a preliminary for deriving the low-rank method LR-Smith(l) presented in section 4.

The squared version of Smith(l), which is derived analogously to that of the standard Smith method, is not considered here because it involves dense $n \times n$ matrices

in the computation. However, such a version may be of interest for large, dense Lyapunov equations. For $l = 2$, such a generalization of the squared Smith method has been proposed by Davison and Man [7].

3. A low-rank phenomenon. This section contains a brief discussion of a phenomenon that motivates low-rank methods. We describe this phenomenon by use of Example 1. Since the pair (A, B) is controllable and $m = 1$, the solution matrix X is invertible [8, section 4], i.e., rank $X = n = 400$. The matrix X is symmetric and positive semidefinite. Hence, its eigenvalues are positive, real numbers. We solved the Lyapunov equation numerically by the Bartels–Stewart method and computed the descending ordered eigenvalues λ_j $(j = 1, \ldots, 400)$ of the matrix X (using MATLAB). These values are plotted in Figure 3.1. Here, the range of the y-axis is $[\epsilon\lambda_1, \lambda_1]$, where λ_1 equals the spectral radius and ϵ is the usual machine epsilon ($\epsilon \approx 2.22 \cdot 10^{-16}$). The vast majority of the eigenvalues are smaller than the very small lower bound of this interval. In other words, the *numerical* rank of X is much smaller than the order n. We call the very accurate approximation of the solution matrices of a certain class of Lyapunov equations by matrices of relatively low rank the *low-rank phenomenon*. It is also reflected in the results of the tests in section 6 and by unpublished numerical experiments with Lyapunov equations, where A and B are generated at random and B consists only of a few columns. The low-rank property of the right-hand side is essential for the low-rank phenomenon. For example, the solution $X = I$ of the Lyapunov equation with $A = -\frac{1}{2}I$ and $B = I$ admits no good low-rank approximations.

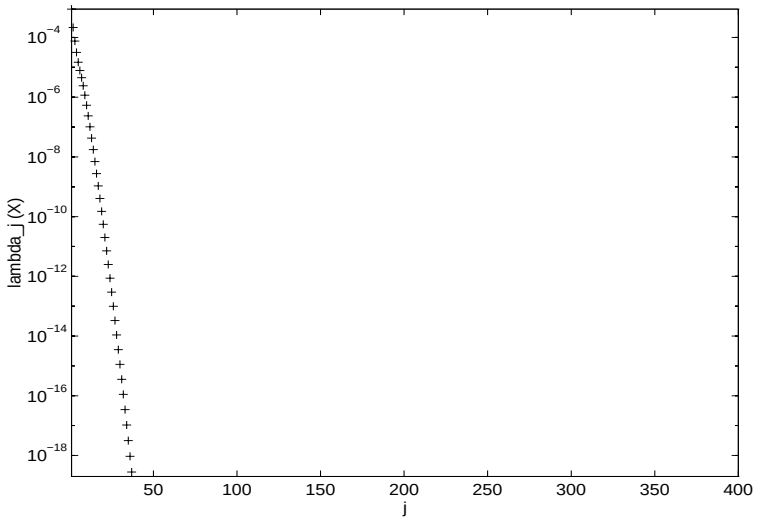

FIG. 3.1. *Example* 2.1. *Eigenvalues* λ_j *of* X *in descending order.*

The question of bounds, which describe the low-rank phenomenon or, more precisely, the fast decay of the eigenvalues of the solution, is still open. There exist a number of bounds for eigenvalues, eigenvalue sums, eigenvalue products, etc. of Lyapunov solutions (see [20] for a survey), but these bounds do not take the low-rank property of the right-hand side into account and cannot be used to explain the phenomenon.

4. LR-ADI and LR-Smith(l)—low-rank versions of ADI and Smith(l). For the remaining part of this paper, we assume $m \ll n$. This enables us to establish a low-rank version of ADI we will refer to as LR-ADI. Theoretically, this iteration could be used to solve the Lyapunov equation (1.1). However, we consider the derivation of LR-ADI as a preliminary for the construction of the cyclic low-rank Smith method, which is more interesting from the practical point of view due to its superior efficiency.

The key idea of the low-rank methods is to substitute the ADI and Smith(l) iterates by the products

$$X_i^A = Z_i^A {Z_i^A}^T \qquad \text{and} \qquad X_{il} = Z_{il} Z_{il}^T, \tag{4.1}$$

respectively. This is always possible because the iterates X_i^A and X_{il} can be shown recursively to be symmetric and positive semidefinite. Although a similar approach is pursued in several methods for Lyapunov equations (e.g., [1, 5, 13, 16, 17, 31]), this has not been done in combination with ADI or Smith-like methods. A combination of the terms "low-rank" and "ADI" can be found in [44], but both terms are used there in a completely different context.

The *low-rank ADI* method (LR-ADI) is based on the ADI single sweep (1.3). Using (4.1) this formula can be rewritten in terms of the matrices Z_i^A as

$$Z_i^A = \left[(A^T - p_i I)(A^T + p_i I)^{-1} Z_{i-1}^A \quad \sqrt{-2p_i}(A^T + p_i I)^{-1} B \right] \tag{4.2}$$

with

$$Z_1^A = \sqrt{-2p_1}(A^T + p_1 I)^{-1} B.$$

The number of columns in the matrices Z_i^A is enlarged by m in each step of the iteration and $\operatorname{rank}(X_i^A) \le mi$. Although the memory requirement and the computational cost per iteration are linearly raising, LR-ADI can be considered as an efficient iterative scheme for solving large, sparse Lyapunov equations because the number of ADI iterations is generally much smaller than the dimension of the problem. In particular, LR-ADI is of interest if a sequence $\{p_i\}_{i=1}^{\infty}$ of different shift parameters (e.g., [3, 36, 38]) is available.

