# Efficient Computation of the Exponential Operator for Large, Sparse, Negative-Definite Matrices

Luca Bergamaschi* and Marco Vianello‡

## 1 Introduction

The problem of approximating the matrix exponential operator, $\exp(-tA)\boldsymbol{v}$, for large, sparse, symmetric negative-definite matrices, is strictly connected to the numerical solution of $n$-dimensional systems of ODEs like

$$\dot{\boldsymbol{y}} = -A\boldsymbol{y} + \boldsymbol{b}(t), \qquad t > 0\,; \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0\,, \tag{1}$$

which arise from the space discretization of linear parabolic PDEs (method of lines, linear Galerkin method).

In this framework, starting from [3, 8] Krylov subspace techniques have been shown to be very effective, as they exhibit remarkable speedups, both in terms of matrix-vector products and total computing time, when compared with the traditional (implicit) time-stepping methods, such as the popular Crank-Nicholson scheme; see for example [2, 4, 6] and the references therein.

As known, the main idea of Krylov-based methods in computing $f(A)\boldsymbol{v}$ for (analytic) matrix functions, consists of approximately projecting $f(A)$ onto a 'small' Krylov subspace $K_m = \{\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{m-1}\boldsymbol{v}\}$. This reduces the problem to compute $f(T_m)$ in view of the approximation

$$f(A)\boldsymbol{v} \approx \|\boldsymbol{v}\|_2 V_m f(T_m)\boldsymbol{e}_1, \tag{2}$$

where $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$ is the orthonormal basis of $K_m$, and $T_m$ the tridiagonal matrix produced by the Lanczos process with $\boldsymbol{v}/\|\boldsymbol{v}\|_2$ as initial vector.

Another polynomial method for the computation of functions of symmetric matrices, is based on the Chebychev series expansion of the corresponding scalar function on the spectral interval of $A$, say $[\lambda_1, \lambda_n]$, assuming $f$ analytic in a neighborhood of such an interval [3]. The approximation of $f(A)\boldsymbol{v}$ takes on the form

$$f(A)\boldsymbol{v} \approx S_m(B)\boldsymbol{v} = \sum_{k=0}^{m} a_k C_k(B)\boldsymbol{v}, \tag{3}$$

where $C_k$ is the $k^{th}$ Chebychev polynomials of the first kind,

$$B = \frac{1}{\lambda_n - \lambda_1}\left(2A - (\lambda_1 + \lambda_n)I\right) \tag{4}$$

*Department of Mathematical Methods for Applied Sciences, University of Padua, Italy (berga@dmsa.unipd.it)
This author was partially supported by: the EC Inco-Copernicus (contract n. ERBIC15CT96-0211) and Fondi Ministeriali 40%.
‡Department of Pure and Applied Mathematics, University of Padua, Italy (marcov@math.unipd.it)

and the $a_k$ are the Chebychev coefficients of $f\left(\frac{\lambda_n-\lambda_1}{2}t+\frac{\lambda_1+\lambda_n}{2}\right), \quad t\in[-1,1]$. In view of the three-term recurrence for the Chebychev polynomials, the expansion in (3) can be conveniently implemented at the main cost per step of a matrix-vector product and without any scalar product.

The number of iterations and the overall computational cost of Chebychev method are usually larger than Krylov's, since the former method is thought to approximate the matrix $f(A)$ and cannot conveniently exploit the spectral structure of $\boldsymbol{v}$. However, if we take into account computer storage, Chebychev method requires only a fixed number of vectors (in addition to $A$), while the standard Krylov process needs to keep in memory all the $m$ Ritz vectors. For this reason, for very large size problems, the standard Krylov method has to be somehow modified.

In the present work we propose an efficient implementation of the Chebychev method for the exponential operator, based on control of the relative error. The key tools are: an efficient approximation of the spectral interval using Lanczos method for the largest eigenvalue and a Conjugate Gradient optimization of the Rayleigh Quotient for the smallest [1], and a priori convergence estimates, recently obtained in [9].

