

Implementation of a Restarted Krylov Subspace Method for the Evaluation of Matrix Functions

Martin Afanasjew, Michael Eiermann, Oliver G. Ernst and Stefan Güttel*
Institut für Numerische Mathematik und Optimierung
TU Bergakademie Freiberg, Germany

July 10, 2007

Dedicated to Richard S. Varga
on the occasion of his 80th birthday

A new implementation of restarted Krylov subspace methods for evaluating $f(A)\mathbf{b}$ for a function f , a matrix A and a vector \mathbf{b} is proposed. In contrast to an implementation proposed previously, it requires constant work and constant storage space per restart cycle. The convergence behavior of this scheme is discussed and a new stopping criterion based on an error indicator is given. The performance of the implementation is illustrated for three parabolic initial value problems, requiring the evaluation of $\exp(A)\mathbf{b}$.

1 Introduction

The interplay of complex approximation theory and matrix computations has long been, and still is, a recurring theme in the work of Richard Varga. The subject of this paper is an instance where this interplay is fundamental, namely the computation of the vector

$$f(A)\mathbf{b} \tag{1}$$

given a matrix $A \in \mathbb{C}^{n \times n}$, a vector $\mathbf{b} \in \mathbb{C}^n$ of unit norm and a function f analytic in a neighborhood of the spectrum $\Lambda(A)$ of A . In particular, we shall find that the successful implementation of a technique which two of the authors recently proposed [6] rests on pioneering work of Richard Varga [3] from the late 1960s.

*This work was partially supported by the Deutsche Forschungsgemeinschaft.

The evaluation of $f(A)\mathbf{b}$ is a common task in scientific computing, and the most familiar case is surely that of the exponential function $f(\lambda) = \exp(\lambda)$, which occurs, e.g., in connection with the linear initial value problem

$$\mathbf{u}'(t) = A\mathbf{u}(t), \quad t > 0, \quad \mathbf{u}(0) = \mathbf{u}_0, \quad (2)$$

with solution $\exp(tA)\mathbf{u}_0$. Initial value problems such as (2) result naturally from the method of lines discretization of parabolic partial differential operators. Such discretizations as well as the construction of time-stepping schemes based on Padé or Chebyshev rational approximation were among the subjects of Richard Varga's earlier work [22].

We are concerned with the situation where A is large and either sparse or structured, such that matrix-vector products with A can be carried out inexpensively, whereas first forming $f(A)$ and then multiplying with \mathbf{b} cannot. Here Krylov subspace approximations of (1) have become popular (cf. [4, 7, 12]) and, with regard to solving initial value problems, have had a large impact on so-called exponential integrators, in which evaluations of the exponential function applied to the Jacobian are incorporated directly into time-stepping schemes (cf. [13]).

Krylov subspace approximations of (1) are based on an Arnoldi-like decomposition

$$AV_m = V_{m+1}\tilde{H}_m = V_m H_m + \eta_{m+1,m} \mathbf{v}_{m+1} \mathbf{e}_m^\top \quad (3)$$

in which the columns of V_m form an ascending basis of the Krylov space

$$\mathcal{K}_m(A, \mathbf{b}) := \text{span}\{\mathbf{b}, A\mathbf{b}, \dots, A^{m-1}\mathbf{b}\},$$

$\tilde{H}_m := [\eta_{i,j}]$ is an $(m+1) \times m$ upper Hessenberg matrix, $H_m := [I_m \ 0] \tilde{H}_m$ and \mathbf{e}_m denotes the m -th unit coordinate vector in \mathbb{R}^m . In the most common situation, (3) is a proper Arnoldi decomposition, i.e., the basis sequence $\{\mathbf{v}_m\}$ consists of orthonormal vectors, as generated by the Arnoldi process, which reduces to the Hermitian Lanczos process when A is Hermitian. In this case H_m is Hermitian tridiagonal. More general Arnoldi-like decompositions arise in restarted Arnoldi schemes, in which the basis vectors are only block orthogonal, or when non-orthogonalizing basis generation schemes are used (see the discussion in [6]). For an orthonormal basis sequence the Arnoldi approximation of (1) is then given by

$$\mathbf{f}_m := V_m f(H_m) V_m^H \mathbf{b} = V_m f(H_m) \mathbf{e}_1, \quad (4)$$

where \mathbf{e}_1 denotes the first unit coordinate vector in \mathbb{R}^m . There are (at least) three ways of motivating the approximation (4) (see [11]). One is as a subspace approximation of (1), since for a proper Arnoldi decomposition (3) the Hessenberg matrix H_m represents the orthogonal section of A onto $\mathcal{K}_m(A, \mathbf{b})$, and in this sense $f(H_m)$ an approximation of the action of $f(A)$ on this space. Alternatively, if Γ is a contour containing $\Lambda(A)$ in its interior $\text{int } \Gamma$ and if f is analytic in the closure $\overline{\text{int } \Gamma}$, we have

$$\begin{aligned} \mathbf{f}_m = V_m f(H_m) V_m^H \mathbf{b} &= \frac{1}{2\pi i} \int_{\Gamma} f(\lambda) V_m (\lambda I - H_m)^{-1} V_m^H \mathbf{b} \, d\lambda \\ &\approx \frac{1}{2\pi i} \int_{\Gamma} f(\lambda) (\lambda I - A)^{-1} \mathbf{b} \, d\lambda = f(A) \mathbf{b}, \end{aligned}$$

which characterizes \mathbf{f}_m as the Galerkin approximation of the resolvent integral representation of (1). Finally, and this is the interpretation closest to the method proposed below, it can be shown that (4) can be written as

$$\mathbf{f}_m = q(A)\mathbf{b},$$

where $q \in \mathcal{P}_{m-1}$ is a polynomial of degree $m - 1$ which interpolates f in the Hermite sense at the eigenvalues of H_m , i.e., at the Ritz values of A with respect to $\mathcal{K}_m(A, \mathbf{b})$.

The evaluation of (4) requires that the complete basis of $\mathcal{K}_m(A, \mathbf{b})$ be available, which can be prohibitive for large A and large values of m . One remedy restricted to the Hermitian case, in which the \mathbf{v}_m can be generated by the three-term Lanczos-recurrence, is to compute the Hessenberg matrix H_m in a first pass to subsequently determine the coefficient vector $f(H_m)\mathbf{e}_1$, followed by a regeneration of the basis vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ in a second pass to form the linear combination (4), which, while feasible, seems nonetheless unelegant.

This was the motivation for the restarting algorithm proposed in [6], which involves Arnoldi decompositions of a fixed length m and generates a sequence of approximations $\{\hat{\mathbf{f}}_k\}_{k \in \mathbb{N}}$ with $\hat{\mathbf{f}}_k \in \mathcal{K}_{km}(A, \mathbf{b})$. The implementation proposed in [6], however, suffered from the deficiency that, although not requiring the storage of more than m basis vectors at a time, the coordinate calculations required the evaluation of f for a block Hessenberg matrix of size km , resulting in computational work which grows with k . In what follows, we introduce a new implementation for which the computational expense as well as the storage requirements are the same for each cycle. In addition, we propose a stopping criterion based on an error indicator, discuss the convergence behavior of the new implementation vs. that proposed in [6], and illustrate their performance in some numerical examples.

2 The Restarted Krylov Subspace Algorithm for Matrix Functions

The restarted Krylov subspace algorithm proposed in [6] proceeds by repeatedly generating a basis of Krylov spaces of fixed dimension m , updating the most recent approximation to (1), and then discarding all but the last basis vector, which is subsequently used as the initial vector for the next Krylov space. Although any procedure which generates a nested basis of a Krylov space can be used, we restrict ourselves to the Arnoldi process in the following.* We recall the basic restart step using two Arnoldi decompositions

$$AV_1 = V_1H_1 + \eta_2\mathbf{v}_{m+1}\mathbf{e}_m^\top, \tag{5a}$$

$$AV_2 = V_2H_2 + \eta_3\mathbf{v}_{2m+1}\mathbf{e}_m^\top, \tag{5b}$$

*When we use terminology such as *Arnoldi decomposition*, *Arnoldi approximation*, *Arnoldi algorithm*, etc. in the context of a Hermitian matrix A , we tacitly assume that computations are carried out with the Hermitian Lanczos process.

in which the columns of V_1 and V_2 are orthonormal bases of $\mathcal{K}_m(A, \mathbf{v}_1)$ and $\mathcal{K}_m(A, \mathbf{v}_{m+1})$, respectively, and H_1 and H_2 are unreduced upper Hessenberg matrices. The columns of $\widehat{V}_2 := [V_1 \ V_2]$ thus form a basis of $\mathcal{K}_{2m}(A, \mathbf{v}_1)$, and we combine the two decompositions in (5) as

$$A\widehat{V}_2 = \widehat{V}_2\widehat{H}_2 + \eta_3\mathbf{v}_{2m+1}\mathbf{e}_{2m}^\top, \quad (6)$$

where

$$\widehat{H}_2 := \begin{bmatrix} H_1 & O \\ E_2 & H_2 \end{bmatrix}, \quad E_2 := \eta_2\mathbf{e}_1\mathbf{e}_m^\top.$$

Note that, since the columns of \widehat{V}_2 are only blockwise orthonormal, we refer to (6) as an Arnoldi-like decomposition. The restarting scheme computes the Krylov subspace approximation

$$\widehat{\mathbf{f}}_2 := \widehat{V}_2 f(\widehat{H}_2) \mathbf{e}_1,$$

associated with the decomposition (6).[†]

Due to the block lower triangular structure of \widehat{H}_2 , the approximation $\widehat{\mathbf{f}}_2$ has the form

$$\widehat{\mathbf{f}}_2 = V_1 f(H_1) \mathbf{e}_1 + V_2 F_{2,1} \mathbf{e}_1 = \widehat{\mathbf{f}}_1 + V_2 F_{2,1} \mathbf{e}_1,$$

in which $\widehat{\mathbf{f}}_1$ denotes the approximation associated with the first decomposition (5a) and the $m \times m$ matrix $F_{2,1}$ is defined by

$$f(\widehat{H}_2) = \begin{bmatrix} f(H_1) & O \\ F_{2,1} & f(H_2) \end{bmatrix}.$$

If $F_{2,1}$, or rather its first column, can be computed, then only $\widehat{\mathbf{f}}_1$ needs to be stored from the first cycle of the algorithm, and V_1 can be discarded after computing $\widehat{\mathbf{f}}_1$.