From the numerical point of view, the iteration (4.2) has two drawbacks if the sequence of the shift parameters contains nonreal numbers. First, the matrices Z_i^A are not real although the solution matrix X is real. Second, the iteration involves complex matrix operations, which are disadvantageous for a computer implementation. These problems can be avoided by assuming that the parameter sequence consists exclusively of real numbers and complex pairs $\{p_i, p_{i+1}\}$ with $p_i = \bar{p}_{i+1}$. In practice, this is not a strong restriction. For example, Algorithm 5.1 in section 5 delivers a set of parameters that meets this requirement. The iterate Z_i^A should only be computed by (4.2) if p_i is real. If p_i is not real and $p_i = \bar{p}_{i+1}$, then Z_i^A is not formed and Z_{i+1}^A is computed directly from Z_{i-1}^A by the double step

$$Z_{i+1} = \left[\begin{array}{ccc} Z_{i+1}^{(I)} & Z_{i+1}^{(II)} & Z_{i+1}^{(III)} \end{array} \right] \tag{4.3}$$

with

$$\begin{aligned}
Z_{i+1}^{(I)} &= (A^T - \bar{p}_i I)(A^T - p_i I)(A^T + \bar{p}_i I)^{-1}(A^T + p_i I)^{-1} Z_{i-1}, \\
Z_{i+1}^{(II)} &= 2\sqrt{-\operatorname{Re} p_i}(A^T + \bar{p}_i I)^{-1}(A^T + p_i I)^{-1} A^T B \quad = \quad \frac{1}{|p_i|} A^T Z_{i+1}^{(III)}, \\
Z_{i+1}^{(III)} &= 2\sqrt{-\operatorname{Re} p_i}|p_i|(A^T + \bar{p}_i I)^{-1}(A^T + p_i I)^{-1} B
\end{aligned}$$

instead of by applying (4.2) twice. In this way, we obtain a subsequence of $\{Z_i^A\}_{i=1}^{\infty}$ consisting of real matrices. Complex operations can be avoided in (4.3) by computing products of complex matrices as

$$(A \pm p_i I)(A \pm \bar{p}_i I) = A^2 \pm 2 \operatorname{Re} p_i \; A + |p_i|^2 I. \tag{4.4}$$

However, a drawback of this procedure is that the nonzero structure of the product (4.4) is mostly less satisfactory than that of $A + p_i I$. This structure has an impact on the numerical cost for the solution of linear systems with these matrices.

If the number of different ADI parameters is limited, the *cyclic low-rank Smith method* (LR-Smith(l)) provides a more efficient alternative to LR-ADI. The algorithm LR-Smith(l) consists of two stages. First, the lth iterate Z_l^A of LR-ADI with the shift parameters $p_1, \ldots, p_l$ is computed. Analogous to (2.2), this matrix is used for the initialization

$$\begin{aligned} Z^{(l)} &= Z_l^A, \\ Z_l &= Z^{(l)}. \end{aligned}$$

Second, the actual LR-Smith(l) iteration is performed by

$$\begin{aligned} Z^{((i+1)l)} &= S^T Z^{(il)}, \\ Z_{(i+1)l} &= \begin{bmatrix} Z_{il} & Z^{((i+1)l)} \end{bmatrix}, \end{aligned} \tag{4.5}$$

where S is given by (2.2). If Z_l^A is real and the set $\{p_1, \ldots, p_l\}$ consists only of real numbers and conjugate complex pairs $\{p_i, p_{i+1}\}$, then all iterates Z_{il} are real matrices. Note that the computational cost per iteration step (4.5) is constant, which is an important advantage of LR-Smith(l) over LR-ADI. It is straightforward to prove that LR-Smith(l) is linked to Smith(l) by (4.1). Moreover, LR-Smith(l) and ADI are mathematically equivalent in the sense of $Z_{il} Z_{il}^T = X_{il}^A$, if the shift parameters $p_1, \ldots, p_l$ are used cyclically in the ADI iteration. However, determining Z_{il} by LR-Smith(l) is generally much more efficient than computing X_{il} by ADI if n is large and m is small.

Obviously, low-rank methods such as LR-ADI or LR-Smith(l) are only of practical interest if they deliver approximate solution factors Z_i^A or Z_{il}, which contain fewer columns than rows. Thus, these methods can fail if m is too large or if the ADI convergence is very slow. The latter may be caused either by the insufficient quality of the ADI shift parameters or by particularly bad algebraic properties of the matrix A (for example, an extremely large condition number in the symmetric case or eigenvalues of A which have very large imaginary parts). Attempts to retain the low-rank structure in case of a slow convergence are described in [27, section 7.5]. The basic idea is to replace the matrices Z_i^A repeatedly by matrices $\tilde{Z}_i^A$ in the LR-ADI iteration, where $\tilde{Z}_i^A$ contains fewer columns than Z_i^A and $\tilde{Z}_i^A \tilde{Z}_i^{A^T}$ approximates $Z_i^A Z_i^{A^T}$. However, for the majority of test examples, this procedure does not improve the results significantly.

The remaining part of this section addresses implementational aspects of LR-ADI and LR-Smith(l). Neither the matrices $(A^T + p_i I)^{-1}$ in LR-ADI, nor the matrix S in LR-Smith(l) are formed explicitly. Instead, the sparse matrices $A^T + p_i I$ are factorized a priori (e.g., by LU factorizations) and the iterations (4.2) and (4.5) involve forward and backward substitutions. Of course, a certain amount of fill-in is mostly produced by these factorizations. Nevertheless, this procedure is generally much more

efficient than computing the inverses of $A^T + p_i I$ explicitly if the matrix A is sparse. Alternatively, iterative methods can be utilized to solve sparse, linear systems of the type $(A^T + p_i I)x = y$, e.g., [32]. Note that these comments also apply to the linear systems with (4.4) as a coefficient matrix.

