Document downloaded from:

http://hdl.handle.net/10251/99868

This paper must be cited as:

Bader, P.; Blanes Zamora, S.; Seydaoglu, M. (2015). The scaling, splitting, and squaring method for the exponential of perturbed matrices. SIAM Journal on Matrix Analysis and Applications. 36(2):594-614. doi:10.1137/14098003X

The final publication is available at

http://doi.org/10.1137/14098003X

Copyright Society for Industrial and Applied Mathematics

Additional Information

Document downloaded from:

http://hdl.handle.net/10251/99868

This paper must be cited as:

Bader, P.; Blanes Zamora, S.; Seydaoglu, M. (2015). The scaling, splitting, and squaring method for the exponential of perturbed matrices. SIAM Journal on Matrix Analysis and Applications. 36(2):594-614. doi:10.1137/14098003X

The final publication is available at

http://doi.org/10.1137/14098003X

Copyright Society for Industrial and Applied Mathematics

Additional Information

THE SCALING, SPLITTING AND SQUARING METHOD FOR THE EXPONENTIAL OF PERTURBED MATRICES

PHILIPP BADER*, SERGIO BLANES†, AND MUAZ SEYDAOĞLU‡

Abstract. We propose splitting methods for the computation of the exponential of perturbed matrices which can be written as the sum $A = D+\varepsilon B$ of a sparse and efficiently exponentiable matrix D with sparse exponential e^D and a dense matrix εB which is of small norm in comparison with D. The predominant algorithm is based on scaling the large matrix A by a small number 2^{-s}, which is then exponentiated by efficient Padé or Taylor methods and finally squared in order to obtain an approximation for the full exponential. In this setting, the main portion of the computational cost arises from dense-matrix multiplications and we present a modified squaring which takes advantage of the smallness of the perturbed matrix B in order to reduce the number of squarings necessary. Theoretical results on local error and error propagation for splitting methods are complemented with numerical experiments and show a clear improvement over existing methods when medium precision is sought.

Key words. matrix exponential, scaling and squaring method, splitting method, Padé approximation, backward error analysis

AMS subject classifications. 65F30, 65F60

1. Introduction.

The efficient computation of matrix exponentials has been extensively considered in the literature and the *scaling and squaring method* is perhaps the most widely used method for matrices of dimension $n \times n$ with n as large as a few hundred (see [9, 15, 18] and references therein). For example, Matlab and Mathematica compute numerically the exponential of matrices using this method where highly efficient algorithms for general matrices exist [1, 7, 9, 10].

Given $A \in \mathbb{C}^{n\times n}$, the method is based on the property

$$e^A = \left(e^{A/2^s}\right)^{2^s} = \underbrace{\left(\cdots\left(e^{A/2^s}\right)^{2}\cdots\right)^{2}}_{s-\text{times}}, \tag{1.1}$$

where typically $e^{A/2^s}$ is replaced by a polynomial approximation (e.g. a mth-order Taylor method, $T_m(A/2^s)$) or a rational approximation (e.g. an $2m$th-order diagonal Padé method, $r_{2m}(A/2^s)$) [9, 10, 17]. The optimal choice of both s and the algorithms to compute $e^{A/2^s}$ usually depend on the value of $\|A\|$ and the desired tolerance, and have been deeply analyzed.

The computational cost, $c(\cdot)$, is usually measured by the number of matrix–matrix products, so $c(e^A) = s + c(e^{A/2^s})$, where $c(e^{A/2^s})$ has to be replaced by the cost of its numerical approximation, e.g. $c(T_m(A/2^s))$ or $c(r_{2m}(A/2^s))$. Given a tolerance, one has to look for the scheme which provides such accuracy with the minimum number of products (see [9, 10] and references therein).

In some cases, if the matrix A has a given structure, more efficient methods can be obtained [4, 5] . For example, to compute the exponential of upper or lower triangular

*INSTITUTO DE MATEMÁTICA MULTIDISCIPLINAR, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, E-46022 VALENCIA, SPAIN. PHIBA@IMM.UPV.ES

†INSTITUTO DE MATEMÁTICA MULTIDISCIPLINAR, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, E-46022 VALENCIA, SPAIN. SERBLAZA@IMM.UPV.ES

‡DEPARTMENT OF MATHEMATICS, FACULTY OF ART AND SCIENCE, MUŞ ALPARSLAN UNIVERSITY, 49100 MUŞ, TURKEY. MUASEY@IMM.UPV.ES

matrices, in [1] the authors show that it is advantageous to exploit the fact that the diagonal elements of the exponential are exactly known. It is then more efficient to replace the diagonal elements obtained using e.g. Taylor or Padé approximations by the exact solution before squaring the matrix (this technique can also be extended to the first super (or sub-)diagonal elements).

On the other hand, in many cases the matrix A can be considered as a small perturbation of a sparse matrix D, i.e., $A = D + B$ with $\|B\| < \|D\|$ (and frequently $\|B\| \ll \|D\|$) where e^D is sparse and exactly solvable (or can be accurately and cheaply approximated numerically), and B is a dense matrix. This is the case, for example, if D is diagonal (or block diagonal with small matrices along the diagonal), or if it is diagonalizable using only a few elementary transforms. This is also the case, for example, if $n = 2k$ and

$$D = \begin{pmatrix} 0 & I \\ -\Omega^2 & 0 \end{pmatrix}$$

where I is the $k \times k$ identity matrix and Ω is a diagonal matrix where e^D is also an sparse and trivial to compute matrix. This problem can be originated from a semidiscretization of a hyperbolic PDE or from a set of k linearly coupled oscillators.

