

KRYLOV APPROXIMATION OF LINEAR ODES WITH POLYNOMIAL PARAMETERIZATION

ANTTI KOSKELA*, ELIAS JARLEBRING*, AND MICHIEL E. HOCHSTENBACH†

Abstract. We propose a new numerical method to solve linear ordinary differential equations of the type $\frac{\partial u}{\partial t}(t, \varepsilon) = A(\varepsilon)u(t, \varepsilon)$, where $A : \mathbb{C} \rightarrow \mathbb{C}^{n \times n}$ is a matrix polynomial with large and sparse matrix coefficients. The algorithm computes an explicit parameterization of approximations of $u(t, \varepsilon)$ such that approximations for many different values of ε and t can be obtained with a very small additional computational effort. The derivation of the algorithm is based on a reformulation of the parameterization as a linear parameter-free ordinary differential equation and on approximating the product of the matrix exponential and a vector with a Krylov method. The Krylov approximation is generated with Arnoldi's method and the structure of the coefficient matrix turns out to be independent of the truncation parameter so that it can also be interpreted as Arnoldi's method applied to an infinite dimensional matrix. We prove the superlinear convergence of the algorithm and provide a posteriori error estimates to be used as termination criteria. The behavior of the algorithm is illustrated with examples stemming from spatial discretizations of partial differential equations.

Key words. Krylov methods, Arnoldi's method, matrix functions, matrix exponential, exponential integrators, parameterized ordinary differential equations, Fréchet derivatives, model order reduction.

AMS subject classifications. 65F10, 65F60, 65L20, 65M22

DOI.

1. Introduction. Let $A_0, A_1, \dots, A_N \in \mathbb{C}^{n \times n}$ be given matrices and consider the parameterized linear time-independent ordinary differential equation

$$(1.1) \quad \frac{\partial u}{\partial t}(t, \varepsilon) = A(\varepsilon)u(t, \varepsilon), \quad u(0, \varepsilon) = u_0,$$

where A is the matrix polynomial $A(\varepsilon) := A_0 + \varepsilon A_1 + \dots + \varepsilon^N A_N$. Although most of our results are general, the usefulness of the approach is more explicit in a setting where N is not very large and the matrices A_0, \dots, A_N are large and sparse, e.g., stemming from a spatial finite-element semi-discretization of a parameterized partial-differential equation of evolutionary type.

We present a new iterative algorithm for the parameterized ODE (1.1), which gives an explicit parameterization of the solution. This parameterization is explicit in the sense that after executing the algorithm we can find a solution

*Department of Mathematics, Royal Institute of Technology (KTH), Stockholm, SeRC Swedish e-Science Research Center, akoskela@kth.se, eliasj@kth.se. The first author was supported by the Dahlquist research fellowship.

†Department of Mathematics and Computer Science, TU Eindhoven, PO Box 513, 5600 MB Eindhoven, The Netherlands www.win.tue.nl/~hochsten/. This author was supported by an NWO Vidi research grant.

to the ODE (1.1) for many different values of ε and $t > 0$ essentially without an additional computational effort. Such explicit parameterizations of solutions are useful in various settings, e.g., in parametric model order reduction and in the field of uncertainty quantification (with a single model parameter); see the discussion of model reduction below and the references in [2]. Numerical methods for parametric ODEs can be found in the literature in application specific contexts, see e.g. [5, 23] and the references therein.

The parameterization of the solution of (1.1) is represented as follows. Let the coefficients of the Taylor expansion of the solution with respect to the parameter ε around 0 be denoted by $c_0(t), c_1(t), \dots$, i.e.,

$$(1.2) \quad u(t, \varepsilon) = \exp(tA(\varepsilon)) u_0 = \sum_{\ell=0}^{\infty} \varepsilon^\ell c_\ell(t).$$

As $\exp(tA(\varepsilon))$ is an entire function of a matrix polynomial, the expansion (1.2) exists for all $\varepsilon \in \mathbb{C}$.

Consider the approximation stemming from the truncation of the Taylor series (1.2) and a corresponding approximation of the Taylor coefficients

$$(1.3) \quad \begin{aligned} u_k(t, \varepsilon) &:= \sum_{\ell=0}^{k-1} \varepsilon^\ell c_\ell(t) \\ &\approx \sum_{\ell=0}^{k-1} \varepsilon^\ell \tilde{c}_\ell(t) =: \tilde{u}_k(t, \varepsilon). \end{aligned}$$

Our approach gives an explicit parameterization with respect to t of the approximate coefficients $\tilde{c}_0(t), \dots, \tilde{c}_{k-1}(t)$ which, via (1.3), gives an approximate solution with an explicit parameterization with respect to ε and t .

The derivation of our approach is based on an explicit characterization of the time-dependent coefficients $c_0(t), \dots, c_{k-1}(t)$. We prove in Section 2 that they are solutions to the linear ordinary differential equation of size nm ,

$$(1.4) \quad \frac{d}{dt} \begin{bmatrix} c_0(t) \\ \vdots \\ c_{k-1}(t) \end{bmatrix} = L_k \begin{bmatrix} c_0(t) \\ \vdots \\ c_{k-1}(t) \end{bmatrix}, \quad \begin{bmatrix} c_0(0) \\ \vdots \\ c_{k-1}(0) \end{bmatrix} = \begin{bmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The matrix L_k in (1.4) is a finite-band block Toeplitz matrix and also a lower block triangular matrix.

Since (1.4) is a non-parametric linear ODE, we can in principle apply any numerical method to compute the solution which results in approximate coefficients $\tilde{c}_0(t), \dots, \tilde{c}_{k-1}(t)$. Exponential integrators combined with Krylov approximation of matrix functions have turned out to be an efficient class of methods for large-scale (semi)linear ODEs arising from PDEs [12, 13]. See also [14] for

a recent summary of exponential integrators. Krylov approximations of matrix functions have a feature which is suitable in our setting: after one run they give parameterized approximations with respect to the time-parameter.

Our derivation is based on approximating the solution of (1.4), i.e., a product of the matrix exponential and a vector, using a Krylov method. This is done by exploiting the structure of the coefficient matrix L_k . We show that when we apply Arnoldi's method to construct a Krylov subspace corresponding to (1.4), the block Toeplitz and lower block triangular property of L_k result in a particular structure in the basis matrix given by Arnoldi's method.

The structure of L_k is such that, in a certain sense, the algorithm can be extended to infinity. For example when $N = 1$, the basis matrix is extended with one block row as well as a block column in every iteration. This is analogous to the infinite Arnoldi method which has been developed for nonlinear eigenvalue problems [16] and linear inhomogeneous ODEs [17]. This feature implies that the algorithm does not require an a priori choice of the truncation parameter k .

We prove convergence of the algorithm (in Section 3) and also provide a termination criteria by giving a posteriori error estimates in Section 4.

The results can be interpreted and related to other approaches from a number of different perspectives. From one viewpoint, our result is related to recent work on computations and theory for Fréchet derivatives of matrix functions, e.g., [11, 20, 19]. As an illustration of a relation, consider the special case $N = 1$. The first-order expansion of the matrix exponential in (1.2) and [10, Chapter 3.1] gives

$$\begin{aligned} u(t, \varepsilon) &= \exp(t(A_0 + \varepsilon A_1)) u_0 \\ &= \exp(tA_0) u_0 + L_{\exp}(tA_0, \varepsilon tA_1) u_0 + o(|\varepsilon| |t| \|A_1\|), \end{aligned}$$

where L_{\exp} is the Fréchet derivative of the matrix exponential. Since the Fréchet derivative is linear in the second parameter, the first coefficient is explicitly given by $c_1(t) = L_{\exp}(tA_0, tA_1) u_0$. The higher order terms c_2, c_3, \dots have corresponding relationships with the higher order Fréchet derivatives. An analysis of higher order Fréchet derivatives is given in [21]. In contrast to the current Fréchet derivative approaches, which are essentially constructed for full matrices, our approach is an iterative Krylov method with a focus on large and sparse matrices and a specific starting vector u_0 .

The general approach to compute parameterized solutions to parameterized problems is very common in the field of model order reduction (MOR). See the recent survey papers [2]. In the terminology of MOR, our approach can be interpreted as a time-domain model order reduction technique for parameterized linear dynamical systems, without input or output. Parametric MOR is summarized in [2]; see also [18, 26]. There are time-domain Krylov methods, e.g., those described in PhD thesis [6]. To our knowledge, none of these methods can be interpreted as exponential integrators.

