BIT manuscript No.
(will be inserted by the editor)

Approximation of the matrix exponential for matrices with skinny fields of values

Marco Caliari · Fabio Cassini · Franco Zivcovich

the date of receipt and acceptance should be inserted later

Abstract The backward error analysis is a great tool which allows to select in an effective way the scaling parameter s and the polynomial degree of approximation m when the action of the matrix exponential $\exp(A)v$ has to be approximated by $\left(p_m(s^{-1}A)\right)^s v = \exp(A+\Delta A)v$. We propose here a rigorous bound for the relative backward error $\|\Delta A\|_2 / \|A\|_2$, which is of particular interest for matrices whose field of values is skinny, such as the discretization of the advection–diffusion or the Schrödinger operators. The numerical results confirm the superiority of the new approach with respect to methods based on the classical power series expansion of the backward error for the matrices of our interest, both in terms of computational cost and achieved accuracy.

Keywords backward error analysis · action of matrix exponential · Leja–Hermite interpolation · skinny field of values

Mathematics Subject Classification (2000) 65D05 · 65F30 · 65F60

1 Introduction

In the recent years, the problem of approximating the action of the matrix exponential on a vector $\exp(A)v$ has attracted an increasing amount of attentions. Among polynomial methods we recall the recent implementations of the Krylov method (see [8,9,20]), the truncated Taylor series expansion [2]

M. Caliari
University of Verona
E-mail: marco.caliari@univr.it

F. Cassini
University of Trento
E-mail: fabio.cassini@unitn.it

F. Zivcovich
University of Verona
E-mail: franco.zivcovich@univr.it

and polynomial interpolation methods (e.g., [5,6]). Among rational methods we recall instead the rational Krylov methods (see [10,19,23]) and the Carathéodory–Fejér approach used in [21]. For a survey on these and other methods we refer to [18] and [13, § 10 and § 13]. This interest is mainly due to the several applications where the action of the matrix exponential plays a fundamental role. Prominent examples are the exponential integrators [15], which constitute effective methods for the time integration of large stiff or oscillatory systems of differential equations. These very practical applications led the authors of this manuscript into refining existing techniques to achieve better accuracy and performances over a fairly specific class of matrices, that is the family of matrices having a skinny field of values. In fact, when it comes to the real applications, very often the spectrum of the matrices of interest is not just a scattered bunch of points on the complex plane. It is, on the contrary, contained in a skinny rectangle centered at the origin of the complex plane, after a proper shift of the matrix. We just mention the spatial discretization of diffusion, advection–diffusion, advection, and Schrödinger operators, among others.

The goal of this paper is to outline an algorithm for the computation of the action of the matrix exponential which exploits the information coming from the field of values.

In particular, we first split the problem into s easier-to-approximate substeps. Then, we employ a polynomial interpolation p_m of degree m of the exponential function at the so-called Leja–Hermite points. Through such approximation, we compute the action of the exponential of a slightly perturbed matrix, i.e. $\exp(A+\Delta A)$. For this approximation to be backward accurate, we have to ensure that m and s are taken so that the inequality $\|\Delta A\|_2 \leq \mathrm{tol}\cdot\|A\|_2$ holds true, for a prescribed tolerance tol. To do so, we rely on a tool called *backward error analysis*, which was employed in [2] for computing the action of the matrix exponential using a truncated Taylor expansion of degree m of the exponential function. By analyzing the norm of A, or some of its powers, the algorithm proposed in [2] can select effectively the number of sub-steps s and the polynomial degree m such that the action of the matrix exponential is approximated up to the desired accuracy. That original idea was then extended to the interpolation of the exponential function at Leja [5] and Leja–Hermite points [6]. These families of interpolation points enjoy useful numerical properties that are going to be crucial in the development of this work. In [5, § 3.2] the idea was introduced that the backward error can be analyzed by a contour integral expansion along a curve which embraces the ε-pseudo-spectrum of the matrix. In this paper, we refine this idea by considering a compact K which resembles the shape of the field of values. This choice produces more effective parameters s and m for the matrices of interest.

The paper is developed as follows. In section 2, we recall the idea of interpolating the exponential function in the Hermite sense. We then show how to equip interpolation polynomials with a backward error analysis as it was done in [6]. In section 3, we recall the classical backward error analysis based on the norms of A and some of its powers. In section 4, we formulate the backward

error analysis based on the field of values of A. We then apply this technique to the interpolation at Leja–Hermite points. Moreover, we overcome the idea of choosing the parameters s and m merely trying to minimize the expected cost $s \cdot m$ in terms of matrix-vector products. This turns out to be a rewarding strategy both in achieved accuracy and in terms of the total number of matrix-vector products, as confirmed by the numerical experiments presented in section 5. We finally draw some conclusions in section 6.

2 Polynomial interpolation of the exponential function

For computing the desired approximation of $\exp(A)v$ for a matrix $A \in \mathbb{C}^{n\times n}$ and a vector $v \in \mathbb{C}^n$, following the papers [2,5,6], we consider the polynomial approximation p_m of degree m of the exponential function coupled with a substepping strategy. Namely, we pick a positive integer s such that $p_m(s^{-1}A)v$ is easier to compute than $p_m(A)v$ and then we recover the wanted approximation by marching as follows

$$v^{(l+1)} = p_m(s^{-1}A)v^{(l)}, \quad l = 0, 1, \ldots, s-1 \tag{2.1}$$

where we set $v^{(0)} = v$. In particular, p_m is the polynomial which interpolates the function e^x in the Hermite sense at $m+1$ points $\{z_i\}_{i=0}^m$ over the real interval $[-c, c]$, $c \in \mathbb{R}_0^+$. We fix $z_0 = 0$, which means that $p_m(0) = \mathrm{e}^0 = 1$. Moreover we assume, without loss of generality, that $z_0 = z_1 = \ldots = z_\ell = 0$ for $0 \le \ell \le m$.

In order to determine if approximation (2.1) is accurate, we represent the *backward error*, that is the matrix ΔA such that

$$(p_m(s^{-1}A))^s = \exp(A + \Delta A),$$

as a function of A. In order to do so, we exploit the properties of the exponential function to obtain

$$\exp(-A)(p_m(s^{-1}A))^s = \exp(\Delta A).$$

Then, by applying the principal logarithm, we get the desired representation of the backward error as a function of A

$$s^{-1}\Delta A = h_{m+1}(s^{-1}A)$$

where

$$h_{m+1}(X) := \log(\exp(-X)p_m(X))$$

is a function defined on the matrix set

$$\{X \in \mathbb{C}^{n\times n} \colon \rho(\exp(-X)p_m(X) - I) < 1\},$$

with $\rho(\cdot)$ denoting the spectral radius of the matrix argument and I denoting the identity matrix of size n.