As a motivational example, let us consider the linear time-dependent system of differential equations

$$\frac{d}{dt}X = M(\varepsilon t)X, \qquad X(t_0) = X_0 \in \mathbb{C}^{n\times n}$$

with $M \in \mathbb{C}^{n\times n}$ and $|\varepsilon| \ll 1$, i.e., $M(\varepsilon t)$ evolves adiabatically with the variable t. Suppose that $M(\varepsilon t)$ is instantaneously diagonalizable, i.e., $M(\varepsilon t) = Q(\varepsilon t)D(\varepsilon t)Q^{-1}(\varepsilon t)$ with D a diagonal matrix. Then, we can consider what it is usually called the adiabatic picture in quantum mechanics (if M is a skew-Hermitian matrix), i.e., the change of variables, $X = Q(\varepsilon t)Y$ where Y is the solution of the differential equation

$$\frac{d}{dt}Y = \left(D - Q^{-1}\frac{d}{dt}Q\right)Y, \qquad Y(t_0) = Q^{-1}(\varepsilon t_0)X_0.$$

A second order method in the time step h which advances the solution from t_i to $t_i + h$, where $Y_i \approx Y(t_i)$, is given by

$$Y_{i+1} = e^{h\left(D_{1/2}+\varepsilon B_{1/2}\right)}Y_i, \tag{1.2}$$

where

$$D_{1/2} = D\left(\varepsilon\left(t_{i+1/2}\right)\right), \qquad \varepsilon B_{1/2} = -Q^{-1}\left(\varepsilon\left(t_{i+1/2}\right)\right)\frac{d}{dt}Q\left(\varepsilon\left(t_{i+1/2}\right)\right),$$

with $t_{i+1/2} = t_i + \frac{h}{2}$. Notice that $\varepsilon B_{1/2}$ is, in general, a dense matrix with a small norm (proportional to ε) due to the term $\frac{d}{dt}Q(\varepsilon t)$.

It is then natural to look for methods that approximate the exponential (1.2) at a low computational cost while providing sufficient accuracy. Notice that in most cases in practice it is not necessary to approximate the exponential up to round-off accuracy since the model/method itself does not reproduce the exact solution within round-off precision. However, the preservation of qualitative properties (e.g. orthogonality, symplecticity, unitarity, etc.) is in some cases of great interest [11].

The aim of this work is the exploration of new and more efficient algorithms which take advantage of the fact that e^D is sparse and known at a cheap computational cost and that B has a small norm. The schemes we analyze in continuation are based on splitting and composition techniques tailored for this particular problem.

For clarity in the presentation, we take the partition $s = s_1 + s_2$, we set $h = 2^{-s_2}, N = 1/h = 2^{s_2}$ and replace B by εB with $\|B\| \sim \|D\|$, and we propose a new recursive procedure that we refer as Modified Squaring

$$X_0 = e^{bh\varepsilon B}, \qquad X_k = X_{k-1} e^{a_k h D} X_{k-1}, \qquad k = 1, \ldots, s_1 \tag{1.3}$$

and $Y_{s_1} = e^{a_{s_1+1} h D} X_{s_1} e^{a_{s_1+1} h D}$ where $b = 1/2^{s_1}$ and the parameters a_k will be chosen properly to improve accuracy. The total cost is

$$c(Y_{s_1}^{s_2}) = s_1 + s_2 + c(e^{bh\varepsilon B})$$

where $c(e^{bh\varepsilon B}) = c(e^{\varepsilon B/2^s})$ is the cost to approximate this exponential. Since $\|h\varepsilon B\|$ is very small, a low-order diagonal Padé approximation can provide sufficient accuracy (for most problems it will suffice just to consider r_2 or r_4 which only require one inversion or one inversion and one product, or even a low-order Taylor approximation can also be used).

The choice $s_1 = 0$ corresponds to the Leapfrog or Strang method,

$$e^{h(D+\varepsilon B)} \approx e^{hD/2} e^{h\varepsilon B} e^{hD/2}, \tag{1.4}$$

where, as already mentioned, $e^{hD/2}$ can be accurately and cheaply computed.

More accurate methods can be obtained using a general composition

$$S_p^{[m]} = \prod_{i=1}^{m} e^{h a_i D} e^{h b_i \varepsilon B} \approx e^{h(D+\varepsilon B)}, \tag{1.5}$$

where the coefficients a_i, b_i are chosen such that $S_p^{[m]}$ is an approximation to the exact solution up to a given order, p, in the parameter h, i.e. $S_p^{[m]} = e^{h(D+\varepsilon B)} + \mathcal{O}(h^{p+1})$. However, to get efficient methods it is crucial to reduce the computational cost. Since the cost is dominated by the exponentials $e^{h b_i \varepsilon B}$, it is advisable to reuse as many exponentials as possible, e.g., letting $b_i = 1/m$, only one exponentiation is necessary. However, this class of methods has some limitations since for orders greater than 2, at least one of the coefficients a_i and one of the b_i must be negative and thus might jeopardize the re-utilization of the exponentials. However, for small perturbations, very accurate results can still be obtained with positive coefficients.

In the particular situation when $A \in \mathbb{C}^{n\times n}$, complex coefficients, $a_i \in \mathbb{C}$, can be used without increasing the computational cost, and then fourth-order methods with all b_i real and equal are achievable. The proposed recursive algorithm (1.3) corresponds to a particular case of an splitting method where the cost has been reduced while still leaving some free parameters for optimisation.

