Memory-efficient Krylov subspace techniques for solving large-scale Lyapunov equations

Daniel Kressner*

***Abstract*— This paper considers the solution of large-scale Lyapunov matrix equations of the form $AX+XA^T = -bb^T$. The Arnoldi method is a simple but sometimes ineffective approach to deal with such equations. One of its major drawbacks is excessive memory consumption caused by slow convergence. To overcome this disadvantage, we propose two-pass Krylov subspace methods, which only compute the solution of the compressed equation in the first pass. The second pass computes the product of the Krylov subspace basis with a low-rank approximation of this solution. For symmetric A, we employ the Lanczos method; for nonsymmetric A, we extend a recently developed restarted Arnoldi method for the approximation of matrix functions. Preliminary numerical experiments reveal that the resulting algorithms require significantly less memory at the expense of extra matrix-vector products.**

I. INTRODUCTION

In this paper, we investigate numerical methods for approximating the solution $X \in \mathbb{R}^{n\times n}$ to a *Lyapunov equation* of the form

$$AX + XA^T = -bb^T, \qquad (1)$$

where $A \in \mathbb{R}^{n\times n}$ is supposed to be stable (i.e., its eigenvalues lie in the open left half plane) and $b \in \mathbb{R}^n$. The solution X exists, is unique and symmetric positive semidefinite.

A variety of applications in systems and control theory lead to such equations. For example, the computationally most expensive step in balanced truncation model reduction [2] consists of solving two Lyapunov equations. Also, the Newton method for solving continuous-time algebraic Riccati equations [21], arising in optimal and robust control, requires the solution of a sequence of Lyapunov equations. In many applications, it is more natural and general to consider a right-hand side in (1) that has rank larger than 1. However, to avoid technical complications, we restrict ourselves to rank 1. In principle, this poses no limitation as any Lyapunov equation with right-hand side of rank k can be written as the sum of k independent equations of the form (1).

There exists a variety of numerical methods to address (1), roughly classified as dense and sparse methods in the following.

Dense methods. The most naive approach is to write (1) as a linear system of n^2 equations and apply Gaussian elimination with partial pivoting. The requires $\mathcal{O}(n^6)$ floating point operations (flops) and $\mathcal{O}(n^4)$ memory; too expensive for any application of practical relevance. Direct solvers, based on the Schur decomposition of A, bring these costs down to $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ memory. The *Bartels-Stewart method* [3], [20] is such a direct method. However, for a Lyapunov equation of the form (1) a variant called *Hammarling's method* [15], [24] is more suitable, as it takes the low rank structure of the right-hand side into account and directly produces a Cholesky factorization of X. As n increases, these methods exceed the capacities of a serial computing environment. Unfortunately, the parallelization of the Bartels-Stewart method [12], [13] is far from being straightforward, particularly because of the need for parallelizing the initial Schur decomposition. The *matrix sign function iteration* [4] offers a conceptually simpler alternative by requiring only matrix multiplication and inversion.

Sparse methods. As n increases further, say $\mathcal{O}(10^5)$ and larger, dense methods become unsuitable. Iterative methods, taking sparsity or other structure in A into account, must be used. Several of these methods are based on projection. Given a search space, the Lyapunov equation is projected onto this space, yielding a much smaller compressed equation which can be solved by any of the dense methods. An approximation to X is obtained by projecting the solution of the compressed equation back to the original space $\mathbb{R}^n$. These methods mainly differ in the way the search space is built. The alternating-direction implicit (ADI) [14], [26], [29], [37] are based on repeated matrix-vector multiplications with $(A-\sigma_j I)^{-1}$ for a suitably chosen set of shifts $\sigma_1, \dots, \sigma_m$. This usually yields quick convergence, particularly for symmetric A, but also requires that the solution of the corresponding linear systems can be performed

*Seminar für angewandte Mathematik, ETH Zürich, Switzerland. kressner@sam.math.ethz.ch

exactly, e.g., by sparse direct methods [6]. On the other hand, *Krylov subspace methods* [17], [18], [19], [32] converge significantly slower but only require the multiplication with A and henceforth no solution of linear systems.