The function h_{m+1} has a power series expansion

$$h_{m+1}(X) = \sum_{k=\ell+1}^{\infty} c_k X^k \tag{2.2}$$

in which the first power of X is precisely $\ell + 1$. We refer to [1,2,5,6,12] for more details. The function h_{m+1} depends on ℓ and c, as well. Here we prefer to keep the notation used, for instance, in [2]. If we consider $c = 0$ or $\ell = m$, then all the points coincide with $z_0 = 0$ and the Hermite interpolation is nothing else than the truncated Taylor series approximation of e^x.

Additionally, we can consider a purely imaginary interpolation interval $[-c, c] = \mathrm{i}[-|c|, |c|]$, for $c \in \mathrm{i}\mathbb{R}^+$. In this case, in order to keep real arithmetic for real input A (see [5,22], for instance), it is necessary to use complex conjugate points defined by

$$\begin{aligned} z_0 = \ldots = z_\ell = 0, &\quad \ell + m \text{ even} \\ z_{i+2} = \overline{z_{i+1}}, &\quad i = \ell, \ell+2, \ldots, m-2. \end{aligned}$$

Clearly, the real and the imaginary interpolations above are more effective when approximating the exponential function around the origin of the complex plane. Since the matrix exponential can be written as a polynomial interpolation in the Hermite sense

$$\exp(A) = \sum_{i=0}^{n-1} \left(\exp[\lambda_0, \lambda_1, \ldots, \lambda_i] \prod_{j=0}^{i-1} (A - \lambda_j I) \right),$$

where $\exp[\lambda_0, \lambda_1, \ldots, \lambda_i]$ denotes the divided difference of order i of the exponential function at the set $\{\lambda_j\}_{j=0}^{i}$ of the eigenvalues of A, it is convenient that the eigenvalues lay around the origin, too. It is usual to shift the matrix A in order to reach such a desired result, for instance, by considering the shift given by $\mathrm{trace}(A)/n$, which corresponds to the average eigenvalue of A (see [2]). Here we prefer to consider the following distinct shift strategy (see [6]). We split the matrix A into its Hermitian A_{H} and skew-Hermitian A_{sH} parts and estimate their extreme eigenvalues (real and pure imaginary values, respectively) by using Geršgorin's disks. We obtain

$$\mathrm{conv}(\sigma(A_{\mathrm{H}})) \subseteq [\alpha, \nu], \quad \mathrm{conv}(\sigma(A_{\mathrm{sH}})) \subseteq \mathrm{i}[\eta, \beta],$$

where conv denotes the convex hull of a set. Therefore, by considering the field of values of a matrix

$$\mathcal{W}(A) = \{z \in \mathbb{C} \colon z = x^* A x, \text{ for } x \in \mathbb{C}^n \text{ with } x^* x = 1\},$$

using its sub-additivity property and the equivalence between field of values and convex hull of the spectrum for normal matrices, we have

$$\begin{aligned} \mathcal{W}(A) &= \mathcal{W}(A_{\mathrm{H}} + A_{\mathrm{sH}}) \subseteq \mathcal{W}(A_{\mathrm{H}}) + \mathcal{W}(A_{\mathrm{sH}}) = \\ &= \mathrm{conv}(\sigma(A_{\mathrm{H}})) + \mathrm{conv}(\sigma(A_{\mathrm{sH}})) \subseteq [\alpha, \nu] + \mathrm{i}[\eta, \beta]. \end{aligned}$$

If we define $R(A) = [\alpha, \nu] + \mathrm{i}[\eta, \beta]$, the previous chain of inclusions becomes

$$\mathcal{W}(A) \subseteq R(A). \tag{2.3}$$

Given the rectangle $R(A)$ containing the field of values of the matrix A, our choice of the shift μ is given by its center, that is

$$\mu = (\alpha + \nu)/2 + \mathrm{i}(\eta + \beta)/2. \tag{2.4}$$

Therefore, we work in practice with $A - \mu I$ whose rectangle

$$R(A - \mu I) = \left[\frac{\alpha - \nu}{2}, \frac{\nu - \alpha}{2}\right] + \mathrm{i}\left[\frac{\eta - \beta}{2}, \frac{\beta - \eta}{2}\right]$$

lays symmetrically about the origin of the complex plane. In order to recover the desired approximation of $\exp(A)v$, thanks to the properties of the exponential function, we can multiply e^{μ} into the vector $v^{(s)}$ obtained by marching as in formula (2.1). On the other hand, if the real part of μ is negative, it is convenient to recover the desired approximation by multiplying $\mathrm{e}^{s^{-1}\mu}$ into $v^{(l)}$ at each sub-step l. Although the two ways of recovering the approximation of $\exp(A)v$ are mathematically equivalent, the second one reduces the possibility to over-flow when the matrix A has eigenvalues with large negative real parts (see [2]). For sake of simplicity, we denote the shifted matrix again by A and the corresponding rectangle by $R(A) = [-\nu, \nu] + \mathrm{i}[-\beta, \beta]$, with $\nu, \beta \geq 0$.

We conclude this section by recalling that there exist more sophisticated ways to enclose the field of values into shapes different from the rectangular one. We refer to [16].

3 Backward error analysis based on the norm of A

In this section, we recall the classical way to perform the backward error analysis, which is based on the norms of the matrices. We refer to [2,5].

If A is the null matrix, we already know that $p_m(A) = \exp(A) = I$. Otherwise, the starting point is the following inequality

$$\frac{\|\Delta A\|}{\|A\|} = \frac{\left\|h_{m+1}(s^{-1}A)\right\|}{\|s^{-1}A\|} \leq \frac{\tilde{h}_{m+1}(s^{-1}\,\|A\|)}{s^{-1}\,\|A\|} \tag{3.1a}$$

where

$$\tilde{h}_{m+1}(X) = \sum_{k=\ell+1}^{\infty} |c_k| X^k.$$

It is possible to compute a priori and with the help of a software for multiple precision arithmetic the scalar value θ_m defined by

$$\theta_m = \max\{\theta \colon \tilde{h}_{m+1}(\theta)/\theta \leq \mathrm{tol}\}$$

where tol is a prescribed tolerance (usually the unit round off for the double or the single precision). If we select 0 as interpolation point at least *twice* ($\ell > 0$), we can write

$$\frac{\tilde{h}_{m+1}(\theta)}{\theta} = \sum_{k=\ell+1}^{\infty} |c_k| \theta^{k-1}.$$

In this way, since $\tilde{h}_{m+1}(\theta)/\theta$ is a monotonic increasing function of θ, θ_m is the unique positive solution of

$$\frac{\tilde{h}_{m+1}(\theta)}{\theta} = \text{tol}.$$

For a given matrix A, if $s^{-1} \|A\| \leq \theta_m$, for a proper integer scaling s, then from (3.1a) we have $\|\Delta A\| / \|A\| \leq \text{tol}$.