In this work, we assume that the product B^2 requires $\mathcal{O}\left(n^3\right)$ operations but DB requires only $\mathcal{O}\left(kn^2\right)$ with $k \ll n$ (e.g. $c(B^2) = 1,\ c(DB) = \delta$, with $\delta \ll 1$). Then, the commutator $\varepsilon[D, B] = \varepsilon(DB - BD)$ can be computed at considerably smaller cost than the product of two dense matrices while retaining a small norm due to the factor ε. It then makes sense to consider the recursive algorithm (1.3) where the exponential $e^{bh\varepsilon B}$ is replaced by

$$e^{bh\varepsilon B + \alpha h^3 \varepsilon [A,[A,B]]} \tag{1.6}$$

whose computational cost is similar, but more accurate results can be obtained if the scalar parameter α is properly chosen. Further exploiting this approach leads to the inclusion of the term $\beta h^5 \varepsilon [A,[A,[A,[A,B]]]]$ in the central exponential, which again, for an appropriate choice of the parameter β, decreases the error at a similar computational cost. The analysis presented in this work is also extended to the case in which not all parameters b_i are taken equal.

This paper is organized as follows: Section 2 considers the computational cost of Padé and Taylor methods as well as the cost of all operations involved in the splitting schemes analyzed in this work in order to develop new algorithms which minimize the whole cost. In Section 3 we analyze the algebraic structure of the different families of methods considered to obtain the order conditions to be satisfied by the coefficients. In Section 4 we propose a recursive algorithms to minimize the cost of the methods and we build new methods. An error analysis is carried in Section 5 and Section 6 illustrates the performance of the methods on several numerical examples. Finally, Section 7 presents the conclusions and the appendix collects, for completeness, several new families of splitting methods which have also been analyzed.

2. Computational cost of matrix exponentiation.

2.1. Computational cost of Taylor and Padé methods. We first review the computational cost of the optimized Taylor and Padé methods which are used in the literature and that are used as reference in the numerical examples.

Taylor methods. We use the Paterson-Stockmeyer scheme (see [8, 10, 16]) to evaluate $T_m = \sum_{k=0}^{m} A^n/n!$ which minimize the required number of products.

From the Horner-scheme-like computation, given a number of matrix products $2k$, the maximal attainable order is $m = (k+1)^2$. In [10], it is indicated that the optimal choice for most cases corresponds to $k = 3$, i.e. order $m = 16$ with just 6 products given by: $A^2 = AA,\ A^3 = A^2A,\ A^4 = A^2A^2$ and

$$T_{16}(A) = g_0 + (g_1 + (g_2 + (g_3 + g_4 A^4)A^4)A^4)A^4,$$

where g_i are linear combinations of already computed matrices, $g_i = \sum_{k=0}^{4} c_{i,k} A^k$, with $c_{i,k} = 1/(4i+k)!$ for $i = 0,1,2,3$ and $g_4 = I/16$ proportional to the identity (matrix).

Diagonal Padé methods. Diagonal Padé methods are given by the rational approximant

$$r_{2m}(A) = \frac{p_m(A)}{p_m(-A)}, \tag{2.1}$$

provided the polynomials p_m are generated by the recurrence

$$\begin{aligned} p_0(A) &= I, \qquad p_1(A) = 2I + A \\ p_m(A) &= 2(2m-1)p_{m-1}(A) + A^2 p_{m-2}(A). \end{aligned} \tag{2.2}$$

Moreover, $r_{2m}(A) = e^A + \mathcal{O}(A^{2m+1})$, whereas for $m = 1, 2$ we have

$$r_2(A) = \frac{I + A/2}{I - A/2}, \qquad r_4(A) = \frac{I + A/2 + A^2/12}{I - A/2 + A^2/12}. \tag{2.3}$$

The recursive algorithm (2.2) is, however, not an efficient way to compute $r_{2m}(A)$. For example, the method $r_{26}(A)$ is considered among the optimal choices (with respect to

accuracy and computational cost) of diagonal Padé methods when round off accuracy is desired and $\|A\|$ takes relatively large values. The algorithm to compute it is given by

$$(-u_{13} + v_{13}) r_{26}(A) = (u_{13} + v_{13}), \tag{2.4}$$

with

$$\begin{aligned} u_{13} &= A[A_6(b_{13}A_6 + b_{11}A_4 + b_9A_2) + b_7A_6 + b_5A_4 + b_3A_2 + b_1I], \\ v_{13} &= A_6(b_{12}A_6 + b_{10}A_4 + b_8A_2) + b_6A_6 + b_4A_4 + b_2A_2 + b_0I, \end{aligned}$$

where $A_2 = A^2, A_4 = A_2^2, A_6 = A_2A_4$. Written in this form, it is evident that only six matrix multiplications and one inversion are required. In a similar way, the method $r_{10}(A)$, which will be used in this work, only requires 3 products and one inversion.

2.2. Computational cost of splitting methods. Recall that we are considering a sparse and sparsely exponentiable matrix D, while B is a dense matrix and responsible for the numerical complexity. In order to build competitive algorithms, it is important to analyze - under these assumptions - the computational cost of all operations involved in the different classes of splitting and composition methods.

Let X, Y be two dense $n \times n$ matrices and denote by $c(\cdot)$ the cost of the operations in brackets as the number of matrix–matrix products of dense matrices, e.g., $c(XY) = 1$ and $c(X+Y) = \delta$, with $\delta \ll 1$, thereby neglecting operations with a lower complexity in the number of operations. According to this criterion, we derive Table 1, where the dominant terms are highlighted in boldface (the cost for the inverse of a matrix is taken as 4/3 the cost of a matrix-matrix product).