The result of k cycles of this restarting scheme is the Krylov subspace approximation associated with the decomposition

$$A\widehat{V}_k = \widehat{V}_k\widehat{H}_k + \eta_{k+1}\mathbf{v}_{km+1}\mathbf{e}_{km}^\top, \quad (7)$$

where $\widehat{V}_k := [V_1 \ V_2 \ \dots \ V_k] \in \mathbb{C}^{n \times km}$,

$$\widehat{H}_k := \begin{bmatrix} H_1 & & & & \\ E_2 & H_2 & & & \\ & \ddots & \ddots & & \\ & & & E_k & H_k \end{bmatrix} \in \mathbb{C}^{km \times km}, \quad E_j := \eta_j \mathbf{e}_1 \mathbf{e}_m^\top \in \mathbb{R}^{m \times m}, \quad j = 2, \dots, k,$$

in which we have collected the quantities of the k Arnoldi decompositions

$$AV_j = V_j H_j + \eta_{j+1} \mathbf{v}_{jm+1} \mathbf{e}_m^\top, \quad j = 1, 2, \dots, k.$$

[†]From now on \mathbf{e}_1 denotes a first unit coordinate vector whose dimension is dictated by the context, whereas, for $j \geq 2$, $\mathbf{e}_j \in \mathbb{R}^j$ shall denote the j -th unit vector.

Setting

$$\widehat{F}_k := f(\widehat{H}_k) = \begin{bmatrix} F_{1,1} & & & \\ F_{2,1} & F_{2,2} & & \\ \vdots & \vdots & \ddots & \\ F_{k,1} & F_{k,2} & \dots & F_{k,k} \end{bmatrix}, \quad \text{where } F_{j,j} = f(H_j), \quad j = 1, 2, \dots, k,$$

the approximation after k restart cycles is given by

$$\widehat{\mathbf{f}}_k := \widehat{V}_k f(\widehat{H}_k) \mathbf{e}_1 = [V_1 \ V_2 \ \dots \ V_k] \widehat{F}_k \mathbf{e}_1 = \sum_{j=1}^k V_j F_{j,1} \mathbf{e}_1 = \widehat{\mathbf{f}}_{k-1} + V_k F_{k,1} \mathbf{e}_1. \quad (8)$$

3 Implementation

In this section we discuss possible implementations of the basic restarting scheme (8). The crucial issue is the fast and stable computation of the coefficient vector $F_{k,1} \mathbf{e}_1$.

3.1 Previously Explored Strategies

There are several possible ways of computing the update (8). In view of the interpolation properties of the Krylov subspace approximation, each additional restart cycle interpolates the function f at m additional nodes, which are the Ritz values of A with respect to the most recent Krylov space. A natural approach would therefore be a block Newton type interpolation scheme, which can be carried out by evaluating matrix polynomials of $m \times m$ matrices: It was shown in [6, Theorem 2.6] that the error of a Krylov subspace approximation (4) with respect to $\mathcal{K}_m(A, \mathbf{b})$ and an Arnoldi-like decomposition (3) has the representation

$$f(A)\mathbf{b} - \mathbf{f}_m = \widetilde{f}(A)\mathbf{v}_{m+1} \quad (9)$$

with a “restart function” $\widetilde{f} := \gamma \Delta_{w_m} f$, in which w_m denotes the characteristic polynomial of H_m , γ is the product of the subdiagonal elements of H_m and $\eta_{m+1,m}$, and the function $\Delta_{w_m} f$ is defined as

$$\Delta_{w_m} f := \frac{f - I_{w_m} f}{w_m}, \quad (10)$$

where $I_{w_m} f$ denotes the polynomial of degree $m-1$ which interpolates f in the Hermite sense at the zeros of the polynomial w_m , i.e., at the Ritz values of A with respect to $\mathcal{K}_m(A, \mathbf{b})$. Since the error (9) has exactly the same form as $f(A)\mathbf{b}$ with \widetilde{f} in place of f , one can proceed by computing corrections to $\widehat{\mathbf{f}}_1 = \mathbf{f}_m$ in the form of Arnoldi approximations to $\widetilde{f}(A)\mathbf{v}_{m+1}$. In view of (9) and (10), this results in a corrected approximation $\widehat{\mathbf{f}}_2 = q(A)\mathbf{b}$, where now $q \in \mathcal{P}_{2m-1}$ interpolates f in the Ritz values of A with respect to $\mathcal{K}_m(A, \mathbf{b})$ as well as those with respect to $\mathcal{K}_m(A, \mathbf{v}_{m+1})$. We note that an alternative expression for the restarted approximations $\widehat{\mathbf{f}}_m$ based on block Newton interpolation for Hermitian A was given in [14].

In [6], the approach based on repeated block Newton interpolation was found to be unstable, and instead it was proposed to evaluate the matrix $f(\widehat{H}_k)$ in each cycle by

standard algorithms such as MATLAB's function `funm`, from which the entries required for the update (8) can be extracted. The resulting scheme is summarized below as Algorithm 1.

Algorithm 1: Restarted Arnoldi approximation for $f(A)\mathbf{b}$ proposed in [6].

Given: A , \mathbf{b} , $\|\mathbf{b}\| = 1$, f

$\mathbf{v}_1 := \mathbf{b}$, $\widehat{\mathbf{f}}_0 := \mathbf{0}$.

for $k = 1, 2, \dots$ until convergence **do**

Compute Arnoldi decomposition $AV_k = V_k H_k + \eta_{k+1} \mathbf{v}_{km+1} \mathbf{e}_m^\top$ of $\mathcal{K}_m(A, \mathbf{v}_{(k-1)m+1})$.

if $k = 1$ **then**

$\widehat{H}_k := H_1$

else

$\widehat{H}_k := \begin{bmatrix} \widehat{H}_{k-1} & O \\ \eta_k \mathbf{e}_1 \mathbf{e}_{(k-1)m}^\top & H_k \end{bmatrix}$.

Update the approximation $\widehat{\mathbf{f}}_k := \widehat{\mathbf{f}}_{k-1} + V_k [f(\widehat{H}_k) \mathbf{e}_1]_{(k-1)m+1:km}$.

Although it allows discarding the basis vectors of previous cycles, Algorithm 1 has the shortcoming that it requires in the k -th cycle the evaluation of f for a matrix of size km . Despite the fact that, typically, $km \ll n$, this can represent substantial computational effort as k gets large. Moreover, it appears wasteful to compute $f(\widehat{H}_k)$ when only the last m entries of its first column are needed.

An alternative approach for computing $f(\widehat{H}_k)$ which promises less work per cycle is to use a recursive scheme [16]: Comparing blocks in the identity

$$\widehat{F}_k \widehat{H}_k = \widehat{H}_k \widehat{F}_k$$

shows that, for $j > \ell$,

$$F_{j,\ell} H_\ell - H_j F_{j,\ell} = E_j F_{j-1,\ell} - F_{j,\ell+1} E_{\ell+1}.$$

Since the diagonal blocks are obtained as $F_{k,k} = f(H_k)$, this relation allows us to recursively compute the last block row of \widehat{F}_k by solving the Sylvester equations

$$X H_{k-j} - H_k X = E_k F_{k-1,k-j} - F_{k,k-j+1} E_{k-j+1}, \quad j = 1, 2, \dots, k-1, \quad (11)$$

for $X = F_{k,k-j}$. The Sylvester equation (11) is easy to solve since its coefficients H_{k-j} and H_k are upper Hessenberg (see [9]). We still, however, have to store \widehat{H}_k , i.e., H_1, H_2, \dots, H_k and $\eta_2, \eta_3, \dots, \eta_k$. In addition, we need $F_{k-1,k-j}$, more precisely, $E_k F_{k-1,k-j}$, i.e., only the last row of $F_{k-1,k-j}$ ($j = 0, 1, \dots, k-1$) has to be saved in the previous cycle. Note further that only the first column of $F_{k,k-j+1}$ enters the equation determining $F_{k,k-j}$, but we still compute $F_{k,j}$ ($j = k, k-1, \dots, 1$), although only $F_{k,1} \mathbf{e}_1$ is needed, and, most importantly, the above Sylvester equation (11) tends to be severely ill-conditioned since H_j and H_{j-k} represent compressions of the same matrix A and thus their spectra are by no means well separated.

3.2 Implementation Based on a Rational Approximation to f

Our new implementation of restarting Krylov subspace algorithms for approximating (1) is based on evaluating $r(\widehat{H}_k)\mathbf{e}_1 \approx f(\widehat{H}_k)\mathbf{e}_1$ using a rational approximation (cf. [7, 19])

$$f(\lambda) \approx r(\lambda) = p(\lambda) + \sum_{\ell=1}^N \frac{\alpha_\ell}{\omega_\ell - \lambda}$$

to f in partial fraction form with polynomial part p , coefficients α_ℓ and poles ω_ℓ which we assume to be simple. In other words, we compute

$$r(\widehat{H}_k)\mathbf{e}_1 = p(\widehat{H}_k)\mathbf{e}_1 + \sum_{\ell=1}^N \alpha_\ell (\omega_\ell I - \widehat{H}_k)^{-1} \mathbf{e}_1. \quad (12)$$

We note that in the most common application $f(\lambda) = \exp(\lambda)$, $\text{Re } \lambda \leq 0$, a polynomial p of degree zero or one usually suffices.

Evaluating $p(\widehat{H}_k)\mathbf{e}_1$ for a polynomial p of low degree is straightforward: Letting $p(\lambda) = \pi_1 \lambda + \pi_0$, for example, yields

$$\widehat{\mathbf{r}}_0 := p(\widehat{H}_k)\mathbf{e}_1 = \begin{bmatrix} (\pi_1 H_1 + \pi_0 I)\mathbf{e}_1 \\ \pi_2 E_2 \mathbf{e}_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

and for higher degrees one can proceed analogously. Evaluating the second expression in (12) consists of summing the vectors $\widehat{\mathbf{r}}_\ell := \alpha_\ell (\omega_\ell I - \widehat{H}_k)^{-1} \mathbf{e}_1$, $\ell = 1, 2, \dots, N$, i.e., solving the linear systems of equations $(\omega_\ell I - \widehat{H}_k)\widehat{\mathbf{r}}_\ell = \mathbf{e}_1$. Due to the sparsity pattern of the right hand side \mathbf{e}_1 and the block lower triangular form of \widehat{H}_k , this can be carried out recursively as

$$(\omega_\ell I - H_1)\mathbf{r}_{\ell,1} = \mathbf{e}_1, \quad (\omega_\ell I - H_j)\mathbf{r}_{\ell,j} = E_j \mathbf{r}_{\ell,j-1}, \quad j = 2, \dots, k,$$

where we have partitioned $\widehat{\mathbf{r}}_\ell = [\mathbf{r}_{\ell,1}^\top, \mathbf{r}_{\ell,2}^\top, \dots, \mathbf{r}_{\ell,k}^\top]^\top$ conformingly with \widehat{H}_k . Note that these are k Hessenberg systems of size m and thus inexpensive to solve. Moreover, in view of (8), we only require the last block of $r(\widehat{H}_k)\mathbf{e}_1$, which is obtained as

$$[O, \dots, O, I] r(\widehat{H}_k)\mathbf{e}_1 = \mathbf{r}_{0,k} + \sum_{\ell=1}^N \alpha_\ell \mathbf{r}_{\ell,k},$$

where $\mathbf{r}_{0,k}$ denotes the last block of $\widehat{\mathbf{r}}_0$.

The resulting algorithm is summarized below as Algorithm 2. (For simplicity, we assume that the polynomial part p of r is the zero polynomial.)

Algorithm 2: Restarted Arnoldi approximation for $f(A)\mathbf{b}$ based on rational approximation.