Theoretically, a squared version of LR-Smith(l) can be derived as well, but it requires forming the dense matrix S explicitly. Hence, such a method should not be applied to large, sparse Lyapunov equations since its memory requirement is $\mathcal{O}(n^2)$ and the computational cost is $\mathcal{O}(n^3)$ even if A is sparse.

In some control theory algorithms, only the product of X with a matrix V containing a few columns is sought instead of the solution matrix X itself. In this case LR-Smith(l) can be very efficient with respect to the memory requirement because the iterates Z_{il} need not be stored. The product XV can be evaluated by accumulating the sum on the right-hand side of

$$XV = \lim_{i\to\infty} Z_{il} Z_{il}^T V = \sum_{i=1}^{\infty} Z^{(il)} \left({Z^{(il)}}^T V \right) \tag{4.6}$$

in the course of the iteration (4.5). Such a procedure has been proposed in [31, section 3] in a similar context.

In general, it is not known a priori how many LR-Smith(l) steps are needed to attain a prescribed accuracy of the approximate solution. Therefore, it is necessary to compute repeatedly the Frobenius norm of the current residual matrix. In practice, this matrix should not be formed explicitly if the dimension of the problem is large. Rather one should compute the norm of the residual by

$$\begin{aligned} \left\| A^T Z_{il} Z_{il}^T + Z_{il} Z_{il}^T A + BB^T \right\|_F &= \left\| \begin{bmatrix} A^T Z_{il} & Z_{il} & B \end{bmatrix} \begin{bmatrix} Z_{il} & A^T Z_{il} & B \end{bmatrix}^T \right\|_F \\ &= \left\| R_i \begin{bmatrix} 0 & I_{ilm} & 0 \\ I_{ilm} & 0 & 0 \\ 0 & 0 & I_m \end{bmatrix} R_i^T \right\|_F , \end{aligned} \tag{4.7}$$

where $R_i \in \mathbb{R}^{(2il+1)m,(2il+1)m}$ is the square, upper, triangular matrix (or, more general, a column-permutation of such a matrix) resulting from an "economy size" QR decomposition $Q_i R_i = W_i = \begin{bmatrix} A^T Z_{il} & Z_{il} & B \end{bmatrix}$. As long as $(2il+1)m \ll n$, this procedure is much more efficient than the direct computation of the residual norm with respect to both computational cost and memory requirement. It can be improved further if the residual norm is computed after each step of the iteration. In this case, the desired factorization of W_i can be obtained from that of W_{i-1} by updating techniques for the QR factorization (e.g., [11, section 12.5]) because W_i is generated by insertion of $2lm$ columns into W_{i-1}. However, the determination of the residual norm is an issue that must not be underestimated. Note that the computation of the residual norm with the improved technique was still more expensive than the whole LR-Smith(l) iteration in some of our numerical experiments!

There are a number of approaches to a parallelization of LR-Smith(l). The factorization of the matrices $A^T + p_i I$ can be realized efficiently in parallel using l processors. Moreover, in (4.5) the products of S^T with the single columns of $Z^{(il)}$ can be computed simultaneously. The computational cost for realizing one step of (4.5) is proportional to m. If $m > 1$, the right-hand side matrix of the Lyapunov equation (1.1) can be split up into

$$-BB^T = -\sum_{j=1}^{m} b_j b_j^T$$

with $B = \begin{bmatrix} b_1 & \dots & b_m \end{bmatrix}$. This has been proposed in [16, section 5] in a similar fashion. For a parallel computer with m processors this offers an ideal parallelization, because the resulting m Lyapunov equations with right-hand side matrices of rank one can be solved simultaneously.

5. A heuristic procedure for determining suboptimal ADI shift parameters. The performance of the ADI-based methods described in sections 1, 2, and 4 depends strongly on the choice of the shift parameters p_i. The conventional approach to the computation of these parameters is to cover the spectrum of A by a domain $\Omega \subset \mathbb{C}_-$ and to solve the ADI minimax problem (1.5) with respect to Ω instead of $\sigma(A)$. For a few shapes of the domain Ω (e.g., intervals, rectangles, circles, trapezoids) optimal, or at least suboptimal, shift parameters have been found, e.g., [36, 41, 44]. Moreover, several procedures for constructing sequences of suboptimal ADI parameters for more general domains have been proposed in [3, 6, 36, 38]. However, all these approaches, which are based on approximation theory, require knowledge of certain bounds of the spectrum. In rare cases such bounds can be computed analytically, but mostly one has to determine bounds a priori by numerical methods. If A is a symmetric, banded matrix it is reasonable to compute the spectrum by the QR method. In the unsymmetric case the most simple approach might be to estimate the extreme eigenvalues of the symmetric and the skew-symmetric part of A by power iteration or inverse iteration (e.g., [11]), which deliver bounds for the spectrum of A by Bendixon's theorem, e.g., [24]. Of course, this procedure cannot be applied if the symmetric part of A is indefinite, since the rectangular obtained this way is not a subset of $\mathbb{C}_-$. Alternatively, estimates for the eigenvalues of A can be obtained by the Arnoldi process. However, this method can fail in the indefinite case, too, because it may deliver estimates with nonnegative real parts.