	Operation	Effort
Sum	$c(D + D) \approx 0$	$\mathcal{O}(k\,n)$, with $k \ll n$
	$c(X + Y) = \delta$	$\mathcal{O}(n^2)$
Product	$c(XY) = \mathbf{1}$	$\mathcal{O}(n^3)$
	$c(DD) = 0$	$\mathcal{O}(k^2\,n)$
	$c(DX) = k\delta$	$\mathcal{O}(k\,n^2)$
Inversion	$c(X^{-1}Y) = \mathbf{1} + \frac{\mathbf{1}}{\mathbf{3}}$	$c(X^{-1}Y) = \frac{4}{3}c(XY)$
Commutation	$c([D, X]) = c(DX - XD) = 2k\delta$	$\mathcal{O}(k\,n^2)$
	$c([D, [D, \ldots, [D, X]\cdots]]) = 2rk\delta$	$\mathcal{O}(k\,n^2)$
Exponentiation	$c(e^D) = wk\delta$	$\mathcal{O}(k^2\,n)$
	$c(r_2(X)) = \mathbf{1} + \frac{\mathbf{1}}{\mathbf{3}}$	$\mathcal{O}(n^3)$
	$c(r_4(X)) = \mathbf{2} + \frac{\mathbf{1}}{\mathbf{3}}$	$\mathcal{O}(n^3)$

TABLE 1

Computational cost of matrix operations for the sparse and sparsely exponentiable matrix D and arbitrary dense matrices $X, Y \in \mathbb{C}^{n\times n}$. The factor w in $c(e^D)$ is assumed to be small, $w \ll 1$.

Based on this analysis, we examine the splitting method (1.5) to identify the computationally relevant aspects. In this work we assume $\delta \ll 1$ and in our computations we will take $\delta = 0$ for simplicity. First, we have to choose how to approximate the exponentials $e^{h\varepsilon b_i B}$ taking into account that

$$r_2(h\varepsilon b_i B) = e^{h\varepsilon b_i B} + \mathcal{O}(h^3\varepsilon^3), \tag{2.5}$$

$$r_4(h\varepsilon b_i B) = e^{h\varepsilon b_i B} + \mathcal{O}(h^5\varepsilon^5). \tag{2.6}$$

A rough estimate for the composition (1.5), assuming all coefficients b_i different, and taking into account the cost shown in Table 1, we have

$$c(S_p^{[m]}, r_2) = m\frac{4}{3} + m - 1 = \frac{7}{3}m - 1, \qquad c(S_p^{[m]}, r_4) = m\frac{7}{3} + m - 1 = \frac{10}{3}m - 1,$$

where $c(S_p^{[m]}, r_i)$ denotes the cost of the method $S_p^{[m]}$ when the exponentials $e^{\varepsilon B}$ are approximated by $r_i(\varepsilon B)$. Repeating the coefficients b_i, i.e., $b_i = 1/m,\ i = 1, \ldots, m$, the computational cost can be reduced considerably, in this case, one gets

$$c(S_p^{[m]}, r_2) = \frac{4}{3} + (m-1) = m + \frac{1}{3}, \qquad c(S_p^{[m]}, r_4) = m + \frac{4}{3}.$$

Further simplifications are applicable and will be discussed in Sect. 4.

3. The Lie algebra of perturbed systems: (p_1, p_2) methods. Following the terminology of [14], we introduce a modified error concept which is suitable for the near-integrable structure of the matrix A at hand.

Letting $S_p^{[m]}$ be a pth-order m-stage consistent ($\sum_i a_i = \sum_i b_i = 1$) splitting method (1.5), we expand its error as

$$S_p^{[m]} - e^{hA} = \sum_{i=p+1} \sum_{j=1} e_{i,j} \varepsilon^j h^i C_{i,j},$$

where $e_{i,j}$ is a polynomial in the splitting coefficients a_k, b_k and $C_{i,j}$ is a sum of matrix products consisting of all combinations containing $(i-j)$ sparse elements D and j times B. Notice that in addition to the scaling h, we also expand in powers of the small parameter ε. The method is said to be of order $p = (p_1, p_2, \ldots)$ if $e_{i_1,1} = e_{i_2,2} = \ldots = 0$ for all $i_k \le p_k$ and $p_1 \ge p_2 \ge \cdots$.

Designing a method now consists of identifying the dominant error terms $e_{i,j}\varepsilon^j h^i$ and finding coefficients a_j, b_j to zero the polynomial $e_{i,j}$. The main tool in this endeavor is the Baker-Campbell-Hausdorff formula which provides a series expansion of the single exponential that has been actually computed when multiplying two matrix exponentials,

$$e^{hA} e^{hB} = e^{\mathrm{bch}(hA, hB)}, \quad \mathrm{bch}(hA, hB) = h(A+B) + \frac{h^2}{2}[A, B] + \mathcal{O}(h^3).$$

Recursive application of this formula to a symmetric splitting (1.5) establishes the concept of a modified matrix $h\tilde{A}$, along the lines of backward-error-analysis,

$$\begin{aligned} (3.1) \quad \log(S_p^{[m]}) = h\tilde{A} = hA &+ \tilde{e}_{3,1}\varepsilon h^3 [D,[D,B]] + \tilde{e}_{3,2}\varepsilon^2 h^3 [B,[D,B]] \\ &+ \tilde{e}_{5,1}\varepsilon h^5 [D,[D,[D,[D,B]]]] + \tilde{e}_{5,2}\varepsilon^2 h^5 [[D,[D,B]],[D,B]] \\ &+ \tilde{e}_{5,3}\varepsilon^2 h^5 [B,[D,[D,[D,B]]]] + \tilde{e}_{7,1}\varepsilon h^7 [D,[D,[D,[D,[D,[D,B]]]]]] + \mathcal{O}\left(\varepsilon^3 h^5 + \varepsilon^2 h^7\right), \end{aligned}$$

where the $\tilde{e}_{i,j}$ are also polynomials in the splitting coefficients a_k, b_k which multiply elements of the Lie algebra and are different from the coefficients $e_{i,j}$. Higher-order terms can be computed by efficient algorithms [3].