According to [1, Thm. 4.2(a)], it is possible to refine estimate (3.1a) with the following

$$\frac{\|\Delta A\|}{\|A\|} = \frac{\left\|h_{m+1}(s^{-1}A)\right\|}{\|s^{-1}A\|} \leq \frac{\tilde{h}_{m+1}(s^{-1}\alpha_q(A))}{s^{-1}\alpha_q(A)}, \quad \text{if } q(q-1) \leq \ell + 1 \tag{3.1b}$$

where

$$\alpha_q(X) = \max\{\|X^q\|^{\frac{1}{q}}, \left\|X^{q+1}\right\|^{\frac{1}{q+1}}\}.$$

In fact

$$\rho(X) \leq \|X^q\|^{\frac{1}{q}} \leq \|X\|, \quad \lim_{q\to\infty} \|X^q\|^{\frac{1}{q}} = \rho(X) \tag{3.2}$$

and the values $\alpha_q(A)$ can be *much* smaller than $\|A\|$, for instance for non-normal matrices. When it happens, the estimate of the relative backward error given in (3.1b) is sharper than (3.1a) and it allows to satisfy the requirement $\|\Delta A\| \leq \text{tol} \cdot \|A\|$ provided that the weaker inequality $s^{-1}\alpha_q(A) \leq \theta_m$ holds. The choice of a smaller scaling parameter s reduces the *over-scaling* phenomenon. In fact, when the scaling parameter s is chosen too large, it is possible that rounding errors seriously affect the final result. For instance, for x sufficiently small, $(1 + x)$ can be closer to e^x than $(1 + x/s)^s$, for s large. The values $\|X^q\|$ can be estimated in the 1-norm by using the algorithm described in [14]. It is important to remark that it is possible to use the inequality in (3.1b) only if the number $\ell + 1$ of zeros among the interpolation points is large enough. In particular, it is not possible to use this technique with pure Chebyshev or Leja interpolation points (see [5]). In [6] it has been shown that the possibility to select the interpolation intervals $[-c, c]$ or $\mathrm{i}[-|c|, |c|]$ and the number $\ell + 1$ of zeros among the points leads to polynomial approximations that often outperform the truncated Taylor series approach.

4 Backward error analysis based on the field of values of A

The use of the $\alpha_q(A)$ values may help a lot to reduce the scaling parameter s. On the other hand, the backward error analysis based on the norms of the matrices cannot distinguish, for instance, matrices with real eigenvalues in an

interval from matrices with eigenvalues spread in a disk with that interval as a diameter. Therefore we aim at exploiting the information contained into the rectangle $R(A)$.

Since 0 is among the interpolation points, we can write

$$h_{m+1}(X) = X \sum_{k=\ell+1}^{\infty} c_k X^{k-1} = X g_{m+1}(X). \tag{4.1}$$

By using the estimate [7, inequality (3)]

$$\|g_{m+1}(X)\|_2 \leq (1+\sqrt{2}) \sup_{z \in \mathcal{W}(X)} |g_{m+1}(z)|,$$

we get

$$\|g_{m+1}(X)\|_2 \leq (1+\sqrt{2}) \|g_{m+1}\|_{\Gamma},$$

where $\Gamma = \partial K$ denotes the contour of a domain $K \subset \mathbb{C}$ that contains the field of values $\mathcal{W}(X)$ of X and

$$\|g_{m+1}\|_{\Gamma} = \max_{z \in \Gamma} |g_{m+1}(z)| = \max_{z \in K} |g_{m+1}(z)|.$$

Therefore

$$\frac{\|\Delta A\|_2}{\|A\|_2} = \frac{\left\|h_{m+1}(s^{-1}A)\right\|_2}{\|s^{-1}A\|_2} \leq \left\|g_{m+1}(s^{-1}A)\right\|_2 \leq (1+\sqrt{2}) \|g_{m+1}\|_{\Gamma} \tag{4.2}$$

if $\Gamma = \partial K$, with K now containing the field of values $\mathcal{W}(s^{-1}A)$ of $s^{-1}A$. Thanks to (2.3), this is certainly true if

$$R(s^{-1}A) = s^{-1}R(A) \subseteq K. \tag{4.3}$$

Now we have to restrict the choice of possible domains K of interest. We consider the domain K circumscribed by an ellipses Γ_γ

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad z = x + \mathrm{i}y$$

whose focal interval is the interpolation interval $[-c, c]$ and the capacity $(a+b)/2$ is γ. Since $c^2 = a^2 - b^2$ (c can be real or purely imaginary), it turns out that the ellipse Γ_γ has semi-axes

$$a = \gamma + \frac{c^2}{4\gamma}, \quad b = \gamma - \frac{c^2}{4\gamma}.$$

Such a choice for the domains K makes it possible to select the ellipse Γ_{γ_m}, for a given c, which realizes

$$(1+\sqrt{2}) \|g_{m+1}\|_{\Gamma_{\gamma_m}} = \mathrm{tol} \tag{4.4}$$

by finding the root (by the secant method, e.g.) of the uni-variate function

$$\gamma \mapsto (1+\sqrt{2}) \|g_{m+1}\|_{\Gamma_\gamma} - \mathrm{tol}.$$

Such a function has at most one positive root. In fact, $\gamma \geq |c|/2$ and the function is monotonically increasing with γ, by the maximum modulus principle. Therefore, either the error estimate exceeds tol already for $\Gamma_{|c|/2} = [-c, c]$ (it means that the interpolation degree m is not large enough for the given interval) or there exists one positive root. Therefore, for each polynomial of interest $p_m \colon [-c, c] \to \mathbb{R}$ ($p_m \colon \mathrm{i}[-|c|, |c|] \to \mathbb{R}$) which interpolates e^x at $m+1$ points containing $\ell + 1$ zeros, it is possible to pre-compute once and for all, with a software for multiple precision arithmetic, the semi-axes a_m and b_m of the ellipse Γ_{γ_m} with capacity γ_m which satisfies (4.4).