The purpose of this paper is to remedy some of the disadvantages caused by the slow convergence of Krylov subspace methods. In particular, slow convergences yields large search space dimensions, which in turn leads to excessive memory requirements for storing a basis of the search space. The solution proposed in this paper is to store only a fraction of this basis during the generation of the search space. In a second pass of the Krylov subspace method the basis is multiplied on-the-fly with a low-rank approximation to the solution of the compressed equation. This concept can be realized in a rather direct way for symmetric A; the Lanczos process produces the basis by storing only 2 vectors at a time [11]. For nonsymmetric matrices, choosing a suitable low-memory Krylov subspace method is much less trivial. For example, there is no experience with using a nonsymmetric Lanczos method in this context. Instead, we will extend a restarted method for matrix functions developed in [9], requiring the storage of only a fixed number of vectors at a time, largely independent of the rate of convergence.

The rest of this paper is organized as follows. In Section II, properties of the basic Arnoldi algorithm for solving Lyapunov equations are recalled. A two-pass Lanczos method for symmetric A is proposed and analyzed in Section III, while a two-pass restarted Arnoldi method for nonsymmetric A is sketched in Section IV.

II. THE BASIC ARNOLDI METHOD

The Arnoldi method for solving (1) was introduced by Saad [32] and extended to right-hand sides of larger rank by Jaimoukha and Kasenally [18]. To describe its basic idea, let the columns of $U_k \in \mathbb{R}^{n\times l}$ span an orthonormal basis for the Krylov subspace

$$\mathcal{K}_k(A, b) = \text{span}(b, Ab, A^2b, \dots, A^{k-1}b).$$

Consider the *compressed Lyapunov equation*

$$H_k Y + Y H_k^T = -(U_k^T b)(U_k^T b)^T \qquad (2)$$

and assume the matrix $H_k = U_k^T A U_k$, sometimes called the *compression of* A, to be stable. Note that the stability of A is not enough to guarantee this to be true. A well-known sufficient (but not necessary) condition for the stability of H_k is the negative definiteness of $(A^T+A)/2$, the symmetric part of A. An approximate solution to (1) is then obtained by setting

$$X_k = U_k Y U_k^T. \qquad (3)$$

As (hopefully) $k \ll n$, the standard dense methods listed above can be applied to solve (2). Specifically, if Hammarling's method is used, we directly obtain the Cholesky factorization $Y = LL^T$ and only the $n \times l$ matrix $U_k L$ needs to be stored to represent X_k.

In [17], [18], other ways of obtaining an approximate solution to (1) from a Krylov subspace have been considered, meeting, e.g., a minimum residual criterion. However, as observed in [32], the Galerkin formulation (3) has the advantage of admitting the integral representation

$$X_k = \int_0^\infty e^{A_k t} b b^T e^{A_k^T t} \, \mathrm{d}t, \qquad (4)$$

with $A_k = U_k H_k U_k^T = (U_k U_k^T) A (U_k U_k^T)$. Since $e^{A_k t}B$ happens to be the standard Krylov subspace approximation to $e^{At}B$, we can facilitate existing theoretical results on matrix exponential approximation. For example, error bounds on the matrix exponential [8], [10], [16], [22], [34] can be turned into bounds on $\|X - X_k\|_2$. Alternatively, in [33] a direct approach for estimating the convergence is proposed.

The convergence of X_k towards X is often painfully slow, especially if A has eigenvalues very close to the imaginary axis. Unfortunately, this is typically the case, e.g., if A is the discretization of a compact self-adjoint operator on a Hilbert space. The following examples illustrates the convergence.

Example 1: Let A be the standard central finite difference discretization of the 2D Laplace operator on the unit square with $n = 30 \times 30 = 900$ and choose $b \in \mathbb{R}^{900}$ randomly. Then A is symmetric negative definite with $\lambda_{\min}(A) \approx -7670$ and $\lambda_{\max}(A) \approx -19$. Figure 1 displays the convergence of the Arnoldi method.