By evaluating (2.3) at $t = 0$, and noting that $u(0, \varepsilon) = u_0$ is independent of ε , it follows that $c_0(0) = u_0$ and $c_\ell(0) = 0$ for $\ell > 0$. The initial value problem (1.1) and the expansion of its solution (1.2) imply that

$$\begin{aligned}
c'_j(t) &= \frac{1}{j!} \frac{\partial^j}{\partial \varepsilon^j} \frac{\partial}{\partial t} u(t, \varepsilon) \Big|_{\varepsilon=0} = \frac{1}{j!} \frac{\partial^j}{\partial \varepsilon^j} \left(\sum_{i=0}^N \varepsilon^i A_i \right) u(t, \varepsilon) \Big|_{\varepsilon=0} \\
&= \frac{1}{j!} \frac{\partial^j}{\partial \varepsilon^j} \sum_{i=0}^N \varepsilon^i A_i \sum_{\ell=0}^{\infty} \varepsilon^\ell c_\ell(t) \Big|_{\varepsilon=0} \\
(2.4) \quad &= \sum_{i=0}^N \sum_{\ell=0}^{\infty} \left(\frac{1}{j!} \frac{\partial^j}{\partial \varepsilon^j} \varepsilon^{i+\ell} \Big|_{\varepsilon=0} \right) A_i c_\ell(t) \\
&= \sum_{i=0}^{\min(N, j)} A_i c_{j-i}(t).
\end{aligned}$$

From (2.3) and (2.4) it follows that the vector $\text{vec}(c_0(t), \dots, c_{k-1}(t))$ satisfies the linear ODE (1.4) with a solution given by (2.1). \square

Algorithm 1: Infinite Arnoldi algorithm for polynomial uncertain ODEs

Input : $u_0 \in \mathbb{C}^n$, $A_0, \dots, A_N \in \mathbb{C}^{n \times n}$

Output: Matrices $Q_p \in \mathbb{C}^{n(1+N(p-1)) \times p}$ and $H_p \in \mathbb{C}^{p \times p}$ representing approximations of the coefficients c_0, \dots, c_{p-1} via (2.7)

- 1 Let $\beta = \|u_0\|$, $Q_1 = u_0/\beta$, $\underline{H}_0 = []$
 - for** $\ell = 1, 2, \dots, p$ **do**
 - 2 Let $x = Q_\ell(:, \ell) \in \mathbb{C}^{n+(\ell-1)nN}$
 - 3 Compute $y := \text{vec}(y_1, \dots, y_{1+\ell N}) \in \mathbb{C}^{n+\ell nN}$ with (2.6)
 - 4 Let $\underline{Q}_\ell := \begin{bmatrix} Q_\ell \\ 0 \end{bmatrix} \in \mathbb{C}^{(n+\ell nN) \times \ell}$
 - 5 Compute $h = \underline{Q}_\ell^* y$
 - 6 Compute $y_\perp := y - \underline{Q}_\ell h$
 - 7 Repeat Steps 5–6 if necessary
 - 8 Compute $\alpha = \|y_\perp\|$
 - 9 Let $\underline{H}_\ell = \begin{bmatrix} \underline{H}_{\ell-1} & h \\ 0 & \alpha \end{bmatrix}$
 - 10 Let $Q_{\ell+1} := [Q_\ell, y_\perp/\alpha] \in \mathbb{C}^{(n+\ell nN) \times (\ell+1)}$
 - end**
 - 11 Set $H_p \in \mathbb{C}^{p \times p}$ to be the leading submatrix of $\underline{H}_p \in \mathbb{C}^{(p+1) \times p}$
-

2.2. Algorithm. Theorem 1 can be used to compute the coefficients $c_\ell(t)$ if we can compute the matrix exponential of L_k times the vector \tilde{u}_0 . We use a Krylov approximation which exploits the structure of the problem. See, e.g., [12, 13] for literature on Krylov approximations of matrix functions.

The Krylov approximation of $v(t) = \exp(tB)v_0$ consists of p steps of the Arnoldi iteration for the matrix B initiated with the vector v_0 . This results in the Arnoldi relation

$$BQ_p = Q_{p+1}\underline{H}_p,$$

where $\underline{H}_p \in \mathbb{C}^{(p+1) \times p}$ is a Hessenberg matrix and Q_p an orthogonal matrix spanning the Krylov subspace $\mathcal{K}_p(B, v_0) = \text{span}(v_0, Bv_0, \dots, B^{p-1}v_0)$. The Krylov approximation of $\exp(tB)v_0$ is given by

$$v(t) = \exp(tB)v_0 \approx Q_p \exp(tH_p)e_1 \|v_0\|,$$

where $H_p \in \mathbb{C}^{p \times p}$ is the leading submatrix of \underline{H}_p , and e_1 is the first unit basis vector.

The only way B appears in the Arnoldi algorithm is in the form of matrix vector products. Moreover, the Arnoldi algorithm is initiated with the vector v_0 . Suppose we apply this Arnoldi approximation to (2.1). In the first step we need to compute the matrix vector product

$$(2.5) \quad L_k \text{vec}(u_0, 0, \dots, 0) = \text{vec}(A_0 u_0, \dots, A_N u_0, 0, \dots, 0),$$

which is more generally given as follows.

LEMMA 2 (Matrix vector product). *Suppose $x = \text{vec}(x_1, \dots, x_j, 0, \dots, 0) = \text{vec}(X) \in \mathbb{C}^{nk}$, where $x_1, \dots, x_j \in \mathbb{C}^n$ and $k > j + N$. Then,*

$$L_k x = \text{vec}(y_1, \dots, y_{j+N}, 0, \dots, 0),$$

where

$$(2.6) \quad y_\ell = \sum_{i=\max(0, \ell-j)}^{\min(N, \ell-1)} A_i x_{\ell-i}, \quad \ell = 1, \dots, j + N.$$

Proof. Suppose $S \in \mathbb{R}^{k \times k}$ is the shift matrix $S := \sum_{\ell=1}^{k-1} e_{\ell+1} e_\ell^T$ which satisfies $(S^i)^T = \sum_{\ell=1}^{k-i} e_\ell e_{\ell+i}^T$. We have

$$\begin{aligned} L_k x &= \sum_{i=0}^N (S^i \otimes A_i) \text{vec}(X) = \sum_{i=0}^N \text{vec}(A_i X (S^i)^T) \\ &= \sum_{i=0}^N \sum_{\ell=1}^{k-i} \text{vec}(A_i X e_\ell e_{\ell+i}^T). \end{aligned}$$

Note that $X e_\ell = x_\ell$ if $\ell \leq j$ and $X e_\ell = 0$ if $\ell > j$. Hence, by using the assumption $k > j + N$, and by reordering the terms in the sum we find the explicit formula $L_k x = \sum_{i=0}^N \sum_{\ell=1}^j \text{vec}(A_i x_\ell e_{\ell+i}^T) = \sum_{\ell=1}^{j+N} \sum_{i=\max(0, \ell-j)}^{\min(N, \ell-1)} \text{vec}(A_i x_{\ell-i} e_\ell^T)$. \square Since the Arnoldi method consists of applying matrix vector products and

orthogonalizing the new vector against previous vectors, we see from (2.5) that the second vector in the Krylov subspace will consist of $N + 1$ nonzero blocks. Repeated application of Lemma 2 results in a structure where the j th column in the basis matrix consists of $(j - 1)N + 1$ nonzero blocks, under the condition that k is sufficiently large. It is natural to store only the nonzero blocks of the basis matrix and use Lemma 2 for the matrix vector products. Then, the Arnoldi method for (2.1) reduces to Algorithm 1. Notice that the orthogonalization cost of Algorithm 1 may become significant for large p 's, as the complexity of the orthogonalization of p steps of Algorithm 1 is $\mathcal{O}(p^3 N n)$ as opposed to the $\mathcal{O}(p^2 n)$ -cost of p steps of the Arnoldi iteration applied to an $n \times n$ matrix.

We note that our construction is equivalent to the Arnoldi method and the output of the algorithm is a basis matrix and a Hessenberg matrix which together form the approximation of the coefficients c_0, \dots, c_{k-1}

$$(2.7) \quad \text{vec}(c_0(t), \dots, c_{k-1}(t)) \approx \text{vec}(\tilde{c}_0(t), \dots, \tilde{c}_{k-1}(t)) := Q_p \exp(tH_p) e_1 \|u_0\|,$$

where by construction $k = N(p - 1) + 1$. The approximation of the solution is denoted as in (1.3), i.e.,

$$\tilde{u}_{k,p}(t, \varepsilon) := \sum_{\ell=0}^{k-1} \varepsilon^\ell \tilde{c}_\ell(t),$$

where we have added an index p to stress the dependence on the iteration. A feature of this construction is that the algorithm does not explicitly depend on k , such that it in a sense can be extended to infinity, i.e., it is equivalent to Arnoldi's method on an infinite dimensional operator. This can be summarized as follows.

THEOREM 3. *The following procedures generate identical results.*

- (i) p iterations of Algorithm 1 started with u_0 and A_0, \dots, A_N ;
- (ii) p iterations of Arnoldi's method applied to L_k with starting vector $e_1 \otimes u_0 \in \mathbb{C}^{nk}$ for any $k \geq Np$;
- (iii) p iterations of Arnoldi's method applied to the infinite matrix L_∞ with the infinite starting vector $e_1 \otimes u_0 \in \mathbb{C}^\infty$.