Given: $A, \mathbf{b}, \|\mathbf{b}\| = 1, f, (\alpha_\ell, \omega_\ell)_{\ell=1}^N$
 $\mathbf{v}_1 := \mathbf{b}, \widehat{\mathbf{f}}_0 := \mathbf{0}$
for $k = 1, 2, \dots$ **until convergence do**
 Compute Arnoldi decomposition $AV_k = V_k H_k + \eta_{k+1} \mathbf{v}_{km+1} \mathbf{e}_m^\top$ of $\mathcal{K}_m(A, \mathbf{v}_{(k-1)m+1})$.
 if $k = 1$ **then**
 for $\ell = 1, 2, \dots, N$ **do**
 Solve $(\omega_\ell I - H_k) \mathbf{r}_{\ell,1} = \mathbf{e}_1$
 else
 for $\ell = 1, 2, \dots, N$ **do**
 Solve $(\omega_\ell I - H_k) \mathbf{r}_{\ell,k} = \eta_k (\mathbf{e}_m^\top \mathbf{r}_{\ell,k-1}) \mathbf{e}_1$
 $\mathbf{h}_k := \sum_{\ell=1}^N \alpha_\ell \mathbf{r}_{\ell,k}$
 $\widehat{\mathbf{f}}_k := \widehat{\mathbf{f}}_{k-1} + V_k \mathbf{h}_k$

In many applications both A and \mathbf{b} are real and f has the property that $f(\bar{\lambda}) = \overline{f(\lambda)}$. In this case $f(A)\mathbf{b}$ is also real and it is natural to approximate this vector using real arithmetic.

The rational approximation to f is usually also real for real arguments, but its poles ω_ℓ and coefficients α_ℓ appear in complex conjugate pairs, say $\omega_{\ell+1} = \overline{\omega_\ell}$ and $\alpha_{\ell+1} = \overline{\alpha_\ell}$. Since all other quantities in the equations $(\omega_\ell I - \widehat{H}_k) \widehat{\mathbf{r}}_\ell = \widehat{\mathbf{e}}_1$ are real, we have $\widehat{\mathbf{r}}_{\ell+1} = \overline{\widehat{\mathbf{r}}_\ell}$ and therefore $\mathbf{r}_{\ell+1,j} = \overline{\mathbf{r}_{\ell,j}}$ for all $j = 1, 2, \dots, k$. For the quantities entering the update of $\widehat{\mathbf{f}}_{k-1}$, we thus have

$$\alpha_\ell \mathbf{r}_{\ell,k} + \alpha_{\ell+1} \mathbf{r}_{\ell+1,k} = \alpha_\ell \mathbf{r}_{\ell,k} + \overline{\alpha_\ell \mathbf{r}_{\ell,k}} = 2 \operatorname{Re}(\alpha_\ell \mathbf{r}_{\ell,k})$$

and there is no need to solve $(\omega_{\ell+1} I - H_k) \mathbf{r}_{\ell+1,k} = E_k \mathbf{r}_{\ell+1,k-1}$. Setting $\mathbf{r}_{\ell,j} = \mathbf{r}_{\ell,j}^{(R)} + i \mathbf{r}_{\ell,j}^{(I)}$ and $\omega_\ell = \omega_\ell^{(R)} + i \omega_\ell^{(I)}$, a straightforward calculation shows that

$$\begin{aligned} (|\omega_\ell|^2 I - 2\omega_\ell^{(R)} H_1 + H_1^2) \mathbf{r}_{\ell,1}^{(R)} &= (\omega_\ell^{(R)} I - H_1) \mathbf{e}_1, \\ \mathbf{r}_{\ell,1}^{(I)} &= \frac{1}{\omega_\ell^{(I)}} \left(\left[\omega_\ell^{(R)} I - H_1 \right] \mathbf{r}_{\ell,1}^{(R)} - \mathbf{e}_1 \right), \end{aligned}$$

while for $j = 2, 3, \dots, k$,

$$\begin{aligned} (|\omega_\ell|^2 I - 2\omega_\ell^{(R)} H_j + H_j^2) \mathbf{r}_{\ell,j}^{(R)} &= \omega_\ell^{(I)} E_j \mathbf{r}_{\ell,j-1}^{(I)} + (\omega_\ell^{(R)} I - H_j) E_j \mathbf{r}_{\ell,j-1}^{(R)}, \\ \mathbf{r}_{\ell,j}^{(I)} &= \frac{1}{\omega_\ell^{(I)}} \left(\left[\omega_\ell^{(R)} I - H_j \right] \mathbf{r}_{\ell,j}^{(R)} - E_j \mathbf{r}_{\ell,j-1}^{(R)} \right). \end{aligned}$$

Finally, $\alpha_\ell \mathbf{r}_{\ell,k} + \alpha_{\ell+1} \mathbf{r}_{\ell+1,k} = 2 \operatorname{Re}(\alpha_\ell \mathbf{r}_{\ell,k}) = 2 \left[\operatorname{Re}(\alpha_\ell) \mathbf{r}_{\ell,k}^{(R)} - \operatorname{Im}(\alpha_\ell) \mathbf{r}_{\ell,k}^{(I)} \right]$ and we have avoided complex arithmetic.

To summarize, the two main ingredients of Algorithm 2 for computing $f(A)\mathbf{b}$ are the Arnoldi process (there is no difference to Algorithm 1) and (the partial fraction decomposition of) a rational function r such that $r(\widehat{H}_k) \approx f(\widehat{H}_k)$. In the following we restrict our attention to the important special case that f is the exponential function and always choose r as its best uniform rational approximation on $(-\infty, 0]$ of type 16 as derived by Richard Varga and co-workers in [3] and [2]. Alternative rational approximations are described in [21].

4 A Stopping Criterion Based on an Error Indicator

To obtain a practical stopping criterion for our restarting schemes, we develop an error indicator using an extension of an idea described in [19, Theorem 5.1] (see also [17, Theorem 3.1]). The approach relies on an interpolation expansion of the approximation error $f(A)\mathbf{b} - \widehat{\mathbf{f}}_k$ obtained by adding a sequence of auxiliary nodes $\{\vartheta_j\}$ in addition to the km Ritz values on which the restarted approximation $\widehat{\mathbf{f}}_k$ is based.

Given \widetilde{m} complex numbers $\vartheta_1, \vartheta_2, \dots, \vartheta_{\widetilde{m}}$ such that $f(\vartheta_j)$ is defined for each j and the associated nodal polynomials

$$w_0(\lambda) := 1, \quad w_j(\lambda) := (\lambda - \vartheta_1)(\lambda - \vartheta_2) \cdots (\lambda - \vartheta_j), \quad j = 1, 2, \dots, \widetilde{m},$$

we denote the associated divided differences of f by

$$\phi_0(\lambda) := f(\lambda), \quad \phi_j(\lambda) := [\Delta_{w_j} f](\lambda), \quad j = 1, 2, \dots, \widetilde{m}.$$

From the interpolation identity

$$I_{w_{j+1}} f = I_{w_j} f + w_j \Delta_{w_j} f, \quad j = 0, 1, \dots, \widetilde{m} - 1,$$

we see that these obey the recursion

$$\phi_j(\lambda) = \frac{\phi_{j-1}(\lambda) - \phi_{j-1}(\vartheta_j)}{\lambda - \vartheta_j}, \quad j = 1, 2, \dots, \widetilde{m}.$$

Finally, for $0 \leq \ell \leq \widetilde{m}$ and $0 \leq j \leq \widetilde{m} - \ell$ we define $\Delta_\ell^j f$ to be the j -th order divided difference of f with respect to the nodes $\vartheta_\ell, \vartheta_{\ell+1}, \dots, \vartheta_{\ell+j}$, i.e.,

$$\Delta_\ell^j f := \frac{1}{2\pi i} \int_\Gamma \frac{f(\lambda)}{(\lambda - \vartheta_\ell) \cdots (\lambda - \vartheta_{\ell+j})} d\lambda.$$

Given the Arnoldi-like decomposition (7), we now consider the matrix

$$W_{\widetilde{m}} := [w_0(A)\mathbf{v}_{km+1}, w_1(A)\mathbf{v}_{km+1}, \dots, w_{\widetilde{m}-1}(A)\mathbf{v}_{km+1}] \in \mathbb{C}^{n \times \widetilde{m}} \quad (13)$$

and the bidiagonal matrix

$$B_{\widetilde{m}} := \begin{bmatrix} \vartheta_1 & & & & \\ 1 & \vartheta_2 & & & \\ & \ddots & \ddots & & \\ & & & 1 & \vartheta_{\widetilde{m}} \end{bmatrix} \in \mathbb{C}^{\widetilde{m} \times \widetilde{m}},$$

in terms of which there holds $AW_{\tilde{m}} = W_{\tilde{m}}B_{\tilde{m}} + [\mathbf{0}, \dots, \mathbf{0}, w_{\tilde{m}}(A)\mathbf{v}_{km+1}]$. Together with (7) we obtain the Arnoldi-like decomposition

$$A \begin{bmatrix} \widehat{V}_k & W_{\tilde{m}} \end{bmatrix} = \begin{bmatrix} \widehat{V}_k & W_{\tilde{m}} \end{bmatrix} \widetilde{H}_k + w_{\tilde{m}}(A)\mathbf{v}_{km+1}\mathbf{e}_{km+\tilde{m}}^\top,$$

where

$$\widetilde{H}_k = \left[\begin{array}{c|c} \widehat{H}_k & O \\ \hline \widetilde{E}_k & B_{\tilde{m}} \end{array} \right] \in \mathbb{C}^{(km+\tilde{m}) \times (km+\tilde{m})} \text{ and } \widetilde{E}_k = \eta_k \mathbf{e}_1 \mathbf{e}_{km}^\top \in \mathbb{R}^{\tilde{m} \times km}.$$

If we now approximate $f(A)\mathbf{b}$ by $\widetilde{\mathbf{f}}_k := \begin{bmatrix} \widehat{V}_k & W_{\tilde{m}} \end{bmatrix} f(\widetilde{H}_k)\mathbf{e}_1$, then the associated error may be represented as (see [6, Theorem 2.6])

$$f(A)\mathbf{b} - \widetilde{\mathbf{f}}_k = \widetilde{f}(A) w_{\tilde{m}}(A)\mathbf{v}_{km+1}, \quad (14)$$

where $\widetilde{f} := \gamma \Delta_{\widetilde{w}} f$ with $\widetilde{w} \in \mathcal{P}_{km+\tilde{m}}$ the characteristic polynomial of \widetilde{H}_k and γ the product of the subdiagonal entries of \widetilde{H}_k .