In this section, we propose a procedure for determining a set $\mathcal{P}$ of l different suboptimal ADI shift parameters without first finding a superset Ω of the spectrum. The resulting algorithm is easy to implement. Although it relies more on heuristics than on approximation theory, the numerical results are quite satisfactory. Our algorithm does not require any a priori knowledge of the spectrum of A. All information about this matrix is obtained by a pair of Arnoldi processes related to the matrix A and its inverse. We choose the initial vector r of these processes at random. The integers k_+ and k_- denote the numbers of Arnoldi steps in the processes for the matrices A and A^{-1}, respectively. Writing the result of $k = k_+$ Arnoldi steps with respect to (w.r.t.) the pair (A, r) as a matrix equation, we get (e.g., [11])

$$AV_k = V_{k+1}\tilde{H}_k$$

with $V_k \in \mathbb{R}^{n,k}$, $\tilde{H}_k \in \mathbb{R}^{k+1,k}$, $V_1 \in \operatorname{span}\{r\}$, $V_{k+1}^T V_{k+1} = I_{k+1}$, $(V_{k+1})_{(1:n,1:k)} = V_k$. Moreover,

$$H_k := (\tilde{H}_k)_{(1:k,1:k)} = V_k^T A V_k$$

is an upper Hessenberg matrix. This matrix and its eigenvalues are called *Ritz matrix* and *Ritz values*, respectively. It is well known that the set $\mathcal{R}_+ := \sigma(H_k)$ represents an approximation of the spectrum of A [2]. Repeating this procedure with the inverse of A delivers the set $\mathcal{R}_-$, the elements of which approximate the eigenvalues of A^{-1}. Consequently, the set $\mathcal{R} := \mathcal{R}_+ \cup 1/\mathcal{R}_-$ can be considered as an improved approximation of the spectrum of A. The elements of $\mathcal{R}$ are assumed to be pairwise distinct. In particular, multiple Ritz values are represented as a single element in

$\mathcal{R}_\pm$. The Ritz values obtained by the Arnoldi process tend to be located near the "outer" eigenvalues, i.e., the eigenvalues near the convex hull of the spectrum. In particular, eigenvalues of large magnitude are usually approximated well. In contrast, the elements of $\mathcal{R}_+$ are generally poor approximations to the eigenvalues near the origin. Therefore, we involve the set $1/\mathcal{R}_-$ to approximate these eigenvalues. In fact, this procedure can have a radical impact on the speed of convergence of the iteration, which is shown by an example at the end of this section.

The key idea of our heuristic procedure is to replace $\sigma(A)$ by $\mathcal{R}$ in (1.5), provided that $\mathcal{R} \subset \mathbb{C}_-$. Moreover, we choose the suboptimal ADI parameters $\mathcal{P} := \{p_1, \ldots, p_l\}$ among the elements of $\mathcal{R}$ because the function

$$s_{\mathcal{P}}(t) = \frac{|r_l(t)|}{|r_l(-t)|} = \frac{|(t-p_1) \times \ldots \times (t-p_l)|}{|(t+p_1) \times \ldots \times (t+p_l)|}$$

becomes small over $\sigma(A)$ if there is one of the shifts p_i in the neighborhood of each eigenvalue. Since the Lyapunov equations to be solved are real, we require $\mathcal{P} = \bar{\mathcal{P}}$. This enables the computation of matrices Z_{il}^A and Z_{il} which are real.

Based on these considerations, we determine the elements of $\mathcal{P}$ as follows. First, we detect the element $\rho_i \in \mathcal{R}$ which minimizes the function $s_{\{\rho_i\}}$ over $\mathcal{R}$. The set $\mathcal{P}$ is initialized by either $\{\rho_i\}$ or $\{\rho_i, \bar{\rho}_i\}$. Next, we successively augment the set $\mathcal{P}$ by the elements or pairs of elements of $\mathcal{R}$, for which the maximum of $s_{\mathcal{P}}$ is attained. In other words, the maximum of $s_{\mathcal{P}}$ with respect to the current set $\mathcal{P}$ is replaced by a zero in the refined function $s_{\mathcal{P}}$. This strategy is summarized in the following algorithm. Note that the heuristic applied in step 7 is related to an algorithm by Bagby [3]. The notation $\mathrm{card}(\mathcal{P})$ is used for the number of elements in the set $\mathcal{P}$. For simplicity we assume that $\mathrm{card}(\mathcal{R}_+) = k_+$, $\mathrm{card}(\mathcal{R}_-) = k_-$, and $\mathcal{R}_+ \cap 1/\mathcal{R}_- = \emptyset$.

ALGORITHM 5.1 (Suboptimal ADI parameters).

INPUT: A, l_0, k_+, k_-

OUTPUT: $\mathcal{P}$

1. Choose $r \in \mathbb{R}^n$ at random.

2. Perform k_+ steps of the Arnoldi process w.r.t. (A, r) and compute the set of Ritz values $\mathcal{R}_+$.

3. Perform k_- steps of the Arnoldi process w.r.t. (A^{-1}, r) and compute the set of Ritz values $\mathcal{R}_-$.

4. $\mathcal{R} = \{\rho_1, \ldots, \rho_{k_+ + k_-}\} := \mathcal{R}_+ \cup (1/\mathcal{R}_-)$

5. IF $\mathcal{R} \not\subset \mathbb{C}_-$, STOP

6. Detect i with $\max_{t \in \mathcal{R}} s_{\{\rho_i\}}(t) = \min_{\rho \in \mathcal{R}} \max_{t \in \mathcal{R}} s_{\{\rho\}}(t)$ and initialize

$$\mathcal{P} := \begin{cases} \{\rho_i\} & : \ \rho_i \text{ real} \\ \{\rho_i, \bar{\rho}_i\} & : \ \text{otherwise} \end{cases}.$$