In order to find the smallest integer value s for which inclusion (4.3) is satisfied, it is now possible to solve the inequality

$$\frac{\nu^2}{s^2 a_m^2} + \frac{\beta^2}{s^2 b_m^2} \leq 1,$$

where $s^{-1}(\nu, \beta)$ is the top-right vertex of the rectangle $s^{-1}R(A)$, which gives

$$s = \left\lceil \sqrt{\frac{\nu^2}{a_m^2} + \frac{\beta^2}{b_m^2}} \right\rceil. \tag{4.5}$$

We briefly sketch again the procedure:

1. For a given matrix A, compute the rectangle which contains its field of values, shift it by μI (see (2.4)) and compute the final centered rectangle $R(A) = [-\nu, \nu] + \mathrm{i}[-\beta, \beta]$.
2. Compute s as in (4.5).
3. Approximate $\exp(A)v$ by $v^{(s)}$ as in (2.1).

The backward error analysis ensures that the result is such that $v^{(s)} = \exp(A + \Delta A)v$, with $\|\Delta A\|_2 \leq \mathrm{tol} \cdot \|A\|_2$.

If approximations at different matrix scales $\exp(t_i A)v_i$, $t_i \geq 0$, are required (this is quite common in the so-called exponential integrators [15]), it is possible to compute the matrix-dependent quantity

$$r(A) = \sqrt{\frac{\nu^2}{a_m^2} + \frac{\beta^2}{b_m^2}}$$

once and for all and later to select the scaling parameter as

$$s_i = \lceil t_i r(A) \rceil.$$

This was not possible in [5, § 3.2] where a contour integral expansion of the backward error on the ε-pseudo-spectrum was employed for a fixed value $\varepsilon = 1/50$ independent of the matrix A.

4.1 Possible refinement of the rectangle $R(A)$

Once the original matrix has been shifted, it would be possible to compute the rectangle $R(A)$ in the following way

$$R(A) = [-\|A_{\mathrm{H}}\|_2, \|A_{\mathrm{H}}\|_2] + \mathrm{i}[-\|A_{\mathrm{sH}}\|_2, \|A_{\mathrm{sH}}\|_2],$$

thanks to the inclusions

$$\begin{aligned}\sigma(A_{\mathrm{H}}) &\subseteq [-\rho(A_{\mathrm{H}}), \rho(A_{\mathrm{H}})] = [-\|A_{\mathrm{H}}\|_2, \|A_{\mathrm{H}}\|_2],\\ \sigma(A_{\mathrm{sH}}) &\subseteq \mathrm{i}[-\rho(A_{\mathrm{sH}}), \rho(A_{\mathrm{sH}})] = \mathrm{i}[-\|A_{\mathrm{sH}}\|_2, \|A_{\mathrm{sH}}\|_2].\end{aligned}$$

Such a rectangle is in general smaller than the one given by the Geršgorin's disks. On the other hand, it requires two 2-norm computations (or estimates). The standard power method can be used to produce two non-decreasing sequences of estimates of the 2-norms of A_{H} and A_{sH} (see [11]), in a fast way and without even forming the matrices A_{H} and A_{sH}. However, running too few iterations could yield to a rectangle which in principle does not contain the eigenvalues. As a remedy, it is possible to exploit the inequalities in (3.2) which ensure that $\rho(A_{\mathrm{H}}) \le \|A_{\mathrm{H}}^q\|_1^{1/q}$ and $\rho(A_{\mathrm{sH}}) \le \|A_{\mathrm{sH}}^q\|_1^{1/q}$, and compute or approximate the values $\|A_{\mathrm{H}}^q\|_1^{1/q}$ and $\|A_{\mathrm{sH}}^q\|_1^{1/q}$ for small values of q. Estimates with few iterations of the required norms are commonly used in the literature (see [2,5,6]).

4.2 Application to Leja–Hermite interpolation points

As an application of the above ideas, we consider the Leja–Hermite interpolation points introduced in [3] and whose classical (norm based) backward error estimate was considered in [6].

Leja–Hermite points are defined by

$$\begin{aligned} &z_0 = z_1 = \ldots = z_\ell = 0,\\ &z_{i+1} \in \arg\max_{x\in[-c,c]} \prod_{j=0}^{i} |x - z_j|, \quad i = \ell, \ell+1, \ldots, m-1, \end{aligned} \tag{4.6a}$$

where $\ell \ge 0$. Points $z_{\ell+1}$, $z_{\ell+2}$, and $z_{\ell+3}$ are not uniquely determined and we select them as $z_{\ell+1} = c$, $z_{\ell+2} = -c$, and $z_{\ell+3} = c\sqrt{(\ell+1)/(\ell+3)}$.

When we work in the complex interval $[-c, c] = \mathrm{i}[-|c|, |c|]$ we use complex conjugate Leja–Hermite points defined by

$$\begin{aligned} &z_0 = \ldots = z_\ell = 0, \quad \ell + m \text{ even},\\ &z_{i+1} \in \arg\max_{x\in[-c,c]} \prod_{j=0}^{i} |x - z_i|, \quad z_{i+2} = \overline{z_{i+1}}, \quad i = \ell, \ell+2, \ldots, m-2 \end{aligned} \tag{4.6b}$$

The points $z_{\ell+1}$, $z_{\ell+2}$, and $z_{\ell+3}$ are selected in a similar way as above. With the introduction of Leja–Hermite points we have plenty of possibilities to select

the degree m of interpolation, the number $\ell+1$ of zeros among the interpolation points, and the interpolation interval $[-c,c]$, both real and imaginary. We propose in this section a way to perform a choice on a limited subset of possibilities.

Let us start with the real case and fix tol $= 2^{-53}$. For each m and ℓ, consider $c_k = k/2$, $k \in \mathbb{N}$, and denote by $c_{\bar{k}}$ the smallest value of this form for which it is not possible to satisfy (4.4). Between $c_{\bar{k}-1}$ and $c_{\bar{k}}$ there may be additional values of $c \in \mathbb{R}^+$ such that (4.4) is satisfied. We look for them by using few iterations of the bisection method. The final set of intervals to be considered for interpolation is made out of c_k, $k = 0, 1, \ldots, \bar{k}-1$ together with those values obtained in the bisection process.