Lemma 4.1. *In terms of the notation introduced above, there holds*

$$f(\widetilde{H}_k) = \left[\begin{array}{c|c} f(\widehat{H}_k) & O \\ \hline \widetilde{F}_{k,\tilde{m}} & f(B_{\tilde{m}}) \end{array} \right], \quad (15)$$

where

$$f(B_{\tilde{m}}) = \left[\begin{array}{cccc} f(\vartheta_1) & & & \\ \Delta_1^1 & f(\vartheta_2) & & \\ \vdots & \vdots & \ddots & \\ \Delta_1^{\tilde{m}-1} & \Delta_2^{\tilde{m}-2} & \dots & f(\vartheta_{\tilde{m}}) \end{array} \right] \in \mathbb{C}^{\tilde{m} \times \tilde{m}} \quad (16)$$

and where $\widetilde{F}_{k,\tilde{m}} \in \mathbb{C}^{\tilde{m} \times km}$ has the rows

$$\mathbf{e}_j^\top \widetilde{F}_{k,\tilde{m}} = \eta_k \mathbf{e}_{km}^\top \phi_j(\widehat{H}_k), \quad j = 1, 2, \dots, \tilde{m}. \quad (17)$$

Proof. The first assertion (16) was proven in [15]. To show (17) we follow the proof of Proposition 3.2 in [17]. Note first that comparing the (2,1) blocks in the identity $f(\widetilde{H}_k)\widetilde{H}_k = \widetilde{H}_k f(\widetilde{H}_k)$ yields

$$\widetilde{F}_{k,\tilde{m}}\widehat{H}_k - B_{\tilde{m}}\widetilde{F}_{k,\tilde{m}} = \eta_k \left[\mathbf{e}_1 \mathbf{e}_{km}^\top f(\widehat{H}_k) - f(B_{\tilde{m}})\mathbf{e}_1 \mathbf{e}_{km}^\top \right]. \quad (18)$$

We obtain (17) by induction on j . For $j = 1$, multiplying by \mathbf{e}_1^\top on both sides of (18) yields

$$\mathbf{e}_1^\top \widetilde{F}_{k,\tilde{m}}(\widehat{H}_k - \vartheta_1 I) = \eta_k \mathbf{e}_{km}^\top \left(f(\widehat{H}_k) - f(\vartheta_1)I \right),$$

or

$$\mathbf{e}_1^\top \tilde{F}_{k,\tilde{m}} = \eta_k \mathbf{e}_{km}^\top \left(f(\hat{H}_k) - f(\vartheta_1 I) \right) (\hat{H}_k - \vartheta_1 I)^{-1} = \eta_k \mathbf{e}_{km}^\top \phi_1(\hat{H}_k).$$

For $j > 1$, multiplying (18) from the left by \mathbf{e}_j^\top leads to

$$\mathbf{e}_j^\top \tilde{F}_{k,\tilde{m}} (\hat{H}_k - \vartheta_j I) = \mathbf{e}_{j-1}^\top \tilde{F}_{k,\tilde{m}} - \eta_k \Delta_1^{j-1} \mathbf{e}_{km}^\top = \eta_k \mathbf{e}_{km}^\top \left(\phi_{j-1}(\hat{H}_k) - \phi_{j-1}(\vartheta_j) \right),$$

from which (17) follows after multiplying by $(\hat{H}_k - \vartheta_j I)^{-1}$. Note that we have tacitly assumed that $\vartheta_j \notin \Lambda(\hat{H}_k)$ for all j . The usual confluent divided difference calculus shows, however, that the assertion is also valid without this restriction. \square

With the expression for the columns of $\tilde{F}_{k,\tilde{m}}$ given in Lemma 4.1 together with the definition of $W_{\tilde{m}}$ (13), we find that

$$\tilde{\mathbf{f}}_k = \hat{V}_k f(\hat{H}_k) \mathbf{e}_1 + \eta_k \sum_{j=1}^{\tilde{m}} \left[\mathbf{e}_{km}^\top \phi_j(\hat{H}_k) \mathbf{e}_1 \right] w_{j-1}(A) \mathbf{v}_{km+1}.$$

Together with the error representation (14) we have obtained

Theorem 4.2. *The error of the restarted Arnoldi approximation (8) can be expanded as*

$$f(A) \mathbf{b} - \hat{V}_k f(\hat{H}_k) \mathbf{e}_1 = \eta_k \sum_{j=1}^{\tilde{m}} \left[\mathbf{e}_{km}^\top \phi_j(\hat{H}_k) \mathbf{e}_1 \right] w_{j-1}(A) \mathbf{v}_{km+1} + \tilde{f}(A) w_{\tilde{m}}(A) \mathbf{v}_{km+1}. \quad (19)$$

Remark 4.3. *By an obvious modification to a result given in [19, Theorem 5.1] one can show that the remainder term in (19) may also be written*

$$\tilde{f}(A) w_{\tilde{m}}(A) \mathbf{v}_{km+1} = w_{\tilde{m}}(A) \left[\phi_{\tilde{m}}(A) \mathbf{b} - \hat{V}_k \phi_{\tilde{m}}(\hat{H}_k) \mathbf{e}_1 \right].$$

The sum in (19) represents the leading \tilde{m} terms of an interpolation series (see [23, Chapter III]) for the function $f - I_w f$, where $w \in \mathcal{P}_{km}$ denotes the nodal polynomial of the eigenvalues of \hat{H}_k . Provided the series converges as $\tilde{m} \rightarrow \infty$ — for $f = \exp$ it is sufficient to assume the nodes are bounded — we obtain the expansion

$$f(A) \mathbf{b} - \hat{V}_k f(\hat{H}_k) \mathbf{e}_1 = \eta_k \sum_{j=1}^{\infty} \left[\mathbf{e}_{km}^\top \phi_j(\hat{H}_k) \mathbf{e}_1 \right] w_{j-1}(A) \mathbf{v}_{km+1}$$

of the error of the restarted Arnoldi approximation. We obtain an error indicator by truncating the series after one or two terms, where we are free to choose the additional nodes $\{\vartheta_j\}_{j=1}^{\tilde{m}}$. In the numerical experiments presented below, we choose $\vartheta_1 = \min \Lambda(\hat{H}_k)$ for $\tilde{m} = 1$ and $\vartheta_2 = \max \Lambda(\hat{H}_k)$ with the same choice for ϑ_1 when $\tilde{m} = 2$. Computationally, all that is required for evaluating these error estimates are the coefficients $\mathbf{e}_{km}^\top \phi_j(\hat{H}_k) \mathbf{e}_1$ and the vectors $w_{j-1}(A) \mathbf{v}_{km+1}$. The former can be extracted from the first column of

$\tilde{F}_{k,\tilde{m}}$ (cf. (17)) after applying f to \tilde{H}_k , a matrix of size $km + \tilde{m}$ only slightly larger than \hat{H}_k , whose evaluation is called for in Algorithm 1 and Algorithm 2. The latter require $\tilde{m} - 1$ additional matrix-vector products of A with \mathbf{v}_{km+1} . We emphasize that for $\tilde{m} = 1$ this means no additional matrix-vector products are necessary and that for $\tilde{m} = 2$ the additional matrix-vector product $A\mathbf{v}_{km+1}$ can be reused in the generation of the Krylov space of the $(k + 1)$ -st restart cycle.

Note that the choices for ϑ_1 and ϑ_2 given above for $\tilde{m} = 1, 2$ correspond to a Gauss-Radau and a Gauss-Lobatto quadrature formula for the error (cf. [8]). For certain functions, among these the exponential, it can be shown that these choices result in lower and upper bounds, respectively, for the error. A rigorous derivation of quadrature-based a-posteriori error bounds will be the subject of future work.

5 Convergence Issues: A Case Study

With Algorithms 1 and 2, we have described two techniques for implementing a restarted Krylov subspace method for the evaluation of $f(A)\mathbf{b}$. These differ mainly in the way the function f is applied to the compressions \hat{H}_k of A . Considering specifically the exponential function $f = \exp$, in Algorithm 1 a standard routine such as the built-in MATLAB function `expm` is applied. For the current implementation[‡] of `expm`, this means that we approximate $\exp(\hat{H}_k)$ by a rational expression of the form $r_1(\hat{H}_k) := r(\hat{H}_k/2^s)^{2^s}$, where r is a $[t/t]$ Padé fraction with $t \leq 13$ for the exponential function and $s \in \mathbb{N}_0$ depends on $\|\hat{H}_k\|_1$ (see [10]). Note that s and r , and thus r_1 , depend on the argument. Therefore, r_1 is *by no means a rational function*—but rather it represents a family of rational functions from which one member is chosen depending on the current argument: If this is a scalar λ then $r_1(\lambda)$ is an accurate approximation to $\exp(\lambda)$ regardless of where in the complex plane λ is located. In floating point arithmetic, r_1 is for all practical purposes indistinguishable from `exp`; in particular, r_1 has no finite poles.

By contrast, Algorithm 2 approximates $\exp(\hat{H}_k)$ by $r_2(\hat{H}_k)$, where r_2 is a *fixed rational function*, namely the best uniform rational approximation of type 16 to the exponential function on $(-\infty, 0]$. In sharp contrast to r_1 above, r_2 approximates `exp` well only in a neighborhood of the negative real axis and therefore lacks the universal approximation property of r_1 . Moreover, r_2 has finite poles. When approximating $\exp(A)$ for matrices with real nonpositive eigenvalues, as arise e.g. in connection with parabolic initial value problems, this may appear as a somewhat academic issue; but, as we shall see, this leads to a tremendous difference in numerical behavior between the two approaches if the restart length m is sufficiently small. The main distinction is that a restarted Krylov approximation to $f(A)\mathbf{b}$ converges superlinearly if f is an entire function (such as `exp`, see [6, Theorem 4.2]), whereas, asymptotically, only a linear rate of convergence results if f has finite singularities (such as r_2).

We demonstrate this using the following model problem. Let

$$A = \text{diag}(-100, -99, \dots, 0) \in \mathbb{R}^{101 \times 101} \quad \text{and} \quad \mathbf{b} = [1, 1, \dots, 1]^\top / \sqrt{101} \in \mathbb{R}^{101}.$$

[‡]Release 2007a

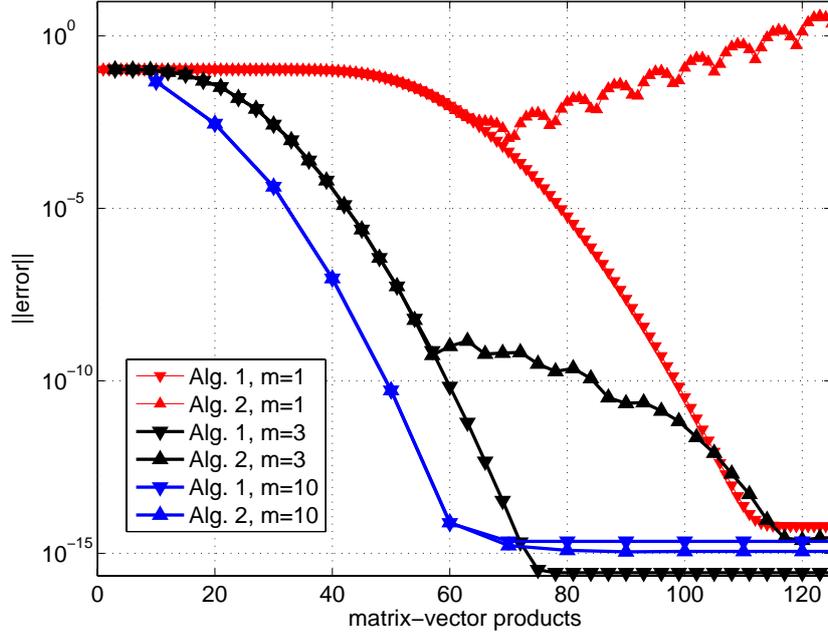


Figure 1: Absolute errors of the approximations of $f(A)\mathbf{b}$ computed with Algorithms 1 and 2 for restart lengths $m = 1, 3$ and 10 .