3. A priori convergence theory. To show the validity of our approach we now bound the total error after p iterations, which is separated into two terms as

$$(3.1) \quad \begin{aligned} \text{err}_p(t, \varepsilon) &:= \|u(t, \varepsilon) - \tilde{u}_{k,p}(t, \varepsilon)\| \\ &\leq \text{err}_{K,k,p}(t, \varepsilon) + \text{err}_{T,k}(t, \varepsilon), \end{aligned}$$

where

$$(3.2) \quad \text{err}_{K,k,p}(t, \varepsilon) := \|\tilde{u}_{k,p}(t, \varepsilon) - u_k(t, \varepsilon)\|$$

$$(3.3) \quad \text{err}_{T,k}(t, \varepsilon) := \|u(t, \varepsilon) - u_k(t, \varepsilon)\|.$$

Notice that when running Algorithm 1, $k = N(p - 1) + 1$ in equation (3.1).

A bound of $\text{err}_{\text{K},k,p}$, which corresponds to the Krylov approximation of the expansion coefficients c_0, \dots, c_{k-1} , is given in Section 3.1 and a bound on $\text{err}_{\text{T},k}$, which corresponds to the truncation of the series, is given in Section 3.2. After combining the main results of Section 3.1 and Section 3.2, in particular formulas (3.8) and (3.9), we reach the conclusion that

$$(3.4) \quad \text{err}_p(t, \varepsilon) \leq C_1(t, \varepsilon) \sum_{\ell=0}^{N-1} \frac{C_2(t, \varepsilon)^{p+\ell-1} e^{C_2(t, \varepsilon)}}{(p + \ell - 2)!} \|u_0\| + 2 \sqrt{\frac{1 - |\varepsilon|^{2N(p-1)}}{1 - |\varepsilon|^2}} \frac{(t\alpha)^p e^{t\gamma}}{p!} \|u_0\|,$$

where α and γ are given in (3.6), and $C_1(t, \varepsilon)$ and $C_2(t, \varepsilon)$ are given in (3.10). Due to the factorial in the denominator of (3.4), for fixed ε and $t > 0$, the bound suggests that the total error eventually decreases superlinearly with respect to the iteration count p .

3.1. A bound on the Krylov error. We first study the error generated by the Arnoldi method to approximate the coefficients $c_0(t), \dots, c_{k-1}(t)$. We define

$$(3.5) \quad E_{k,p}(t) = [c_0(t), \dots, c_{k-1}(t)] - [\tilde{c}_0(t), \dots, \tilde{c}_{k-1}(t)].$$

where $\tilde{c}_0, \dots, \tilde{c}_{k-1}$ are the approximations given by the Arnoldi method after p steps, i.e., by the vector

$$\hat{c}_k(t) := \begin{bmatrix} \tilde{c}_0(t) \\ \vdots \\ \tilde{c}_{k-1}(t) \end{bmatrix} = Q_p \exp(tH_p) e_1 \|u_0\|.$$

Using existing bounds for the Arnoldi approximation of the matrix exponential [9], we get a bound for the error of this approximation, as follows.

LEMMA 4 (Krylov coefficient error bound). *Let $t > 0$, $A_0, \dots, A_N \in \mathbb{C}^{n \times n}$, and $u_0 \in \mathbb{C}^n$. Let $\tilde{c}_0(t), \dots, \tilde{c}_{k-1}(t)$ be the result of Algorithm 1, and let $E_{k,p}(t)$ be defined by (3.5). Then, the total error in the coefficients $\|\text{vec}(E_k(t))\|$ satisfies*

$$\|\text{vec}(E_{k,p}(t))\| = \|\exp(tL_k) \tilde{u}_0 - \hat{c}_k(t)\| \leq 2 \frac{(t\alpha)^p}{p!} e^{t \max\{1, \beta\}} \|u_0\|$$

where

$$(3.6) \quad \alpha = \sum_{\ell=0}^N \|A_\ell\| \quad \text{and} \quad \beta = \mu(A_0) + \sum_{\ell=1}^N \|A_\ell\|,$$

and $\mu(B)$ denotes the logarithmic norm defined in (1.5)

Proof. The result follows directly from [9, Thm. 2.1], and Lemma 8 and Corollary 10 in Appendix A. \square

The coefficient error bound in Lemma 4, implies the following bound on the error $\text{err}_{\text{K},k,p}$, via the relation

$$(3.7) \quad \text{err}_{\text{K},k,p}(t, \varepsilon) = \|E_{k,p}(t)[1, \varepsilon, \dots, \varepsilon^{k-1}]^T\|.$$

THEOREM 5 (Krylov error bound). *Let $\text{err}_{\text{K},k,p}$ be defined in (3.2) corresponding to applying p steps of Algorithm 1 to $A_0, \dots, A_N \in \mathbb{C}^{n \times n}$ and $u_0 \in \mathbb{C}^n$. Then,*

$$(3.8) \quad \text{err}_{\text{K},k,p}(t, \varepsilon) \leq 2 \sqrt{\frac{1 - |\varepsilon|^{2k}}{1 - |\varepsilon|^2}} \frac{(t\alpha)^p e^{t \max\{1, \beta\}}}{p!} \|u_0\|$$

where α and β are given in (3.6).

Proof. By (3.7) and the Cauchy–Schwarz inequality we have that

$$\begin{aligned} \text{err}_{\text{K},k,p}(t, \varepsilon) &= \left\| \sum_{\ell=0}^{k-1} \varepsilon^\ell (c_\ell(t) - \tilde{c}_\ell(t)) \right\| \\ &\leq \sqrt{\sum_{\ell=0}^{k-1} |\varepsilon^\ell|^2} \sqrt{\sum_{\ell=0}^{k-1} \|c_\ell(t) - \tilde{c}_\ell(t)\|^2} \\ &= \sqrt{\frac{1 - |\varepsilon|^{2k}}{1 - |\varepsilon|^2}} \|\text{vec}(E_{k,p}(t))\|. \end{aligned}$$

The claim follows now from Lemma 4. \square

3.2. A bound on the truncation error. The previous subsection gives us an estimate for the error in the coefficient vectors $c_\ell(t)$. To characterize the total error of the approximation $\tilde{u}_{k,p}(t, \varepsilon)$, we now analyze the second term in the error splitting (3.1), i.e., the remainder

$$\text{err}_{\text{T},k}(t, \varepsilon) := \|u(t, \varepsilon) - \sum_{\ell=0}^{k-1} \varepsilon^\ell c_\ell(t)\|.$$

Lemma 13 of Appendix B gives a bound for the norms of $c_\ell(t)$ and can be used to derive the following theorem which bounds $\text{err}_{\text{T},k}(t, \varepsilon)$.

THEOREM 6 (Remainder bound). *Let $u(t, \varepsilon)$ be the solution of the initial value problem (1.1), and consider its ε -expansion (1.2). Then, the error $\text{err}_{\text{T},k}(t, \varepsilon)$ is bounded as*

$$(3.9) \quad \text{err}_{\text{T},k}(t, \varepsilon) \leq C_1(t, \varepsilon) \sum_{\ell=0}^{N-1} \frac{C_2(t, \varepsilon)^{\lfloor \frac{k}{N} \rfloor + \ell}}{(\lfloor \frac{k}{N} \rfloor + \ell - 1)!},$$

where

$$(3.10) \quad \begin{aligned} C_1(t, \varepsilon) &= |\varepsilon|^{\text{sign}(|\varepsilon|-1)} e^{t(\mu(A_0)+eNa)+C_2(t,\varepsilon)-1} \|u_0\|, \\ C_2(t, \varepsilon) &= |\varepsilon|^N eNta. \end{aligned}$$

Proof. From Lemma 13 it follows that

$$\text{err}_{T,k}(t, \varepsilon) = \left\| \sum_{\ell=k}^{\infty} \varepsilon^\ell c_\ell(t) \right\| \leq \sum_{\ell=k}^{\infty} |\varepsilon|^\ell \|c_\ell(t)\| \leq \tilde{C}_1(t, \varepsilon) \sum_{\ell=k}^{\infty} |\varepsilon|^\ell \frac{(eNta)^{\lceil \frac{\ell}{N} \rceil}}{(\lceil \frac{\ell}{N} \rceil - 1)!},$$

where $\tilde{C}_1(t, \varepsilon) = e^{t(\mu(A_0)+eNa)-1} \|u_0\|$.