We compare two-step Lanczos and Chebychev methods on very large size matrices arising from both finite difference and finite element space discretization of 3-D parabolic PDEs. Numerical results demonstrate that the Chebychev method can be an effective alternative to Krylov techniques, especially when memory bounds do not allow the storage of all Ritz vectors.

## 2 Chebychev series approach

One of the main features in using Chebychev series, is the computation of the Chebychev coefficients $a_k$ cf. (3), which depend on the scalar function $f$, and the spectral interval of $A$. In the case of the exponential function $f(\lambda)=\exp(-\tau\lambda)$, $\lambda\in[\lambda_1,\lambda_n]$, $\tau>0$, it is known that the coefficients are related to Bessel functions of pure imaginary argument. More precisely, we have

$$\begin{aligned}
a_0 &= \exp\left(-\tau\frac{\lambda_1+\lambda_n}{2}\right)\, I_0\left(-\tau\frac{\lambda_n-\lambda_1}{2}\right)\\
a_k &= 2\exp\left(-\tau\frac{\lambda_1+\lambda_n}{2}\right)\, I_k\left(-\tau\frac{\lambda_n-\lambda_1}{2}\right)
\end{aligned} \tag{5}$$

cf. [7, §6.5]. A sharp approximation (from outside) of the spectral interval is therefore required, which represents a drawback of Chebychev method with respect to Krylov method. In practice, however, the cost of this approximation on very large size problems, can be made negligible when compared with the overall cost. This is obtained by the following approach: the largest eigenvalue is computed by the Lanczos process, at the price of a few matrix-vector multiplications [5], while the smallest can be conveniently approximated with a diagonally-preconditioned conjugate gradient technique, applied to the minimization of the Rayleigh Quotient, as in [1]. It is worth noting that both these procedures provide (monotone) approximations from the inside of the spectral interval, while convergence of Chebychev series is guaranteed if an estimate from outside is at hand. We had also experimental evidence of the bad behavior of the algorithm even when the approximation from inside are accurate. This problem can be easily solved, taking into account the linear convergence character of both methods. It is sufficient to estimate *a posteriori* the error via the approximate convergence factor, and then to add (or subtract) a multiple of it to the last approximation.

The well known three-term recurrence for Bessel functions, cannot be used forward, in view of its instability when applied to arguments with large absolute value. This problem can be circumvented using backward the three-term recurrence, once a realistic *a priori* upper bound of the required number $m$ of coefficients is available. In fact, the two starting values $I_m$ and $I_{m-1}$ of the recurrence can be easily computed at a high accuracy by a well-known asymptotic expansion. Such an *a priori* upper bound can be obtained starting from the absolute error estimate:

$$\begin{aligned} \left\| e^{-\tau A}\boldsymbol{v} - \sum_{k=0}^{m} a_k C_k(B)\boldsymbol{v} \right\| = \left\| \sum_{k=m+1}^{\infty} a_k C_k(B)\boldsymbol{v} \right\| &\leq \|\boldsymbol{v}\| \sum_{k=m+1}^{\infty} |a_k| \leq \\ &\leq 2\|v\| \exp(-\tau\lambda_1) E(m) \end{aligned} \tag{6}$$

with $B$ is defined in (4) and $a_k$ in (3), and

$$E(m) = \begin{cases} \exp\left(\frac{-b(m+1)^2}{\tau\lambda_n}\right)\left[1 + \sqrt{\frac{\tau\lambda_n\pi}{4b}}\right] + \frac{d^{\tau\lambda_n}}{1-d} & \text{if } \ m \leq \tau\lambda_n \\ \frac{d^m}{1-d} & \text{if } \ m > \tau\lambda_n \end{cases} \tag{7}$$

where $b \approx 0.618$, $d \approx 0.438$; these values, together with (7) are taken from [9]. Whenever a lower bound of $\|\exp(-\tau A)\boldsymbol{v}\|$ is available, we can estimate the number $m(\varepsilon)$ such that the relative error is less than a prescribed tolerance $\varepsilon$. An upper bound of such a norm can be obtained by writing $\boldsymbol{v}$ as a linear combination of the orthonormal eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ of $A$: $\boldsymbol{v} = \sum_{i=1}^{n} c_i \boldsymbol{u}_i$, so that

$$\|e^{-\tau A}\boldsymbol{v}\|_2 = \sqrt{c_1^2 e^{-2\tau\lambda_1} + \ldots + c_n^2 e^{-2\tau\lambda_n}} \geq |c_1| e^{-\tau\lambda_1} \tag{8}$$