In particular, Figure 1 reveals that the observed convergence of Krylov subspace methods is much too slow to capture the rapid decay of the singular values of X [1], [30]. This implies that k, the rank of the obtained approximation X_k, is much larger than needed for actually approximating X. In other words, the memory requirements of the Arnoldi method are unnecessarily high.

A. Implementation details

Before attempting to reduce the memory requirements of the Arnoldi method, it is important

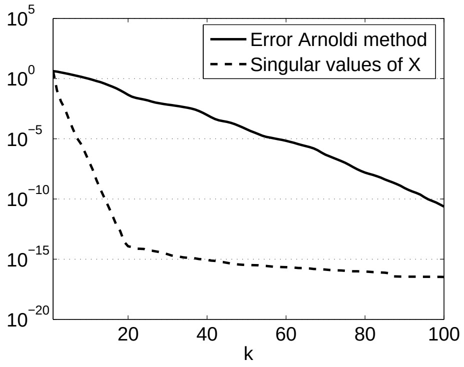

Fig. 1. Error $\|X - X_k\|_2$ in kth step of Arnoldi method vs. kth singular value of X

Algorithm 1 Arnoldi method for solving Lyapunov equations

Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, an integer k.
Output: Matrix $V_k \in \mathbb{R}^{n\times k}$ such that $X_k = V_k V_k^T$ approximates the solution of (1).

```
$u_1 \leftarrow b/\|b\|_2, \quad U_1 \leftarrow u_1, \quad \hat{H}_0 \leftarrow [\,]$
for $j \leftarrow 1, 2, \dots, k$ do
    $w \leftarrow Au_j$
    $h_j \leftarrow U_j^T w, \quad w \leftarrow w - U_j h_j$                (O)
    $h_{j+1,j} \leftarrow \|w\|_2, \quad u_{j+1} \leftarrow w/h_{j+1,j}$    (N)
    $\hat{H}_j \leftarrow \begin{bmatrix} \hat{H}_{j-1} & h_j \\ 0 & h_{j+1,j} \end{bmatrix}$
    $U_{j+1} \leftarrow [U_j, u_{j+1}]$
end for
Partition $\hat{H}_k = \begin{bmatrix} H_k \\ h_{k+1,k} e_k^T \end{bmatrix}$.
Apply Hammarling's method to compute $L$ such that $Y = LL^T$ solves the compressed Lyapunov equation $H_k Y + Y H_k^T = -\|b\|_2^2 e_1 e_1^T$.
Set $V_k = U_k L$.
```

to explain some of its implementation details. Algorithm 1 contains the pseudocode for computing the matrix X_k defined in (3). Some remarks are in order:

1) In exact arithmetic, the columns of the matrix U_k form an orthogonal basis. To maintain orthogonality in finite-precision arithmetic, it is necessary to reorthogonalize a newly produced column u_{j+1} against all previously produced vectors $u_1, \dots, u_j$. In practice, this can be performed by repeating step (O) in Algorithm 1 once, see [35, Sec. 5.1] for more details.
2) Note that the matrix H_k is in upper Hessenberg form. This can be exploited in the first step of Hammarling's method, when reducing H_k to Schur form.
3) The quality of the obtained approximation X_k can be estimated using the residual
$$R_k = AX_k + X_k A^T + bb^T. \tag{5}$$
In [18], it was shown that the Arnoldi decomposition $AU_k = U_k H_k + h_{k+1,k} u_{k+1} e_k^T$ implies
$$\begin{aligned} \|R_k\|_2 &= h_{k+1,k} \|e_k^T Y\|_2, \\ \|R_k\|_F &= \sqrt{2} h_{k+1,k} \|e_k^T Y\|_2. \end{aligned} \tag{6}$$
Thus the norm of the residual can be computed at almost no extra cost.
4) If some $h_{j+1,j}$ happens to be zero then step (N) in Algorithm 1 becomes ill-defined. This turns out to be a rare and fortunate event, since $AU_j = U_j H_j$ holds in this case and hence the corresponding X_j solves (1) *exactly*.