3.1. Error propagation by squaring. The splitting method (3.1) can also formally be written as

$$(3.2) \qquad S_{(p_1,p_2)}^{[m]} = \exp\left(h(D + \varepsilon B) + \varepsilon \sum_{k > p_1} c_k h^k [D^k, B] + \mathcal{O}\left(\varepsilon^2 h^{p_2+1}\right) \right)$$

where $[D^k, B] = [D, [D, [\ldots, [D, B]\ldots]]]$ and there is only one term proportional to ε at each power of h. We can then define a *processor*, a close to the identity map

$$P = \exp\left(-\varepsilon \sum_{k>p_1} c_k h^{k-1} [D^{k-1}, B]\right), \tag{3.3}$$

such that the method can be written as

$$S^{[m]}_{(p_1,p_2)} = PKP^{-1}, \tag{3.4}$$

with

$$K = \exp\left(h(D + \varepsilon B) + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right)\right). \tag{3.5}$$

Suppose now that the matrix A can be diagonalized, $A = QD_AQ^{-1}$, then clearly

$$e^A = Qe^{D_A}Q^{-1}.$$

The *kernel* K of the numerical method, on the other hand, can be diagonalized for sufficiently small $h = 1/n$ and ε using

$$\hat{Q} = Q + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right), \qquad \hat{D}_A = hD_A + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right),$$

such that, after n integration steps, we obtain

$$K^n = \hat{Q}e^{\tilde{D}_A}\hat{Q}^{-1}. \tag{3.6}$$

with $\tilde{D}_A = D_A + \mathcal{O}\left(nh^{p_2+1}\varepsilon^2\right)$. The size estimates of the above considerations lead to a favorable error propagation result which is stated in the following theorem.

THEOREM 1. *Let $A = D + \varepsilon B$ a diagonalizable matrix such that e^A is bounded and let $S^{[m]}_{(p_1,p_2)}$ be a splitting method that approximates the scaled exponential e^{hA} with $h = 1/n$. Then, for sufficiently small values of h and ε we have that*

$$\left\| e^A - \left(S^{[m]}_{(p_1,p_2)}\right)^n \right\| \leq C_1 h^{p_1+1}\varepsilon + nC_2 h^{p_2+1}\varepsilon^2. \tag{3.7}$$

where C_1, C_2 are constants which do not depend on h and ε.

Proof. From (3.4) and (3.6) we have that

$$\left(S^{[m]}_{(p_1,p_2)}\right)^n = P\hat{Q}e^{\tilde{D}_A}\hat{Q}^{-1}P^{-1} = \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1} \tag{3.8}$$

where now $\tilde{Q} = P\hat{Q} = Q + \mathcal{O}\left(h^{p_1+1}\varepsilon\right)$. Then

$$\begin{aligned}
\left\| e^A - \left(S^{[m]}_{(p_1,p_2)}\right)^n \right\| &= \left\| Qe^{D_A}Q^{-1} - \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1} \right\| \\
&= \left\| Qe^{D_A}Q^{-1} - \tilde{Q}e^{D_A}Q^{-1} + \tilde{Q}e^{D_A}Q^{-1} - \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1} \right\| \\
&\leq \|Q - \tilde{Q}\|\, \|e^{D_A}Q^{-1}\| + \|\tilde{Q}\|\, \|e^{D_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\|.
\end{aligned}$$

The right summand is expanded in a similar way to

$$\begin{aligned}
\|e^{D_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\| &= \|e^{D_A}Q^{-1} - e^{\tilde{D}_A}Q^{-1} + e^{\tilde{D}_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\| \\
&\leq \|e^{D_A} - e^{\tilde{D}_A}\|\, \|Q^{-1}\| + \|e^{\tilde{D}_A}\|\, \|Q^{-1} - \tilde{Q}^{-1}\|.
\end{aligned} \tag{3.9}$$

Taking into account that $\tilde{D}_A = D_A + \mathcal{O}\left(nh^{p_2+1}\varepsilon^2\right)$, $\tilde{Q} = Q + \mathcal{O}\left(h^{p_1+1}\varepsilon\right)$, and that e^A is bounded we obtained the desired result for sufficiently small values of h and ε. □

This result indicates that the error is the sum of a local error of order $\mathcal{O}(\varepsilon)$ plus a global error of order $\mathcal{O}\left(\varepsilon^2\right)$. For problems which require a relatively large number of squaring (a large value of $n = 2^s$) the dominant error of the splitting methods is proportional to ε^2. Then, to build methods which are accurate for different values of s it seems convenient to look for methods of effective order (p_1, p_2) with $p_1 > p_2$

The following numerical example illustrates the results obtained.