Since the Rayleigh Quotient minimization yields also the eigenvector $\boldsymbol{u}_1$, the coefficient $c_1$ can be easily computed taking into account the orthonormality of the eigenvectors of $A$:

$$\boldsymbol{u}_1^T \boldsymbol{v} = \boldsymbol{u}_1^T \sum_{i=1}^{n} c_i \boldsymbol{u}_i = c_1 \boldsymbol{u}_1^T \boldsymbol{u}_1 = c_1.$$

We can now write the following estimate of the relative error:

$$\frac{\|e^{-\tau A}\boldsymbol{v} - \sum_{k=0}^{m} a_k C_k(B)\boldsymbol{v}\|}{\|e^{-\tau A}\boldsymbol{v}\|} \leq \frac{2\|\boldsymbol{v}\|}{c_1} E(m), \tag{9}$$

(cf. 7), thus obtaining $m(\varepsilon) = \min\{m : E(m) < c_1\varepsilon/(2\|\boldsymbol{v}\|)\}$.

At this point, once the first $m(\varepsilon)$ Chebychev coefficients have been computed, using backward the Bessel recurrence as described above, we can start the evaluation of the sum in (3) as:

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + a_{k+1}\boldsymbol{v}_{k+1} \tag{10}$$

where $\boldsymbol{v}_k = C_k(B)\boldsymbol{v}$ satisfies the three-term recurrence:

$$\boldsymbol{v}_{k+1} = 2B\boldsymbol{v}_k - \boldsymbol{v}_{k-1} \qquad\qquad k = 1, 2, \ldots \tag{11}$$

Our stopping criterion is not based on the over-estimate as in (9), which is aimed only at computing a sufficient number of Chebychev coefficients. A sharper estimate of the absolute error (the left hand side in (6)), can be *a priori* obtained from:

$$\left\| \sum_{k=m+1}^{\infty} a_k C_k(B)\boldsymbol{v} \right\| \leq \|\boldsymbol{v}\| \sum_{k=m+1}^{\infty} |a_k| \approx \|\boldsymbol{v}\| \sum_{k=m+1}^{m(\varepsilon)} |a_k|. \tag{12}$$

## 3 Krylov subspace approach

Krylov-based method for computing matrix functions $f(A)\boldsymbol{v}$ has become the most popular, thanks to the following reasons: it is tailored on the spectral structure of both $A$ and $\boldsymbol{v}$, thus achieving a faster convergence than Chebychev, without requiring any *a priori* knowledge of the spectral interval; it works for general (non symmetric) matrices provided that the Lanczos process is used to generate a biorthogonal basis of the subspaces $K_m(A, \boldsymbol{v}_1)$ and $K_m(A, \boldsymbol{w}_1)$, $\boldsymbol{w}_1$ being a suitable initial vector [8].

The main drawback of the Krylov approach is represented by the memory requirement: all the $m$ Lanczos vectors should be kept in memory to get the final approximation of $\exp(-\tau A)\boldsymbol{v}$. A simple way to overcome this problem is to run *twice* the Lanczos process. The first step aims to evaluate the coefficients of (2) i. e. the vector $\boldsymbol{c}_m = \|\boldsymbol{v}\| \exp(-\tau T_m)\boldsymbol{e}_1$, by keeping in memory only the $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\nu$, $\nu$ depending on actual storage limitations. The second step generates again the Lanczos vectors (starting form the $(\nu+1)^{th}$), updating the linear combination as in (2). The cost of the first run is generally higher than that of the second one, essentially depending on the ratio $\frac{\nu}{m}$: in particular the number of matrix-vector products is reduced in the second step by a factor $1 - \frac{\nu}{m}$. Moreover every iteration in the second run of the Lanczos process can be performed without computing the two scalar products.