WHILE $\mathrm{card}(\mathcal{P}) < l_0$

7. Detect i with $s_{\mathcal{P}}(\rho_i) = \max_{t \in \mathcal{R}} s_{\mathcal{P}}(t)$ and set

$$\mathcal{P} := \begin{cases} \mathcal{P} \cup \{\rho_i\} & : \ \rho_i \text{ real} \\ \mathcal{P} \cup \{\rho_i, \bar{\rho}_i\} & : \ \text{otherwise} \end{cases}.$$

END WHILE

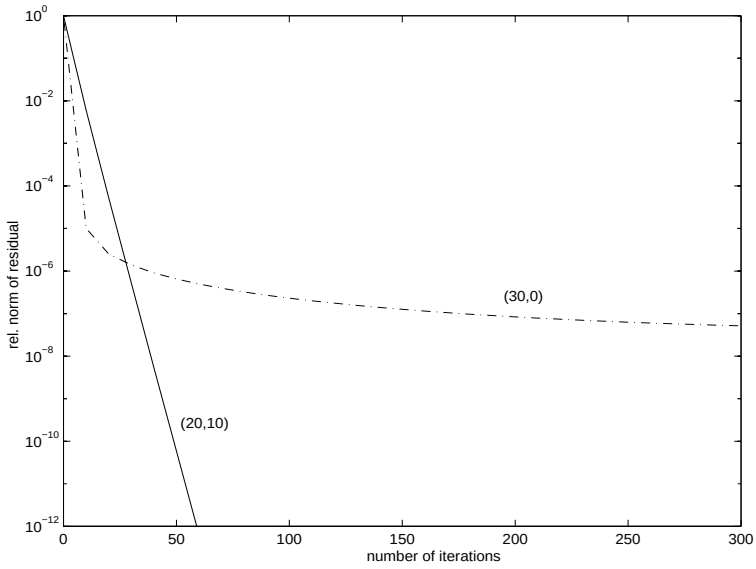

FIG. 5.1. *Example* 2.1. *Convergence of the LR-Smith*(10) *iteration with two different sets of shift parameters. These sets are determined by Algorithm* 5 *with* $(k_+, k_-) = (20, 10)$ *and* $(k_+, k_-) = (30, 0)$.

Step 5 can be omitted if $A + A^T$ is negative definite. This enables us to prove that $A^{-1} + A^{-T}$ and the symmetric parts of the Ritz matrices w.r.t. A and A^{-1} are negative definite. If $A + A^T$ is indefinite, the algorithm can fail, although this has never been observed in our numerical experiments. However, it is possible to construct a starting vector r which makes it fail in the indefinite case.

If Algorithm 5.1 stops at step 5, it can be restarted with a new random vector r or the values of k_+ or k_- can be increased. The latter is motivated by the observation that Ritz values obtained by the Arnoldi process tend to approximate the spectrum of a matrix better if the number of Arnoldi steps is enlarged. More sophisticated approaches may involve implicit restart techniques [35] to purge the sets $\mathcal{R}_+$ and $\mathcal{R}_-$ of elements with nonnegative real parts.

In the remainder of this section, we show the importance of employing the Ritz values w.r.t. A^{-1} in Algorithm 5.1. In Figure 5.1, we compare two runs of LR-Smith(10) applied to Example 2.1 with different sets of ADI shift parameters. For the first run, we include the Ritz values w.r.t. A^{-1} in the computation of the shift parameters by Algorithm 5.1. More precisely, we choose $(k_+, k_-) = (20, 10)$. In contrast, we set $(k_+, k_-) = (30, 0)$ for determining the parameter set $\mathcal{P}$ for the second run. In either case, $\mathcal{P}$ consists of $l = l_0 = 10$ elements.

Figure 5.1 shows that the convergence of LR-Smith(10) is fast and linear for the set $\mathcal{P}$ obtained by use of the parameters $(k_+, k_-) = (20, 10)$. In contrast, for $(k_+, k_-) = (30, 0)$ the convergence is very fast in the first stage, but it almost stagnates in the second. This phenomenon can be explained as follows. There are a few eigenvalues which are poorly approximated by the set $\mathcal{R}$. As a consequence, the function $s_{\mathcal{P}}(t)$ delivered by Algorithm 5.1 is almost 1 if t is equal to one of these eigenvalues, but it is relatively small if t belongs to the majority of eigenvalues which are approximated well. Thus, the component of the residual related to the latter type of eigenvalues is

quickly damped in the first stage of the iteration, whereas the iteration is delayed in the second stage by a small number of eigenvalues approximated poorly by $\mathcal{R}_+$. These eigenvalues, which are typically of small magnitude, are usually well represented by elements of $1/\mathcal{R}_-$.

6. Numerical experiments. In this section, we provide four examples of large-scale Lyapunov equations. We display the results of numerical experiments with LR-Smith(l) applied to these examples. For comparison, we also show the test results obtained by FOM-L. All experiments were carried out using MATLAB 5.1 and IEEE double precision arithmetic (machine precision $\epsilon \approx 2.22 \cdot 10^{-16}$) on an HP9000/800 workstation at the TU Chemnitz, Germany. We characterize the performance of the iterative methods by both the number of iterations and the number of flops required to attain prescribed tolerances for the accuracy, which is measured by the relative Frobenius norm of the residual

$$\frac{\left\|A^T X_i + X_i A + BB^T\right\|_F}{\left\|BB^T\right\|_F},$$

where X_i denotes the ith iterate of LR-Smith(l) or FOM-L. Note that for LR-Smith(l) each sweep of (4.5) is counted as l iterations.

The residual norm is determined after each of these sweeps. In our implementation, the norms of the residual are computed by (4.7) because the dimensions of the examples are too large to form the residual matrix explicitly. We consider the following examples in our tests.