(We choose such a simple example in order that the quantities entering our analysis be explicitly known.) Applying Algorithms 1 and 2 yields approximations to $\exp(A)\mathbf{b}$ which we shall denote by $\hat{\mathbf{f}}_k^{(1)}$ and $\hat{\mathbf{f}}_k^{(2)}$, respectively. We consider restart cycles consisting of $m = 1$, $m = 3$ and $m = 10$ steps. Figure 1 shows the corresponding absolute errors $\|\hat{\mathbf{f}}_k^{(\mu)} - \exp(A)\mathbf{b}\|$ ($\mu = 1, 2$).

For restart length $m = 1$, we observe that Algorithm 1 converges while Algorithm 2 does not. To explain why we recall that both algorithms are based on interpolation processes. The interpolation nodes are the eigenvalues of the Hessenberg matrices \hat{H}_k which, for restart length $m = 1$, are bidiagonal. It can be shown[§] that, in our example, the diagonal entries (and thus the eigenvalues) of \hat{H}_k are all equal to $\vartheta = -50$. Thus for restart length $m = 1$, both algorithms are based on (truncated) Taylor expansions of

[§]Let $p_k(\lambda) = \lambda^k \in \mathcal{P}_k$ denote the k -th monomial and define the vectors $\mathbf{w}_k = [p_k(-50), p_k(-49), \dots, p_k(50)]^\top$. Then $\sum_{j=-50}^{50} j p_k(j)^2 = 2 \sum_{j=0}^{50} (j-j)p_k(j)^2 = 0$ and therefore, $\rho_k := (\mathbf{w}_k^\top A \mathbf{w}_k) / (\mathbf{w}_k^\top \mathbf{w}_k) = (\sum_{j=-50}^{50} (j-50)p_k(j)^2) / (\sum_{j=-50}^{50} p_k(j)^2) = -50$. Consider the vector $A\mathbf{w}_k - \rho_k \mathbf{w}_k$ whose j -th component (indexed by $j \in \{-50, -49, \dots, 50\}$) is $(j-50)p_k(j) + 50p_k(j) = j p_k(j) = p_{k+1}(j)$, i.e., $A\mathbf{w}_k - \rho_k \mathbf{w}_k = \mathbf{w}_{k+1}$. Since for the first Arnoldi vector \mathbf{v}_1 , there holds $\mathbf{v}_1 = \mathbf{b} = \mathbf{w}_0 / \|\mathbf{w}_0\|$ and since the Arnoldi vectors satisfy the recursion $\mathbf{v}_{k+1} = (A\mathbf{v}_k - (\mathbf{v}_k^\top A \mathbf{v}_k)\mathbf{v}_k) / \|A\mathbf{v}_k - (\mathbf{v}_k^\top A \mathbf{v}_k)\mathbf{v}_k\|$, we see that $\mathbf{v}_k = \mathbf{w}_k / \|\mathbf{w}_k\|$ for all k . Thus, $\mathbf{v}_k^\top A \mathbf{v}_k = (\mathbf{w}_k^\top A \mathbf{w}_k) / (\mathbf{w}_k^\top \mathbf{w}_k) = -50$, but these are the diagonal entries of \hat{H}_k .

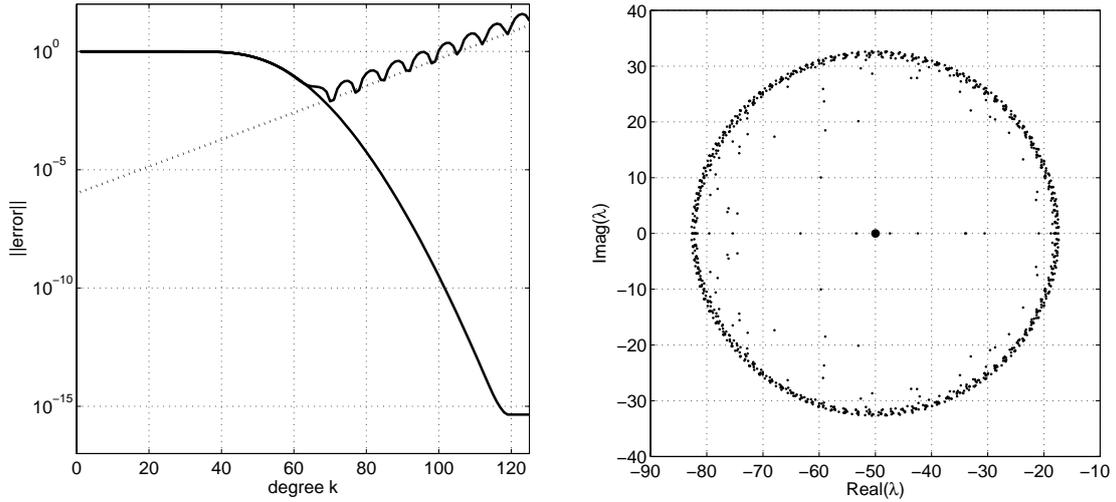


Figure 2: Left: Errors $\max_{\lambda \in \Lambda(A)} |q_{k-1}^{(\mu)}(\lambda) - \exp(\lambda)|$ for the truncated Taylor series $q_{k-1}^{(\mu)}$ of \exp ($\mu = 1$) and r_2 ($\mu = 2$).
 Right: Eigenvalues of $\widehat{H}_{100} + E$ for ten random matrices of norm $\|E\| = 2^{-52}$.

$f = \exp$ and $f = r_2$, respectively, about $\vartheta = -50$: We have

$$\widehat{\mathbf{f}}_k^{(1)} = \widehat{V}_k \exp(\widehat{H}_k) \mathbf{e}_1 = q_{k-1}^{(1)}(A) \mathbf{b} = \sum_{j=0}^{k-1} \frac{\exp(-50)}{j!} (A + 50I)^j \mathbf{b} \text{ and}$$

$$\widehat{\mathbf{f}}_k^{(2)} = \widehat{V}_k r_2(\widehat{H}_k) \mathbf{e}_1 = q_{k-1}^{(2)}(A) \mathbf{b} = \sum_{j=0}^{k-1} \frac{r_2^{(j)}(-50)}{j!} (A + 50I)^j \mathbf{b}.$$

The Taylor polynomials $\{q_{k-1}^{(1)}\}_{k \geq 1}$ converge (albeit slowly) to \exp uniformly on compact subsets of \mathbb{C} . Figure 2 shows the errors $\|q_{k-1}^{(1)} - \exp\|_{\infty, \Lambda(A)} := \max_{\lambda \in \Lambda(A)} |q_{k-1}^{(1)}(\lambda) - \exp(\lambda)|$ of these Taylor polynomials, and these are seen to agree perfectly with the errors of Algorithm 1. The Taylor polynomials $\{q_{k-1}^{(2)}\}_{k \geq 1}$, on the other hand, converge to r_2 in a disk with center $\vartheta = -50$ and radius $\min_{\omega} |\omega + 50|$, where ω runs over all poles of r_2 , and they diverge outside this disk. The poles of r_2 closest to $\vartheta = -50$ are $\omega \approx -11 \pm 19i$ with $|\omega - \vartheta| \approx 44 < 50$. In other words, the Taylor series of r_2 (with expansion point $\vartheta = -50$) has $\tau \approx 44$ as its radius of convergence. Since some of the eigenvalues of A lie outside the convergence disk, the sequence $\{\widehat{\mathbf{f}}_k^{(2)}\}_{k \geq 1}$ must ultimately diverge like $[\max_{\lambda \in \Lambda(A)} |\lambda + 50|/44]^k = (50/44)^k \approx 1.14^k$ (cf. the dotted line in Figure 2). Moreover, Figure 2 shows that also $\|q_{k-1}^{(2)} - \exp\|_{\infty, \Lambda(A)}$ is in perfect agreement with the errors of Algorithm 2. (Note that this error curve cannot be distinguished from $\|q_{k-1}^{(2)} - r_2\|_{\infty, \Lambda(A)}$ because $\|r_2 - \exp\|_{\infty, \Lambda(A)} < 10^{-15}$.)

The matrices \widehat{H}_k are highly nonnormal ($\vartheta = -50$ is their only eigenvalue, with algebraic multiplicity k but geometric multiplicity 1). This raises the question of whether

it is justified to base our analysis on this eigenvalue of \widehat{H}_k , which is extremely sensitive to perturbations, or whether an approach using pseudo-eigenvalues would not be more appropriate. Our answer is that it does not matter. This is a consequence of a theorem due to J. L. Walsh [23, Theorem 7.1] on the overconvergence of differences of interpolating polynomials (which by the way, is another area where Richard Varga has made significant contributions). As mentioned above, the diagonal entries of \widehat{H}_k are all equal to -50 . We observe that its subdiagonal entries $\eta_{j,j-1}$ converge to 50 (see [1] for a proof), i.e., asymptotically \widehat{H}_k resembles a Toeplitz matrix with symbol $50(\lambda - 1)$, and thus its pseudo-spectra (asymptotically) are disks with center -50 (cf. [18]). In Figure 2, the eigenvalues of $\widehat{H}_{100} + E$ are shown for ten random perturbations of norm $\|E\| = \varepsilon = 2^{-52}$. The large majority of these pseudo-eigenvalues lies on a circle with center -50 and some radius $\delta > 0$ (here $\delta \approx 32$). If we now consider $\widehat{\mathbf{f}}_k^{(3)} = q_{k-1}^{(3)}(A)\mathbf{b}$, where q_{k-1} interpolates r_2 at the nodes $-50 + \delta \exp(2\pi i j/k)$ ($j = 0, 1, \dots, k-1$), then Walsh's overconvergence result tells us that, as long as $\delta < 44$, the difference of the interpolating polynomials $|q_{k-1}^{(2)}(\lambda) - q_{k-1}^{(3)}(\lambda)|$ tends to zero for $|\lambda + 50| \leq 44^2/\delta$. This convergence is linear, more precisely $\limsup_{k \rightarrow \infty} \max_{|\lambda+50| \leq \tau} |q_{k-1}^{(2)}(\lambda) - q_{k-1}^{(3)}(\lambda)|^{1/k} = (\delta\tau)/44^2$. We set $\delta = 32$ and observe that $\limsup_{k \rightarrow \infty} \|q_{k-1}^{(2)} - q_{k-1}^{(3)}\|_{\infty, \Lambda(A)}^{1/k} \leq (50 \cdot 32)/(44^2) \approx 0.83$. In other words, $\|\widehat{\mathbf{f}}_k^{(2)} - \widehat{\mathbf{f}}_k^{(3)}\|$ tends to zero like 0.83^k which is negligible compared to the size of $\|\widehat{\mathbf{f}}_k^{(2)} - \exp(A)\mathbf{b}\|$.