Example Let

$$A = \begin{pmatrix} \varepsilon & 1+\varepsilon \\ -1+\varepsilon & -\varepsilon \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{3.10}$$

with $\varepsilon = 10^{-1}, 10^{-3}$, and approximate $e^{2^s A} = \left(\cdots \left(e^A\right)^2 \cdots\right)^2$ to a relatively low accuracy. To approximate e^A, we consider a fourth-order Taylor method, $T_4(A)$ (that only requires 2 products) and a fourth-order Padé approximation, $r_4(A)$ (with a cost of one product and one inversion, equivalent to $1 + 4/3$ products). We compare the obtained results with the second-order splitting method (1.4), which we denote by $S_2^{[2,a]}$ or, since in this case $p_1 = p_2 = 2$, $S_{(2,2)}^{[2,a]}$, where the exponential e^D is computed exactly and εB is approximated with the second order diagonal Padé method, $r_2(\varepsilon B)$. The exact solution is given by

$$e^{2^s A} = \begin{pmatrix} \cos(2^s\mu) + \frac{\varepsilon}{\mu}\sin(2^s\mu) & \frac{1+\varepsilon}{\mu}\sin(2^s\mu) \\ -\frac{1-\varepsilon}{\mu}\sin(2^s\mu) & \cos(2^s\mu) - \frac{\varepsilon}{\mu}\sin(2^s\mu) \end{pmatrix}$$

with $\mu = \sqrt{1-2\varepsilon^2}$ and we analyze the error growth due to the squaring process in Fig. 1. We observe that neither Padé nor Taylor methods are sensitive w.r.t. the small parameter, whereas the splitting method drastically improves when decreasing ε. The splitting method is only of second order and thus used with a second order Padé method r_2 (using the fourth order method r_4 leaves error plot unchanged). Notice that for the small perturbation $\varepsilon = 10^{-3}$, the splitting with $r_2(\varepsilon B)$ is more accurate than the fourth-order Padé $r_4(A)$ which comes at nearly twice the computational cost (1 inversion vs. 1 inversion and 1 dense product). According to Theorem 1, the error of $S_{(2,2)}^{[2,a]}$ is the sum of a local error proportional to $h^3\varepsilon$ and a global error proportional to $nh^3\varepsilon^2$, with $n = 2^s$. Fig. 1 shows the results obtained for different values of ε^2 and s which clearly show both error sources.

4. Splitting methods for scaling and squaring. Taking into account the numerical effort established in the introduction, we derive methods which are optimal for the problem at hand. The optimization principle becomes clear at the example of the two versions of Strang's second-order splitting method

$$S_2^{[2,a]} = e^{\frac{h}{2}D}\, e^{h\varepsilon B}\, e^{\frac{h}{2}D} = \mathcal{D}_{h/2}\mathcal{B}_h\mathcal{D}_{h/2}, \tag{4.1}$$

and

$$S_2^{[2,b]} = e^{\frac{h}{2}\varepsilon B}\, e^{hD}\, e^{\frac{h}{2}\varepsilon B} = \mathcal{B}_{h/2}\mathcal{D}_h\mathcal{B}_{h/2}, \tag{4.2}$$

which differ in computational cost: Using the notation $\mathcal{D}_h = e^{hD}$, $\mathcal{B}_h = e^{hB}$, and keeping in mind that $\mathcal{D}_h$ is a sparse matrix while $\mathcal{B}_h$ is dense, the dominant numerical cost amounts to a single exponential with $c(S_2^{[2,a]}) = c(\mathcal{B}_h)$ for the first version, whereas the latter requires an additional matrix product, $c(S_2^{[2,b]}) = c(\mathcal{B}_{h/2}) + c(\mathcal{B}\mathcal{B})$.

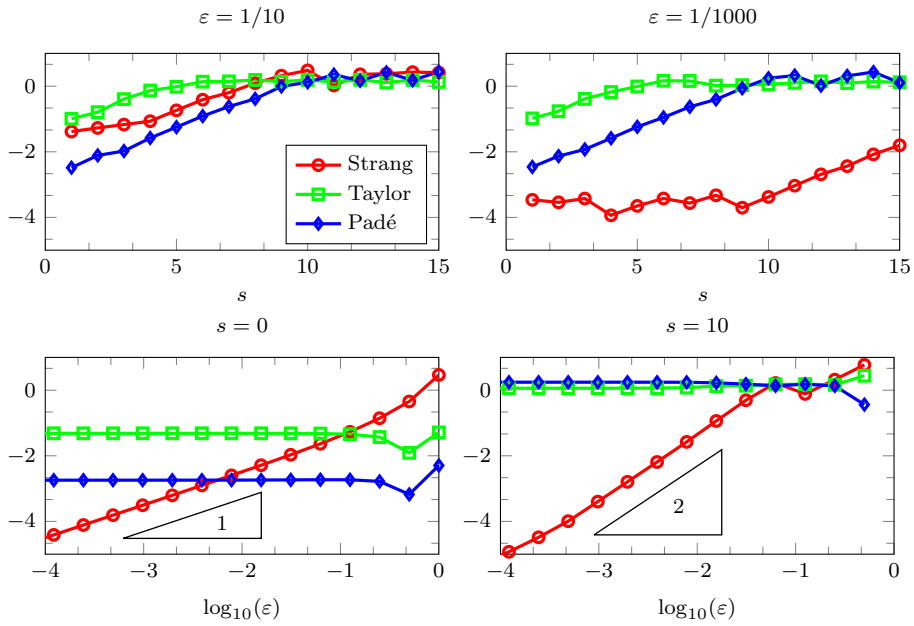

FIG. 1. *Error in the approximation to $e^{2^s A}$ with A given by (3.10) for different values of ε and number of squaring, s, in double-logarithmic axes. The bottom figures show that the error of the splitting methods is proportional to ε for small s (local error) and proportional to ε^2 for large values of s (global error)*

Furthermore, the large dominant part D is multiplied by $1/2$ before exponentiation in the cheaper variant which is advantageous in the sense of the scaling process.