Example 6.1. The structure of this example accords with that of the medium scale Example 2.1, but here the Lyapunov equation is of order $n = 10{,}000$.

Example 6.2. [15] This example corresponds to a second-order model of dimension n_0 which is equivalent to a dynamical system (2.1) of dimension $n = 2n_0$. The matrices $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,1}$ are given as

$$A = \begin{bmatrix} 0_{n_0} & I_{n_0} \\ A_{21} & -dI_{n_0} \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1/h^2 \end{bmatrix}$$

with

$$A_{21} = \begin{bmatrix} -k/h^2 & k/h^2 & 0 & \cdots & 0 \\ k/h^2 & -2k/h^2 & k/h^2 & \ddots & \vdots \\ 0 & k/h^2 & -2k/h^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & k/h^2 \\ 0 & \cdots & 0 & k/h^2 & -2k/h^2 \end{bmatrix} \in \mathbb{R}^{n_0,n_0},$$

where $n = 2n_0 = 3{,}000$, $h = 1/(n+1)$, $k = 10$, and $d = 1$. For our numerical experiments this system has been reordered by applying the permutation $(n_0 + 1, 1, n_0 + 2, 2, \ldots, 2n_0, n_0)$ to the columns and rows of A, such that it becomes a banded matrix with a very small bandwidth.

Example 6.3. This example describes a model of heat flow with convection in the domain $\Omega = (0,1)^2$. The underlying parabolic differential equation has the structure

$$\dot{\mathbf{x}} = \Delta \mathbf{x} - \mathbf{f_1}(\xi)\frac{\partial \mathbf{x}}{\partial \xi_1} - \mathbf{f_2}(\xi)\frac{\partial \mathbf{x}}{\partial \xi_2} + \mathbf{b}(\xi)u(\tau)$$

with $\mathbf{x} = \mathbf{x}(\xi, \tau)$, $\xi = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}^T \in \Omega$, $\tau \in [0, \infty)$. The coefficient functions in the convection term are defined as $\mathbf{f_1}(\xi) = 10\xi_1$ and $\mathbf{f_2}(\xi) = 1\,000\xi_2$. The differential equation is discretized by finite differences using a grid with equidistant spacing and 50×50 grid points. The resulting stiffness matrix $A \in \mathbb{R}^{2500,2500}$ is sparse and stable. Its bandwidth is 50. The matrix $B \in \mathbb{R}^{2500,1}$ is chosen at random.

Example 6.4. This example originates from a nonlinear descriptor system arising in chromatography. For more background information, we refer to [19]. This descriptor system has been linearized at a working point and transformed into a dynamical system (2.1). To reduce the bandwidth of the matrix A, we applied the reverse Cuthill–McKee algorithm, which is provided as the built-in function `SYMRCM` in MATLAB. The reordered matrix has the bandwidth $\max_{i,j:(A)_{ij}\neq 0} |i-j| = 31$. It is stable, but its symmetric part is indefinite. The Lyapunov equation is of order $n = 3{,}600$. The underlying dynamical system has four inputs ($m = 4$). In our numerical experiment we used the first column of the matrix B in (2.1) to create the right-hand side of the Lyapunov equation. Thus, the actual computational cost for solving the Lyapunov equation is about four times the number of flops given in Table 6.1.

Next, we investigate the convergence speed of LR-Smith(l) applied to Examples 6.1–6.4. It should be noted that these examples pose problems of very large scale. The properties of these examples are not favorable for iterative methods. Whereas the matrix A is symmetric, but ill-conditioned in Example 6.1, it has an indefinite symmetric part in Examples 6.2–6.4. Note that we do not restrict ourselves to examples with very small bandwidth, for which our method is most effective. For all examples we determine the shift parameters by Algorithm 5.1, where the input parameters (k_+, k_-, l_0) of this algorithm are chosen as (40,20,10) for Examples 6.1, 6.3, 6.4, and (60,0,20) for Example 6.2. The latter is one of the few examples where slightly better results are obtained by ignoring the Ritz values w.r.t. A^{-1} in Algorithm 5.1. It is worth noting that this algorithm did not fail for Examples 6.1–6.4 or in any of our further numerical tests, which are not reported here.

Table 6.2 displays the numbers of iterations required by LR-Smith(l) to attain different relative residual norms. For each example, LR-Smith(l) delivers, with reasonable convergence speed, a solution of satisfactory accuracy. Consequently, the rank of the approximate solutions and the memory size needed to store them are comparably low. Table 6.1 shows the computational costs in terms of the number of flops needed for the determination of the shift parameters and for the iteration itself. In this table, we do not take the costs for the computation of the residual norm into account.

TABLE 6.1
LR-Smith(l). Number of flops required to attain different relative residual norms.

Example	Relative residual norm				
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
6.1	$9.5 \cdot 10^7$	$1.3 \cdot 10^8$	$1.5 \cdot 10^8$	$1.9 \cdot 10^8$	$2.2 \cdot 10^8$
6.2	$2.2 \cdot 10^8$	$3.4 \cdot 10^8$	$3.4 \cdot 10^8$	$4.6 \cdot 10^8$	$> 1.8 \cdot 10^9$
6.3	$1.4 \cdot 10^9$	$1.7 \cdot 10^9$	$2.1 \cdot 10^9$	$2.3 \cdot 10^9$	$2.5 \cdot 10^9$
6.4	$7.0 \cdot 10^8$	$8.6 \cdot 10^8$	$1.2 \cdot 10^9$	$1.3 \cdot 10^9$	$> 4.9 \cdot 10^9$

For comparison, we provide the results delivered by FOM-L in Table 6.3. In contrast to LR-Smith(l), FOM-L fails to compute accurate solutions within a reasonable number of iterations in two cases. In Table 6.4, the corresponding numbers of flops are displayed. Furthermore, the last column of this table shows estimates for the expected

TABLE 6.2
LR-Smith(l). Number of iterations required to attain different relative residual norms.