The behavior of Algorithms 1 and 2 for the restart lengths $m = 3$ and $m = 10$ (cf. Figure 1) can be explained along the same lines: Algorithm 1 relies on an interpolation process for the exponential function whereas in Algorithm 2 its uniform best rational approximation r_2 is interpolated. The nodes are again the eigenvalues of the block bidiagonal matrices \widehat{H}_k , i.e., the eigenvalues of its diagonal blocks H_1, H_2, \dots, H_k which are symmetric tridiagonal matrices of dimension $m = 3$ and $m = 10$, respectively. Now the following observation is crucial: There holds

$$\lim_{j \rightarrow \infty} H_{2j-1} = \widetilde{H}_1 \quad \text{and} \quad \lim_{j \rightarrow \infty} H_{2j} = \widetilde{H}_2$$

(see [1] for a theoretical justification). The nodal sequence $\vartheta_1, \vartheta_2, \vartheta_3, \dots$ therefore has the property

$$\lim_{j \rightarrow \infty} \vartheta_{2mj+\nu} = \widetilde{\vartheta}_\nu \quad \text{for } \nu = 1, 2, \dots, 2m.$$

Asymptotically, we interpolate \exp and r_2 at $2m$ nodes which are repeated cyclically. In our example, these nodes are given by

$$\widetilde{\vartheta}_1 \approx -87, \quad \widetilde{\vartheta}_2 = -50, \quad \widetilde{\vartheta}_3 \approx -13, \quad \widetilde{\vartheta}_4 \approx -92, \quad \widetilde{\vartheta}_5 = -50, \quad \widetilde{\vartheta}_6 \approx -8$$

if $m = 3$. The convergence properties of the corresponding interpolation polynomials can be described in terms of the lemniscates

$$L_\tau := \{\lambda \in \mathbb{C} : |w_{2m}(\lambda)| = \tau^{2m}\}, \quad \tau > 0, \quad w_{2m}(\lambda) = \prod_{\nu=1}^{2m} (\lambda - \widetilde{\vartheta}_\nu).$$

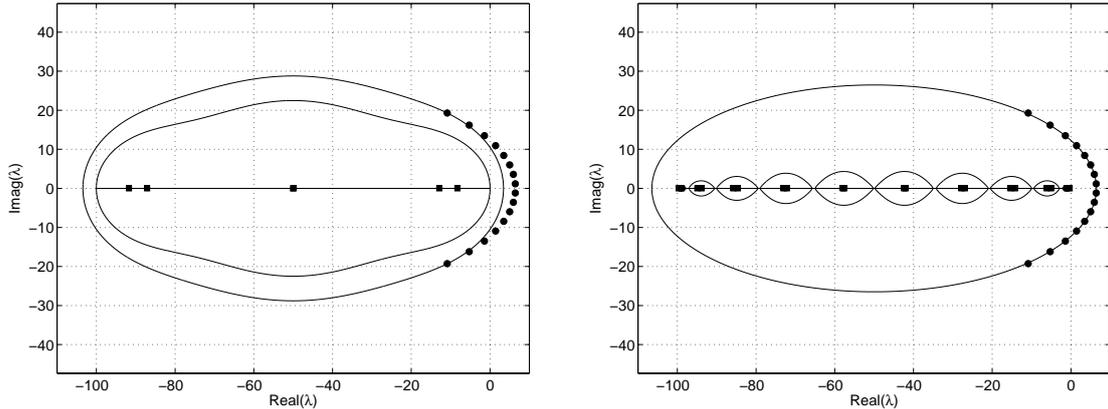


Figure 3: Lemniscates L_τ for $\tau = \tau_s$ and $\tau = \tau_A$ governing the convergence rate of Algorithm 2 for restart lengths $m = 3$ (left) and $m = 10$ (right). The squares mark the nodes ϑ_j , the dots the poles of the rational approximation r_2 .

From a theorem of Walsh [23, Theorem 3.6] it follows that

$$\limsup_{k \rightarrow \infty} \|\widehat{\mathbf{f}}_k^{(2)} - \exp(A)\mathbf{b}\|^{1/km} = \frac{\tau_s}{\tau_A},$$

where

$$\tau_s = \max\{\tau : r_2 \text{ is analytic in the interior of } L_\tau\},$$

i.e., L_{τ_s} is the largest lemniscate with foci $\tilde{\vartheta}_\nu$ ($\nu = 1, 2, \dots, 2m$) such that r_2 is analytic in the interior of L_{τ_s} , and where

$$\tau_A = \min\{\tau : \Lambda(A) \text{ is contained in the closed interior of } L_\tau\},$$

i.e., L_{τ_A} is the smallest lemniscate with foci $\tilde{\vartheta}_\nu$ such that all eigenvalues of A are contained in the closed interior of L_{τ_A} . Figure 3 shows these lemniscates for our example.

To summarize, if we replace the exponential function by its rational best approximation then the convergence of the restarted Lanczos method shows two phases. Initially, we observe the error behavior of a polynomial approximation to an entire function, i.e., after a start-up phase, where the error is not reduced, the polynomial converges super-linearly. But there is a point from where on the poles become visible and then we have slower linear convergence or even linear divergence. This point is fairly independent of the restart length, whereas the linear rate of convergence/divergence depends on it (of course, it also depends on the eigenvalues of A). The aim is to choose the restart length large enough such that at the point of transition the desired accuracy is reached or nearly reached.

6 Numerical Examples

In this section we illustrate the performance of the two restart algorithms for some initial-boundary value problems. All computations were carried out in MATLAB Release

2007a on an Intel Xeon 5160 at 3 GHz with 16 GB RAM running SuSE Linux Enterprise Server (SLES) Version 10.

6.1 The Heat Equation

Our first numerical experiment is based on a standard example in this area (see, e.g., [7] and [6]): We consider the initial-boundary value problem

$$\partial_t u - \Delta u = 0 \quad \text{in } \Omega = (0, 1)^3, \quad t > 0, \quad (20a)$$

$$u(x, t) = 0 \quad \text{on } \Gamma = \partial\Omega, \quad t > 0, \quad (20b)$$

$$u(x, 0) = u_0(x) \quad \text{in } \Omega. \quad (20c)$$

When the Laplacian is discretized by the usual seven-point stencil on a uniform grid involving n_1 interior grid points in each Cartesian direction, problem (20) reduces to the initial value problem

$$\mathbf{u}'(t) = A\mathbf{u}(t), \quad t > 0,$$

$$\mathbf{u}(0) = \mathbf{u}_0,$$

with an $n \times n$ matrix A ($n = n_1^3$) and an initial vector \mathbf{u}_0 consisting of the values $u_0(x)$ at the grid points x , the solution of which is given by

$$\mathbf{u}(t) = \exp(tA)\mathbf{u}_0. \quad (21)$$

As in [7], we give the initial vector in terms of its expansion in eigenfunctions of the discrete Laplacian as

$$\mathbf{u}_0^{i,j,k} = \sum_{i',j',k'} \frac{1}{i' + j' + k'} \sin(ii'\pi h) \sin(jj'\pi h) \sin(kk'\pi h).$$

Here $h = 1/(n_1 + 1)$ is the mesh size and the triple indexing is relative to the lexicographic ordering of the mesh points in the unit cube.

We consider this problem for a discretization with $n_1 = 50$, resulting in a matrix of dimension $n = 125,000$. We apply the restarted Lanczos approximation with restart lengths $m = 10, 20, 30$, and 50 as well as the full Lanczos algorithm ($m = \infty (I)$) and the two-pass Lanczos method ($m = \infty (II)$). Each iteration is run until the accuracy no longer improves. Note that, in the full methods ($m = \infty$), the evaluation of $\exp(H_k)$ is only performed once, when final accuracy is reached. The resulting execution times are shown in Table 1.

For $m = 10$, Algorithm 2 converges so slowly that the final accuracy was not reached even after 2000 matrix-vector multiplications. The fastest method is clearly the full Lanczos algorithm ($m = \infty (I)$), but note that it requires storing 282 vectors of dimension $n = 125,000$ (282 MB). To our surprise, the restarted method (for $m = 30$ and $m = 50$) is faster than the double-pass Lanczos algorithm. One also observes that Algorithm 2 is generally faster than Algorithm 1, but that the final accuracy of Algorithm 1 is higher.

m	Algorithm 1			Algorithm 2		
	time [s]	mvp	acc.	time [s]	mvp	acc.
$\infty (I)$	5.6	282	5e-14	3.9	282	5e-12
$\infty (II)$	9.7	564	5e-14	7.6	564	5e-12
50	4.8	350	3e-14	4.1	300	6e-12
30	7.9	360	2e-14	6.0	330	5e-12
20	9.5	380	5e-15	7.2	400	6e-12
10	17.4	430	9e-15	—	—	—

Table 1: Execution times, number of matrix-vector products and final accuracy for the solution of the heat equation ($n = 50$) for different restart lengths m . Here, $m = \infty (I)$ refers to the standard (unrestarted) Lanczos method, while $m = \infty (II)$ stands for the (unrestarted) two-pass Lanczos algorithm.

This loss of accuracy for Algorithm 2 stems from the eight linear systems we have to solve in each cycle. For $m = 50$ the condition numbers of their coefficient matrices vary between $2 \cdot 10^2$ and $4 \cdot 10^2$.

We conclude this example by applying the error indicators described in Section 4 for the restart lengths $m = 20$ and $m = 50$ (see Figure 4). We obtain lower ($\tilde{m} = 1$) and upper ($\tilde{m} = 2$) bounds for our two restart algorithms. By their construction it is clear that the error indicators cannot detect when the method begins to stagnate because final accuracy has been reached (but this can easily be detected by other means). Specifically, for Algorithm 2 the point where the convergence behavior changes from superlinear to linear is located precisely (even in cases where this occurs long after final accuracy is reached).