Setting $\tilde{k} = k - N \lfloor \frac{k}{N} \rfloor$ and using the bound $c^\ell = (c^N)^{\frac{\ell}{N}} \leq (c^N)^{\lceil \frac{\ell}{N} \rceil} c^{\text{sign}(c-1)}$ for $c > 0$, we get

$$\begin{aligned} \sum_{\ell=k}^{\infty} |\varepsilon|^\ell \frac{(eNta)^{\lceil \frac{\ell}{N} \rceil}}{(\lceil \frac{\ell}{N} \rceil - 1)!} &\leq |\varepsilon|^{\text{sign}(|\varepsilon|-1)} \sum_{\ell=\tilde{k}}^{\infty} \frac{(|\varepsilon|^N eNta)^{\lceil \frac{\ell}{N} \rceil}}{(\lceil \frac{\ell}{N} \rceil - 1)!} \\ &= |\varepsilon|^{\text{sign}(|\varepsilon|-1)} \sum_{j=0}^{N-1} \sum_{\ell=\lfloor \frac{k}{N} \rfloor + j}^{\infty} \frac{(|\varepsilon|^N eNta)^\ell}{(\ell - 1)!}. \end{aligned}$$

Using the inequality [22, Lemma 4.2]

$$(3.11) \quad \sum_{\ell=k}^{\infty} \frac{x^\ell}{\ell!} \leq \frac{x^k e^x}{k!} \quad \text{for } x > 0,$$

the claim follows. \square

We also give a bound for the special case $N = 1$ since it is in this case considerably lower than the one given in Theorem 6.

THEOREM 7 (Remainder bound $N = 1$). *Let $N = 1$. Then the remainder $\text{err}_{T,k}$ is bounded as*

$$(3.12) \quad \text{err}_{T,k}(t, \varepsilon) \leq \frac{e^{t(\mu(A_0)+|\varepsilon|\|A_1\|)} (|\varepsilon| \|tA_1\|)^k}{k!} \|u_0\|.$$

Proof. From Lemma 11 and (B.3) we see that $c_\ell(t)$ consists now of one integral term which can be bounded by Lemma 12 giving

$$\|c_\ell(t)\| \leq \frac{\|tA_1\|^k}{k!} e^{t\mu(A_0)} \|u_0\|.$$

Therefore

$$\text{err}_{T,k}(t, \varepsilon) \leq \sum_{\ell=k}^{\infty} |\varepsilon|^\ell \|c_\ell(t)\| \leq e^{t\mu(A_0)} \sum_{\ell=k}^{\infty} \frac{(|\varepsilon| \|tA_1\|)^\ell}{\ell!} \|u_0\|,$$

and the claim follows from the inequality (3.11). \square

3.3. Expansion around a fixed ε and choice of ε_{\max} . In case one is interested in solutions of (1.1) for values of ε around some fixed value $\tilde{\varepsilon}$, it may be beneficial to perform the expansion (1.2) around $\tilde{\varepsilon}$ instead of 0. As an example, consider the case $N = 2$, defined by the coefficient matrices A_0, A_1 and A_2 . Then, the ODE can be reformulated as follows

$$\begin{aligned} u'(t) &= (A_0 + \varepsilon A_1 + \varepsilon^2 A_2) u(t) \\ &= (A_0 + (\varepsilon - \tilde{\varepsilon})A_1 + \tilde{\varepsilon}A_1 + (\varepsilon - \tilde{\varepsilon})^2 A_2 + 2\varepsilon\tilde{\varepsilon}A_2 - \tilde{\varepsilon}^2 A_2) u(t) \\ &= ((A_0 + \tilde{\varepsilon}A_1 + \tilde{\varepsilon}^2 A_2) + (\varepsilon - \tilde{\varepsilon})(A_1 + 2\tilde{\varepsilon}A_2) + (\varepsilon - \tilde{\varepsilon})^2 A_2) u(t). \end{aligned}$$

By running Algorithm 1 for the matrices $\tilde{A}_0 = A_0 + \tilde{\varepsilon}A_1 + \tilde{\varepsilon}^2 A_2$, $\tilde{A}_1 = (A_1 + 2\tilde{\varepsilon}A_2)$ and $\tilde{A}_2 = A_2$ one may obtain considerably faster convergence of the ε -expansion for values of ε around $\tilde{\varepsilon}$. This is also reflected by the error bounds (3.4), as ε is then replaced by $\varepsilon - \tilde{\varepsilon}$.

If there is a constraint for the value of p , due to, e.g., memory constraints, the a priori bound (3.4) can be used to determine the largest value of $|\varepsilon|$ that guarantees a given error tolerance tol for $\text{err}_p(t, \varepsilon)$. One readily sees that the right-hand side of the bound (3.4) is a monotonically increasing function of $|\varepsilon|$. Fixing p and tol , and using, e.g., the bisection method, the maximum value of $|\varepsilon|$ which satisfies the p constraint can be directly computed numerically.

4. An a posteriori error estimate for the Krylov approximation.

Although the previous section provides a proof of convergence, the final bound is not necessarily very useful to estimate the error. We therefore also propose the following a posteriori error estimates, which appear to work well in the simulations of Section 5.

Let $Q_p \exp(H_p)e_1$ be the approximation of $e^{A}b$, $\|b\| = 1$, by p steps of the Arnoldi method. Then, due to the fact that our algorithm is equivalent to the standard Arnoldi method, the following expansion holds [22]

$$(4.1) \quad e^{A}b - Q_p \exp(H_p)e_1 = h_{p+1,p} \sum_{\ell=1}^{\infty} e_p^T \varphi_{\ell}(H_p)e_1 A^{\ell-1} q_{p+1},$$

where $\varphi_{\ell}(z) = \sum_{j=0}^{\infty} \frac{z^j}{(j+\ell)!}$ and q_{p+1} is the $(p+1)$ st basis vector given by the Arnoldi iteration.

We estimate the error of the Arnoldi approximation of $\exp(tL_k)\tilde{u}_0$, i.e., the approximation of the vector $\text{vec}(E_{k,p}(t))$, by using the norm of the first two terms in (4.1). This gives us the estimate

$$(4.2) \quad \begin{aligned} \text{vec}(E_{k,p}(t)) &\approx h_{p+1,p} (e_p^T \varphi_1(tH_p)e_1 q_{p+1} + e_p^T \varphi_2(tH_p)e_1 (tL_k)q_{p+1}) \|u_0\| \\ &:= \widetilde{\text{err}}_{k,p}(t). \end{aligned}$$

Then, for the Krylov error $\text{err}_{K,k,p}(t, \varepsilon)$ in the total error (3.1), we obtain an

estimate $\widetilde{\text{err}}_{\text{K},k,p}(t, \varepsilon)$ directly using (3.7):

$$\begin{aligned}
(4.3) \quad \text{err}_{\text{K},k,p}(t, \varepsilon) &= \|E_{k,p}(t)[1, \varepsilon, \dots, \varepsilon^{k-1}]^T\| \\
&= \|(I_n \otimes [1, \varepsilon, \dots, \varepsilon^{k-1}]) \text{vec}(E_{k,p}(t))\| \\
&\approx \|(I_n \otimes [1, \varepsilon, \dots, \varepsilon^{k-1}]) \widetilde{\text{err}}_{k,p}(t)\| =: \widetilde{\text{err}}_{\text{K},k,p}(t, \varepsilon),
\end{aligned}$$

where $k = 1 + (N - 1)p$. Notice that the scalars $e_p^T \varphi_1(tH_p)e_1$ and $e_p^T \varphi_2(tH_p)e_1$ in (4.2) can be obtained with a small extra cost using the fact that [24, Thm. 1]

$$[I_p \ 0] \exp \left(\begin{bmatrix} tH_p & e_1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right) = [\exp(tH_p) \ \varphi_1(tH_p)e_1 \ \varphi_2(tH_p)e_1].$$

Since the a priori bound given by Theorem 6 is rather pessimistic in practice, in numerical experiments we only use the Krylov error estimate (4.3) as a total error estimate when $N \geq 2$. For $N = 1$, we use also the truncation bound given in Theorem 7, i.e., the total estimate is then

$$\begin{aligned}
(4.4) \quad \|u(t, \varepsilon) - \tilde{u}_{k,p}(t, \varepsilon)\| &\leq \text{err}_{\text{T},k}(t, \varepsilon) + \text{err}_{\text{K},k,p}(t, \varepsilon) \\
&\approx \widetilde{\text{err}}_{\text{K},k,p}(t, \varepsilon) + \frac{e^{t(\mu(A_0) + |\varepsilon| \|A_1\|)} (|\varepsilon| \|tA_1\|)^p}{p!}.
\end{aligned}$$

5. Numerical examples. The behavior of the algorithm is now illustrated for two test problems: one stemming from spatial discretization of an advection-diffusion equation and the other one appearing in the literature [18] corresponding to the discretization of a damped wave equation. In all of the numerical examples the reference solution is computed using the `expmv` command described in [1].

