Document downloaded from:

http://hdl.handle.net/10251/192628

This paper must be cited as:

Bader, P.; Blanes Zamora, S.; Casas, F.; Seydaoglu, M. (2022). An efficient algorithm to compute the exponential of skew-Hermitian matrices for the time integration of the Schrodinger equation. Mathematics and Computers in Simulation. 194:383-400. https://doi.org/10.1016/j.matcom.2021.12.002

The final publication is available at

https://doi.org/10.1016/j.matcom.2021.12.002

Copyright Elsevier

Additional Information

arXiv:2103.10132v2 [math.NA] 7 Dec 2021

An efficient algorithm to compute the exponential of skew-Hermitian matrices for the time integration of the Schrödinger equation

Philipp Bader[a], Sergio Blanes[b], Fernando Casas[c], Muaz Seydaoğlu[d,b,*]

[a] *Departament de Matemàtiques, Universitat Jaume I, 12071 Castellón, Spain.*
[b] *Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, E-46022 Valencia, Spain.*
[c] *IMAC and Departament de Matemàtiques, Universitat Jaume I, 12071 Castellón, Spain.*
[d] *Department of Mathematics, Faculty of Art and Science, Muş Alparslan University, 49100, Muş, Turkey.*

Abstract

We present a practical algorithm to approximate the exponential of skew-Hermitian matrices up to round-off error based on an efficient computation of Chebyshev polynomials of matrices and the corresponding error analysis. It is based on Chebyshev polynomials of degrees 2, 4, 8, 12 and 18 which are computed with only 1, 2, 3, 4 and 5 matrix-matrix products, respectively. For problems of the form $\exp(-iA)$, with A a real and symmetric matrix, an improved version is presented that computes the sine and cosine of A with a reduced computational cost. The theoretical analysis, supported by numerical experiments, indicates that the new methods are more efficient than schemes based on rational Padé approximants and Taylor polynomials for all tolerances and time interval lengths. The new procedure is particularly recommended to be used in conjunction with exponential integrators for the numerical time integration of the Schrödinger equation.

Keywords: Matrix exponential, Matrix sine, Matrix cosine, Matrix polynomials, Schrödinger equation.

1. Introduction

Given a skew-Hermitian matrix, $X \in \mathbb{C}^{N\times N}$, $X^H = -X$, we propose in this paper an algorithm to evaluate e^X up to round off accuracy that is more efficient than standard procedures implemented in computing packages for dimensions N up to few hundreds or thousands.

Computing exponentials of skew-Hermitian matrices is very often an intermediate step in the formulation of numerical schemes used for simulating the evolution of different problems in Quantum Mechanics. Thus, suppose one needs to solve numerically the time-dependent Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t}\psi(x,t) = \hat{H}(t)\psi(x,t), \qquad \psi(x,0) = \psi_0(x). \tag{1}$$

Here $\hat{H}(t)$ is in general a time-dependent Hamiltonian operator, $\psi : \mathbb{R}^d \times \mathbb{R} \longrightarrow \mathbb{C}$ is the wave function representing the state of the system, and $\psi_0(x)$ is the initial state.

*Corresponding author
Email addresses: bader@uji.es (Philipp Bader), serblaza@imm.upv.es (Sergio Blanes), casas@uji.es (Fernando Casas), m.seydaoglu@alparslan.edu.tr (Muaz Seydaoğlu)

One possible approach consists in expressing the solution in terms of an orthonormal basis $\{\phi_k(x)\}_{k=1}^{\infty}$ that is truncated up to, say, the first N terms. Then, one has

$$\psi_0(x) = \sum_{k=1}^{N} c_k\, \phi_k(x), \qquad \text{and} \qquad \psi(t,x) = \sum_{k=1}^{N} c_k(t)\, \phi_k(x),$$

where the coefficients $c(t) = (c_1(t), \ldots, c_N(t))^T$ satisfy

$$i\frac{d}{dt}c(t) = H(t)\, c(t), \qquad c(0) = c_0 \in \mathbb{C}^N, \tag{2}$$

and $H(t)$ is a Hermitian matrix with elements $H_{\ell m} = \langle \phi_\ell | \hat{H}(t) | \phi_m \rangle$, $\ell, m = 1, \ldots, N$. One then subdivides the time integration interval in a number of subintervals of length τ, and finally computes approximations $c_k \simeq c(t_k)$ at times $t_k = k\tau$, $k = 1, 2, 3, \ldots$.

Exponential integrators can be used to solve this problem (see [8, 19] and references therein) and they require the computation at each time step of one or several matrix exponentials $\mathrm{e}^{-i\,\tau\, H_k}$, $k = 1, 2, \ldots$, where H_k is a Hermitian matrix depending on $H(t)$ at different times. Although efficient algorithms exist to carry out this task by diagonalizing the constant matrix H_k, we will show that it is indeed possible to compute the exponential in a very efficient way with a different procedure when $\|\tau\, H_k\|$ is not too large . This is typically the situation one encounters when exponential integrators are applied to this class of problems [4].

The goal of this work is thus to present an efficient algorithm for computing e^X, with X a skew-Hermitian matrix, up to round off accuracy with a minimum number of matrix-matrix products. The algorithm is based on Chebyshev polynomials and an efficient procedure to evaluate polynomials of matrices. If $\|X\|$ is large enough, this technique can be combined with scaling-and-squaring. Even then, diagonalizing is only superior when a large number of squarings is necessary.

Since the algorithm can also be used to compute e^{-iA} when A is a Hermitian matrix, just by taking $A = iX$, in the sequel and without loss of generality we address this problem.

Our approach for computing $\mathrm{e}^{-i\,A}$ is based on approximations of the form

$$\mathrm{e}^{-i\,A} \approx P_m(A), \tag{3}$$

where $P_m(y)$ is a polynomial in y that approximates the exponential $\mathrm{e}^{-i\,y}$. Different choices for such $P_m(y)$ are available, namely truncated Taylor or Chebyshev series expansions in an appropriate real interval of y. Rational approximations, like Padé approximants, are also a standard technique to compute the exponential in combination with scaling and squaring [16, 18]. In the autonomous case, when $H(t)$ is constant, this is basically equivalent to solve (2) using a Gauss-Legendre-Runge-Kutta method [12] or a Cayley transform [13].

Specifically, the scaling and squaring technique is based on the property

$$\mathrm{e}^{-iA} = \left(\mathrm{e}^{-iA/2^s}\right)^{2^s}, \quad s \in \mathbb{N}. \tag{4}$$

The exponential $\mathrm{e}^{-iA/2^s}$ is then replaced by a polynomial (or rational) approximation $P_m(A/2^s)$. Both parameters, s and m, are determined in such a way that full machine accuracy is achieved with the minimal computational cost.

An important ingredient in our procedure consists in designing an efficient way to evaluate the approximation P_m. In this respect, the technique we propose can be considered as a direct descent of the procedure presented in [9] for reducing the

number of commutators appearing in different exponential integrators. It was later generalized in [5] to reduce the number of products necessary to compute the Taylor polynomials for approximating the exponential of a generic matrix (see also [6, 26] for a more detailed treatment).

In fact, the theoretical analysis carried out here and supported by numerical experiments performed for different Hermitian matrices A, indicates that our new schemes are more efficient than those based on rational Padé approximants (as used e.g. in MATLAB) or on Taylor polynomials for all tolerances. The algorithm computes the parameter

$$\beta = \|A\|_1$$

as an upper bound to the spectrum of A. As an optional choice, the user can provide upper and lower bounds for the eigenvalues of the matrix A, $E_{\min}$ and $E_{\max}$, and this allows one to consider a shift for reducing the overall cost. Then, the algorithm automatically selects the most efficient polynomial approximation for a prescribed error tolerance.

Although the algorithms based on Taylor polynomial approximations and the use of scaling-and-squaring constructed in [6, 26] can of course be applied also here, it turns out that in the particular case of skew-Hermitian matrices (with purely imaginary eigenvalues) it is more convenient instead to apply a similar procedure based on Chebyshev polynomials. Here only polynomials of degree $m = 2, 4, 8, 12$ and 18 are considered, since the number of matrix-matrix products is minimized in those particular cases. Although higher degrees could in principle be taken, it turns out that applying the scaling-and-squaring technique to lower degree polynomials renders a similar or higher performance.

In many cases, when solving different quantum mechanical or quantum control problems [4] one ends up with a real and symmetric matrix, $A^T = A \in \mathbb{R}^{N\times N}$, so that

$$\mathrm{e}^{-iA} = \cos(A) - i\sin(A),$$

and we also provide an algorithm for computing $\cos(A)$ and $\sin(A)$ simultaneously only involving products of real symmetric matrices. This new algorithm is more efficient than the approach (3) since that scheme usually requires products of complex matrices, and other existing algorithms for the simultaneous computation of the matrix sine and cosine [2, 27]. The squaring (4) (also involving products of complex matrices) is then replaced by the double angle formulae

$$\cos(2A) = 2\cos^2(A) - I = I - 2\sin^2(A), \qquad \sin(2A) = 2\sin(A)\cos(A),$$

so that only two products of real symmetric matrices per squaring are required.

In [7] an algorithm for approximating $\mathrm{e}^{-iA}v$ for any real symmetric matrix A and any complex vector v was proposed. It is based on the idea of splitting and only requires matrix-vector products Av in such a way that the real and imaginary parts of $\mathrm{e}^{-iA}v$ are approximated in a different way, with a considerable saving in the computational cost with respect to the usual Chebyshev approximation. Here, by contrast, we focus on problems where the actual computation of e^{-iA} for any Hermitian matrix is required.