III. Symmetric A

For a symmetric matrix A, the Arnoldi process for producing an orthogonal basis of the Krylov subspace $\mathcal{K}_k(A, b)$ simplifies considerably. In particular, the Hessenberg matrix $H_k = U_k^T A U_k$ is symmetric and hence a tridiagonal matrix
$$H_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix}. \tag{7}$$
This shows that in step (O) of Algorithm 1 the vector w needs to be orthogonalized only against u_{j-1} and u_j, at least in exact arithmetic. The Arnoldi method performed with such a short recurrence is usually called *Lanczos method*. This method provides the potential to discard older vectors during the computation of U_k. However, in order to benefit from this potential, two issues need to be addressed.

First, U_k is needed to define V_k in the last step of Algorithm 1. Even worse, V_k is also $n \times k$ and thus requires the same amount of storage as U_k. Second, in finite-precision arithmetic w cannot assumed to be already orthogonal to $u_1, \dots, u_{j-2}$. Moreover, the reorthogonalization explained in Remark 1) above cannot be performed if older vectors of U_k are discarded. It is not clear how the corresponding loss of orthogonality affects the performance of Algorithm 1. In the following, we fix the first issue algorithmically and the second issue by Paige's analysis of the Lanczos process.

A. A two-pass Lanczos method

The concept of two-pass Lanczos methods for eigenvector computation [5] can be extended to the Lyapunov equation as follows.

In the first pass, the compressed Lyapunov equation is determined. Since the compressed equation also has a right-hand side of rank 1 it can be expected that the singular values of its solution Y decay rapidly, just as the singular values of X. To benefit from this effect, we compute a singular value decomposition (SVD) of the Cholesky factor $L \in \mathbb{R}^{k\times k}$:

$$L = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}^T,$$

where $\Sigma_1 \in \mathbb{R}^{l\times l}$ and $\|\Sigma_2\|_2 \le \epsilon$ for some small user-specified tolerance ϵ. This allows us to replace L by

$$\widetilde{L} = Q_1\Sigma_1 Z_1^T \in \mathbb{R}^{k\times l} \tag{8}$$

without sacrificing the accuracy of $V_k = U_k L$:

$$\begin{aligned} \|V_kV_k^T - U_k\tilde{L}\tilde{L}^TU_k^T\|_2 &= \|LL^T - \widetilde{L}\widetilde{L}^T\|_2 \\ &= \|Q_2\Sigma_2^2Q_2^T\|_2 \le \epsilon^2. \end{aligned}$$

If we have $l \ll k$, it requires significantly less memory to store $\widetilde{V}_k = U_k\widetilde{L} \in \mathbb{R}^{n\times l}$ in place of $V_k \in \mathbb{R}^{n\times k}$.

In the second pass, the matrix-matrix product $U_k\widetilde{L}$ is computed on-the-fly, during the recomputation of the columns of U_k.

Algorithm 2 provides the pseudo-code of the proposed two-pass Lanczos method. Again, some comments on its implementation are in order:

1) In the first pass, only the three vectors u_j, u_{j+1}, w as well as all coefficients of H_k need to be stored during the process. In the second pass, the coefficients of H_k can be dismissed after each loop.
2) The tridiagonal structure of the matrix H_k can be exploited to speed up the solution of the compressed solution. By a careful inspection of Hammarling's method for this case it is possible to reduce the cost from $\mathcal{O}(k^3)$ to almost $\mathcal{O}(k^2)$. However, it turns out that the ADI method is even more effective and requires only $\mathcal{O}(k \log k)$ flops, see [25] for more details.

Example 2: The Spiral Inductor PEEC Model from the Oberwolfach model reduction benchmark collection [23] gives rise to a Lyapunov equation (1) with symmetric A and $n = 1434$. Algorithm 2 with $\epsilon = 10^{-8}$ was applied. No difference in the convergence of the Arnoldi method with full reorthogonalization and the Lanczos method with no reorthogonalization was observed. However, Figure 2 reveals a significant difference in the memory consumption. While the number of vectors stored by the two-pass Lanczos method saturates at a value of 23*, the memory required by the Arnoldi method grows linearly with* n.