5.1. Scaling of L_k . It turns out that the performance of the algorithm can be improved by performing a transformation which scales the coefficient matrices. This scaling can be carried out as follows. Let $A_0, A_1, \dots, A_N \in \mathbb{C}^{n \times n}$ and L_k be the corresponding block-Toeplitz matrix of the form (2.2). Let $\gamma > 0$ and define $\Sigma_k := \text{diag}(1, \gamma, \dots, \gamma^{k-1}) \otimes I_n$. Then it clearly holds

$$\begin{aligned}
\hat{c}(t) &= \exp(tL_k) \tilde{u}_0 = \Sigma_k \exp(t\Sigma_k^{-1} L_k \Sigma_k) \tilde{u}_0 \\
&= \Sigma_k \exp(t\hat{L}_k) \tilde{u}_0,
\end{aligned}$$

where \hat{L}_k is the matrix (2.2) corresponding to $A_0, \gamma^{-1}A_1, \dots, \gamma^{-N}A_N$.

Thus, we see that using this scaling strategy corresponds to the changes

$$(5.1) \quad \varepsilon \rightarrow \gamma\varepsilon \quad \text{and} \quad A_\ell \rightarrow \gamma^{-\ell} A_\ell$$

when performing the Arnoldi approximation of the product $\exp(t\hat{L}_k) \tilde{u}_0$. This is also evident from the original ODE (1.1).

The performance of the algorithm appears to improve when we scale the norms of coefficients A_ℓ , $1 \leq \ell \leq N$, such that they are of the order 1 or less. To balance the norms, we used the heuristic choice

$$(5.2) \quad \gamma = \max_{1 \leq \ell \leq N} \|A_\ell\|^{1/\ell}.$$

This was found to work well in all of our numerical experiments, giving both good convergence and a posteriori error estimates.

We note that scaling has also been exploited for polynomial eigenvalue problems, e.g., in [8]. Our scaling (5.2) can be interpreted as a slight variation of the scaling proposed in [3, Thm. 6.1]. Another related scaling, one for the matrix exponential of an augmented matrix, can be found in [1, p. 492].

5.2. Advection-diffusion operator. Consider the 1-d advection-diffusion equation

$$(5.3) \quad \frac{\partial}{\partial t} y(t, x) = a \frac{\partial^2}{\partial x^2} y(t, x) + \varepsilon \frac{\partial}{\partial x} y(t, x), \quad y(0, x) = y_0(x)$$

with Dirichlet boundary conditions on the interval $[0, 1]$ and $y_0(x) = 16((1-x)x)^2$. The spatial discretization using central finite differences gives the ordinary differential equation $u' = (A_0 + \varepsilon A_1)u$, $u(0) = u_0 \in \mathbb{R}^n$, where the matrices A_0 and A_1 are of the form

$$A_0 = \frac{a}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}, \quad A_1 = \frac{1}{2\Delta x} \begin{bmatrix} & -1 & & & \\ 1 & & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & & -1 \\ & & & 1 & \end{bmatrix},$$

where $\Delta x = (n+1)^{-1}$ and u_0 is the discretization of $y_0(x)$. We set $n = 200$ and $a = 3 \cdot 10^{-4}$, and approximate at $t = 0.5$. Then, $\|tA_0\| \approx 95$. We compute the approximations $\tilde{u}_{k,p}(t, \varepsilon)$ for $\varepsilon = 10^{-3}, 1.5 \cdot 10^{-2}$, and $3 \cdot 10^{-2}$. Then, respectively, $\|t\varepsilon A_1\| \approx 0.4, 6.0$ and 12.0 . Figure 5.1 shows the 2-norm errors of these approximations and the corresponding a posteriori error estimates using (4.4). We observe superlinear convergence for the error and the estimate.

To illustrate the generality of our approach we now consider the case $N = 2$, namely a modification of (5.3)

$$(5.4) \quad \frac{\partial}{\partial t} y(t, x) = a \frac{\partial^2}{\partial x^2} y(t, x) + \varepsilon \frac{\partial}{\partial x} y(t, x) + \varepsilon^2 b y(t, 1-x), \quad y(0, x) = y_0(x);$$

the extra term can be interpreted as a non-localized feedback. We set the parameter $a = 3 \cdot 10^{-4}$ and $b = 2 \cdot 10^2$. The spatial discretization with finite differences gives the ODE $u' = (A_0 + \varepsilon A_1 + \varepsilon^2 A_2)u$, $u(0) = u_0 \in \mathbb{R}^n$, where

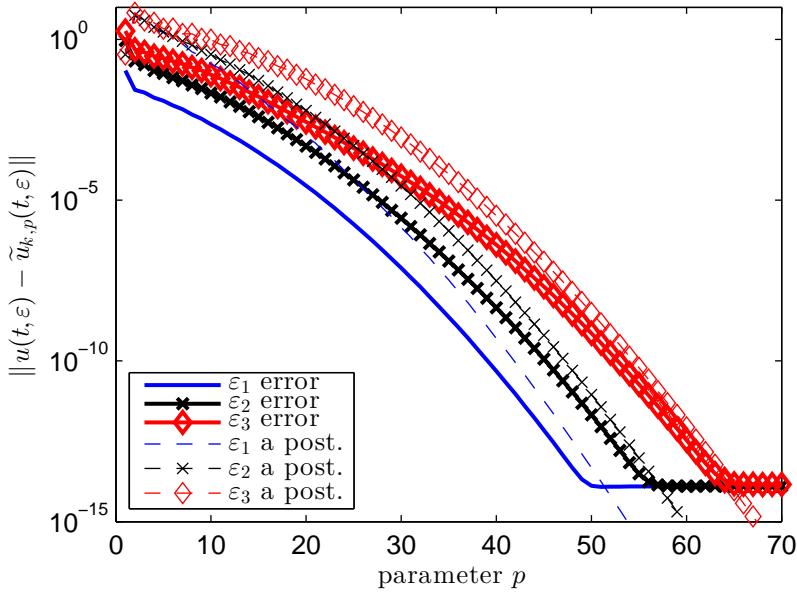


FIGURE 5.1. 2-norm errors of approximations $\tilde{u}_{k,p}(t, \varepsilon)$ and the estimate (4.4) when ε has the values $\varepsilon_1 = 1 \cdot 10^{-3}$, $\varepsilon_2 = 1.5 \cdot 10^{-2}$ and $\varepsilon_3 = 3 \cdot 10^{-2}$.

u_0 and the matrices A_0 and A_1 are as above, and

$$A_2 = b \cdot \begin{bmatrix} & & 1 \\ & \dots & \\ 1 & & \end{bmatrix}.$$

We compute the approximations $\tilde{u}_{k,p}(t, \varepsilon)$ for $\varepsilon = 10^{-3}$, $1.5 \cdot 10^{-2}$ and $3 \cdot 10^{-2}$, for which respectively, $\|t\varepsilon^2 A_2\| \approx 5.0 \cdot 10^{-4}$, 0.11, and 0.45. Figure 5.2 shows the 2-norm errors of these approximations and the corresponding a posteriori error estimates using (4.4).

In Figure 5.3 we illustrate the dependence of the convergence on the scaling. Clearly the choice (5.2) results in the fastest convergence for this example. Simulations with γ larger than what is suggested by (5.2) did not result in substantial improvement of the convergence.

5.3. Wave equation. Consider next the damped wave equation inside the 3-d unit box given in [18, Section 5.2]. The governing $2n$ -dimensional first-order differential equation is given by

$$(5.5) \quad \frac{d}{dt} \begin{bmatrix} u(t) \\ u'(t) \end{bmatrix} \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C(\gamma) \end{bmatrix} \begin{bmatrix} u(t) \\ u'(t) \end{bmatrix}, \quad \begin{bmatrix} u(0) \\ u'(0) \end{bmatrix} = \begin{bmatrix} u_0 \\ u'_0 \end{bmatrix} \in \mathbb{R}^{2n},$$

where $C(\gamma_1, \gamma_2) = \gamma_1 C_1 + \gamma_2 C_2$. The model is obtained by finite differences with 15 discretization points in each dimension, i.e., $n = 15^3$. The matrix K denotes the discretized Laplacian, $C(\gamma_1, \gamma_2)$ the damping matrix stemming

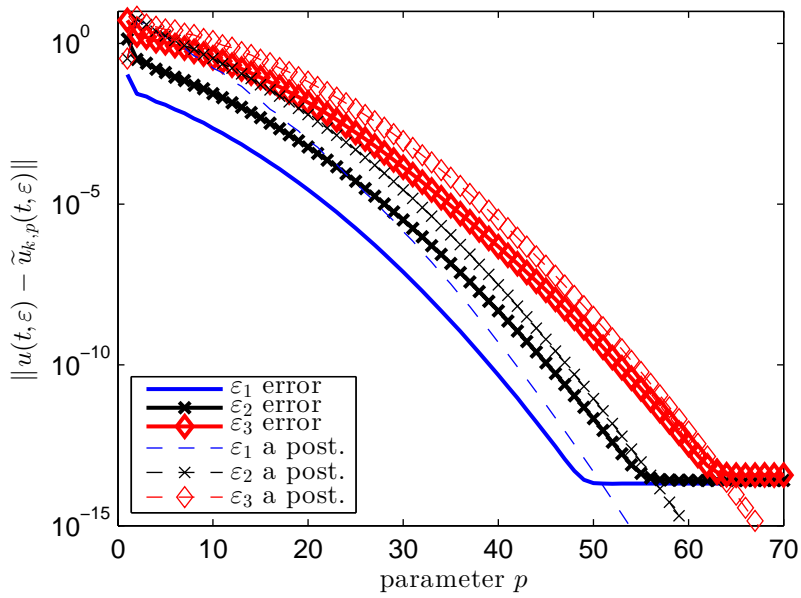


FIGURE 5.2. 2-norm errors of approximations $\tilde{u}_{k,p}(t, \varepsilon)$ for the equation (5.4) and the error estimate (4.3) when ε has the values $\varepsilon_1 = 1 \cdot 10^{-3}$, $\varepsilon_2 = 1.5 \cdot 10^{-2}$ and $\varepsilon_3 = 3 \cdot 10^{-2}$.