The plan of the paper is the following. In section 2 we analyze the approximation of the exponential by Taylor and Chebyshev polynomials and by Padé approximants as well as their error bounds. In section 3 we obtain explicitly the Chebyshev polynomials of the degree previously chosen and for the parameters that ensure the error bound previously studied, and next we present the algorithms to evaluate these polynomials with a reduced number of products. The algorithm for the case of a real-symmetric matrix A is also considered. Section 4 contains numerical experiments illustrating the performance of the new methods and some future lines of research are enumerated in the final Section 5.

2. Polynomial approximations

Assume that $P_m(y)$ is a mth degree polynomial (or a rational function) approximating the function $\mathrm{e}^{-i\,y}$. Then, the error is bounded (in Euclidean norm) as

$$\|P_m(A) - \mathrm{e}^{-i\,A}\| \leq \max_{j=0,1,\ldots,N-1} |P_m(E_j) - \mathrm{e}^{-i\,E_j}|$$

in terms of the real eigenvalues $E_0, \ldots, E_{N-1}$ of the Hermitian matrix A. If the spectrum $\sigma(A) = \{E_0, \ldots, E_{N-1}\}$ is contained in an interval of the form $[E_{\min}, E_{\max}]$, then

$$\|P_m(A) - \mathrm{e}^{-i\,A}\| \leq \sup_{E_{\min} \leq y \leq E_{\max}} |P_m(y) - \mathrm{e}^{-i\,y}|.$$

The quantities $E_{\max}$ and $E_{\min}$ can be estimated in different ways depending on the particular problem (see e.g. [20]). Once they have been determined, by introducing the quantities

$$\alpha = \frac{E_{\max} + E_{\min}}{2}, \qquad \beta = \frac{E_{\max} - E_{\min}}{2}, \qquad \text{and} \qquad \overline{A} = A - \alpha I, \tag{5}$$

it is clear that the spectrum of the shifted operator $\overline{A}$ is contained in an interval centered at the origin, namely $\sigma(\overline{A}) = \{E_0 - \alpha, \ldots, E_{N-1} - \alpha\} \subset [-\beta, \beta]$, so that

$$\mathrm{e}^{-i\,A} = \mathrm{e}^{-i\,\alpha}\,\mathrm{e}^{-i\,\beta(\overline{A}/\beta)}, \tag{6}$$

with $\sigma(\overline{A}/\beta) \subset [-1, 1]$.

If the bounds $E_{\min}$ and $E_{\max}$ cannot be estimated in a convenient way, one can always take $\beta = \|A\|_1$, so that $\sigma(A) \leq \beta$, and no shift is considered.

In any event, and without loss of generality, our problem consists now in approximating $\mathrm{e}^{-i\,A}$ for a Hermitian matrix A with $\sigma(A) \subset [-\beta, \beta]$ by means of $P_m(A)$. In that case,

$$\frac{\|P_m(A) - \mathrm{e}^{-i\,A}\|}{\|\mathrm{e}^{-i\,A}\|} = \|P_m(A) - \mathrm{e}^{-i\,A}\| \leq \epsilon_m(\beta), \tag{7}$$

where

$$\epsilon_m(\theta) := \sup_{-\theta \leq y \leq \theta} |P_m(y) - \mathrm{e}^{-i\,y}| \tag{8}$$

and $\|\mathrm{e}^{-i\,A}\| = 1$.

2.1. Taylor polynomial approximation

An upper bound for the error estimate (8) of the mth degree Taylor polynomial

$$P_m^T(y) \equiv \sum_{k=0}^{m} \frac{(-i)^k}{k!} y^k \tag{9}$$

approximating $\mathrm{e}^{-i\,y}$ can be obtained by computing the Lagrange form of the remainder in the Taylor series expansion:

$$|P_m^T(y) - \mathrm{e}^{-i\,y}| = \frac{1}{(m+1)!} |\mathrm{e}^{-i\xi}(-iy)^{m+1}| = \frac{1}{(m+1)!} |y|^{m+1}$$

for $\xi \in (0, y)$ so that, from eq. (8),

$$\epsilon_m^T(\theta) := \frac{\theta^{m+1}}{(m+1)!}. \tag{10}$$

Therefore, $P_m^T(A)$ is guaranteed to approximate $\mathrm{e}^{-i\,A}$ up to round-off error as long as $\beta \leq \theta$ with θ such that $\epsilon_m^T(\theta) \leq u = 2^{-53}$. We collect in Table 1 the largest

Table 1: θ values for Taylor and Chebyshev polynomials of degree m that can be computed with π matrix-matrix products to approximate e^{-iA} with A skew-Hermitian and guaranteeing that $\epsilon_m(\theta) \leq u = 2^{-53}$. The θ value in the column $m = 15+$ corresponds to the polynomial of degree 16 built in [26] that approximates the Taylor expansion up to order 15 with 4 products (see section 5 for details).

m :	2	4	8	12	15+	18
π :	1	2	3	4	4	5
Taylor pol. :	8.73E-6	1.67E-3	0.0699	0.336	0.709	1.147
Chebyshev pol. :	1.38E-5	2.92E-3	0.1295	0.636		2.212

θ verifying this restriction for the values of m considered in this work. As stated before, only polynomials of degree $m \leq 18$ will be employed in practice.

Remark: Notice that we can write the polynomial function in the exponential form $P_m^T(\theta) = \mathrm{e}^{-i\,(\theta+\Delta\theta)}$ where $\Delta\theta = \mathcal{O}(\theta^{m+1})$, so condition $|P_m^T(\theta) - \mathrm{e}^{-i\,\theta}| \leq 2^{-53}$ for $\theta \sim \mathcal{O}(1)$ implies that $|\Delta\theta| \sim 2^{-53}$. However, when backward error analysis is considered one looks for the largest value of θ such that $\frac{\Delta\theta}{\theta} \leq 2^{-53}$ so different values for θ are obtained (smaller values when $\theta < 1$ and greater values when $\theta > 1$).

2.2. Chebyshev polynomial approximation

The mth degree truncation of the Chebyshev series expansion of $\mathrm{e}^{-i\,y}$ in the interval $y \in [-\theta, \theta]$ reads

$$P_{m,\theta}^C(y) := J_0(\theta) + 2\sum_{k=1}^{m}(-i)^k J_k(\theta)\, T_k(y/\theta), \tag{11}$$

in terms of the Bessel function of the first kind $J_k(t)$ [1, formula 9.1.21] and the kth Chebyshev polynomial $T_k(x)$ generated from the recursion [23, section 3.11]

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \qquad k \geq 1 \tag{12}$$

with $T_0(x) = 1$, $T_1(x) = x$.

At least three estimates for $\epsilon_m(\theta)$ may be considered when dealing with Chebyshev polynomial approximations. According with the analysis in [22, section III.2.1], one can take

$$\epsilon_m^{C_1}(\theta) := 4\left(\mathrm{e}^{1-\theta^2/(2m+2)^2}\,\frac{\theta}{2m+2}\right)^{m+1}. \tag{13}$$

On the other hand, in [29, Theorem 8.2] it is shown that

$$\max_{|y|\leq 1}\left|e^{\theta y} - P_{m,\theta}^C(y)\right| \leq \frac{2M}{\rho^n(\rho-1)} = \epsilon_m^{C_2}(\theta), \tag{14}$$

where $M = \max_{z\in\mathcal{E}_\rho}\left|e^{\theta z}\right| = e^{\frac{\theta}{2}(\rho+1/\rho)}$, and $\mathcal{E}_\rho$ denotes the Bernstein ellipse in the complex plane [29, chapter 8],

$$\mathcal{E}_\rho = \left\{ z \in \mathbb{C} \;\middle|\; z = \frac{1}{2}(r + r^{-1}),\ r = \rho\,\mathrm{e}^{i\phi},\ -\pi \leq \phi \leq \pi \right\}.$$

Here, ρ is any positive number with $\rho > 1$, and the optimal value that minimizes the right hand side of (14) has to be computed numerically for each choice of θ.

Finally, one can also take the tail of the whole Chebyshev series expansion as an upper bound of the error, i.e.,

$$\|P_{m,\theta}^C(A) - \mathrm{e}^{-i\,A}\| \leq \left\| \sum_{k=m+1}^{\infty} 2(-i)^k J_k(\theta)\, T_k(y/\theta) \right\| \leq \sum_{k=m+1}^{\infty} 2\,|J_k(\theta)| \equiv \epsilon_m^{C_3}(\theta). \tag{15}$$

We have evaluated the three estimates (13)-(15) for the relevant degrees m and compared with the observed behaviour of the corresponding polynomials. From these computations we conclude that the bound (15) exhibits the sharpest result, i.e., larger values of θ for all m considered. Thus, in particular, for $m = 18$ bound (13) leads to $\theta = 1.8843$, bound (14) gives $\theta = 1.939$, whereas bound (15) provides the largest value $\theta = 2.212$. The corresponding values for θ obtained with (15) are also collected in Table 1. Notice that these values are almost twice larger than those associated to Taylor approximations.