Table 4.1 Values of a_m and b_m for degrees $m = 30$ and $m = 50$ ($\ell = 1$) and selected values of c.

$m = 30$			$m = 50$		
c	a_m	b_m	c	a_m	b_m
0	3.447e0	3.447e0	0	8.419e0	8.419e0
0.5	3.457e0	3.421e0	0.5	8.430e0	8.414e0
⋮	⋮	⋮	⋮	⋮	⋮
4	4.523e0	2.111e0	10	11.19e0	5.027e0
⋮	⋮	⋮	⋮	⋮	⋮
5.5	5.573e0	0.902e0	11.5	12.13e0	3.874e0
6	6.013e0	0.390e0	12.5	12.53e0	0.878e0
6.5	/	/	13	/	/
⋮	⋮	⋮	⋮	⋮	⋮
6.172e0	6.173e0	0.107e0	12.52e0	12.53e0	0.515e0
6.180e0	/	/	12.53e0	/	/

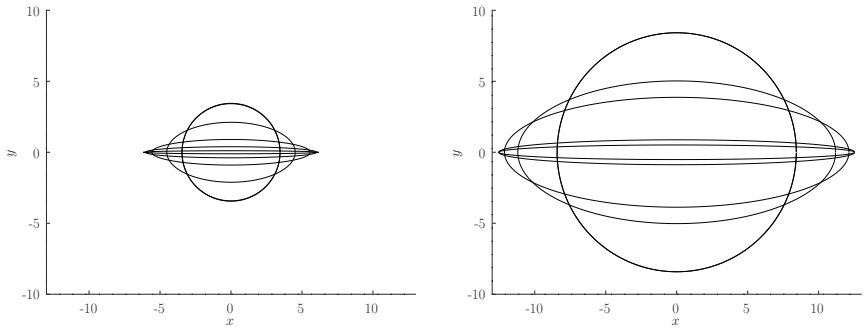

Fig. 4.1 Ellipses corresponding to degree 30 (left) and degree 50 (right) of Table 4.1. The ellipses corresponding to $c = 0$ and $c = 0.5$ cannot be distinguished in the plot.

For instance, for degrees $m = 30$ and $m = 50$ and $\ell = 1$, we can see in Table 4.1 that the largest possible values for a real c are 6.172e0 and 12.52e0, respectively (values reported with the exponential notation were rounded). The ellipses corresponding to the values displayed in Table 4.1 are drawn in Figure 4.1.

In a totally analogous way, we obtained the set of imaginary interpolation intervals.

For a given matrix A it remains now open the choices of the degree m, the number ℓ, and the interval $[-c, c]$ of interpolation. Since for a given number of sub-steps s and degree of interpolation m the cost of the polynomial evaluation, in terms of matrix-vector multiplications, is $s \cdot m$, we select m and c in order to minimize it. To give an example, we consider the discretization of the one-dimensional advection–diffusion operator $\partial_x + 0.02 \cdot \partial_{xx}$ with homogeneous Dirichlet boundary conditions and 149 inner nodes in the interval $[0, 1]$. In MATLAB syntax it is

```
n = 149;
h = 1/(n+1);
A = toeplitz(sparse([1,1],[1,2],[-2,1]/h^2,1,n))/50+...
    toeplitz(sparse(1,2,-1/(2*h),1,n),sparse(1,2,1/(2*h),1,n));
```

We have $\alpha = -1800$, $\nu = 0$, $\eta = -150$, and $\beta = 150$ and after the shift the rectangle $R(A)$ is $[-900, 900] + \mathrm{i}[-150, 150]$. If we compute s as in (4.5) for the values a_m and b_m as in Table 4.1, we get 265, 265, 212, 232, 413, and 1410, for $m = 30$ and the different values of c, and 109, 109, 86, 84, 186, and 300 for $m = 50$. If we now multiply these values for the corresponding degree m, we find out that the smallest evaluation cost ($s \cdot m = 4200$) is attained for $m = 50$ and $c = 11.5$. The strategy of minimizing the expected cost $s \cdot m$ is already used in [2,5,6].

Although the backward error analysis guarantees the required accuracy in *exact arithmetic*, there is a phenomenon that we should take into account: the *hump* phenomenon. We observe that the interpolation error does not always decrease monotonically with the degree of interpolation (see [5, § 4.3]). This behavior can lead to a significant loss of accuracy due to round-off errors. This cannot be predicted by the backward error analysis since it is a pure finite arithmetic side effect. It was already observed in [5] that the hump is related to a bad distribution of the interpolation points with respect to the localization of the eigenvalues. If we consider again the example in this section, we see that the selection of $s = 84$ and $m = 50$ as the one minimizing the cost $s \cdot m$ has the drawback to scale the original rectangle $R(A)$ to $R(s^{-1}A) = [-10.71\text{e}0, 10.71\text{e}0] + \mathrm{i}[-1.786\text{e}0, 1.786\text{e}0]$, while the corresponding interpolation interval is $[-c, c]$ with $c = 11.5$. Therefore, it is not contained into $R(s^{-1}A)$. With the additional constraint that $[-c, c] \subseteq R(s^{-1}A)$, we get that the optimal values are $s = 86$ and $m = 50$ (corresponding to $c = 10$), with a computational cost $s \cdot m = 4300$. We show in Figure 4.2 the two scaled rectangles and corresponding ellipses for the choices $m = 50$, $s = 86$, and $c = 10$ and $m = 50$, $s = 84$, and $c = 11.5$, respectively. In the case that differ-