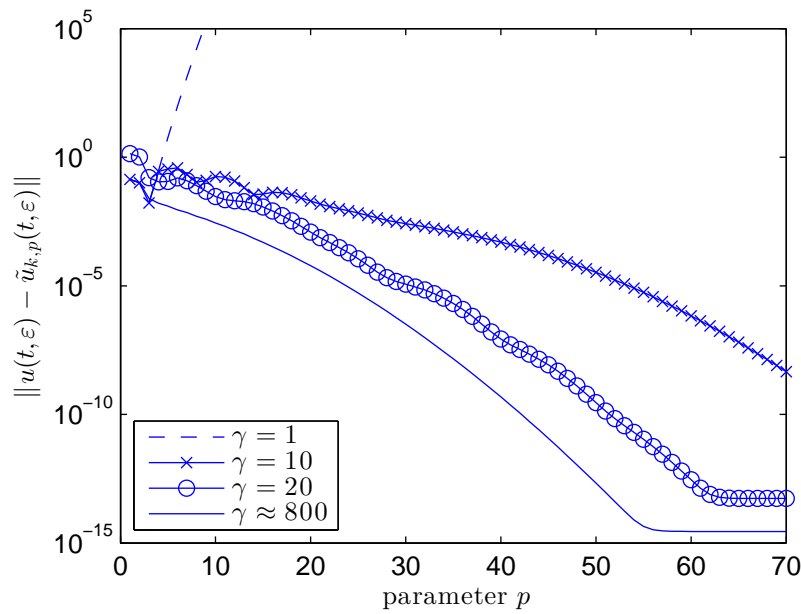


FIGURE 5.3. 2-norm errors of approximations $\tilde{u}_{k,p}(t, \varepsilon)$, when $\varepsilon = 1.5 \cdot 10^{-2}$ using different scalings (5.1). The last option corresponds to the scaling (5.2).

from Robin boundary conditions, and M the mass matrix which is a diagonal matrix. We carry out numerical experiments for parameter values $\gamma_1 = 0, 1, 2$

and $\gamma_2 = 0, 1, 2$.

We reformulate (5.5) in the form (1.1) by setting

$$A_0 = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}\gamma_1 C_1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 \\ 0 & -M^{-1}C_2 \end{bmatrix}.$$

Then, the variable ε in (1.1) corresponds to γ_2 . This means that by running the algorithm for a fixed value of γ_1 , we may efficiently obtain solutions for different values of t and γ_2 .

Figure 5.4 shows the contour plots of the numerical solutions of (5.5) at $t = 9$ on the plane $\{(x, y, z) \in [0, 1]^3 : z = 0.5\}$ for different values of (γ_1, γ_2) . Note that for a fixed value of γ_1 , only one run of the algorithm is required to compute the solution for many different γ_2 .

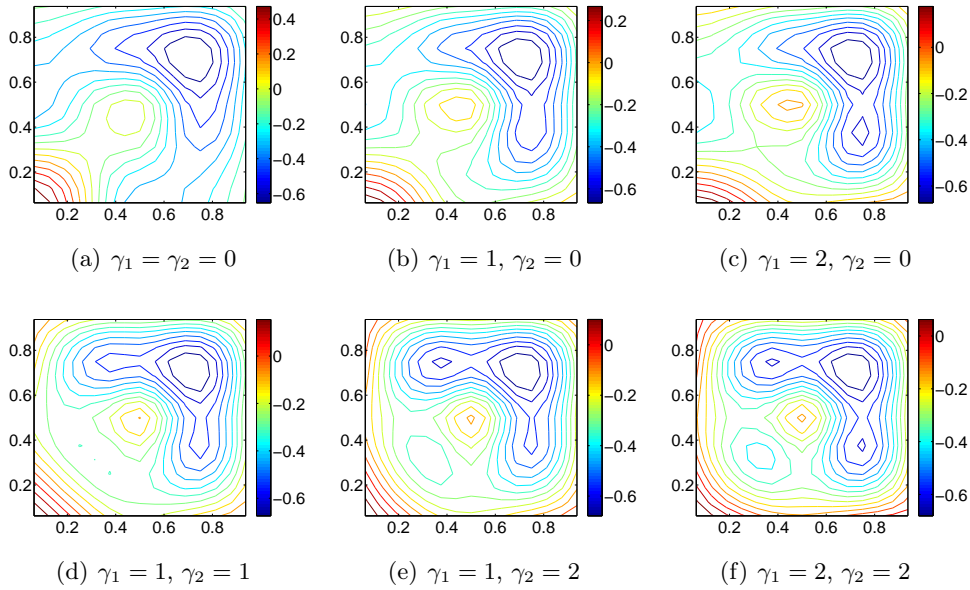


FIGURE 5.4. The solution to (5.5) in the plane $z = 0.5$, for different values of (γ_1, γ_2) at $t = 9$.

In Figure 5.5 we illustrate the 2-norm errors of the approximations, when $\gamma_1 = 2$ and $\gamma_2 = 0, 1$ and 2 . Then, $\|tA_0\| \approx 108$, and, respectively, $\|t\gamma_2 A_1\| \approx 0, 9.6$ and 12.9 . We again observe superlinear convergence, and, moreover, the a posteriori error estimate is very accurate for this example.

Table 5.1 illustrates the costs of the different parts of the algorithm when producing the approximation $\tilde{u}_{k,p}(t, \varepsilon)$ for the solution of (5.5), when $\gamma_1 = 2$ and $\gamma_2 = 1$. We see that the computational cost of evaluating the vectors $\tilde{c}_\ell(t)$ (i.e., evaluating the product $Q_p \exp(tH_p)e_1$) for different t is very small compared to the cost of producing the matrices Q_p and H_p using Algorithm 1. And as illustrated by these timings, the cost of evaluating the solution for different ε for given t (i.e., evaluating the sum $\sum_\ell \tilde{c}_\ell(t)\varepsilon^\ell$) is very moderate.

p	Time for Arnoldi	Time for $Q_p \exp(tH_p)e_1$	Time for $\sum_{\ell} \tilde{c}_{\ell}(t)\varepsilon^{\ell}$
10	0.067	0.0031	$1.75 \cdot 10^{-4}$
30	0.912	0.0316	$3.33 \cdot 10^{-4}$
50	3.942	0.0833	$5.40 \cdot 10^{-4}$
70	10.53	0.1591	$7.59 \cdot 10^{-4}$

TABLE 5.1

Table of CPU timings (in seconds) for the example (5.5): CPU time of running p steps of Algorithm 1, evaluating the vectors $\tilde{c}_{\ell}(t)$, $\ell = 0, \dots, p-1$, and of evaluating the series $\sum_{\ell=0}^{p-1} c_{\ell}(t)\varepsilon^{\ell}$, when $p = 10, 30, 50, 70$.