In practice, we have constructed the Chebyshev polynomial approximations for each pair (m, θ) specified in Table 1 as in [15] (which is in fact equivalent to Eq. (11))

$$P^C_{m,\theta}(y) = \frac{1}{2}c_0 + \sum_{k=1}^{m} c_k T_k(y/\theta), \tag{16}$$

with

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{\mathrm{e}^{-i\theta y}\, T_k(y)}{\sqrt{1-y^2}}\, dx \tag{17}$$

and all the calculations have been carried out with 30 digits of accuracy. In Figure 1 we show both the absolute error $|P^C_{m,\theta}(y) - \mathrm{e}^{-iy}|$ for $(m = 18, \theta = 2.212)$ and the value of $u \approx 1.11\text{E-}16$. Notice how the error is always smaller than u for the whole interval $y \in [-\theta, \theta]$.

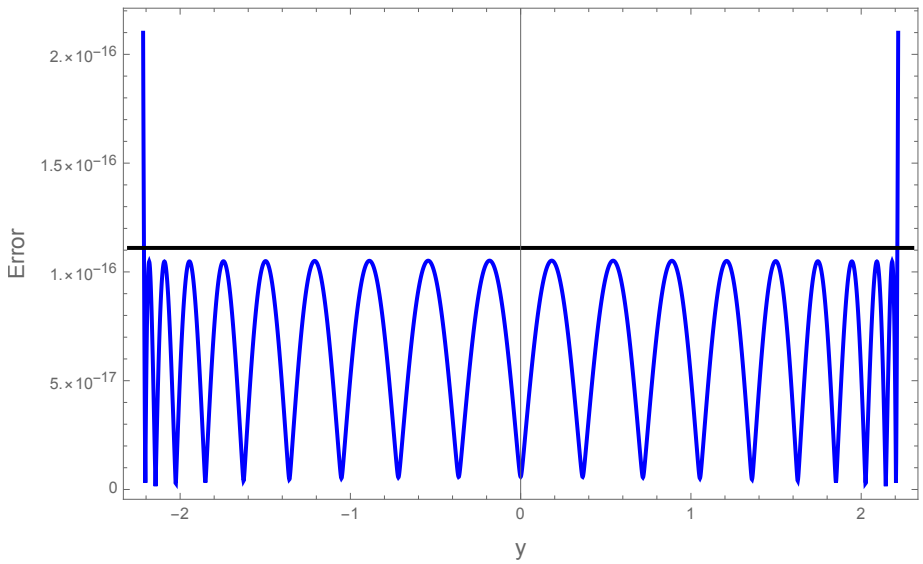

Figure 1: Absolute error $|P^C_{m,\theta}(y) - \mathrm{e}^{-iy}|$ for $(m = 18, \theta = 2.212)$ (blue) and the value of $u \approx 1.11\text{E-}16$ (black). The error is always smaller than u for $y \in [-\theta, \theta]$.

2.3. Padé approximations

Most popular computing packages such as MATLAB (`expm`) and *Mathematica* (`MatrixExp`) use Padé approximants (in combination with scaling-and-squaring) to compute numerically the exponential of a generic matrix [16, 18].

Diagonal $[m/m]$ Padé approximants are of the form

$$r_m(-i\,A) = p_m(-iA)\big[p_m(iA)\big]^{-1}, \tag{18}$$

Table 2: Values of θ for diagonal Padé approximants of the highest order $2m$ that are computed with π products (we take the computation of the inverse of a matris as 4/3 products).

m :	2	3	5	7	9	13
π :	$2+\frac{1}{3}$	$3+\frac{1}{3}$	$4+\frac{1}{3}$	$5+\frac{1}{3}$	$6+\frac{1}{3}$	$7+\frac{1}{3}$
Padé	2.4007E-3	2.715E-2	2.803E-1	0.8983	1.833	4.316

where

$$p_m(x) = \sum_{j=0}^{m} \frac{(2m-j)!m!}{(2m)!(m-j)!} \frac{x^j}{j!}, \tag{19}$$

and they verify that $r_m(-i\,A) = \mathrm{e}^{-i\,A} + \mathcal{O}(A^{2m+1})$. In practice, the evaluation of $p_m(-i\,A)$ and $p_m(i\,A)$ is carried out so as to keep the number of matrix products at a minimum. The previous notation $\mathcal{O}(A^n)$ is defined next, since it will be helpful in the sequel.

Definition 1. *We say that a given function $f(A)$ of the matrix A satisfies $f(A) = \mathcal{O}(A^n)$ if it can be written as a convergent Taylor expansion, $f(A) = \sum_{k=n}^{\infty} c_k A^k$, for $\|A\| < \alpha$, with α a positive constant.*

For skew-Hermitian matrices, we can use, instead of the generic backward error bounds obtained e.g. in [17], an error estimate of the form (7) with $\epsilon_m(\theta)$ in (8) replaced by its upper bound:

$$\|r_m(-i\,A) - \mathrm{e}^{-i\,A}\| \leq \left| \sum_{k=2m+1}^{\infty} d_k \theta^k \right| \leq \sum_{k=2m+1}^{\infty} |d_k|\,\theta^k \equiv \epsilon_m^P(\theta). \tag{20}$$

In practice, for a given m, we have computed $s \equiv \sum_{k=2m+1}^{2000} |d_k| y^k$ and determined the largest y for which $s \leq u = 2^{-53}$. This value is taken then as the bound θ. The values for θ are collected in Table 2 for those m for which the diagonal Padé approximant can be computed with the minimum number of products. The function `expm` in MATLAB uses the corresponding bound θ obtained from relative backward error with a cost of 2,3,4,5 and 6 products, respectively, in addition to one matrix inverse (we take the cost of one inverse as 4/3 products[1]). In order to compare with our methods under the same conditions, we have used the function `expm` from MATLAB but taking the θ values from Table 2. One should notice that the corresponding backward error bounds are smaller up to $m = 7$.

3. Evaluating Chebyshev polynomial approximations with a reduced number of products

Our next goal is to reproduce the Chebyshev polynomial approximations considered in section 2.2 with a reduced number of matrix products in comparison with the *de facto* standard Paterson–Stockmeyer method for polynomial evaluation. Since the technique has been already explained in detail in the context of Taylor polynomials approximating the exponential of a generic matrix in [6] (see also [25] for a closely related procedure), here we only collect its most salient features and refer to [6] for a comprehensive treatment.

[1]For a $N \times N$ matrix, it requires one LU factorization at the cost of 1/3 products plus N solutions of upper and lower triangular systems by forward and backward substitution at the cost of one product.

Essentially, the idea is a modification of a procedure designed in [9] to reduce the number of commutators appearing in exponential integrators, and consists in taking a sequence of products of the form

$$
\begin{aligned}
&A_0 := I, \qquad A_1 := A \\
&A_2 := z_0 A_0 + z_1 A_1 + \big(x_1 A_0 + x_2 A_1\big)\big(x_3 A_0 + x_4 A_1\big) \\
&A_4 := z_2 A_0 + z_3 A_1 + z_4 A_2 + \big(x_5 A_0 + x_6 A_1 + x_7 A_2\big)\big(x_8 A_0 + x_9 A_1 + x_{10} A_2\big) \\
&A_8 := z_5 A_0 + \cdots + z_8 A_4 + \big(x_{11} A_0 + \cdots + x_{14} A_4\big)\big(x_{15} A_0 + \cdots + x_{18} A_4\big), \\
&\quad \vdots
\end{aligned}
\tag{21}
$$

to rewrite any polynomial $P_m(A)$ as $P_m(A) = \sum_{k\geq 0} \alpha_k A_k$. Proceeding in this way there might be both redundancies in the coefficients (for instance, it suffices to take $A_2 = A_1 A_1$ since any polynomial of degree two can be writen in terms of A_0, A_1 and A_2) and also not enough parameters to reproduce some powers in A (e.g. to compute $P_7(A) = A^7$). For this reason, one includes new terms of the form, say,

$$(w_0 A_0 + w_1 A_1)(w_3 A_0 + w_4 A_1 + w_5 A_2),$$

in the procedure for computing A_k, $k > 2$, so that one has additional parameters. The price to be paid is of course that it is necessary to evaluate some extra products.

Concerning the particular class of polynomials and degrees we are interested in, $P_m(A)$ with $m = 2, 4$ can be obtained with just 1 and 2 matrix products, in a similar way as the Paterson–Stockmeyer technique.

Degree $m = 2$. The quadratic Chebyshev polynomial with $\theta = 1.38\text{E-}5$ can be trivially computed with one product, and is given by

$$A_2 = A^2, \qquad P^C_{2,\theta}(A) = \alpha_0 I + \alpha_1 A + \alpha_2 A_2,$$

with

$$
\begin{aligned}
&\alpha_0 = 0.99999999999999999999998, \qquad \alpha_1 = -0.99999999997619500000001\, i \\
&\alpha_2 = -0.49999999999920650000000.
\end{aligned}
$$

Degree $m = 4$. The Chebyshev polynomial of degree four with $\theta = 2.92\text{E-}3$ can be computed with two products as follows:

$$
\begin{aligned}
&A_2 = A^2, \qquad A_4 = A_2(x_1 A + x_2 A_2), \\
&P^C_{4,\theta}(A) = \alpha_0 I + \alpha_1 A + \alpha_2 A_2 + A_4,
\end{aligned}
$$

with

$$
\begin{aligned}
&\alpha_0 = 0.99999999999999999997, \qquad \alpha_1 = -0.99999999999981067844\, i \\
&\alpha_2 = -0.49999999999994320353, \qquad x_1 = 0.16666657785001893215\, i \\
&x_2 = 0.04166664890333648869.
\end{aligned}
$$

Although we report here 20 digits for the coefficients, they can be in fact determined with arbitrary accuracy.