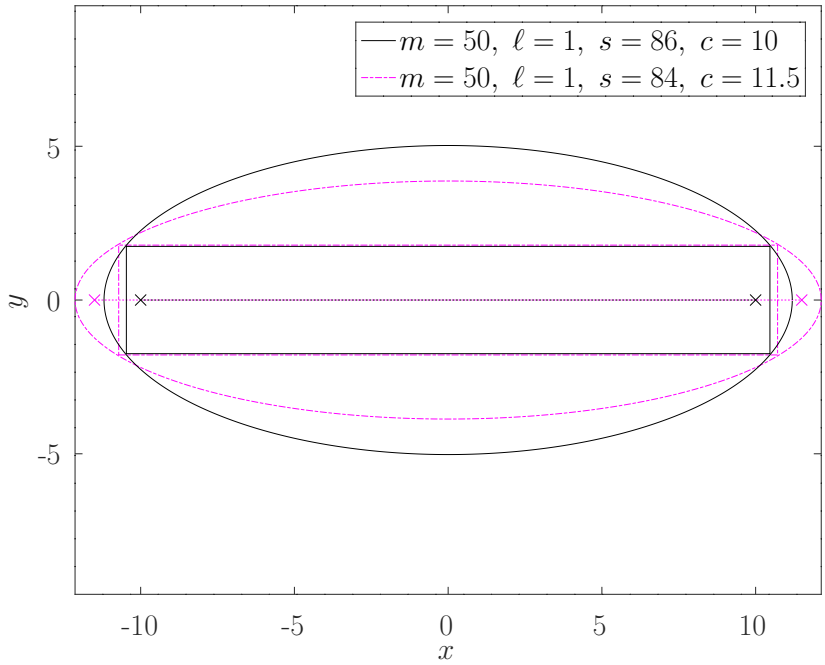

Fig. 4.2 Two scaled rectangles containing the field of values of an advection–diffusion matrix with the corresponding ellipses. The interpolation interval of the magenta dashed-dotted ellipse is not included in the corresponding rectangle and thus has to be excluded.

ent combinations of degrees (m, ℓ, c) lead to the same computational cost $s \cdot m$ and $[-c, c] \subseteq R(s^{-1}A)$, we select the one corresponding to a ratio a_m/b_m as close as possible to the ratio of the corresponding edges of the rectangle $R(A)$.

5 Numerical experiments

5.1 Technical details

For an efficient and reliable implementation of interpolation at Leja–Hermite points, some considerations have to be made. First of all, Leja–Hermite points (both in their standard form (4.6a) and in the complex conjugate form (4.6b)) can be computed once and for all in reference intervals, say $[-2, 2]$ and $\mathrm{i}[-2, 2]$, and later scaled to $[-c, c]$ and $\mathrm{i}[-|c|, |c|]$, respectively. Due to their recursive definition, the proper way to implement the polynomial interpolation is the Newton form

$$p_m(A)v = \sum_{i=0}^{m} \left(d_i \prod_{j=0}^{i-1} (A - z_j I)v \right).$$

It requires an accurate evaluation of the divided differences $\{d_i\}_{i=0}^m$, $d_i = \exp[z_0, z_1, \ldots, z_i]$, for which the algorithm described in [4,17] or the very recent in [24] can be used. When using complex conjugate points for real inputs A and v, Newton evaluation scheme has to be adapted in order to keep real arithmetic, see [22]. Both in the real and the complex case, it is possible to interrupt the scheme before reaching the desired degree m. This feature is called *early termination* and it is usually triggered when the magnitude of the updates is much smaller than the current approximation. Another feature that may help the early termination is a good ordering of the interpolation points. Leja points naturally have such a property, but Leja–Hermite points have the leading set of zeros with multiplicity $\ell + 1$. With the strategy described in [6, § 3.2] we order à la Leja the set $\{z_i\}_{i=\ell}^m$ and add the remaining ℓ zeros at the end of the Newton interpolation process. We refer to [2,5,6] for further details.

In the next sections, we will show the effectiveness of the new approach with respect to the classical backward error analysis based on norms for the truncated Taylor series [2] and interpolation at Leja or Leja–Hermite points [6]. To be fair, all the methods have been limited to the maximum degree polynomial $m = 55$, as it is in [2]. The computation of the parameters θ_m and γ_m for each selected degree m and interval of interpolation $[-c, c]$ was done once and for all. The result is a long table of candidate interpolation polynomials. For each of them, it is possible to compute in advance the corresponding divided differences up to the needed degree.

For each example, we will report the number of iterations (matrix-vector products) for the evaluation of the polynomial $\left(p_m(s^{-1}A)\right)^s v$ as an indication of the overall computational cost. The algorithms share the same idea of scaling the matrix and approximating its exponential using a polynomial. Therefore, the number of matrix-vector products is a reliable indication of their total cost. On the other hand, since we compare the methods on matrices of modest dimension, the measured CPU times are prone to fluctuations and cannot be used to infer the behavior of the algorithms on larger matrices. The cost of approximating the values $\alpha_q(A)$ or $R(A)$ is not reported, since it is negligible for the examples we consider. On the other hand, it is a pre-processing cost common to all the methods which bound the backward error.

5.2 Examples with real Leja–Hermite points

In the first five numerical examples, we consider interpolation at real Leja–Hermite points.

5.2.1 Two-dimensional advection–diffusion matrices

We consider the discretization by second-order finite differences of the advection–diffusion partial differential equation

$$\frac{\partial u}{\partial t} + \boldsymbol{b} \cdot \nabla u = d \nabla^2 u$$

defined in the two-dimensional spatial domain $[0,1]^2$, subject to homogeneous Dirichlet boundary conditions and initial solution $u_0(x,y) = 16x(1-x)y(1-y)$. The discretization is done with 49 inner points and thus $h = 1/50$. The diffusion coefficient is fixed to $d = 1/100$ and the advection term is $\boldsymbol{b} = (b,b)$, $b \geq 0$. The grid Péclet number turns out to be

$$\mathrm{Pe} = \frac{hb}{2d} = b.$$

We first consider the diffusion case ($b = 0$, corresponding to a symmetric matrix). After the shift, the "rectangle" $R(A)$ is the real interval $[-100, 100]$. The 1-norm, together with the values $\alpha_q(A)$ up to $q = 8$, is 100. Table 5.1 contains the results for the truncated Taylor series, for interpolation at pure Leja points based on the 1-norm of the matrix, and for interpolation at Leja–Hermite points based on the field of values of the matrix. We report the scaling parameter s, the degree of interpolation m, the value ℓ, the endpoint c (in exponential notation when rounded) of the symmetric interpolation interval, the values θ_m (for methods based on the norm of the matrix) or γ_m (for the presented method), the expected number of iterations $s \cdot m$, the actual number of iterations due to the early termination criterion and the relative error measured in the 1-norm with respect to the Matlab® R2017b command `expm(A)*v` on the original un-shifted matrix.

Table 5.1 Results for the diffusion case ($b = 0$).

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\|A\|_1$	11	53	53	0	9.3e0	583	495	4.4e-14
L $\|A\|_1$	10	55	0	4.8e0	1.0e1	550	460	4.9e-14
LH $\mathcal{W}(A)$	7	55	30	14.2e0	7.8e0	385	235	1.5e-14

From the table, we see that the expected number of iterations is much smaller with the presented approach (denoted by LH $\mathcal{W}(A)$) than with the methods based on the norm of the matrix (Taylor truncated series, denoted by T $\|A\|_1$ and interpolation at Leja points, denoted by L $\|A\|_1$). This is even more clear from the number of actual iterations: all the three methods benefit a lot from the early termination criterion, but the new method takes less than half of the iterations of the Taylor approach. The final errors are comparable, with a small advantage for the new method.