Note that in the first run the computation of $\exp(-\tau T_m)\boldsymbol{e}_1$ is required for the final tridiagonal matrix $T_m$, and, possibly, for some intemediate matrices $T_k$, with $k < m$, when the stopping criterion is based on an *a posteriori* bounds [8]. If the dimension of the Krylov subspace $m$ is not too large, this step can be performed using standard methods. In our applications this number can be of order of the hundreds, and hence *ad hoc* techniques are required. In e. g. [8] the use of rational Chebychev approximation is proposed; our implementation makes use of the Chebychev series method, as described in §2. The cost of this operation will be shown to be 'negligible' as compared to that of the whole algorithm.

We sketch now the implementation of the first step of the Lanczos algorithm, recalling that the Kyrlov subspace approximation of the exponential operator is given by:

$$\exp(-\tau A)\boldsymbol{v} \approx \|\boldsymbol{v}\| V_m \exp(-\tau T_m)\boldsymbol{e}_1 = V_m \boldsymbol{c}_m \tag{13}$$

where the tridiagonal matrix $T_m$ is defined as:

$$T_m = \text{tridiag}[\beta_k, \alpha_k, \beta_{k+1}], \quad k = 1, \ldots, m$$

The stopping criterion for the first Lanczos run is based on the *a posteriori* estimate provided in [8] in which the absolute error is bounded by the quantity:

$$e_m = \beta_{m+1} \left| \boldsymbol{e}_m^T \phi(-\tau T_m)\boldsymbol{e}_1 \right| \|\boldsymbol{v}\| \tag{14}$$

where

$$\phi(z) = \frac{\exp(z) - 1}{z}$$

The evaluation of (14) is performed every $p$ iterations via the following steps: first we compute

$$\boldsymbol{c}_m = \|\boldsymbol{v}\| \exp(-\tau T_m)\boldsymbol{e}_1$$

by the Chebychev algorithm, as described in §2.

In order to get an *a posteriori* bound of the relative error, an estimate of the solution norm can be easily obtained by $\|\boldsymbol{c}_m\|$ in view of the identity (valid at least in exact arithmetics):

$$\|V_m \boldsymbol{c}_m\|^2 = \boldsymbol{c}_m^T V_m^T V_m \boldsymbol{c}_m = \boldsymbol{c}_m^T \boldsymbol{c}_m = \|\boldsymbol{c}_m\|^2 \tag{15}$$

The first Lanczos run is then stopped at the smallest index $m$ such that:

$$\frac{e_m}{\|\boldsymbol{c}_m\|} < \varepsilon$$

where $e_m = \frac{\beta_{m+1}}{\tau} \left| \boldsymbol{e}_m^T T_m^{-1} \left( \boldsymbol{c}_m - \|\boldsymbol{v}\| \boldsymbol{e}_1 \right) \right|$ is computed by solving

$$T_m \boldsymbol{y} = \boldsymbol{c}_m - \|\boldsymbol{v}\| \boldsymbol{e}_1$$

and computing

$$e_m = \frac{\beta_{m+1}}{\tau} |y_m|$$

# 4 Numerical Results on 3D Finite Difference Matrices

This section is devoted to the validation of the code with test cases arising from the space discretization by Finite Differences of a 3-D linear diffusion equation on a unitary cubic domain $\Omega$. Our purpose is to demonstrate the effectiveness of the Chebychev approach with respect to the Krylov technique, on problems of very large size. Consider the standard linear diffusion problem with homogeneous Dirichlet conditions:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} + \dfrac{\partial^2 u}{\partial z^2} & \vec{x} \in \Omega \\ u(\vec{x}, t) = 0, \qquad \vec{x} \in \partial\Omega, & u(\vec{x}, 0) = u_0(\vec{x}) \end{cases} \tag{16}$$