The situation is more involved, however, for higher degrees. We next collect the results for the Chebyshev polynomial approximations to the exponential of degrees $m = 8, 12$ and 18. Although more values of m could be considered, it turns out that these polynomials can be constructed with only 3, 4 and 5 products, respectively.

Degree $m = 8$. As is the case with Taylor polynomials [6], the following sequence allows one to evaluate $P_8(A) \equiv P^C_{8,\theta}(A)$, with $\theta = 0.1295$:

$$\begin{aligned} &A_2 = A^2, \qquad\qquad A_4 = A_2(x_1 A + x_2 A_2), \\ &A_8 = (x_3 A_2 + A_4)(x_4 I + x_5 A + x_6 A_2 + x_7 A_4), \\ &P^C_{8,0.1295}(A) = \alpha_0 I + \alpha_1 A + \alpha_2 A_2 + A_8. \end{aligned} \tag{22}$$

Notice that this is a particular example of the sequence (21) with some of the coefficients fixed to zero to avoid redundancies. The parameters x_i, α_i are determined such that $P^C_{8,0.1295}(A)$ agrees with the corresponding expression (16). One has 10 parameters to solve 9 nonlinear equations and this results in two families of solutions depending on a free parameter, x_1. All solutions provide the same polynomial (if exact arithmetic is considered), and we have chosen x_1 to (approximately) minimize the 1-norm of the vector of parameters. The corresponding coefficients in (22) for the Chebyshev polynomial are given by

$$\begin{aligned} &x_1 = 431/4000, && x_2 = -0.02693906873598870733\, i, \\ &x_3 = 0.66321004441662438593\, i, && x_4 = 0.54960853911436015786\, i, \\ &x_5 = 0.16200952846773660904, && x_6 = -0.01417981805211804396\, i, \\ &x_7 = -0.03415953916892111403, && \alpha_0 = 0.99999999999999999928, \\ &\alpha_1 = -0.99999999999999233987\, i, && \alpha_2 = -0.13549409636220703066. \end{aligned}$$

Degree $m = 12$. Here the situation is identical to what happens with Taylor polynomials approximating e^X for a generic matrix [6]: although polynomials up to degree 16 could in principle be constructed with 4 products by applying the sequence (21), in practice the highest degree we are able to get is $m = 12$ with the following sequence:

$$\begin{aligned} &A_2 = A^2, && A_3 = A_2 A, \\ &B_1 = a_{0,1} I + a_{1,1} A + a_{2,1} A_2 + a_{3,1} A_3, && B_2 = a_{0,2} I + a_{1,2} A + a_{2,2} A_2 + a_{3,2} A_3, \\ &B_3 = a_{0,3} I + a_{1,3} A + a_{2,3} A_2 + a_{3,3} A_3, && B_4 = a_{0,4} I + a_{1,4} A + a_{2,4} A_2 + a_{3,4} A_3, \\ &A_6 = B_3 + B_4^2 \\ &P^C_{12,0.636}(A) = B_1 + (B_2 + A_6) A_6. \end{aligned} \tag{23}$$

This ansatz has four families of solutions with three free parameters. A judicious choice leading to a small value for $\sum_{i,j} |a_{i,j}|$ is:

$$\begin{aligned} &a_{0,1} = -6.26756985350202252845, && a_{1,1} = 2.52179694712098096140\, i, \\ &a_{2,1} = 0.05786296656487001838, && a_{3,1} = -0.07766686408071870344\, i, \\ &a_{0,2} = 0, && a_{1,2} = 1.41183797496250375498\, i, \\ &a_{2,2} = 0, && a_{3,2} = -0.00866935318616372016\, i, \\ &a_{0,3} = 2.69584306915332564689, && a_{1,3} = -1.35910926168869260391\, i, \\ &a_{2,3} = -0.09896214548845831754, && a_{3,3} = 0.01596479463299466666\, i, \\ &a_{0,4} = 0, && a_{1,4} = 0.13340427306445612526\, i, \\ &a_{2,4} = 0.02022602029818310774, && a_{3,4} = -0.00674638241111650999\, i. \end{aligned}$$

Degree $m = 18$. We have been able to write the Chebyshev polynomial approximation of degree $m = 18$ with 5 products. This is done by expressing $P^C_{18,2.212}(A)$ as the product of two polynomials of degree 9, that are further decomposed into polynomials of lower degree. The polynomial is evaluated through the following

sequence:

$$
\begin{aligned}
&A_2 = A^2, \qquad A_3 = A_2 A, \qquad A_6 = A_3^2,\\
&B_1 = a_{0,1} I + a_{1,1} A + a_{2,1} A_2 + a_{3,1} A_3,\\
&B_2 = b_{0,1} I + b_{1,1} A + b_{2,1} A_2 + b_{3,1} A_3 + b_{6,1} A_6,\\
&B_3 = b_{0,2} I + b_{1,2} A + b_{2,2} A_2 + b_{3,2} A_3 + b_{6,2} A_6,\\
&B_4 = b_{0,3} I + b_{1,3} A + b_{2,3} A_2 + b_{3,3} A_3 + b_{6,3} A_6,\\
&B_5 = b_{0,4} I + b_{1,4} A + b_{2,4} A_2 + b_{3,4} A_3 + b_{6,4} A_6,\\
&A_9 = B_1 B_5 + B_4,\\
&P^C_{18,2.212}(A) = B_2 + (B_3 + A_9) A_9,
\end{aligned}
\tag{24}
$$

with coefficients

$$
\begin{aligned}
&a_{0,1} = 0, && a_{1,1} = 3/25,\\
&a_{2,1} = -0.00877476096879703859\, i, && a_{3,1} = -0.00097848453523780954,\\
&b_{0,1} = 0, && b_{1,1} = -0.66040840760771318751\, i,\\
&b_{2,1} = -1.09302278471564897987, && b_{3,1} = 0.25377155817710873323\, i,\\
&b_{6,1} = 0.00054374267434731225, && b_{0,2} = -2.58175430371188142440,\\
&b_{1,2} = -1.73033278310812419209\, i, && b_{2,2} = -0.07673476833423340755,\\
&b_{3,2} = -0.00261502969893897079\, i, && b_{6,2} = -0.00003400011993049304,\\
&b_{0,3} = 2.92377758396553673559, && b_{1,3} = 1.44513300347488268510\, i,\\
&b_{2,3} = 0.12408183566550450221, && b_{3,3} = -0.01957157093642723948\, i,\\
&b_{6,3} = 0.00002425253007433925, && b_{0,4} = 0,\\
&b_{1,4} = 0, && b_{2,4} = -0.12395369585828313148 0\, i,\\
&b_{3,4} = -0.01120269484108559237 3, && b_{6,4} = -0.00001236724053825989 6\, i.
\end{aligned}
$$

3.1. The case of real symmetric matrices

In the particular case when A is a real symmetric matrix, we can write

$$
\mathrm{e}^{-iA} = \cos(A) - i \sin(A) \tag{25}
$$

and it is possible to construct algorithms for approximating the real symmetric matrices $\cos(A)$ and $\sin(A)$ simultaneously by means of a reduced number of products of real symmetric matrices, as shown in [2, 27].

The polynomial (16) can be decomposed into real and imaginary parts,

$$
P^C_{m,\theta}(A) = c^C_{m,\theta}(A) - i s^C_{m,\theta}(A),
$$

with

$$
c^C_{m,\theta}(A) = \mathrm{Re}(P^C_{m,\theta}(A)), \qquad s^C_{m,\theta}(A) = -\mathrm{Im}(P^C_{m,\theta}(A))
$$

and the goal is to compute exactly $c^C_{m,\theta}(A)$ with a reduced number of products. Then, by using all computations carried out in this process, one also obtain approximations $S^C_{m,\theta}(A)$ to the imaginary part in such a way that it is a polynomial of degree $k > m$ such that $S^C_{m,\theta}(A) = s^C_{m,\theta}(A) + \mathcal{O}(A^{m+1})$. Taking into account that

$$
\begin{aligned}
\left\| \left(c^C_{m,\theta}(A) + i S^C_{m,\theta}(A) \right) - \mathrm{e}^{-i\,A} \right\| &\leq \left\| P^C_{m,\theta}(A) - \mathrm{e}^{-i\,A} \right\| + \left\| s^C_{m,\theta}(A) - S^C_{m,\theta}(A) \right\| \\
&\leq \epsilon^{C_3}_m(\theta) + \left\| s^C_{m,\theta}(A) - S^C_{m,\theta}(A) \right\|
\end{aligned}
\tag{26}
$$

we need to check if $\|s^C_{m,\theta}(A) - S^C_{m,\theta}(A)\| \leq \sum_{\ell=m+1}^{k} |c_\ell|\, |\theta|^\ell \leq 2^{-53}$ for these values of m and θ. If this is not the case, one has to find the maximum value of ϑ such that $\|s^C_{m,\theta}(\vartheta) - S^C_{m,\theta}(\vartheta)\| \leq 2^{-53}$ and to take this value as the value for θ, i.e. the largest value of $\|A\|$ that guarantees an error smaller that roundoff. If the value obtained for ϑ is considerably smaller than the value of θ obtained for the cosine

function, we will look for a new and more accurate approximation to $s^C_{m,\theta}(A)$ by taking e.g. one extra product in the numerical scheme.