In the second example we consider $b = 0.25$. The rectangle is $[-100, 100] + \mathrm{i}[-25, 25]$. The values $\alpha_q(A)$ are still constant and equal to 100. Table 5.2 collects the results.

Since the values $\alpha_q(A)$ are the same as in the previous case, it is not surprising that the Taylor method and interpolation at pure Leja points adopt the same strategies as before. The new method selects a different ellipse and turns out to be still the best in terms of both expected and actual number of iterations.

Table 5.2 Results for the advection–diffusion case ($b = 0.25$).

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\|A\|_1$	11	53	53	0	9.3e0	583	495	9.1e-15
L $\|A\|_1$	10	55	0	4.8e0	1.0e1	550	460	1.5e-14
LH $\mathcal{W}(A)$	9	55	4	11	9.3e0	495	315	1.9e-14

The final example in this series is relative to the case $b = 0.5$. The rectangle $R(A)$ is $[-100, 100] + \mathrm{i}[-50, 50]$. The $\alpha_q(A)$ values are still constant and equal to 100. Table 5.3 collects the results.

Table 5.3 Results for the advection–diffusion case ($b = 0.5$).

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\|A\|_1$	11	53	53	0	9.3e0	583	495	9.1e-15
L $\|A\|_1$	10	55	0	4.8e0	1.0e1	550	456	1.5e-14
LH $\mathcal{W}(A)$	11	55	2	9	9.6e0	605	375	2.6e-14

The new approach still takes a considerably smaller number of iterations with respect to the competitors, although the expected number of iterations is larger.

5.2.2 `triw` *matrices*

The next two experiments show the behavior of the new approach in case the rectangle containing the field of values is not skinny. The first matrix we consider is given by the Matlab® command `-gallery('triw',20,4)` which corresponds to a 20×20 matrix with -1 on the main diagonal and -4 on the remaining upper triangular part. The initial vector has components $v_j = \cos(j)$. After the shift with $\mu = -1$, the matrix is nilpotent and A^{20} is the null matrix. Its field of values is contained in the square $R(A) = [-38, 38] + \mathrm{i}[-38, 38]$ and the sequence of the $\alpha_q(A)$ values decreases from $\alpha_1(A) = \|A\|_1 = 76$ to $\alpha_8(A) =$1.63e1. Since this matrix is nilpotent, the truncated Taylor series with degree m at least 20 can compute exactly (in exact arithmetic) the matrix exponential at any sub-step s. This example is taken from [2, Experiment 6].

Table 5.4 Results for the `triw` example, dimension $n = 20$.

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\alpha_7(A)$	2	54	54	0	9.6e0	108	42	3.1e-14
LH $\mathcal{W}(A)$	6	53	53	0	9.2e0	318	109	1.0e-15

From Table 5.4 we see that the Taylor method requires much fewer iterations than the new approach, which, with the choice $c = 0$, is, in fact, the Taylor method too. But the former uses the information coming from the sequence of the $\alpha_q(A)$ values (in particular it uses the value $\alpha_7(A) =$1.88e1) and

manages to select a much smaller number of sub-steps s. The analysis based on the field of values cannot use this information and performs an over-scaling of the matrix. Of course, it would be possible to endow all the candidate interpolation polynomials p_m with the value θ_m associated to the classical backward error analysis based on the norms.

The second matrix is still a `triw` but with dimension 110×110 and the initial vector has constant components equal to 1. The rectangle $R(A)$ is the square $[-218, 218] + \mathrm{i}[-218, 218]$. The sequence of the $\alpha_q(A)$ values decreases from $\alpha_1(A) = \|A\|_1 = 436$ to $\alpha_8(A)$ =1.12e2. This time, the truncated Taylor series of maximum degree $m = 55$ cannot compute the solution up to machine precision.

Table 5.5 Results for the `triw` example, dimension $n = 110$.

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\alpha_8(A)$	12	55	55	0	9.9e0	660	313	3.2e-12
LH $\mathcal{W}(A)$	32	55	55	0	9.7e0	1760	608	9.1e-15

From Table 5.5, we see again that the Taylor method uses the information coming from the sequence of values $\alpha_q(A)$ and selects a number of sub-steps $s = 12$ much smaller than the other method. However, the final error is two orders of magnitude larger due to the hump phenomenon. In this case, the use of the value $\alpha_8(A)$ makes the Taylor method to *under-scale* the matrix.

5.3 Examples with complex conjugate Leja–Hermite points

In the final two numerical examples, we consider interpolation at complex conjugate Leja–Hermite points.

5.3.1 Advection matrix

We consider the one-dimensional discretization by central second-order finite differences of the advection operator ∂_x with periodic boundary conditions on the spatial domain $[0, 1]$. The length of the discretization step is $h = 1/70$. The application vector is the discretization of the initial solution $u_0(x) = 1/(2+\cos(2\pi x))$. The resulting matrix of dimension 70×70 is skew-symmetric. The "rectangle" $R(A)$ is $\mathrm{i}[-70, 70]$ and $\alpha_q(A) = 70$ up to $q = 8$. Table 5.6 collects the results.

Table 5.6 Results for the advection case.

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\|A\|_1$	8	51	51	0	8.8e0	408	304	4.2e-15
LH $\|A\|_1$	9	53	1	8.0e0	8.0e0	477	297	4.6e-15
LH $\mathcal{W}(A)$	6	48	38	11.5	7.0e0	288	246	4.5e-15

The ellipse selected by the new approach corresponds to the interpolation on a set with quite a lot of repeated zeros ($\ell + 1 = 39$). This probably did not help into triggering the early termination criterion. On the other hand, the new method is the most efficient in terms of actual number of iterations.

5.3.2 Schrödinger matrix

In this example, we consider the discretization by second order central finite differences of the free Schrödinger operator $\mathrm{i}\partial_{xx}$ in the one-dimensional spatial domain $[-1, 1]$ with homogeneous Dirichlet boundary conditions. The space step size h is $1/35$, the matrix has dimension 69×69 and is skew-Hermitian. The application vector is the discretization of $u_0(x) = 1/(2 + \cos(2\pi x)) - 1/3$. The "rectangle" $R(A)$ is $\mathrm{i}[-2450, 2450]$. The values $\alpha_q(A)$ are constant and equal to 2450. Table 5.7 collects the results.