Algorithm 2 Two-pass Lanczos method for solving Lyapunov equations

Input: $A \in \mathbb{R}^{n\times n}$ symmetric negative definite, $b \in \mathbb{R}^n$, an integer k, and a tolerance ϵ.
Output: Matrix $\widetilde{V}_k \in \mathbb{R}^{n\times l}$ such that $\widetilde{X}_k = \widetilde{V}_k\widetilde{V}_k^T$ approximates the solution of (1).

```
for pass ← $1, 2$ do
    $u_1 \leftarrow b/\|b\|_2$
    $w \leftarrow Au_1$
    for $j \leftarrow 1, 2, \dots, k$ do
        $\alpha_j \leftarrow u_j^T w, \quad w \leftarrow w - \alpha_j u_j$
        $\gamma_{j+1} \leftarrow \|w\|_2, \quad u_{j+1} \leftarrow w/\gamma_{j+1}$
        if pass $= 2$ then
            $\widetilde{V}_k \leftarrow \widetilde{V}_k + u_j e_j^T \widetilde{L}$
        end if
        $w \leftarrow Au_{j+1}$
        $w \leftarrow w - \gamma_{j+1}u_j$
    end for
    if pass $= 1$ then
        Compute $L$ such that $Y = LL^T$ solves the compressed Lyapunov equation $H_kY + YH_k = -\|b\|_2^2 e_1 e_1^T$. with $H_k$ as in (7).
        Compute low-rank approximation $\tilde{L} \in \mathbb{R}^{l\times\tilde{l}}$ using (8) such that $\|LL^T - \widetilde{L}\widetilde{L}^T\|_2 \le \epsilon^2$.
        Initialize $V \leftarrow 0$.
    end if
end for
```

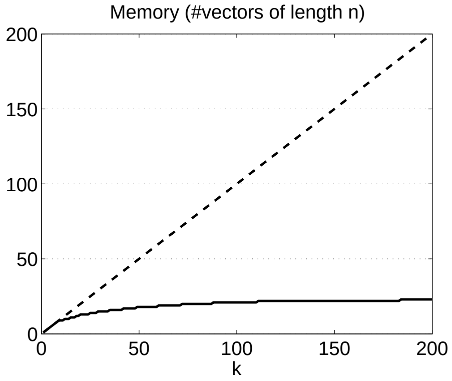

Fig. 2. Memory requirements of two-pass Lanczos method (solid line) vs. Arnoldi method (dashed line) for Example 2.

B. Loss of orthogonality

As an eigenvalue of H_k converges towards an eigenvalue of A during the Lanczos process, it is well known that the Krylov subspace basis U_k

quickly loses its orthogonality in finite-precision arithmetic [28]. However, it has been observed and analyzed in [7], [36] that eigenvalue-based bounds for the Krylov subspace approximation of matrix functions are not affected by this loss of orthogonality. From the integral representation (4) we can expect that a similar result for the solution of Lyapunov equations. In fact, such a result is proved in [25] using Paige's theory [27] on the finite-precision properties of the Lanczos process; eigenvalue-based convergence bounds on $\|X - X_k\|_2$ are only slightly affected by the loss of orthogonality. Also, the expressions (6) remain valid upper bounds on the norm of the residual.

The Strakoš matrix demonstrates that the invariance of the convergence *bounds* does not imply that the actually observed convergence rate is not affected. Let

$$A = Q\mathrm{diag}\left(0.1 + \frac{i-1}{199}\cdot 99.9 \cdot 0.8^{200-i}\right)_{i=1}^{200} Q^T, \tag{9}$$

where Q is a random orthogonal matrix, and let b be a random vector. Figure 3 reveals that loss of orthogonality indeed spoils convergence. Nevertheless, (linear) convergence itself is maintained; only the observed superlinear convergence of the Arnoldi method is destroyed.