For example, we can compute simultaneously $c^C_{4,\theta}(A)$ and $s^C_{4,\theta}(A)$, i.e. $P^C_{4,\theta}(A)$, with three products of real symmetric matrices. However, with the same number of products we can also compute $c^C_{5,\theta}(A)$, $s^C_{5,\theta}(A)$, i.e. $P^C_{5,\theta}(A)$ that has a larger value of θ, so that we only consider this last case which is evaluated as follows (in this case $S^C_{m,\theta}(A) = s^C_{m,\theta}(A)$).

Degree $m = 5$. The polynomial $c^C_{5,\theta}(A)$, $\theta = 1.17\text{E-}2$, is computed with 2 products by taking $B = A^2$ as

$$c^C_{5,\theta}(A) = \alpha_0 I + \alpha_1 B + \alpha_2 B^2,$$

with

$$\begin{aligned} \alpha_0 &= 0.99999999999999988866, \qquad \alpha_1 = -0.49999999998536031183 \\ \alpha_2 &= 0.04166638147997997916, \end{aligned}$$

whereas for evaluating $s^C_{5,\theta}(A)$ only one additional product is required:

$$s^C_{5,\theta}(A) = A(z_0 I + z_1 B + z_2 B^2)$$

with

$$\begin{aligned} z_0 &= 0.99999999999999994433, \qquad z_1 = -0.16666666666341340086 \\ z_2 &= 0.00833328580219952161. \end{aligned}$$

Degree $m = 8$. The polynomial $c^C_{8,\theta}(A)$, $\theta = 0.1295$, is computed with 3 products as:

$$\begin{aligned} &B = A^2, \qquad B_2 = B^2, \qquad B_4 = B_2(x_1 B + x_2 B_2), \\ &c^C_{8,\theta}(A) = \alpha_0 I + \alpha_1 B + \alpha_2 B_2 + B_4, \end{aligned}$$

with

$$\begin{aligned} \alpha_0 &= 0.99999999999999999928, \qquad \alpha_1 = -0.49999999999999787210, \\ \alpha_2 &= 0.04166666666565156615, \qquad x_1 = -0.00138888871939942118, \\ x_2 &= 0.00002479003614491668, \end{aligned}$$

and $s^C_{8,\theta}(A)$ is approximated with error $\mathcal{O}(A^9)$ with one additional product by

$$S^C_{8,\theta}(A) = A(z_0 I + z_1 B + z_2 B_2 + z_3 c^C_{8,\theta}(A))$$

with

$$\begin{aligned} z_0 &= 0.85721768947064012466, \qquad z_1 = -0.09527551139590047256, \\ z_2 &= 0.00238406908730568850, \qquad z_3 = 0.14278231052935221530. \end{aligned}$$

Notice that the condition

$$|s^C_{8,\theta}(\vartheta) - S^C_{8,\theta}(\vartheta)| = x_2\, z_3\, \vartheta^9 \le 2^{-53}$$

is satisfied only for $\vartheta \le 0.06807$. This is a significant reduction with respect to θ and for this reason we look for an approximation which involves one extra product. With five products, however, it is possible to exactly compute the polynomials for $m = 9$, with $\theta = 0.2143$.

Degree $m = 9$. The polynomial $c^C_{9,\theta}(A)$, $\theta = 0.2143$, is computed with 4 products in the same way:

$$\begin{aligned} &B = A^2, \qquad B_2 = B^2, \qquad B_3 = B_2 B, \qquad B_4 = B_3 B, \\ &c^C_{9,\theta}(A) = \alpha_0 I + \alpha_1 B + \alpha_2 B_2 + \alpha_3 B_3 + \alpha_4 B_4, \end{aligned}$$

with

$$\begin{aligned} &\alpha_0 = 0.99999999999999989168, \qquad \alpha_1 = -0.49999999999988173685, \\ &\alpha_2 = 0.04166666664600636231, \qquad \alpha_3 = -0.00138888762558264513, \\ &\alpha_4 = 0.00002477005498155486, \end{aligned}$$

and $s^C_{9,\theta}(A)$ can be computed with one additional product:

$$s^C_{9,\theta}(A) = A(z_0 I + z_1 B + z_2 B_2 + z_3 B_3 + z_4 B_4),$$

with

$$\begin{aligned} &z_0 = -0.999999999999999945837, \qquad z_1 = 0.166666666666643012068, \\ &z_2 = -0.008333333330440664914, \qquad z_3 = 0.000198412554024823435, \\ &z_4 = -2.75257852630876250884 \cdot 10^{-6}. \end{aligned}$$

Degree $m = 16$. We can compute $c^C_{16,\theta}(A)$, $\theta = 1.5867$, with only four products as follows. We first take $B = A^2$, so that $c^C_{16,\theta}$ is indeed a polynomial of degree eight in B, that can be computed with only three products in a similar way to $P^C_{8,0.1295}(A)$ with the sequence

$$\begin{aligned} &B = A^2, \qquad B_2 = B^2, \qquad B_4 = B_2(x_1 B + x_2 B_2), \\ &B_8 = (x_3 B_2 + B_4)(x_4 I + x_5 B + x_6 B_2 + x_7 B_4), \\ &c^C_{16,\theta}(A) = \alpha_0 I + \alpha_1 B + \alpha_2 B_2 + B_8, \end{aligned} \tag{27}$$

where

$$\begin{aligned} &x_1 = 1/100, \qquad x_2 = -0.00008035854055477845, \\ &x_3 = -0.10743065643419630630, \qquad x_4 = -0.12491372919298427513, \\ &x_5 = 0.00130085397953037838, \qquad x_6 = -0.00001633763177694857, \\ &x_7 = 7.13215089463286614820 \cdot 10^{-6}, \qquad \alpha_0 = 0.99999999999999999530, \\ &\alpha_1 = -0.49999999999999969795, \qquad \alpha_2 = 0.02824710274181773 4721. \end{aligned}$$

With two extra products we can approximate the matrix $s^C_{16,\theta}(A)$:

$$\begin{aligned} C_{24} &= (z_5 I + z_5 B + z_6 B_2 + z_7 B_4 + z_8\, c_{16,\theta}(A)) B_4, \\ S^C_{16,\theta}(A) &= A\big(z_0 I + z_1 B + z_2 B_2 + z_3 B_4 + z_4\, c_{16,\theta}(A) + C_{24}\big), \end{aligned} \tag{28}$$

with

$$\begin{aligned} &z_0 = 33/50, \qquad z_1 = 0.00333333333335438849, \\ &z_2 = -0.00583333333345309522, \qquad z_3 = 0.02773310749258735833, \\ &z_4 = 0.33999999999999886261, \qquad z_5 = -0.00034915267907803119, \\ &z_6 = 4.19573036995827807213 \cdot 10^{-6}, \qquad z_7 = -2.63931697420854364428 \cdot 10^{-6}, \\ &z_8 = -3.00240279002259730782 \cdot 10^{-6}. \end{aligned}$$

In this way $S^C_{16,\theta}(A)$ is a polynomial of degree 25 in A where the condition

$$\|s^C_{16,\theta}(\vartheta) - S^C_{16,\theta}(\vartheta)\| \le 2^{-53}$$

is satisfied for $\vartheta \le 0.7563$. One extra product (7 products in total) suffices to exactly compute $s^C_{16,\theta}(A)$ (and then to keep the value of θ). We do not show this scheme because, as we will see, with 7 products one can find an improved approximation.

Degree $m = 24$. The same strategy can be applied to the polynomials $c^C_{24,\theta}(A)$ and $s^C_{24,\theta}(A)$, with $\theta = 4.5743$. Thus, $c^C_{24,\theta}(A)$ is computed by taking $B = A^2$ and computing the corresponding polynomial of degree 12 with only four additional products as previously:

$$\begin{array}{ll} D = A^2, \qquad D_2 = D^2 & D_3 = D_2 D, \\ B_1 = a_{0,1}I + a_{1,1}D + a_{2,1}D_2 + a_{3,1}D_3, & B_2 = a_{0,2}I + a_{1,2}D + a_{2,2}D_2 + a_{3,2}D_3, \\ B_3 = a_{0,3}I + a_{1,3}D + a_{2,3}D_2 + a_{3,3}D_3, & B_4 = a_{0,4}I + a_{1,4}D + a_{2,4}D_2 + a_{3,4}D_3, \\ D_6 = B_3 + B_4^2 & \\ c^C_{24,\theta}(A) = B_1 + (B_2 + D_6)D_6, & \end{array} \tag{29}$$

with

$$\begin{array}{ll} a_{0,1} = 0.39272620931352327385, & a_{1,1} = -0.08760637124112618048, \\ a_{2,1} = 0.01962064507143601071, & a_{3,1} = -0.00013421604022829771, \\ a_{0,2} = 1/5, & a_{1,2} = -0.54235659842328961975, \\ a_{2,2} = 679/100000, & a_{3,2} = -0.00002902999756981724, \\ a_{0,3} = 0.68566773555140770915, & a_{1,3} = -0.02578520551577453856, \\ a_{2,3} = 0.00019815665089300452, & a_{3,3} = -1.10083330495602029332 \cdot 10^{-6}, \\ a_{0,4} = 0, & a_{1,4} = -0.03931944346958836562, \\ a_{2,4} = 0.00017839382197658767, & a_{3,4} = -1.06908694221941432625 \cdot 10^{-6}, \end{array} \tag{30}$$

whereas with two extra products the following approximation to $s^C_{24,\theta}(A)$ is obtained

$$\begin{aligned} C_{48} &= (z_6 I + z_7 D + z_8 D_2 + z_9 D_3 + z_{10} D_6 + z_{11}\, c_{24,\theta}(A))c_{24,\theta}(A), \\ S^C_{24,\theta}(A) &= A\big(z_0 I + z_1 D + z_2 D_2 + z_3 D_3 + z_4 D_6 + z_5\, c_{24,\theta}(A) + C_{48}\big) \end{aligned} \tag{31}$$

with

$$\begin{array}{ll} z_0 = -0.01238438326981811663, & z_1 = -0.06180067679127220638, \\ z_2 = 0.00046275599640408615, & z_3 = -9.92990416300441584763 \cdot 10^{-6}, \\ z_4 = 1.26307934615308708610, & z_5 = 9.10439014880980346565 \cdot 10^{-15}, \\ z_6 = 0.14610549096048524519, & z_7 = 0.00087697762149660844, \\ z_8 = 4.12092186281469998191 \cdot 10^{-6}, & z_9 = 2.23743615053828476204 \cdot 10^{-8}, \\ z_{10} = 0.00033015662857238333, & z_{11} = -2.40537107176685232329 \cdot 10^{-7}. \end{array}$$