Example	Relative residual norm				
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
6.1	30	50	60	80	100
6.2	42	63	63	84	> 315
6.3	50	60	80	90	100
6.4	40	50	70	80	> 300

computational cost of the Bartels–Stewart method. Note that it was impossible to solve the large scale Lyapunov equations of Examples 6.1–6.4 by the Bartels–Stewart method due to the extensive memory requirement of this method. However, even if the available memory was not the limiting factor for the application of this method, a comparison of the flop estimate for the Bartels–Stewart method with the number of flops required by LR-Smith(l) shows the superiority of our low-rank method.

TABLE 6.3
FOM-L. Number of iterations required to attain different relative residual norms.

Example	Relative residual norm				
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
6.1	49	> 300	> 300	> 300	> 300
6.2	61	95	203	> 300	> 300
6.3	166	207	254	298	> 300
6.4	> 300	> 300	> 300	> 300	> 300

TABLE 6.4
FOM-L and Bartels–Stewart method. Number of flops required by FOM-L to attain different relative residual norms. Estimates for the flop count of the Bartels–Stewart method are shown in the last column.

Ex.	Relative residual norm for FOM-L					Bartels–Stewart
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}	
6.1	$1.1 \cdot 10^8$	$> 4.2 \cdot 10^9$	$> 4.2 \cdot 10^9$	$> 4.2 \cdot 10^9$	$> 4.2 \cdot 10^9$	$2.5 \cdot 10^{13}$
6.2	$6.6 \cdot 10^7$	$1.8 \cdot 10^8$	$1.1 \cdot 10^9$	$> 2.7 \cdot 10^9$	$> 2.7 \cdot 10^9$	$6.8 \cdot 10^{11}$
6.3	$6.0 \cdot 10^8$	$1.0 \cdot 10^9$	$1.8 \cdot 10^9$	$2.7 \cdot 10^9$	$> 2.7 \cdot 10^9$	$3.9 \cdot 10^{11}$
6.4	$> 3.1 \cdot 10^9$	$> 3.1 \cdot 10^9$	$> 3.1 \cdot 10^9$	$> 3.1 \cdot 10^9$	$> 3.1 \cdot 10^9$	$1.2 \cdot 10^{12}$

7. Conclusions. This paper addresses the numerical solution of large, sparse, stable Lyapunov equations with right-hand-side matrices of low rank. We have presented the iterative methods LR-ADI and LR-Smith(l), which deliver low-rank approximations to the solution matrix. LR-ADI and LR-Smith(l) are mathematically equivalent to the ADI iteration with a sequence of arbitrary shift parameters or with a set of l cyclic shift parameters, respectively. In this paper, LR-Smith(l) is of particular interest because the proper choice of l different shift parameters, where l is of moderate size (say, $l = 10$), generally ensures a rapid convergence of ADI. The computational cost per iteration is constant for LR-Smith(l), but it is increasing for LR-ADI. Furthermore, we have presented a heuristic algorithm for determining a set of l suboptimal ADI parameters. The algorithm is easy to implement and does not require any a priori knowledge about the spectrum of the matrix A. All information about this matrix is gained from a pair of Arnoldi processes. Thus, LR-Smith(l) combined with the algorithm for determining the ADI parameters can be considered as a "black box" solver for large, sparse, stable Lyapunov equations. In general, the

computational costs of LR-Smith(l) and LR-ADI are much smaller than that of the classical implementation of the ADI iteration. Mostly, the memory requirements of both methods are moderate because the low rank iterates are efficiently stored in factored form. This even allows the solution of Lyapunov equations the order of which is so large that the explicit solution matrix cannot be stored in computer memory. In general, LR-Smith(l) converges fast compared to other low-rank methods, such as FOM-L. As a consequence, it tends to deliver approximate solutions of very low rank. For instance, in Example 6.1, the largest of our test examples, the solution matrix of order 10,000 is approximated quite accurately by a matrix of rank 100. Considering this example, a comparison of LR-Smith(l) with direct standard methods, such as Bartels–Stewart method or Hammarling method, reveals the efficiency of the low-rank method. The estimated computational and memory costs of the direct standard methods exceed those of LR-Smith(l) by factors 100,000 and 300, respectively!

Lyapunov equations are a part of several algorithms in control theory. If LR-Smith(l) is applied to determine the low-rank solutions of these equations, then one should check carefully if the subsequent computation of the explicit solution matrix is really necessary. In some cases there exist efficient alternative formulations of the "outer" algorithm which profit by the factored form of the solution matrix. This is discussed in more detail in a forthcoming paper [28].

Finally, we should point out two aspects of our low-rank methods which can become disadvantageous in some situations. Both LR-ADI and LR-Smith(l) require the solution of systems of linear equations $(A^T + pI)x = y$. If the nonzero pattern of the matrix A is unfavorable, the solution of these systems and the iterative methods may be expensive with respect to both memory and computation. However, most sparse matrices arising in applications are either banded matrices or matrices which can be reordered to achieve this structure. The second drawback of our methods is the restriction to Lyapunov equations with right hand side matrices of small rank. Nevertheless, if $m \gg 1$, splitting up the right hand side matrix into a sum of low-rank matrices enables an efficient parallelization of our method.

Acknowledgments. The author is grateful to E. Kloppenburg for providing Example 6.4 and to E. Quintana-Orti for useful comments.

REFERENCES

[1] F. Aliev and V. Larin, *Construction of square root factor for solution of the Lyapunov matrix equation*, Systems Control Lett., 20 (1993), pp. 109–112.