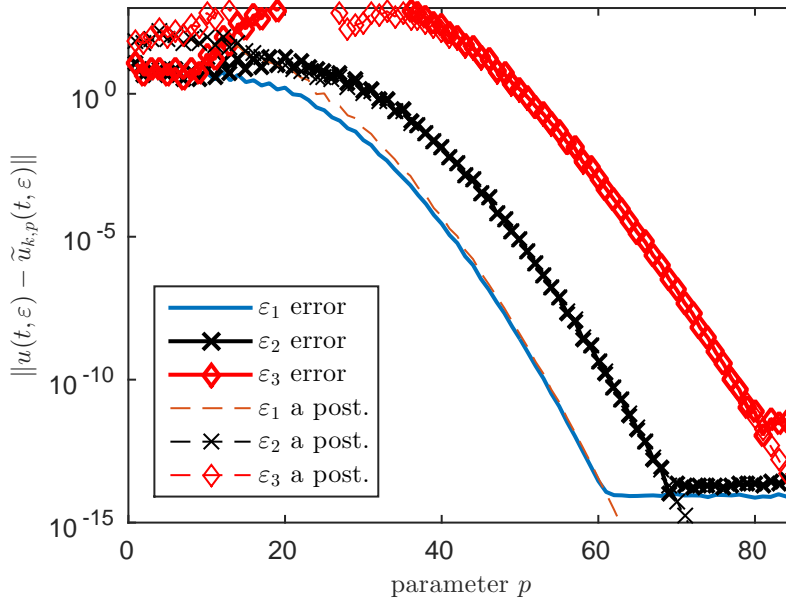


FIGURE 5.5. 2-norm errors of approximations $\tilde{u}_{k,p}(t, \varepsilon)$ for the equation (5.4) and the error estimate (4.3), when $\gamma_1 = 2$ and γ_2 has the values $\varepsilon_1 = 0$, $\varepsilon_2 = 1$ and $\varepsilon_3 = 2$.

6. Conclusions and outlook. The focus of this paper is an algorithm for parameterized linear ODEs, which is shown to have superlinear convergence in theory and perform convincingly in several examples. The behavior is consistent with what is expected from an Arnoldi method. Due to the equivalence with the Arnoldi method, the algorithm may suffer from the typical disadvantages of the Arnoldi method, for instance, the fact that the computation time per iteration increases with the iteration number. The standard approach to resolve this issue is by using restarting [7], which we leave for future work. We note that the technique we have presented is in principle applicable also to ODEs with several perturbation variables. Such constructions would however lead to a more severe growth in the basis matrix and requires further work to be used as a competitive algorithm.

Acknowledgments. The authors thank anonymous reviewers for their constructive comments.

REFERENCES

- [1] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM J. Sci. Comput.*, 33:488–511, 2011.
- [2] P. Benner, S. Gugercin, and K. Willcox. A survey of model reduction methods for parametric systems. Technical report, Max Planck Institute Magdeburg, 2013.
- [3] T. Betcke. Optimal scaling of generalized and polynomial eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 30(4):1320–1338, 2008.
- [4] S. Blanes, F. Casas, J. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Physics Reports*, 470:151–238, 2009.
- [5] H. G. Bock, S. Körkel, and J. P. Schlöder. Parameter estimation and optimum experimental design for differential equation models. In H. G. Bock, Th. Carraro, W. Jäger, S. Körkel, R. Rannacher, and J. Schlöder, editors, *Model Based Parameter Estimation*, pages 1–30. Springer, 2013.
- [6] R. Eid. *Time Domain Model Reduction By Moment Matching*. PhD thesis, TU München, 2008.
- [7] M. Eiermann and O. G. Ernst. A restarted Krylov subspace method for the evaluation of matrix functions. *SIAM J. Numer. Anal.*, 44(6):2481–2504, 2006.
- [8] H.-Y. Fan, W.-W. Lin, and P. Van Dooren. Normwise scaling of second order polynomial matrices. *SIAM J. Matrix Anal. Appl.*, 26(1):252–256, 2004.
- [9] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Stat. Comput.*, 13(5):1236–1264, 1992.
- [10] N. J. Higham. *Functions of Matrices. Theory and Computation*. SIAM, Philadelphia, 2008.
- [11] N. J. Higham and S. D. Relton. Higher order Fréchet derivatives of matrix functions and the level-2 condition number. *SIAM J. Matrix Anal. Appl.*, 35(3):1019–1037, 2014.
- [12] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997.
- [13] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19(5):1552–1574, 1998.
- [14] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.
- [15] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1991.
- [16] E. Jarlebring, W. Michiels, and K. Meerbergen. A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numer. Math.*, 122(1):169–195, 2012.
- [17] A. Koskela and E. Jarlebring. The infinite Arnoldi exponential integrator for linear inhomogeneous ODEs. Technical report, KTH Royal institute of technology, 2014. arxiv preprint.
- [18] P. Lietaert and K. Meerbergen. Interpolatory model order reduction by tensor Krylov methods. Technical report, KU Leuven, 2015.
- [19] R. Mathias. A chain rule for matrix functions and applications. *SIAM J. Matrix Anal. Appl.*, 17(3):610–620, 1996.
- [20] I. Najfeld and T. F. Havel. Derivatives of the matrix exponential and their computation. *Adv. Appl. Math.*, 16(3):321–375, 1995.
- [21] S. D. Relton. *Algorithms for Matrix Functions and their Fréchet Derivatives and Condition Numbers*. PhD thesis, Univ. Manchester, 2014.
- [22] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential

- operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.
- [23] B. A. Schmitt and E. Kostina. Peer two-step methods with embedded sensitivity approximation for parameter-dependent ODEs. *SIAM J. Numer. Anal.*, 50(5):2182–2207, 2012.
- [24] R. B. Sidje. Expokit: a software package for computing matrix exponentials. *ACM Trans. Math. Software*, 24(1):130–156, 1998.
- [25] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005.
- [26] Y. Yue and K. Meerbergen. Using Krylov-Padé model order reduction for accelerating design optimization of structures and vibrations in the frequency domain. *Int. J. Numer. Methods Eng.*, 90(10):1207–1232, 2012.

Appendix A. Technical lemmas for the norm and the field of values of L_m .

We now provide bounds on the norm and the field of values of L_m , which are needed in Section 3.1. The derivation is done with properties of field of values. Recall that the field of values of a matrix $A \in \mathbb{C}^{n \times n}$ is defined as

$$\mathcal{F}(A) = \{x^*Ax : x \in \mathbb{C}^n, \|x\| = 1\}.$$

The bounds for the norm and the field of values of \mathcal{A}_N follow from the block structure of L_m .

LEMMA 8. *Let $N \geq 0$ and L_m be given by (2.2). Then,*

$$\|L_m\| \leq \sum_{\ell=0}^N \|A_\ell\|$$

Proof. Let $x = [x_1^T \dots x_m^T]^T \in \mathbb{C}^{nm}$ such that $x_i \in \mathbb{C}^n$ for all $1 \leq i \leq m$ and $\|x\| = 1$. From the block Toeplitz structure of L_m we see that

$$\|L_m x\| \leq \sum_{\ell=0}^N \sqrt{\sum_{k=1}^{n-\ell} \|A_\ell x_k\|^2} \leq \sum_{\ell=0}^N \|A_\ell\| \sqrt{\sum_{k=1}^n \|x_k\|^2} = \sum_{\ell=0}^N \|A_\ell\|.$$

□

Next, we give a bound for the field of values of the matrix L_m . Let $d(\mathcal{S}, z)$ denote the distance between a closed set \mathcal{S} and a point z .

LEMMA 9. *Let $N \geq 0$ and L_m be given by (2.2). Then,*

$$\mathcal{F}(L_m) \subset \{z \in \mathbb{C} : d(\mathcal{F}(A_0), z) \leq \sum_{\ell=1}^N \|A_\ell\|\}.$$

Proof. Let $x = [x_1^T \dots x_m^T]^T \in \mathbb{C}^{nm}$, where $x_i \in \mathbb{C}^n$ for all $1 \leq i \leq m$ and $\|x\| = 1$. Then

$$(A.1) \quad x^* L_m x = \sum_{\ell=1}^m x_\ell^* A_0 x_\ell + x^* \tilde{L}_m x,$$

where \tilde{L}_m equals L_m on the subdiagonal blocks and is otherwise zero. By the convexity of the field of values [15, Property 1.2.2], we know that $\sum_{\ell=1}^m x_\ell^* A_0 x_\ell \in \mathcal{F}(A_0)$. The second term in (A.1) can be bounded as in proof of Lemma 8, giving

$$|x^* \tilde{L}_m x| \leq \|\tilde{L}_m\| \leq \sum_{\ell=1}^N \|A_\ell\|.$$

□

As a corollary of Lemma 9, we have the following bound, which follows directly from the fact that the logarithmic norm of a matrix in 2-norm equals the real part of the rightmost point in its field of values.

COROLLARY 10. *Let L_m be given by (2.2). Then,*

$$(A.2) \quad \mu(L_m) \leq \mu(A_0) + \sum_{\ell=1}^N \|A_\ell\|,$$

where $\mu(A)$ denotes the logarithmic norm of A defined in (1.5).

Appendix B. Technical lemmas for coefficient bounds. We now derive bounds needed for the a priori analysis of the truncation error in Section 3.2. The following result provides an explicit characterization of the expansion coefficients. The proof technique is based on the same type of reasoning as what is commonly used in the analysis of Magnus series expansions for time-dependent ODEs; see e.g. [4].