The approximation $S^C_{24,\theta}(A)$ given by (31) is a polynomial of degree 48 in A verifying the condition

$$\|s^C_{24,\theta}(\vartheta) - S^C_{24,\theta}(\vartheta)\| \le 2^{-53}$$

for $\vartheta \le 2.1556$, which is smaller than the value of θ for this case but larger than the value of θ for $m = 16$ that requires the same number of products, and for this reason the previous scheme is not considered in practice.

One extra product suffices to exactly compute $s^C_{24,\theta}(A)$ (and then to keep the value of θ) as follows

$$\begin{aligned} D_5 &= D_2(z_{11} D_2 + z_{12} D_3), \\ C_{24} &= (z_6 I + z_7 D + z_8 D_2 + z_9 D_3 + D_5 + z_{13} D_6)(D_6 + z_{10} D), \\ s^C_{24,\theta}(A) &= A\big(z_0 I + z_1 D + z_2 D_2 + z_3 D_3 + z_4 D_5 + z_5\, c_{24,\theta}(A) + C_{24}\big) \end{aligned} \tag{32}$$

In this case we can solve all the equations (including the corresponding to A^{25}). We

Table 3: θ values for Chebyshev polynomials of degree m that can be computed with π products of symmetric real matrices to simultaneously compute the sine and cosine matrix functions to approximate e^{-iA} with A a real symmetric matrix (in parenthesis it is indicated the cost and maximum value of $\|A\|$ when the Chebyshev polynomial for the sine function is approximated with a higher degree polynomial).

m :	5	8	9	16	24
π :	3	4	5	(6)7	(7)8
Chebyshev pol. :	1.17E-2	0.068	0.214	(0.7563)1.587	(2.1556)4.574

have now one free parameter and one solution is:

$$z_0 = 2.85247650396873609664, \quad z_1 = -0.23838922984354509797,$$
$$z_2 = 0.01254735251131974478, \quad z_3 = -0.00003184984233834954,$$
$$z_4 = -7.91411934357932811110, \quad z_5 = -0.45584956828766694538,$$
$$z_6 = -2.34944723110594310069, \quad z_7 = -0.34315650534099675485,$$
$$z_8 = 0.00379529409295014610, \quad z_9 = -0.00001509312002244718,$$
$$z_{10} = -17/1000, \quad z_{11} = 7.68145795118100472945 \cdot 10^{-9},$$
$$z_{12} = -2.71896175810263278764 \cdot 10^{-11} \quad z_{13} = 0.45584956828766694538$$

Table 3 collects the values of θ for the selected approximations to the sine and consine functions and their cost in terms of products of real symmetric matrices.

3.2. The algorithm

In previous sections we have computed a number of Chebyshev polynomials of different degrees for some values of θ that provide errors below roundoff when approximating e^{-iy} for $y \in [-\theta, \theta]$. These polynomials are computed by applying a particular sequence in order to reduce the number of products. To approximate $\mathrm{e}^{-i\,A}$ one has to select the most appropriate polynomial that leads to an error below the prescribed tolerance at the smallest computational cost.

The user has to provide the matrix A and, as an optional input, the values for $E_{\min}$ and $E_{\max}$. The algorithm then computes β and determines the normalized matrix A. If $E_{\min}$ and $E_{\max}$ are not given, the algorithm takes $\beta = \|A\|_1$ as an upper bound to $|E_{\min}|$ and $|E_{\max}|$ and no shift is considered.

Next, the algorithm determines the most efficient method (among the list of available schemes) leading to the desired result: it chooses the cheapest method with error bounds below round off error.

If none of the methods provides an error below tolerance, then the scaling and squaring technique is used. In that case, the value of θ for the polynomial of the highest degree 18 (or 24 for the trigonometric matrix functions) is taken to obtain the number of squarings that will be necessary.

As an illustration, suppose one is interested in computing e^{-iA}, where A is a complex Hermitian matrix such that $E_{\min}$, $E_{\max}$ are not known and, in addition

1. $\|A\|_1 = 8$. (i) With Padé one checks that $\|A/2\|_1 = 4 < 4.316$ and the exponential is computed with one scaling and the approximant with $m = 13$ that involves 6 products and one inverse ($8+1/3$ products in total). (ii) With Taylor we have $\|A/2^3\|_1 = 1 < 1.1468$, the exponential is computed with three scalings and the polynomial with $m = 18$, requiring 5 products (for a total of 8 products). Finally, (iii) with Chebyshev, since $\|A/2^2\|_1 = 2 < 2.212$, the exponential is computed with two scalings and 5 products (for a total of 7 products).
2. $\|A\|_1 = 0.1$. (i) With Padé, the exponential is computed with 3 products and one inverse; (ii) Taylor requires 4 products, and (iii) Chebyshev needs only 3 products.

3. $\|A\|_1 = 0.0025$. (i) With Padé, the exponential is computed with 2 products and one inverse; (ii) Taylor requires 3 products, and (iii) Chebyshev needs 2 products.

Notice that, whereas the reduction in computation is roughly the same in all cases, the relative saving increases as the norm of the matrix is smaller.

This strategy has been implemented as a MATLAB code which is freely available for download at the website [14], together with some notes and examples illustrating the whole procedure.

4. Numerical examples

In this section we report on two numerical experiments carried out by applying the previous algorithm based on Chebyshev polynomials. We also compare their main features with Taylor polynomials and Padé approximants.

Example 1: A high dimensional Rosen–Zener model.. This is a generalization of the well known Rosen–Zener model for a quantum system of two levels [24] which is closely related to the problem analyzed in [21]. The corresponding Schrödinger equation (2) for the evolution operator (in the interaction picture) is

$$\begin{cases} U'(t) = -\,\mathrm{i}\, H(t)\, U(t)\,, & t \in (t_0, t_f)\,, \\ U(t_0) = I, \end{cases} \tag{33a}$$

where the time-dependent Hamiltonian reads, after normalization,

$$H(t) = f_1(t)\, \sigma_1 \otimes I + f_2(t)\, \sigma_2 \otimes R \in \mathbb{C}^{d\times d}\,, \qquad d = 2\,k\,, \tag{33b}$$

with the identity matrix $I \in \mathbb{R}^{k\times k}$, Pauli matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{33c}$$

and

$$R = \mathrm{tridiag}\big(1, 0, 1\big) \in \mathbb{R}^{k\times k}\,. \tag{33d}$$

We take in particular

$$f_1(t) = V_0\, \cos(\omega\, t)\, \big(\cosh\big(\tfrac{t}{T_0}\big)\big)^{-1}, \qquad f_2(t) = -\,V_0\, \sin(\omega\, t)\big(\cosh\big(\tfrac{t}{T_0}\big)\big)^{-1}, \tag{33e}$$

with $\omega = 5$, $d = 20$, $V_0 = 2$. We then integrate from $t_0 = -4T_0$ until the final time $t_f = 4\,T_0$ for $T_0 = 1$ and determine numerical approximations, $U_{\mathrm{app}}(t_f, t_0)$ at $t = t_f$ for different time step sizes $\tau = \frac{t_f - t_0}{M}$; (a reference solution $U_{\mathrm{ref}}(t_f, t_0)$ is computed numerically to high accuracy).

In this example we illustrate the performance of the new algorithm as applied to two different exponential integrators: (i) the well-known 2nd-order exponential midpoint rule

$$U_{n+1} = \exp\Big(-i\tau H(t_n + \frac{\tau}{2})\Big), \qquad n = 0, 1, 2, \ldots, M-1$$

and (ii) the 4th-order commutator-free Magnus integrator given by

$$U_{n+1} = \exp\Big(-i\tau(\beta H_1 + \alpha H_2)\Big)\exp\Big(-i\tau(\alpha H_1 + \beta H_2)\Big),$$

where $H_i = H(t_n + c_i\tau), i = 1, 2$, and

$$c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}, \qquad \alpha = \frac{1}{4} + \frac{\sqrt{3}}{6}, \qquad \beta = \frac{1}{4} - \frac{\sqrt{3}}{6}.$$

(See [3, 10, 11] for more details of this scheme as for other higher order methods of the same class). The exponential matrix is computed in all cases with Padé approximants and the new algorithm based on Chebyshev polynomials. Since the results obtained with Taylor polynomials lie in between both of them, they are not shown in the figures for clarity). We compute

$$\tilde{U}_h = U_M\, U_{M-1} \cdots U_2 U_1,$$

and measure the 2-norm of the error, $\|\tilde{U}_h - U_{\rm ref}(t_f, t_0))\|$ for different values of τ. The total cost is taken as the sum of the number of the matrix-matrix products that are required for the calculation of $U_1, \cdots, U_M$, and we depict the error as a function of this total number of matrix-matrix product evaluated by each procedure. Figure 2 shows the corresponding results obtained by new procedure based on Chebyshev (expmC) and Padé approximants (expmP) for the exponential midpoint rule (top) and the 4th-order commutator-free Magnus integrator (bottom). We see that the relative saving in the computational cost is similar in both cases but the improvement in the accuracy increases with the order of the method.