Table 5.7 Results for the Schrödinger case.

Method	s	m	ℓ	c	θ_m or γ_m	$s \cdot m$	Act. its.	Rel. err.
T $\|A\|_1$	249	55	55	0	9.9e0	13695	13197	5.1e-11
LH $\|A\|_1$	292	55	1	8.4e0	8.4e0	16060	10220	6.1e-13
LH $\mathcal{W}(A)$	176	55	49	13.9e0	7.3e0	9680	9680	3.5e-13

Once again the new method outperforms the other two methods in terms of the actual number of iterations, although the early termination criterion is not triggered. The chosen scaling parameter s is much smaller with respect to the competitors. Moreover, as already observed in [6], the Taylor method loses more than two orders of magnitude in the relative error. This result, as analyzed in [5, § 4.3] is due to a strong hump phenomenon which affects the Taylor truncated series in this case.

6 Conclusions

In this paper, we overcame the analysis of contour integral expansion of the backward error for the action of the matrix exponential $\exp(A)v$, initially developed in [5]. In particular, we have shown that it is possible to bound the backward error by considering ellipses which enclose the field of values $\mathcal{W}(A)$. The analysis can be performed for any interpolation (or even approximation) polynomial or sets of interpolation polynomials which take the value 1 at 0. We applied it to polynomials interpolating the exponential function at the so-called Leja–Hermite points. For matrices whose field of values has a skinny shape, our numerical experiments show a neat advantage in the actual number of iterations (matrix-vector products) with respect to methods based on the classical expansion into a power series of the backward error, such as the truncated Taylor method [2] and interpolation at Leja [5] and Leja–Hermite points [6].

We briefly highlight the main features and novelties of our approach.

- Differently from the Taylor method, that is a particular case of Hermite interpolation where all the interpolation points are taken in 0, our main priority is to interpolate near the eigenvalues of the matrix. This not only mitigates the hump phenomenon, but it also helps triggering earlier the termination criterion. It is for these reasons that we dedicated special attention to the choice of the interpolation points as it was done in [5,6].
- We introduced a new estimate of the backward error based on the field of values which outperforms the power series expansion used in the Taylor method. Also, we proposed a refinement for the estimation of the rectangle containing the field of values.
- Moreover, we did not base the choice of the interpolant on the mere minimization of $s \cdot m$ as in [2], but we also tried to choose it accordingly with the shape of the field of values.

Therefore, the method is effective for several important matrices arising from the spatial discretization of partial differential equations (diffusion, advection–diffusion, advection, Schrödinger, for instance).

Our numerical examples show that the expected number of iterations $s \cdot m$ is almost always an over-estimate (sometimes a large one) of the actual number of iterations, both with the error expanded as a power series or a contour integral. Moreover, the hump phenomenon turns out to be hard to detect. We would like, in a future work, to address these issues and assemble a method which can profitably take the best from the two backward error analysis and reduce the risk of over- or under-scaling.

References

1. A. H. Al-Mohy and N. J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal. Appl.*, 31(3):970–989, 2009.
2. 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(2):488–511, 2011.
3. L. P. Bos and M. Caliari. Application of modified Leja sequences to polynomial interpolation. *Dolomites Res. Notes Approx.*, 8:66–74, 2015.
4. M. Caliari. Accurate evaluation of divided differences for polynomial interpolation of exponential propagators. *Computing*, 80(2):189–201, 2007.
5. M. Caliari, P. Kandolf, A. Ostermann, and S. Rainer. The Leja method revisited: backward error analysis for the matrix exponential. *SIAM J. Sci. Comput.*, 38(3):A1639–A1661, 2016.
6. M. Caliari, P. Kandolf, and F. Zivcovich. Backward error analysis of polynomial approximations for computing the action of the matrix exponential. *BIT Numer. Math.*, 58(4):907–935, 2018.
7. M. Crouzeix and C. Palencia. The numerical range is a $(1+\sqrt{2})$-spectral set. *SIAM J. Matrix Anal. Appl.*, 38(2):649–655, 2017.
8. A. Frommer, S. Güttel, and M. Schweitzer. Efficient and stable Arnoldi restarts for matrix functions based on quadrature. *SIAM J. Matrix Anal. Appl.*, 35(2):661–683, 2014.
9. S. Gaudreault, G. Rainwater, and M. Tokman. KIOPS: A fast adaptive Krylov subspace solver for exponential integrators. *J. Comput. Phys.*, 372(1):236–255, 2018.

10. S. Güttel. Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitt.*, 36(1):8–31, 2013.
11. N. J. Higham. Estimating the matrix p-norm. *Numer. Math.*, 62:539–555, 1992.
12. N. J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM J. Matrix Anal. Appl.*, 26(4):1179–1193, 2005.
13. N. J. Higham. *Functions of Matrices*. SIAM, Philadelphia, 2008.
14. N. J. Higham and F. Tisseur. A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra. *SIAM J. Matrix Anal. Appl.*, 21(4):1185–1201, 2000.
15. M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.
16. C. R. Johnson. Numerical determination of the field of values of a general complex matrix. *SIAM J. Numer. Anal.*, 15(3):595–602, 1978.
17. A. McCurdy, K. C. Ng, and B. N. Parlett. Accurate computation of divided differences of the exponential function. *Math. Comp.*, 43(168):501–528, 1984.
18. C. B. Moler and C. F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.
19. I. Moret and P. Novati. RD-rational approximation of the matrix exponential operator. *BIT Numer. Math.*, 44:595–615, 2004.
20. J. Niesen and W. M. Wright. Algorithm 919: A Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Trans. Math. Software*, 38(3):1–19, 2012.
21. T. Schmelzer and L. N. Trefethen. Evaluating matrix functions for exponential integrators via Carathéodory–Fejér approximation and contour integrals. *Electron. Trans. Numer. Anal.*, 29:1–18, 2007.
22. H. Tal-Ezer. High degree polynomial interpolation in Newton form. *SIAM J. Sci. Stat. Comput.*, 12(3):648–667, 1991.
23. J. van den Eshof and M. Hochbruck. Preconditioning Lanczos approximations to the matrix exponential. *SIAM J. Sci. Comp.*, 27(4):1438–1457, 2006.
24. F. Zivcovich. Fast and accurate computation of divided differences for analytic functions, with an application to the exponential function. *Dolomites Res. Notes Approx.*, 12:28–42, 2019.