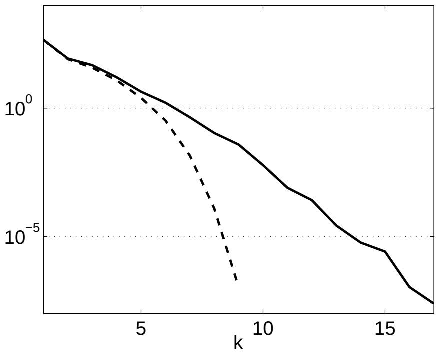

Fig. 3. Convergence of $\|X - X_k\|_2$ for the Lanczos method without reorthogonalization (solid line) and the Arnoldi method with full reorthogonalization (dashed line) applied to (9).

IV. Nonsymmetric A

For a nonsymmetric matrix A, Eiermann and Ernst [9] proposed restarted Arnoldi methods to approximate $f(A)B$ for certain functions f, including the exponential function. The basic idea is to restart the Arnoldi method every k iterations with the last column of the Krylov subspace basis. After pk iterations, this leads to a Krylov decomposition

$$A[U^{(1)}, \dots, U^{(p)}] = AH_{kp} + u_{k+1}^{(p)} h_{pk+1,pk} e_{pk}^T,$$

where each of $U^{(1)}, \dots, U^{(p)} \in \mathbb{R}^{n\times k}$ contains an orthogonal basis. However, the matrices are $U^{(j)}$ not mutually orthogonal. Note that H_{kp} is upper Hessenberg and at the same block lower triangular, see also 4. This structure can be exploited to solve the compressed Lyapunov equation

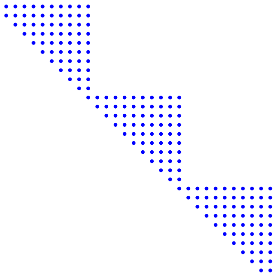

Fig. 4. Structure of H_{kp}, the compression of A for the restarted Arnoldi method.

$$H_{kp}Y + YH_{kp}^T = -\|b\|_2^2 e_1 e_1^T,$$

using only $\mathcal{O}(pk^3)$ (instead of $\mathcal{O}(p^3k^3)$) flops by combining block substitution with Hammarling's method.

The implementation of the restarted Arnoldi method requires the storage of at most k vectors. To achieve reduced storage for Lyapunov equations, a two-pass approach analogous to Algorithm 2 can be used. A critical aspect of the restarted Arnoldi method is the choice of k. If k is chosen too small, convergence is severely affected.

Example 3: The following mildly nonsymmetric example is taken from LYAPACK [31]. Let A be the standard finite difference discretization ($n = 10000$) of

$$\triangle u(x,y) - 10x\frac{\partial u(x,y)}{\partial x} - 20y\frac{\partial u(x,y)}{\partial y}$$

on $\Omega = [0,1]\times[0,1]$, and let b be the vector of all ones. The following table contains the required number of Arnoldi iterations to attain a residual of norm $\leq 10^{-5}$, compared to the restart length k:

k	kp
∞	300
100	400
50	450
20	540
10	> 1000

V. CONCLUSIONS

Modifications that reduce the memory requirements of Krylov subspace methods for solving matrix equations have been investigated. While the use of the Lanczos method for the symmetric case is already well understood (a more detailed analysis will appear in a forthcoming publication [25]), we have only sketched a possible approach for the nonsymmetric case.

VI. ACKNOWLEDGMENT

The author thanks Darija Marković for performing some of the presented numerical experiments.

REFERENCES

[1] A. C. Antoulas, Sorensen D. C., and Zhou Y. On the decay rate of the Hankel singular values and related issues. *Sys. Control Lett.*, 46(5):323–342, 2002.

[2] Athanasios C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6 of *Advances in Design and Control*. SIAM, Philadelphia, PA, 2005.

[3] R. H. Bartels and G. W. Stewart. Algorithm 432: The solution of the matrix equation $AX - BX = C$. *Communications of the ACM*, 8:820–826, 1972.

[4] P. Benner, J. M. Claver, and E. S. Quintana-Orti. Parallel distributed solvers for large stable generalized Lyapunov equations. *Parallel Process. Lett.*, 9(1):147–158, 1999.

[5] J. K. Cullum and R. A. Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations. Vol. 1*, volume 41 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 2002. Reprint of the 1985 original.