[2] W. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[3] T. Bagby, *On interpolation by rational functions*, Duke Math. J., 36 (1969), pp. 95–104.

[4] R. Bartels and G. Stewart, *Solution of the matrix equation $AX + XB = C$: Algorithm* 432, Comm. ACM, 15 (1972), pp. 820–826.

[5] P. Benner and E. Quintana-Orti, *Solving stable generalized Lyapunov equations with the matrix sign function*, Numer. Algorithms, 20 (1999), pp. 75–100.

[6] 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.

[7] E. Davison and F. Man, *The numerical solution of $A^TQ + QA = -C$*, IEEE Trans. Automat. Control, 13 (1968), pp. 448–449.

[8] E. de Souza and S. Bhattacharyya, *Controllability, observability and the solution of $AX - XB = C$*, Linear Algebra Appl., 39 (1981), pp. 167–188.

[9] N. Ellner and E. Wachspress, *Alternating direction implicit iteration for systems with complex spectra*, SIAM J. Numer. Anal., 28 (1991), pp. 859–870.

[10] B. A. Francis, *A Course in H_∞ Control Theory*, Lecture Notes in Control and Information Sciences 88, Springer-Verlag, New York, 1987.

[11] G. Golub and C. V. Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[12] T. Gudmundsson and A. Laub, *Approximate solution of large sparse Lyapunov equations*, IEEE Trans. Automat. Control, 39 (1994), pp. 1110–1114.
[13] S. Hammarling, *Numerical solution of the stable, non–negative definite Lyapunov equation*, IMA J. Numer. Anal., 2 (1982), pp. 303–323.
[14] A. Hodel and K. Poolla, *Heuristic approaches to the solution of very large sparse Lyapunov and algebraic Riccati equations*, in Proceedings of the 27th IEEE Conference Decis. Cont., Austin, TX, December 1988.
[15] A. Hodel, K. Poolla, and B. Tenison, *Numerical solution of the Lyapunov equation by approximate power iteration*, Linear Algebra Appl., 236 (1996), pp. 205–230.
[16] D. Hu and L. Reichel, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.
[17] I. Jaimoukha and E. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[18] D. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.
[19] E. Kloppenburg and E. D. Gilles, *Automatic control of the simulated moving bed process for* C_8 *aromatics separation using asymptotically exact input/output-linearization*, J. Process Control, 9 (1999), pp. 41–50.
[20] W. Kwon, Y. Moon, and S. Ahn, *Bounds in algebraic Riccati and Lyapunov equations: A survey and some new results*, Internat. J. Control, 64 (1996), pp. 377–389.
[21] P. Lancaster and M. Tismenetsky, *Theory of Matrices*, 2nd ed., Academic Press, Orlando, FL, 1985.
[22] J. LaSalle and S. Lefschetz, *Stability of Lyapunov's Direct Method*, Academic Press, New York, 1961.
[23] N. Levenberg and L. Reichel, *A generalized ADI iterative method*, Numer. Math., 66 (1993), pp. 215–233.
[24] M. Marcus and M. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Dover Publications, New York, 1992.
[25] B. C. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–31.
[26] D. Peaceman and H. Rachford, *The numerical solution of elliptic and parabolic differential equations*, J. Soc. Indust. Appl. Math., 3 (1955), pp. 28–41.
[27] T. Penzl, *Numerische Lösung großer Lyapunov-Gleichungen*, Ph.D. thesis, Logos-Verlag, Berlin, 1998.
[28] T. Penzl, *Algorithms for Model Reduction of Large Dynamical Systems*, Numer. Algorithms, submitted.
[29] I. Rosen and C. Wang, *A multi-level technique for the approximate solution of operator Lyapunov and algebraic Riccati equations*, SIAM J. Numer. Anal., 32 (1995), pp. 514–541.
[30] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.
[31] Y. Saad, *Numerical solution of large Lyapunov equations*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. Kaashoek, J. V. Schuppen, and A. Ran, eds., Birkhäuser, Boston, MA, 1990, pp. 503–511.
[32] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[33] M. Safonov and R. Chiang, *A Schur method for balanced-truncation model reduction*, IEEE Trans. Automat. Control, 34 (1989), pp. 729–733.
[34] R. Smith, *Matrix equation* $XA + BX = C$, SIAM J. Appl. Math., 16 (1968), pp. 198–201.
[35] D. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[36] G. Starke, *Optimal alternating direction implicit parameters for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1431–1445.
[37] G. Starke, *Near-circularity for the rational Zolotarev problem in the complex plane*, J. Approx. Theory, 70 (1992), pp. 115–130.
[38] G. Starke, *Fejer–Walsh points for rational functions and their use in the ADI iterative method*, J. Comput. Appl. Math., 46 (1993), pp. 129–141.
[39] M. Tombs and I. Postlethwaite, *Truncated balanced realization of stable, non-minimal state-space systems*, Internat. J. Control, 46 (1987), pp. 1319–1330.
[40] E. Wachspress, *Extended application of alternating direction implicit iteration model problem theory*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 994–1016.

[41] E. Wachspress, *Iterative Solution of Elliptic Systems*, Prentice–Hall, New York, 1966.
[42] E. Wachspress, *The ADI minimax problem for complex spectra*, Appl. Math. Lett., 1 (1988), pp. 311–314.
[43] E. Wachspress, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Lett., 1 (1988), pp. 87–90.
[44] E. Wachspress, *The ADI minimax problem for complex spectra*, in Iterative Methods for Large Linear Systems, D. Kincaid and L. Hayes, eds., Academic Press, San Diego, 1990, pp. 251–271.
[45] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.