Notice that the accuracy improves when the time step τ decreases so, the number of exponentials increases, but the cost to compute each exponential can decrease because $\|\tau H_k\|$ takes smaller values. The slope of the curves is then higher than expected from the order of the numerical integrator used.

Example 2: The Walker–Preston model.. This constitutes a standard model for a diatomic molecule in a strong laser field [30]. The system is described by the one-dimensional Schrödinger equation (in units such that $\hbar = 1$)

$$i\frac{\partial}{\partial t}\psi(x,t) = \left(-\frac{1}{2\mu}\frac{\partial^2}{\partial x^2} + V(x) + f(t)x\right)\psi(x,t), \tag{34}$$

with $\psi(x,0) = \psi_0(x)$. Here $V(x) = D\,(1-\mathrm{e}^{-\alpha x})^2$ is the Morse potential and $f(t)x = A\cos(\omega(t))x$ accounts for the laser field. As an initial condition, we take the ground state of the Morse potential

$$\psi_0(x) = \sigma\, \exp\left(-(\gamma - \frac{1}{2})\alpha x\right)\, \exp(-\gamma\, \mathrm{e}^{-\alpha x}), \tag{35}$$

where $\gamma = 2D/\omega_0$, $\omega_0 = \alpha\sqrt{2D/\mu}$, and σ is a normalizing constant.

We define the wave function ψ in a certain domain $x \in [x_0, x_N]$ that is subdivided into N parts of length $\Delta x = (x_N - x_0)/N$ with $x_i = x_0 + i\Delta x$, and then the vector $u(t) \in \mathbb{C}^N$ with components $u_i = (\Delta x)^{1/2}\psi(x_{i-1}, t),\ i = 1, \ldots, N$, is formed.

If second-order central differences are applied to discretize the equation in space and periodic boundary conditions are considered, one ends up with the differential equation

$$i\frac{du}{dt} = H(t)u = (T + B(t))u, \qquad u(0) = u_0$$

with

$$T = \frac{N^2}{2\mu(x_N - x_0)^2}\begin{pmatrix} 2 & -1 & & & 1 \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ 1 & & & -1 & 2 \end{pmatrix}$$

and $B(t) = \mathrm{diag}\big(V(x_1) + f(t)x_1, \ldots, V(x_N) + f(t)x_N\big)$. Notice that H is a real symmetric matrix, $H^T = H \in \mathbb{R}^{N\times N}$, so that

$$\exp\left(-i\tau H(t_n + \tau/2)\right) = \cos\left(\tau H(t_n + \tau/2)\right) - i\sin\left(\tau H(t_n + \tau/2)\right). \tag{36}$$

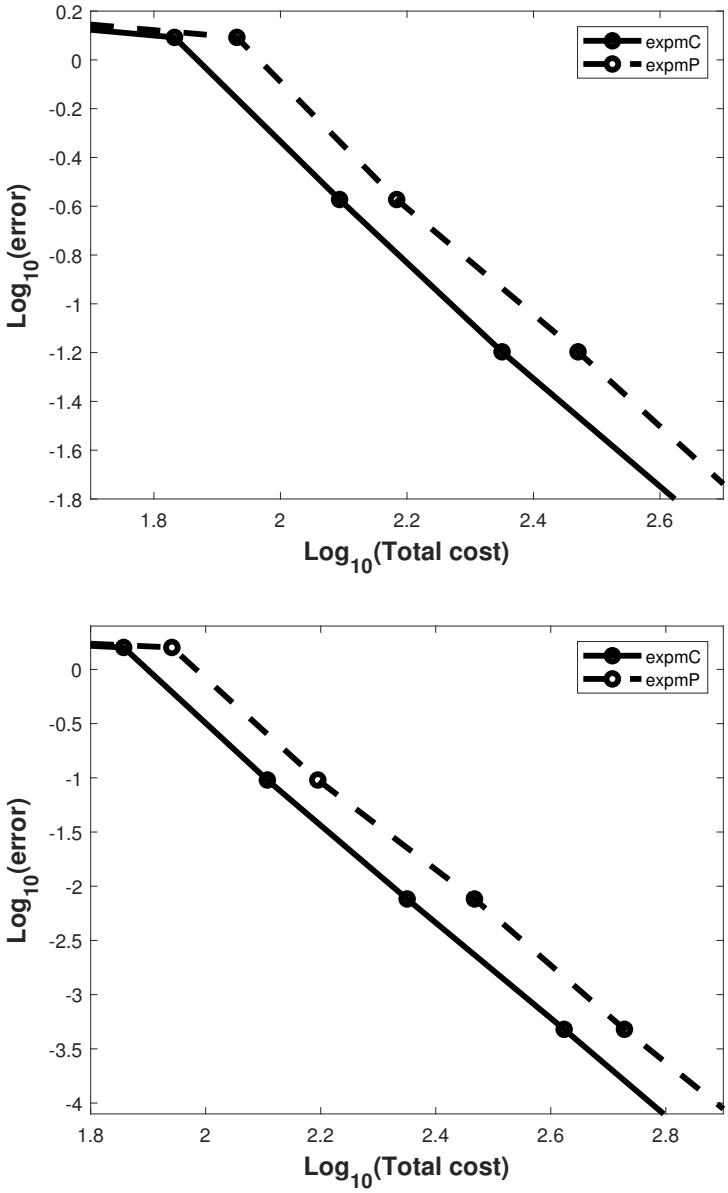

Figure 2: 2-norm error in the unitary matrix evolution at the final time versus the cost (measured as the number of matrix-matrix products required to compute the exponentials at each step) for Example 1: (top) results for the second order exponential midpoint rule and (bottom) results for the fourth order commutator-free exponential Magnus integrator.

Moreover, we can take

$$E_{\min} = \min_{1\le j\le N} B(t)_{jj}, \qquad E_{\max} = \frac{2N^2}{\mu(x_N - x_0)^2} + \max_{1\le j\le N} B(t)_{jj}$$

and so we shift the original matrix according with Eq. (6). For our experiments we take $x \in [-0.8, 4.32]$, the interval is subdivided into $N = 64$ parts of length $\Delta x = 0.08$, and the parameters are chosen as follows (in atomic units): $\mu = 1745$, $D = 0.2251$ and $\alpha = 1.1741$ (corresponding to the HF molecule). Concerning the interaction with the laser field, we take $A = 0.011025$ and the laser frequency

$\omega = 0.01787$.

As before, to check the performance of the different procedures, we compute the 2-norm error in the evolution matrix solution at the final time $t_f = \frac{2\pi}{\omega}$. To do that, we compare with a reference solution computed with high accuracy. The total cost of each procedure is measured as the total number of matrix-matrix products required to approximate the matrix cosine and sine for the total integration interval. In this way we get Figure 3, where the results achieved by Chebyshev approximations (denoted by 'cosmsinmC') and Padé approximants ('cosmsinmP', obtained with the algorithm of [2]) are collected. The top diagram corresponds to the 2nd-order exponential mid-point rule and the bottom graph is obtained with the 4th-order commutator-free Magnus integrator. Here again, the new algorithm based on Chebyshev polynomials leads to more accurate results with a reduced computational cost.

5. Conclusions and future work

We have presented an algorithm to approximate the exponential of skew-Hermitian matrices based on an improved computation of Chebyshev polynomials of matrices and the corresponding error analysis. For problems of the form $\exp(-iA)$, when A is a real and symmetric matrix, an improved version is presented that computes the sine and cosine of A with a reduced number of products of real and symmetric matrices. In both cases, the new procedures turn out to be more efficient than schemes based on rational Padé approximants or Taylor polynomials for all tolerances and time interval lengths.

The Chebyshev methods presented in this paper can be further improved along different lines that will be explored in our future work:

- As we have seen, with only three products it is possible to evaluate most polynomials to order eight (this is, in fact, the highest degree one can reach with three products). With four products one can build polynomials of degree sixteen, but there are not enough free parameters to obtain the Taylor and Chebyshev polynomials approximating the exponential. For this reason, we have limited ourselves here to polynomials of order twelve, which can be obtained with four products. On the other hand, in [26] a polynomial of degree 16 is presented in terms of only 4 products that coincides with the Taylor expansion up to order 15 (this method is denoted in Table 1 as $m = 15+$). In this way, with the computational cost as the method of degree $m = 12$, it provides a larger value for θ that is even slightly larger that the value of the Chebyshev polynomial of degree 12. The same procedure can of course be carried out with Chebyshev polynomials: one could construct a polynomial of degree 16 that coincides with the Chebyshev polynomial up to degree 15 and analyze whether this new polynomial has a larger value of θ. Notice that the procedure is largely similar to the search of polynomials $S^C_{m,\theta}$ that coincide with $s^C_{m,\theta}$. With five products it is also possible to build a polynomial of degree 24 that approximates the Chebyshev polynomial up to order 21, and we expect an improvement with respect to the result obtained for $m = 18$, in the same way as in [26] for the Taylor polynomial.