[6] T. A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms*. SIAM, Philadelphia, PA, 2006.

[7] V. Druskin, A. Greenbaum, and L. Knizhnerman. Using nonorthogonal Lanczos vectors in the computation of matrix functions. *SIAM J. Sci. Comput.*, 19(1):38–54, 1998.

[8] V. Druskin and L. Knizhnerman. Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic. *Numer. Linear Algebra Appl.*, 2(3):205–217, 1995.

[9] M. Eiermann and O. Ernst. A restarted Krylov subspace method for the evaluation of matrix functions, 2006. To appear in *SIAM J. Numer. Anal.*

[10] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Statist. Comput.*, 13(5):1236–1264, 1992.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[12] R. Granat and B. Kågström. Parallel solvers for Sylvester-type matrix equations with applications in condition estimation. part I: Theory and algorithms, 2007. Preprint.

[13] R. Granat and B. Kågström. Parallel solvers for Sylvester-type matrix equations with applications in condition estimation. part II: The SCASY software library, 2007. Preprint.

[14] S. Gugercin, D. C. Sorensen, and A. C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms*, 32(1):27–55, 2003.

[15] S. Hammarling. Numerical solution of the stable, nonnegative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2(3):303–323, 1982.

[16] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997.

[17] D. Y. Hu and L. Reichel. Krylov-subspace methods for the Sylvester equation. *Linear Algebra Appl.*, 172:283–313, 1992.

[18] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.*, 31:227–251, 1994.

[19] K. Jbilou and A. J. Riquet. Projection methods for large Lyapunov matrix equations. *Linear Algebra Appl.*, 415:344–358, 2006.

[20] I. Jonsson and B. Kågström. Recursive blocked algorithm for solving triangular systems. I. one-sided and coupled Sylvester-type matrix equations. *ACM Trans. Math. Software*, 28(4):392–415, 2002.

[21] D. L. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. Automat. Control*, AC-13:114–115, 1968.

[22] L. A. Knizhnerman. Calculation of functions of nonsymmetric matrices by means of Arnoldi's method. *Zh. Vychisl. Mat. i Mat. Fiz.*, 31(1):5–16, 1991.

[23] J. G. Korvink and B. R. Evgenii. Oberwolfach benchmark collection. In P. Benner, V. Mehrmann, and D. C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 311–316. Springer, Heidelberg, 2005.

[24] D. Kressner. Block variants of Hammarling's method for solving Lyapunov equations. Report, Department of Computing Science, Umeå University, Umeå, Sweden, 2006.

[25] D. Kressner and D. Marković. On krylov subspace methods for solving symmetric matrix equations, 2008. In preparation.

[26] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[27] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Inst. Math. Appl.*, 18(3):341–349, 1976.

[28] B. N. Parlett. *The Symmetric Eigenvalue Problem*, volume 20 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

[29] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999.

[30] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Systems Control Lett.*, 40(2):139–144, 2000.

[31] T. Penzl. Lyapack users guide. Technical report SFB393/00-33, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, 09107 Chemnitz, FRG, 2000.

[32] Y. Saad. Numerical solution of large Lyapunov equations. In *Signal processing, scattering and operator theory, and numerical methods (Amsterdam, 1989)*, volume 5 of *Progr. Systems Control Theory*, pages 503–511. Birkhäuser Boston, Boston, MA, 1990.

[33] V. Simoncini and V. Druskin. Convergence analysis of projection methods for the numerical solution of large lyapunov equations, 2007. Preprint.

[34] D. E. Stewart and T. S. Leyk. Error estimates for Krylov subspace approximations of matrix exponentials. *J. Comput. Appl. Math.*, 72(2):359–369, 1996.

[35] G. W. Stewart. *Matrix Algorithms. Vol. II.* SIAM, Philadelphia, PA, 2001. Eigensystems.

[36] H. A. van der Vorst. An iterative solution method for solving $f(A)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix A. *J. Comput. Appl. Math.*, 18(2):249–263, 1987.

[37] E. L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.*, 1(1):87–90, 1988.