LEMMA 11 (Explicit integral form). *Let ℓ and N be positive integers such that $N \leq \ell$. Denote by C_ℓ the set of compositions of ℓ , i.e.,*

$$(B.1) \quad C_\ell = \{(i_1, \dots, i_r) \in \mathbb{N}_+^r : i_1 + \dots + i_r = \ell\},$$

and further denote

$$(B.2) \quad C_{\ell,N} := \{(i_1, \dots, i_r) \in C_\ell : i_s \leq N \text{ for all } 1 \leq s \leq r\}.$$

Then,

$$(B.3) \quad \begin{aligned} c_0(t) &= e^{tA_0} u_0, \\ c_\ell(t) &= \sum_{(i_1, \dots, i_r) \in C_{\ell,N}} \int_0^t e^{(t-t_{i_1})A_0} A_{i_1} \int_0^{t_{i_1}} e^{(t_{i_1}-t_{i_2})A_0} A_{i_2} \\ &\quad \dots \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t_{i_r}) dt_{i_1} \dots dt_{i_r} \quad \text{for } \ell > 0. \end{aligned}$$

Proof. From the ODE (2.4) and the variation-of-constants formula it follows that

$$(B.4a) \quad c_0(t) = e^{tA_0} u_0,$$

$$(B.4b) \quad c_\ell(t) = \sum_{k=1}^{\min\{N,\ell\}} \int_0^t e^{(t-s)A_0} A_k c_{\ell-k}(s) ds \quad \text{for } \ell > 0.$$

Using (B.4) we now prove (B.3) by induction. For $\ell = 1$, we have $C_1 = \{(1)\}$ and $C_{1,1} = \{(1)\}$. From (B.4a) and (B.4b) we directly conclude that

$$c_1(t) = \int_0^t e^{(t-t_1)A_0} A_1 c_0(t_1) dt_1.$$

Suppose (B.3) holds for $\ell = 1, \dots, p-1$ for some value $p > 1$. From Definition (B.1), we know that the row sum of any element of C_p is p , and the row sum of any element in C_{p-k} is $p-k$. Therefore, C_p satisfies the recurrence relation

$$(B.5) \quad C_p = \bigcup_{k=1}^p \bigcup_{(i_1, \dots, i_r) \in C_{p-k}} (k, i_1, \dots, i_r)$$

and

$$(B.6) \quad C_{p,N} = \bigcup_{k=1}^{\min(N,p)} \bigcup_{(i_1, \dots, i_r) \in C_{p-k,N}} (k, i_1, \dots, i_r).$$

By using (B.4b) with $\ell = p$ and the fact that (B.3) is assumed to be satisfied for $\ell = 1, \dots, p-1$ we have

$$\begin{aligned} c_p(t) &= \sum_{k=1}^{\min\{N,p\}} \int_0^t e^{(t-s)A_0} A_k c_{p-k}(s) ds \\ &= \sum_{k=1}^{\min\{N,p\}} \int_0^t e^{(t-s)A_0} A_k \sum_{(i_1, \dots, i_r) \in C_{p-k,N}} \int_0^s e^{(s-t_{i_1})A_0} A_{i_1} \dots \\ &\quad \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t_{i_r}) dt_{i_1} \dots dt_{i_r} ds. \end{aligned}$$

By rearranging the terms as a double sum and using (B.6), we have

$$\begin{aligned}
c_p(t) &= \sum_{k=1}^{\min\{N,p\}} \sum_{(i_1, \dots, i_r) \in C_{p-k, N}} \int_0^t e^{(t-s)A_0} A_k \int_0^s e^{(s-t_{i_1})A_0} A_{i_1} \dots \\
&\quad \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t_{i_r}) dt_{i_1} \dots dt_{i_r} ds \\
&= \sum_{(i_1, \dots, i_r) \in C_{p, N}} \int_0^t e^{(t-t_{i_1})A_0} A_{i_1} \dots \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t_{i_r}) dt_{i_1} \dots dt_{i_r}
\end{aligned}$$

which shows that (B.3) holds for $\ell = p$ and completes the proof.

□

LEMMA 12. *Let m, N be positive integers, $N \leq m$, and let $C_{m, N}$ be defined as in (B.2). Let $(i_1, i_2, \dots, i_r) \in C_{m, N}$, $a = \max_{j=1, \dots, N} \|A_j\|$, and assume that for all $1 \leq j \leq N$. Then, one corresponding term in (B.3) is bounded as*

$$\begin{aligned}
\text{(B.7)} \quad & \left\| \int_0^t e^{(t-t_{i_1})A_0} A_{i_1} \dots \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t_{i_r}) dt_{i_1} \dots dt_{i_r} \right\| \\
& \leq e^{t\mu(A_0)} \frac{(ta)^r}{r!} \|u_0\|.
\end{aligned}$$

Proof. By using the Dahlquist bound [25, p. 138] for the matrix exponential, the rightmost integral on the left-hand side of (B.7) can be bounded as

$$\begin{aligned}
& \left\| \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} e^{t_{i_r}A_0} u_0 dt_{i_r} \right\| \\
& \leq \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})\mu(A_0)} \|A_{i_r}\| e^{t_{i_r}\mu(A_0)} dt_{i_r} \|u_0\| = t_{i_{r-1}} a e^{t_{i_{r-1}}\mu(A_0)} \|u_0\|.
\end{aligned}$$

The claim (B.7) follows by applying the same bounding technique $r - 1$ times for the remaining integrals, and using that $t_i \leq t$ for any i . □

LEMMA 13 (Coefficient bound). *Let c_0, c_1, \dots be the ε -expansion of $u(t, \varepsilon)$ in (1.2) for $N \geq 1$ in (1.2), and let $a = \max_{j=1, \dots, N} \|A_j\|$. Then, for any $\ell \geq 0$ such that $k := \lceil \frac{\ell}{N} \rceil \geq 2$,*

$$\|c_\ell(t)\| \leq e^{t(\mu(A_0) + eNa) - 1} \frac{(eNta)^k}{(k-1)!} \|u_0\|.$$

Proof. We first note that the maximum length of any vector in $C_{m,N}$ is m , and the vector with the shortest length has length at least $k = \lceil \frac{m}{N} \rceil$. Hence, Lemma 11 can be rephrased as

$$(B.8) \quad c_m(t) = \sum_{\widehat{r}=k}^m \sum_{\substack{(i_1, \dots, i_r) \in C_{m,N} \\ r=\widehat{r}}} \int_0^t e^{(t-t_{i_1})A_0} A_{i_1} \dots \int_0^{t_{i_{r-1}}} e^{(t_{i_{r-1}}-t_{i_r})A_0} A_{i_r} c_0(t) dt_{i_1} \dots dt_{i_r}.$$

Since, $C_{m,N} \subset C_m$ we can bound the number of elements in $C_{m,N}$

$$\#\{(i_1, \dots, i_r) \in C_{m,N} : r = \ell\} \leq \#\{(i_1, \dots, i_r) \in C_m : r = \ell\} = \binom{m-1}{\ell-1},$$

and Lemma 12 and (B.8) imply that

$$\|c_m(t)\| \leq e^{t\mu(A_0)} \sum_{\ell=k}^m \binom{m-1}{\ell-1} \frac{(ta)^\ell}{\ell!} \|u_0\|.$$

Moreover,

$$\binom{m-1}{r-1} = \frac{(m-1)(m-2)\dots(m-r+1)}{(r-1)!} \leq \frac{m^r}{(r-1)!}$$

and therefore

$$(B.9) \quad \|c_m(t)\| \leq e^{t\omega(A_0)} \sum_{r=k}^m \frac{m^r (ta)^r}{(r-1)! r!} \|u_0\| \leq e^{t\mu(A_0)} \sum_{r=k}^m \frac{r^r (Nta)^r}{(r-1)! r!} \|u_0\|.$$

In the second inequality in (B.9) we use $m = N\frac{m}{N} \leq N\lceil \frac{m}{N} \rceil = Nk \leq Nr$. Using the inequality $e\left(\frac{r}{e}\right)^r \leq r!$, $r \geq 1$, we see that for $k \geq 2$

$$\begin{aligned} \|c_m(t)\| &\leq e^{t\mu(A_0)} \sum_{r=k}^m \frac{r^r (Nta)^r}{e\left(\frac{r}{e}\right)^r (r-1)!} \|u_0\| \leq e^{t\mu(A_0)} \sum_{r=k}^{\infty} \frac{(eNta)^r}{e(r-1)!} \|u_0\| \\ &= e^{t\mu(A_0)-1} (eNta)^k \sum_{r=0}^{\infty} \frac{r!}{(r+k-1)!} \frac{(eNta)^r}{r!} \|u_0\| \\ &\leq \frac{e^{t(\mu(A_0)+eNa)-1} (eNta)^k}{(k-1)!} \|u_0\|, \end{aligned}$$

where in the last inequality we use $\frac{r!}{(r+k-1)!} \leq \frac{1}{(k-1)!}$. \square