- One could also build a new set of methods aimed to be used with different accuracies, and in particular in single precision. From the error bound formulas for the chosen values of m, the new values for θ have to be obtained and then the corresponding Chebyshev polynomials of degree m have to be obtained that will be then computed with a reduced number of products.

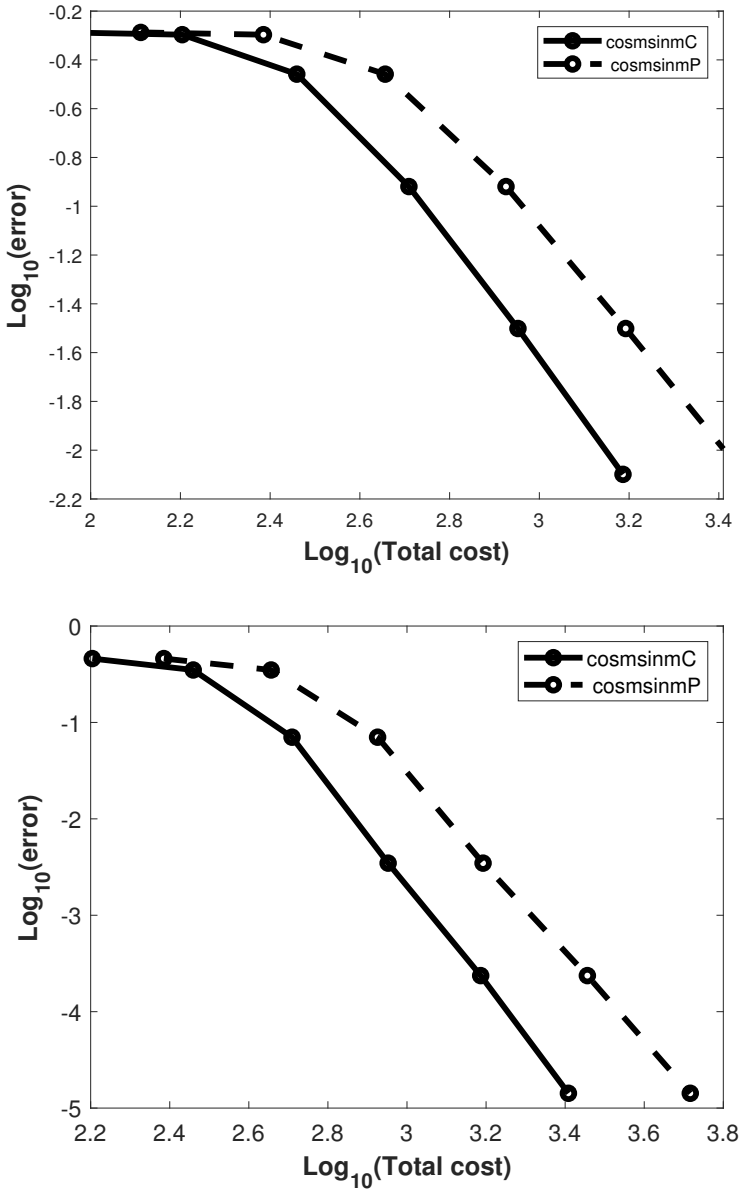

Figure 3: 2-norm error in the unitary matrix evolution at the final time versus the cost (measured as the number of matrix-matrix products required to compute the exponentials at each step) for the real symmetric matrix H of Example 2: (top) results for the second order exponential mid-point rule and (bottom) results for the fourth order commutator-free exponential Magnus integrator.

- When lower accuracies are desired then the preservation of unitarity is also lost to such accuracy. It is well known that diagonal Padé methods preserve unitarity unconditionally and one can look for similar rational Chebyshev approximations to analyze the preservation of unitarity as well as to reduce the cost of these schemes. Rational Chebyshev approximations have been successfully used in [28] to compute the action of the exponential of skew-Hermitian matrices on vectors.

- Finally, there are of course a number of efficient procedures for the diagonalization of Hermitian or skew-Hermitian matrices that might be also employed for evaluating the matrix exponentials required for the application to exponential integrators to certain classes of differential equations. In that case the norm of the matrices involved is usually quite small (since they involve the step size of the integrator) and thus our algorithms are particularly well suited for this purpose. In any case, a future line of research consists in determining precisely under which circumstances related with the size and norm of the matrix the algorithms presented here are competitive with other procedures based on direct diagonalization.

Acknowledgements

SB and FC have been supported by Ministerio de Ciencia e Innovación (Spain) through project PID2019-104927GB-C21 (AEI/FEDER, UE). The work of MS has been funded by the Scientific and Technological Research Council of Turkey (TUBITAK) with grant number 1059B191802292. SB and FC would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme "Geometry, compatibility and structure preservation in computational differential equations", when work on this paper was undertaken. This work was been additionally supported by EPSRC grant number EP/R014604/1. The authors wish to thank the referee for his/her detailed list of comments and suggestions which were most helpful to improve the presentation of the paper.

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover, 1965.

[2] A.H. Al-Mohy, N.J. HIgham, and S.D. Renton, *New algorithms for computing the sine and cosine separately or simultaneously*, SIAM J. Sci. Comput., 37 (2015), A456-A487.

[3] A. Alvermann and H. Fehske, *High-order commutator-free exponential time-propagation of driven quantum systems*, J. Comput. Phys., 230 (2011), pp. 5930–5956.

[4] T. Auckenthaler, M. Bader, T. Huckle, A. Spörl, and D. K. Waldherr, *Matrix exponentials and parallel prefix computation in a quantum control problem*, Parallel Computing 36 (2010), pp. 359–369.

[5] P. Bader, S. Blanes, and F. Casas, *An improved algorithm to compute the exponential of a matrix.* arXiv:1710.10989 [math.NA], 2017.

[6] P. Bader, S. Blanes, and F. Casas, *Computing the matrix exponential with an optimized Taylor polynomial approximation*, Mathematics, 7 (2019), p. 1174.

[7] S. Blanes, F. Casas, and A. Murua, *An efficient algorithm based on splitting for the time integration of the Schrödinger equation*, J. Comput. Phys., 303 (2015), pp. 396–412.

[8] S. Blanes, F. Casas, J. Oteo, and J. Ros, *The Magnus expansion and some of its applications*, Phys. Rep., 470 (2009), pp. 151–238.

[9] S. Blanes, F. Casas, and J. Ros, *High order optimized geometric integrators for linear differential equations*, BIT, 42 (2002), pp. 262–284.

[10] S. Blanes, F. Casas, and M. Thalhammer, *High-order commutator-free quasi-Magnus exponential integrators for non-autonomous linear evolution equations*, Comput. Phys. Comm., 220 (2017), pp. 243–262.

[11] S. Blanes and P. Moan, *Fourth- and sixth-order commutator-free Magnus integrators for linear and non-linear dynamical systems*, Appl. Numer. Math., 56 (2006), pp. 1519–1537.

[12] L. Dieci, R. D. Russell, and E. S. van Vleck, *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM J. Numer. Anal., 31, No. 1 (1994), pp. 261–281.

[13] F. Diele, L. Lopez, and R. Peluso, *The Cayley transform in the numerical solution of unitary differential systems*, Appl. Numer. Math., 8 (1998), pp. 317–334.

[14] Geometric Integration Research Group. Available online: `http://www.gicas.uji.es/Research/ExpSkewHermitian.html`

[15] A. Gil, J. Segura, and N. Temme, *Numerical Methods for Special Functions*, SIAM, 2007.

[16] N. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193.

[17] N. Higham, *Functions of Matrices*, SIAM, 2008.

[18] N. Higham and A. Al-Mohy, *Computing matrix functions*, Acta Numerica, 19 (2010), pp. 159–208.

[19] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numerica, 19 (2010, pp. 209–286.

[20] T. Huang and R. Rau, *A simple estimation for the spectral radius of (block) H-matrices*, J. Comput. Appl. Math, 177 (2005), pp. 455–459.

[21] E. Kyoseva, N. Vitanova, and B. Shore, *Physical realization of coupled Hilbert-space mirrors for quantum-state engineering*, J. Modern Optics, 54 (2007), p. 2237.

[22] C. Lubich, *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*, European Mathematical Society, 2008.

[23] F. Olver, D. Lozier, R. Boisvert, and C. Clark, *NIST Handbook of Mathematical Functions*, Cambridge University Press,, 2010.

[24] N. Rosen and C. Zener, *Double Stern–Gerlach experiment and related collision phenomena*, Phys. Rev., 40 (1932), pp. 502–507.

[25] J. Sastre, *Efficient evaluation of matrix polynomials*, Linear Algebra Appl., 539 (2018), pp. 229–250.

[26] J. Sastre, J. Ibáñez, and E. Defez, *Boosting the computation of the matrix exponential*, Appl. Math. Comput., 340 (2019), pp. 206–220.

[27] M. Seydaoğlu, P. Bader, S. Blanes, and F. Casas, *Computing the matrix sine and cosine simultaneously with a reduced number of products*, Appl. Numer. Math., 163 (2021), pp. 96–107.

[28] R.B. Sidje, *Expokit: a software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.

[29] L. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, 2013.

[30] R. Walker and K. Preston, *Quantum versus classical dynamics in the treatment of multiple photon excitation of the anharmonic oscillator*, J. Chem. Phys., 67 (1977).