6.2 Maxwell's Equations

We next consider a problem which occurs in geoelectrical exploration and for which Lanczos-based Krylov subspace approximations have been very successful, see [5] and the references given there. In the absence of impressed source currents the time-evolution of an electric field $\mathbf{E} = \mathbf{E}(x, t)$ from a given initial state $\mathbf{E}_0 = \mathbf{E}_0(x)$ at time t_0 is the solution of the initial value problem

$$\partial_t(\sigma \mathbf{E}) + \nabla \times (\mu^{-1} \nabla \times \mathbf{E}) = \mathbf{0} \quad \text{in } \Omega, t > t_0, \quad (22a)$$

$$\mathbf{n} \times \mathbf{E} = \mathbf{0} \quad \text{on } \partial\Omega, \quad (22b)$$

$$\mathbf{E}(x, t_0) = \mathbf{E}_0(x) \quad \text{in } \Omega \quad (22c)$$

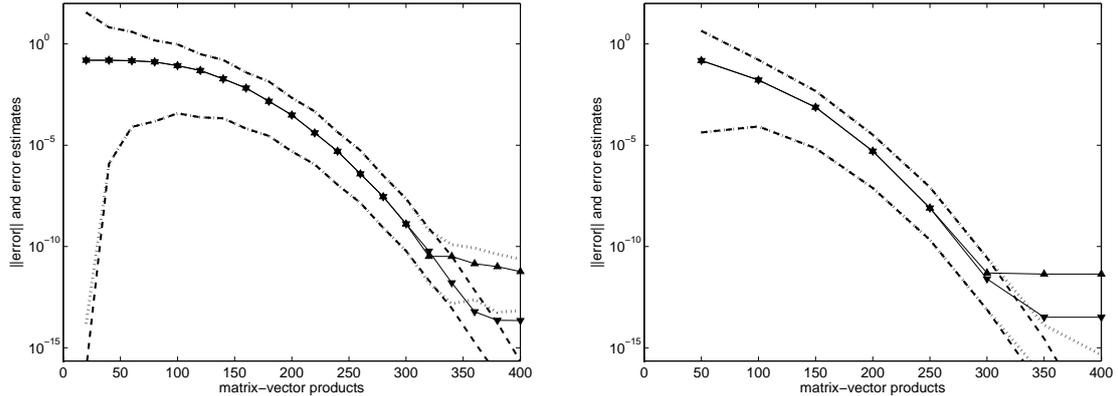


Figure 4: The error indicators described in Section 4 for the restart lengths $m = 20$ (left) and $m = 50$ (right) and for Algorithm 1 (dashed line) and 2 (dotted line).

for the quasi-static Maxwell's equations. Here the domain is a cube $\Omega = (-L, L)^3$, where $L = 2000$ and the magnetic permeability μ and electric conductivity σ are taken to have the constant values $\mu = 4\pi \cdot 10^{-7}$ and $\sigma = 0.1$. The initial data \mathbf{E}_0 is the field at time $t_0 = 10^{-6}$ of a vertical magnetic dipole of unit strength located at the origin for which an analytic expression is known (see [24]).

We discretize the operator $\sigma^{-1}\nabla \times (\mu^{-1}\nabla \times \cdot)$ in space using the Yee finite difference scheme [25] on a graded tensor product mesh (see Figure 5). After symmetrization (cf. [5]) this yields a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{e}_0 \in \mathbb{R}^n$ which is a sampled version of the initial electric field. The semi-discretized system (22) thus reduces to a linear linear system of ordinary differential equations with constant coefficients

$$\mathbf{e}'(t) = -A\mathbf{e}(t), \quad \mathbf{e}(t_0) = \mathbf{e}_0,$$

with solution

$$\mathbf{e}(t) = \exp(-(t - t_0)A)\mathbf{e}_0.$$

The eigenvalues of A are contained in $[0, \lambda_{\max}]$, $\lambda_{\max} \leq 13/(h_{\min}^2 \sigma \mu)$, where h_{\min} is the minimal distance of two adjacent grid-points.

In our example the matrix A is of size $n = 565,326$ and $\lambda_{\max} \approx 10^8$. We approximate $\mathbf{E}(t_1)$, $t_1 = 10^{-3}$, using the Lanczos algorithm (I), the two-pass Lanczos algorithm (II), as well as Algorithm 1 and Algorithm 2 given above. As a reference solution $\mathbf{e}(t_1)$ we used the approximation obtained by the Lanczos method when it stagnates at final accuracy.

In Algorithm 2 we solve 8 tridiagonal linear systems in each cycle instead of calling MATLAB's `expm` which is much more expensive. As a consequence the execution time of Algorithm 2 is dominated by the number of matrix-vector products. This explains why the restarted Krylov methods perform slightly faster than the full two-pass Lanczos method if Algorithm 2 is applied. We terminated our timing measurements for all algorithms when the absolute error to $\mathbf{e}(t_1)$ fell below 10^{-12} .

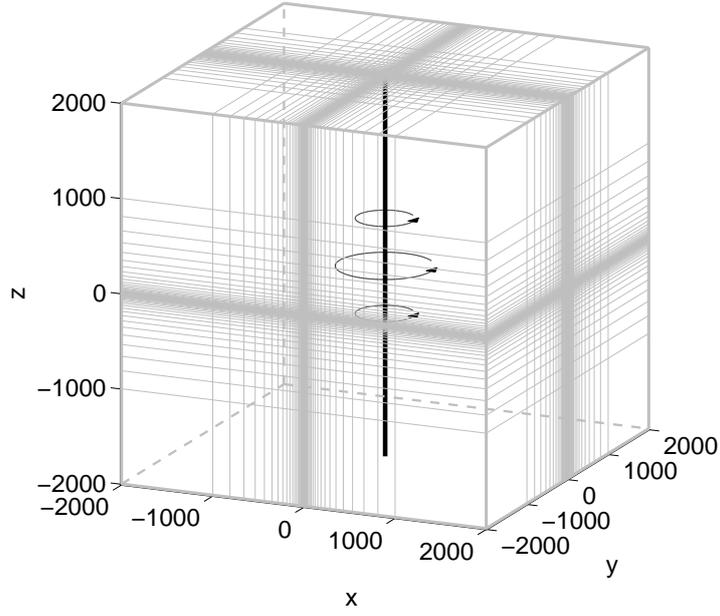


Figure 5: Computational domain and tensor product grid lines shown on the boundary for problem (22). The axis indicates the orientation of the magnetic dipole and the circular lines indicate the direction of the electric field.

m	Algorithm 1			Algorithm 2		
	time [s]	mvp	acc.	time [s]	mvp	acc.
$\infty (I)$	118	1072	9.93e-13	86	1072	9.93e-13
$\infty (II)$	176	2144	9.93e-13	144	2144	9.93e-13
90	273	1350	1.92e-13	118	1350	2.01e-13
70	339	1400	3.28e-13	112	1400	9.13e-13
50	613	1600	2.10e-13	very slow convergence		
30	2014	2040	5.64e-13	Krylov approximations diverge		

Table 2: Performance comparison of several variants of Lanczos and restarted Lanczos methods.

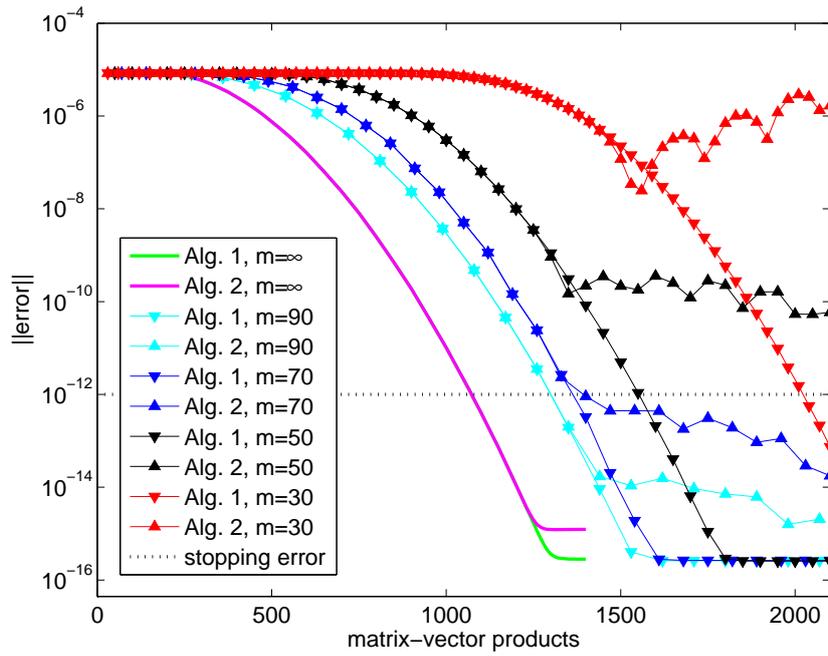


Figure 6: Absolute errors of the approximations of $f(A)\mathbf{b}$ for different restart lengths m . Here, $m = \infty (I)$ refers to the standard (unrestarted) Lanczos method, while $m = \infty (II)$ stands for the (unrestarted) two-pass Lanczos algorithm.

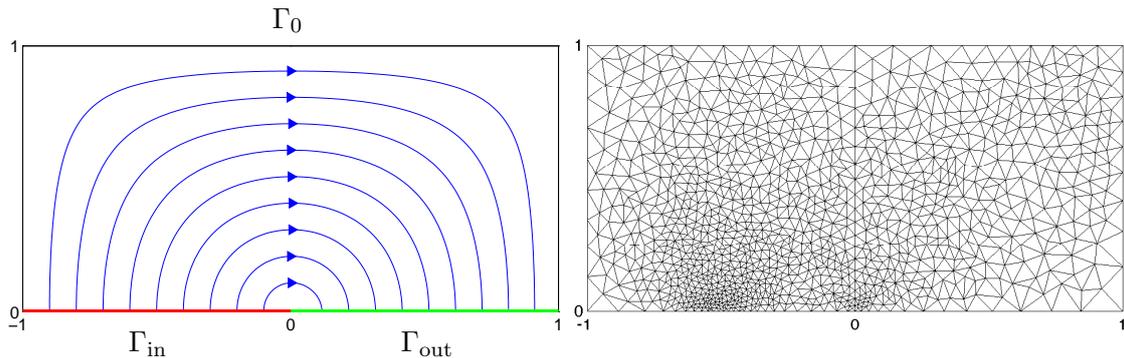


Figure 7: Geometry and streamlines for the advection-diffusion problem (23).

6.3 The Advection-Diffusion Equation

We consider the initial value problem

$$\partial_t u - \frac{1}{\text{Pe}} \Delta u + \mathbf{a} \cdot \nabla u = 0 \quad \text{in } \Omega = (-1, 1) \times (0, 1), \quad (23a)$$

$$u = 1 - \tanh(\text{Pe}) \quad \text{on } \Gamma_0, \quad (23b)$$

$$u = 1 + \tanh((2x + 1) \text{Pe}) \quad \text{on } \Gamma_{\text{in}}, \quad (23c)$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_{\text{out}}, \quad (23d)$$

$$u(x, 0) = u_0(x) \quad \text{in } \Omega \quad (23e)$$

for the advection-diffusion equation, which is a popular benchmark problem for discretizations of advection-dominated problems, see [20]. The convective field is given as

$$\mathbf{a}(x, y) = \begin{bmatrix} 2y(1 - x^2) \\ -2x(1 - y^2) \end{bmatrix}, \quad 0(x, y) \in \Omega,$$

and the boundary $\Gamma = \partial\Omega$ is divided into the inflow boundary $\Gamma_{\text{in}} := [-1, 0] \times \{0\}$, the outflow boundary $\Gamma_{\text{out}} := [0, 1] \times \{0\}$ and the remaining portion Γ_0 (cf. Figure 7, left). The Péclet number Pe is a nondimensional parameter describing the strength of advection relative to diffusion and therefore also how far the discrete operators are from symmetric.