We follow a variety of strategies in order to develop new methods and group them according to the splitting terminology, keeping in mind that the costly parts are products and exponentials of the dense matrices $\mathcal{B}$ and B, respectively.

4.1. Standard splittings. As we have discussed for the Strang splitting $S_2^{[2,b]}$, despite the appearance of B in two exponents, only one exponential actually has to be computed which is then stored and reused for the second identical exponent.

Generalizing this principle, we search for splitting methods a_i, b_j where all $b_j = b$ are identical to reduce the computational effort which now comes solely from the dense-matrix multiplications. A composition that is also symmetric in the coefficients a_j will reduce a great number of error terms (since even powers in h disappear) and additionally the amount of (cheap) exponentials $\mathcal{D}$ to be computed.

Next, we derive a particular family of splittings which can be understood in analogy to squarings and allows to reduce the necessary products.

4.1.1. Modified squarings. We propose to replace a given number of squarings by a one-step splitting method which has the benefit of free parameters to minimize the error. For illustration, let us compute a squaring step, $h = 2^{-1}$, of the standard Strang method,

$$(e^{h/2A} e^{hB} e^{h/2A})^2 = e^{\frac{1}{4}A} e^{\frac{1}{2}B} e^{\frac{1}{2}A} e^{\frac{1}{2}B} e^{\frac{1}{4}A}, \tag{4.3}$$

which we then contrast with a general splitting method at the same cost (one exponential and one product) without squaring ($h = 1$),

$$e^{a_2 A} e^{\frac{1}{2}B} e^{a_1 A} e^{\frac{1}{2}B} e^{a_2 A}. \tag{4.4}$$

It is evident that (4.4) includes (4.3) as a special case (choosing $a_1 = 1/2, a_2 = 1/4$) and we use the example (3.10) to illustrate the gains in accuracy. Fig. 2 shows that the performance is very sensitive to the choice of the free parameter and the method of effective order $(4, 2)$ is very close to the optimal one. A larger number of squarings

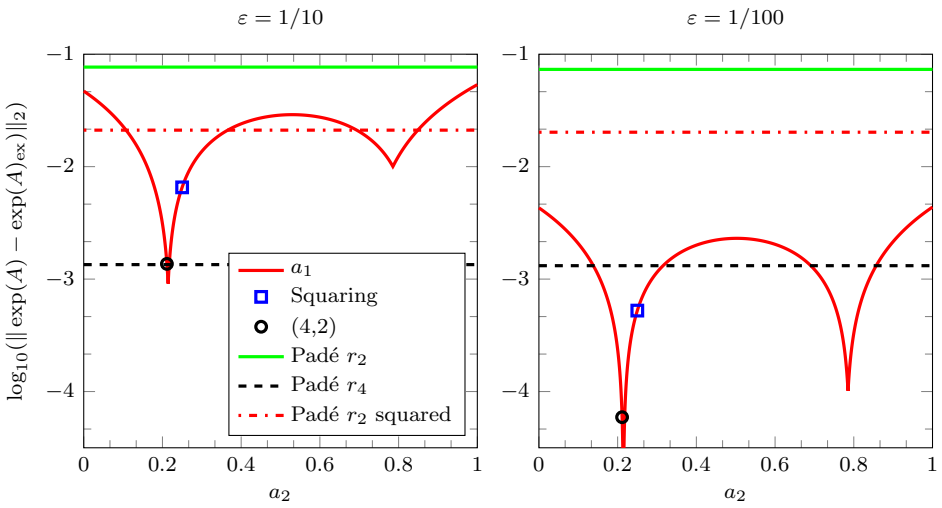

FIG. 2. *Modified squarings. All methods apart from $r_2(A)$ (green solid) have approximately the same numerical cost since the split uses 2nd order padé*

s can be replaced by a recursive procedure,

$$X_0 = e^{hb\varepsilon B}, \qquad X_k = X_{k-1} e^{a_k h D} X_{k-1}, \qquad k = 1, \ldots, s$$

and $Y_s = e^{a_{s+1}h} X_s e^{a_{s+1}h}$ where $b = 1/2^s$. The costly multiplications occur in the consecutive steps, X_k, where we recycle already computed blocks while introducing free parameters a_k at negligible extra effort. As a result, the cost of the algorithm is

$$c(Y_s) = s + c(e^{hb\varepsilon B})$$

where it usually suffices to approximate $e^{hb\varepsilon B}$ with a second or fourth-order Padé method, so $c(e^{hb\varepsilon B}, r_2) = \frac{4}{3}$ and $c(e^{hb\varepsilon B}, r_4) = 1 + \frac{4}{3}$. For consistency, the coefficients a_k have to satisfy

$$\left(2^{s-1}a_1 + \cdots + 2a_{s-1} + a_s\right) + 2a_{s+1} = \sum_{k=1}^{s} 2^{s-k} a_k + 2a_{s+1} = 1.$$

Notice that the choice $a_{s+1} = 1/2^{s+1}$, $a_k = 1/2^s$ for $k = 1, \ldots, s$, corresponds to the standard scaling and squaring applied to the Strang method (4.1). In the following, we have collected the most efficient splitting methods for an increasing numbers of

products $s = 0, 1, 2, 3, 4$. We have observed in the numerical experiments that for $s > 4$, the gain w.r.t. to standard scaling and squaring is marginal, and they are not considered in this work.