Discretization of the Laplace operator by Finite Differences with constant step size $h = 1/p$ gives raise to a $n(= p^3)-$dimensional system of the form (1), (with $\boldsymbol{b} \equiv \boldsymbol{0}$) where $A$ is a symmetric positive definite eptadiagonal matrix. We choose the initial vector to be $\boldsymbol{y}_0 = [1, 1, \ldots, 1]^T$. Four different grids were considered with size $p = 40, 60, 80$ and $100$,

| $p$ | d.o.f. | $\lambda_1^{(p)}$ | $\lambda_n^{(p)}$ | $\|A_p\|_\infty$ |
|---|---|---|---|---|
| 10 | 1,000 | .24305e+02 | .11884e+04 | .12000e+04 |
| ... | ... | ... | ... | ... |
| 40 | 64,000 | .28168e+02 | .19087e+05 | .19200e+05 |
| 60 | 216,000 | .28640e+02 | .43086e+05 | .43200e+05 |
| 80 | 512,000 | .28879e+02 | .76692e+05 | .76800e+05 |
| 100 | 1,000,000 | .29023e+02 | .11988e+06 | .12000e+06 |

Table 1: Extreme eigenvalues for the Finite Difference matrices with different grid sizes

respectively. The extreme eigenvalues of the respective matrices are reported in Table 1. For this case we have used as an approximation of $\lambda_1^{(p)}$ the value of $\lambda_1$ on a coarser mesh since the smallest eigenvalue of the discrete problem converges, as $p$ tends to infinity, to the smallest eigenvalue of the continuous problem. For our numerical tests it was sufficient to calculate $\lambda_1^{(10)}$ with a good saving of the computing time. As for the largest eigenvalue instead of using the costly Lanczos method we simply took the infinity norm of the matrix $A$ as an upper bound for $\lambda_n$. The table accounts for the reliability of this approximation.

We have compared the two algorithms described in the previous sections in the computation of $\exp(-\tau A_p)$ for three values of $\tau = 0.01, 0.03, 0.1$. All the computations were made on an Alpha station 600Mhz with 512 Mb RAM; however the simulations were run assuming

| | | Lanczos | | | | | Chebychev | | |
|---|---|---|---|---|---|---|---|---|---|
| $p$ | $\tau$ | $\nu$ | # it. | 1st run | 2nd run | CPU (s) | $\lambda_n(\tau A_p)$ | # it. | CPU (s) |
| 40 | 0.01 | 200 | 55 | 2.76 | 0.00 | 2.76 | 191 | 58 | 2.54 |
| | 0.03 | 200 | 80 | 3.97 | 0.00 | 3.97 | 573 | 100 | 4.40 |
| | 0.10 | 200 | 85 | 4.22 | 0.00 | 4.22 | 1,909 | 182 | 8.04 |
| 60 | 0.01 | 53 | 80 | 14.78 | 4.50 | 19.28 | 431 | 85 | 13.99 |
| | 0.03 | 53 | 115 | 20.69 | 10.32 | 31.01 | 1,293 | 150 | 24.81 |
| | 0.10 | 53 | 125 | 22.29 | 11.91 | 34.21 | 4,309 | 274 | 45.49 |
| 80 | 0.01 | 8 | 100 | 43.37 | 37.43 | 80.80 | 767 | 115 | 50.34 |
| | 0.03 | 8 | 150 | 64.92 | 57.69 | 122.61 | 2,301 | 199 | 88.05 |
| | 0.10 | 8 | 165 | 71.34 | 63.79 | 135.13 | 7,670 | 365 | 160.79 |
| 100 | 0.01 | 2 | 125 | 104.21 | 98.54 | 202.74 | 1,199 | 144 | 115.43 |
| | 0.03 | 2 | 180 | 150.00 | 142.17 | 292.17 | 3,396 | 249 | 200.18 |
| | 0.10 | 2 | 200 | 166.43 | 158.00 | 324.43 | 11,989 | 455 | 366.67 |

Table 2: Number of iterations and CPU time for the Lanczos and Chebychev algorithms applied ad the FD sample problems with $\epsilon = 10^{-8}$.