We discretize the advection-diffusion operator for $\text{Pe} = 10$ in space using linear finite elements on a triangulation generated by the adaptive mesh generation facility in the COMSOL MULTIPHYSICS finite element software (version 3.3a). In the resulting mesh, shown on the right of Figure 7, one can observe refinements near the sharp transition in the inflow profile and near the origin, around which the advection field rotates. After

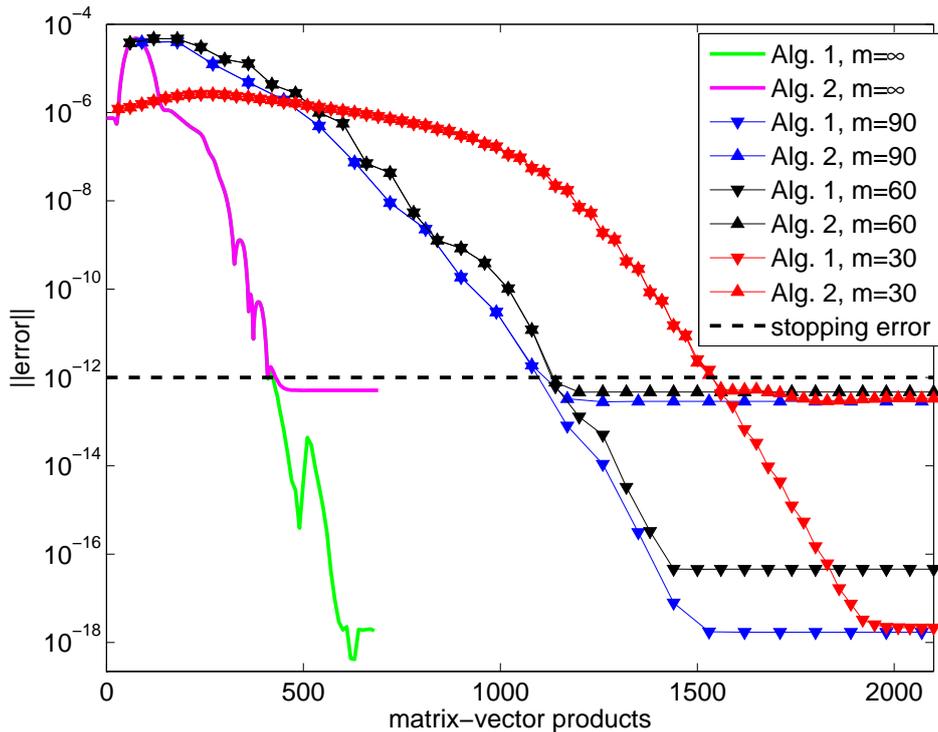


Figure 8: Relative errors of the full and restarted Arnoldi approximation for the advection-diffusion problem.

multiplying from the left by the square root of the (lumped) mass matrix[¶], the semi-discretized problem is a system of ODEs

$$\mathbf{u}'(t) = A\mathbf{u}(t) + \mathbf{g}, \quad \mathbf{u}(0) = \mathbf{u}_0,$$

where A is of size $n = 2157$ and the constant inhomogeneous term \mathbf{g} results from the inhomogeneous Dirichlet boundary condition. We then approximate the matrix exponential part of the solution

$$\mathbf{u}(t) = \exp(tA)(\mathbf{u}_0 + A^{-1}\mathbf{g}) - A^{-1}\mathbf{g}$$

at time $t = 6$, at which the flow has reached a steady state, starting from rest $\mathbf{u}_0 = \mathbf{0}$, using the unrestarted Arnoldi approximation as well as the restarted schemes of Algorithms 1 and 2.

The error curves for restart lengths $m = 30, 60$ and 90 are shown in Figure 8. We observe that, despite the rather small system size, even the unrestarted Arnoldi method requires roughly 500 steps to reach a final accuracy of around 10^{-12} . This slow convergence as well as the nonmonotonic, somewhat more erratic convergence curve indicate

[¶]This makes the Euclidean norm of the transformed vectors coincide with the L^2 -norm on the finite element space.

that this is a harder problem than the preceding two. We also observe that the methods of restart lengths $m = 60$ and $m = 90$ converge at nearly the same rate, and it is noticeable that Algorithm 2 reaches a final accuracy of only around 10^{-6} in all cases. The main reason for this loss of accuracy lies in the fact that the spectrum of A extends into the complex plane, and on the spectrum we have $\|r_2 - \exp\|_{\infty, \Lambda(A)} \approx 10^{-14}$ for the best uniform rational approximation r_2 on $(-\infty, 0]$. Moreover, additional loss of accuracy is to be expected from the solution of the 8 linear systems of equations solved in each restart cycle of Algorithm 2, the condition numbers of which was found to lie near 10^3 for all three restart lengths. Nonetheless, the restarted schemes converge also with the relatively small restart length of $m = 30$.

	Algorithm 1			Algorithm 2		
m	time [s]	mvp	acc.	time [s]	mvp	acc.
∞	8.6	410	9e-13	6.7	420	9e-13
90	90.1	1170	8e-14	12.0	1170	3e-13
60	121.0	1140	6e-13	11.8	1140	8e-13
30	685.7	1560	4e-13	15.3	1560	6e-13

Table 3: Execution times, number of matrix-vector products and final accuracy for the solution of the advection-diffusion problem for different restart lengths m .

Table 3 gives execution times for this example. We observe that the efficiency of Algorithm 2 over Algorithm 1 is most pronounced here. The fact that both restarted variants took longer than the full versions is attributed to the small size of the problem. For large problems, where the orthogonalization effort of full Arnoldi becomes more noticeable, we expect the timings to increasingly favor the restarted versions. Note that the advantage of requiring less storage is present also for these small problems.

7 Conclusions

Restarting Krylov subspace approximations of $f(A)\mathbf{b}$ is of interest because short recurrences for Krylov basis vectors do not translate to short recurrences for the quantity being approximated. We have introduced an efficient implementation for a restarted Krylov subspace approximation and compared it against the approach introduced in [6], both in terms of execution time and convergence properties. The new approach, Algorithm 2, is faster, as it solves a fixed number of linear systems of equations the size of the restart length m instead of, as is the case with Algorithm 1, evaluating a function of a matrix of increasing size km in the k -th cycle. Algorithm 2, while sometimes considerably faster, has the disadvantage that the restart length may need to be chosen

somewhat larger to insure convergence. Moreover, the solution of the linear equations in each cycle of Algorithm 2 can introduce some ill-conditioning which limits the final attainable accuracy. For the examples considered here involving the exponential propagation of semi-discretized partial differential operators subject to discretization errors, the requirements on final accuracy are usually sufficiently low that this is not a severe limitation. We have further introduced an error indicator which allows the termination of the iteration once sufficient or final accuracy has been reached. Moreover, we have pointed out the fundamental limitations of using a fixed rational approximation of f to evaluate $f(\widehat{H}_k)$ when the poles of the former approach the spectrum of A .

For the case of the exponential function, the new method was seen to be competitive with established methods for two symmetric problems, and a viable solution approach for a difficult non-Hermitian problem. In summary, Algorithm 2 is an attractive scheme, particularly when memory is limited so that the unrestarted Arnoldi method is not an option.

References

- [1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel. On the steepest descent method for matrix functions. In preparation, 2007.
- [2] A. J. Carpenter, A. Ruttan, and R. S. Varga. Extended numerical computations on the “1/9” conjecture in rational approximation theory. In P. R. Graves-Morris, E. B. Saff, and R. S. Varga, editors, *Rational Approximation and Interpolation, Proceedings, Tampa, Florida 1983*, number 1105 in Lecture Notes in Mathematics, pages 383–411, Heidelberg, 1984. Springer-Verlag.
- [3] W. J. Cody, G. Meinardus, and R. S. Varga. Chebyshev rational approximation to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems. *J. Approx. Theory*, 2:50–65, 1969.
- [4] V. L. Druskin and L. A. Knizhnerman. Two polynomial methods of calculating functions of symmetric matrices. *Comput. Math. Math. Phys.*, 29(6):112–121, 1989.
- [5] V. L. Druskin and L. A. Knizhnerman. Spectral approach to solving three-dimensional Maxwell’s diffusion equations in the time and frequency domains. *Radio Science*, 29(4):937–953, 1994.
- [6] M. Eiermann and O. G. Ernst. A restarted Krylov subspace method for the evaluation of matrix functions. *SIAM J. Numer. Anal.*, 44:2481–2504, 2006.
- [7] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Statist. Comput.*, 13(5):1236–1264, September 1992.
- [8] G. H. Golub and G. Meurant. Matrices, moments and quadrature. In *Numerical Analysis 1993 (Dundee 1993)*, number 303 in Pitman Research Notes in Mathematics, pages pp. 105–156. Longman Sci. Tech., Harlow, UK, 1994.

- [9] G. H. Golub, S. G. Nash, and C. F. V. Loan. A Hessenberg-Schur method for the matrix problem $AX + XB = C$. *IEEE Trans. Automat. Control*, AC-24:909–913, 1979.
- [10] N. J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM J. Matrix Anal. Appl.*, 26(4):1179–1193, 2005.
- [11] M. Hochbruck and M. E. Hochstenbach. Subspace extraction for matrix functions. Preprint, Dept. of Mathematics, Case Western Reserve University, 2005. Submitted.
- [12] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, October 1997.
- [13] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Statist. Comput.*, 19(5):1552–1574, 1998.
- [14] M. Ilić, I. W. Turner, and D. P. Simpson. A restarted Lanczos approximation to functions of a symmetric matrix. Preprint 8011, Queensland University of Technology, 2007.
- [15] G. Opitz. Steigungsmatrizen. *Z. Angew. Math. Mech.*, 44:T52–T54, 1964.
- [16] B. N. Parlett. A recurrence among the elements of functions of triangular matrices. *Linear Algebra Appl.*, 14:117–121, 1976.
- [17] B. Philippe and R. B. Sidje. Transient solutions of Markov processes by Krylov subspaces. Research Report, RR-1989, INRIA, 1995.
- [18] L. Reichel and L. N. Trefethen. Eigenvalues and pseudo-eigenvalues of Toeplitz matrices. *Linear Algebra Appl.*, 162–164:153–185, 1992.
- [19] Y. Saad. Analysis of some Krylov subspace approximations to the exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, February 1992.
- [20] R. M. Smith and A. G. Hutton. The numerical treatment of advection: A performance comparison of current methods. *Numer. Heat Transfer*, 5:439–461, 1982.
- [21] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer. Talbot quadratures and rational approximations. *BIT*, 46:653–670, 2006.
- [22] R. S. Varga. On higher order stable implicit methods for solving parabolic partial differential equations. *J. Math. Physics*, 40:220–231, 1961.
- [23] J. L. Walsh. *Interpolation and Approximation by Rational Functions in the Complex Domain*, volume XX of *Colloquium Publications*. American Mathematical Society, Providence, R.I., 5th edition, 1969.
- [24] S. H. Ward and G. W. Hohmann. Electromagnetic theory for geophysical applications. In M. N. Nabighian, editor, *Electromagnetic Methods in Applied Geophysics*, volume 1, chapter 4, pages 131–311. Soc. Expl. Geophys., 1987.

- [25] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag.*, AP-14:302–307, 1966.