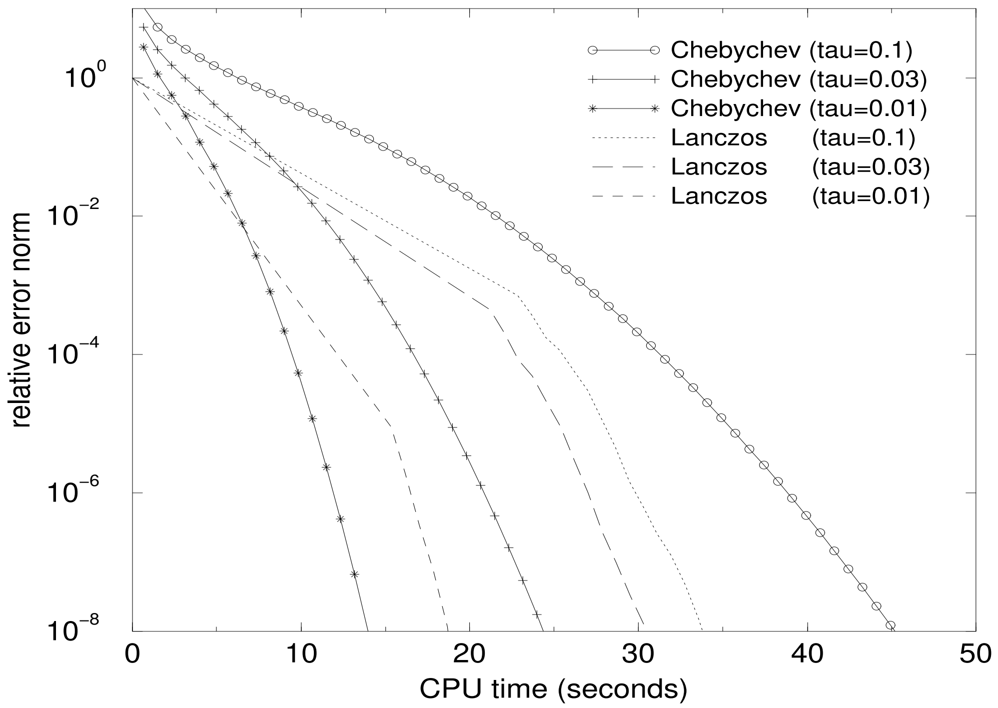


Figure 1: Relative error norm versus CPU time in the evaluation of $\exp(-\tau A)\boldsymbol{v}$ for the $60^3$ mesh and for the three different values of $\tau$. The first part of the Lanczos profile accounts for the initial Lanczos run.

that no more than 128 Mb RAM are avaliable and the maximum number $\nu$ of storable Ritz vectors for the Lanczos algorithm is computed consequently. Table 2 shows the results of the simulations in terms of number of iterations and computing time for both the procedures. The implementation of the Lanczos algorithm, starting from the $p = 60$ case, could not store all the Ritz vectors within the avaliable memory, and hence requires a number of iterations in the second run. The Chebychev code, however, could run using less than 128 Mb RAM also when addressing the largest problem (with $p = 100$).

The Figure 1 displays the results for the intermediate $p = 60$ mesh. From this plot and the Table we can conclude that the Chebychev agorithm may be competitive with Lanczos procedure either when the spectral radius of the matrix is not too large ($\tau = 0.01$ case) or when the size of the problem is sufficiently large to require the execution of the second Lanczos run.

Numerical tests on Finite Element matrices in both 2 and 3 space dimensions are currently under work.

## References


[1] L. Bergamaschi, G. Gambolati, and G. Pini, *Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem*, Numer. Lin. Alg. Appl., 4 (1997), pp. 69–84.

[2] V. L. Druskin, A. Greenbaum, and L. A. Knizhnerman, *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM J. Sci. Comput., (1998). to appear.

[3] V. L. Druskin and L. A. Knizhnerman, *Two polynomial methods for calculating functions of symmetric matrices*, U.S.S.R. Comput. Maths. Math. Phys., 6 (1989), pp. 112–121.

[4] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov subspace methods*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1236–1264.

[5] G. H. Golub and C. F. van Loan, *Matrix Computation*, Johns Hopkins University Press, Baltimore, 1991.

[6] M. Hochbruck and J. J. Lubich, *On Krylov subspace approximations to the matrix exponential*, SIAM J. Num. Anal., 34 (1997), pp. 1911–1925.

[7] G. Meinardus, *Approximation von Funktionen und ihre numerische Behandlung*, Springer-Verlag, 1964.

[8] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Num. Anal., 29 (1992), pp. 209–228.

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