However, the parameter h demonstrates how any such method can be combined with standard scaling and squaring.

This procedure is equivalent to consider the partition $s = s_1 + s_2$ where the first s_1 squarings are carried out with the recursive algorithm with $b = 1/2^{s_1}$ and we continue with the remaining standard s_2 squarings with $h = 1/2^{s_2}$.

$s_1 = 0$. Strang $S_2^{[2,a]}$ with local order $\mathcal{O}(\varepsilon h^3)$.

$s_1 = 1$. After imposing symmetry, one free parameter remains and is used to obtain (4,2) methods [13, 14],

$$Y_1 = \mathcal{D}_{ha_2}\mathcal{B}_{h/2}\mathcal{D}_{ha_1}\mathcal{B}_{h/2}\mathcal{D}_{ha_2}, \tag{4.5}$$

where $a_2 = (3 - \sqrt{3})/6$, $a_1 = 1 - 2a_2$ and with local order $\mathcal{O}(\varepsilon h^5 + \varepsilon^2 h^3)$.

$s_1 = 2$. Allowing an additional product, at $b = 1/4$, we have

$$Y_2 = \mathcal{D}_{a_3 h}(\mathcal{B}_{h/4}\mathcal{D}_{a_2 h}\mathcal{B}_{h/4})\mathcal{D}_{a_1 h}(\mathcal{B}_{h/4}\mathcal{D}_{a_2 h}\mathcal{B}_{h/4})\mathcal{D}_{a_3 h}. \tag{4.6}$$

Optimizing the free parameters a_3, a_2, (where for consistency $a_1 = 1 - 2(a_3 + a_2)$) we can construct fourth-order methods, although complex-valued, with $a_3 = \frac{1}{10}(1 - i/3), a_2 = \frac{2}{15}(2 + i)$ and their complex conjugates a_i^* [2]. Alternatively, there are six real-valued (6,2) methods, the best of which is given in Table 2.

$s_1 = 3$. The three parameters for Y_3 can be used to produce complex-valued methods of order (6,4) or real-valued methods of order (8,2), the ones with smallest error coefficients can be found in Table 2.

$s_1 = 4$. The next iteration yields a 17-stage method Y_4. Its four parameters can be used to cancel the error coefficients $e_{3,1}, e_{3,2}, e_{5,1}, e_{7,1}$ for 48 complex (8,4) methods, or a $(10, 2)$ method with positive real coefficients, see Table 2.

4.2. Modified splittings. A drastic improvement on the previous methods can be made through the use of commutators. The special structure of the matrix allows for the fast computation of certain commutators, namely the ones that contain the matrix B only once. The inclusion of these commutators in the scheme will not only allow to reduce the number of error terms but also to reach order 4 using only real coefficients. Since we are interested in symmetric methods of up to order (6,4), the relevant terms are

$$[D, [D, B]] = DDB - 2DBD + BDD,$$
$$[D^4, B] = DDDDB - 4DDDBD + 6DDBDD - 4DBDDD + BDDDD,$$

and neglecting the numerical cost of summation and multiplication by a sparse matrix D, it is clear that the exponential

$$e^{\alpha h B + \beta h^3 [D,[D,B]] + \gamma h^5 [D,[D,[D,[D,B]]]]} = \tilde{\mathcal{B}}_{\alpha,\beta,\gamma}$$

can be evaluated at the same cost as $\mathcal{B}_{\alpha h}$. Along the lines of the modified squarings, we have derived the following compositions which require only one exponentials $\tilde{B}$ at a fixed number of products. The substitution $Y_s \to \tilde{Y}_s$ indicates the replacement of B by $\tilde{B}$.

TABLE 2

Modified squarings with and without commutators. In the right column, the corresponding computational cost is given together with the number of omitted solutions of the order conditions.

Y_2, order (6,2)	$c(\mathcal{B}_{h/4}) + 2c(\mathcal{BB})$
$a_1 = \sqrt{(5-\sqrt{5})/30},\ a_2 = \sqrt{(5-2\sqrt{5})/15}$	[7 solutions omitted]
Y_3, order (8,2)	
$a_1 = 0.153942020841153420134790213164$	only positive solution
$a_2 = 0.089999237645462605679630986655$	[47 omitted]
$a_3 = 0.102244554291437558627161030779$	
$a_4 = \frac{1}{2} - (4a_1 + 2a_2 + a_3)/2.$	
Y_3, order (6,4)	$c(\mathcal{B}_{h/8}) + 3c(\mathcal{BB})$
$a_1 = 0.13534452760420860194 + 0.06201309787740406230i$	[7 omitted]
$a_2 = 0.13027125534284511606 - 0.10310039626441585374i$	
$a_3 = 0.09906233274082533725 1 - 0.015885424766237390724i$	
$a_4 = \frac{1}{2} - (4a_1 + 2a_2 + a_3)$	
Y_4, order (10,2)	$c(\mathcal{B}_{h/16}) + 4c(\mathcal{BB})$
$a_1 = 0.077255933048297137202077893145$	only positive solution
$a_2 = 0.044492632239320424518905937035 4$	[383 omitted]
$a_3 = 0.051080773613693429438027986467$	
$a_5 = 0.025455365984130899045839064650 8$	
$a_4 = 1 - 8a_1 - 4a_2 - 2a_3 - 2a_5$	
Y_4, order (8,4)	
$a_1 = 0.06782965853562196485274129 + 0.03038453954138687801299186i$	[47 omitted]
$a_2 = 0.06477414774829711915884478 - 0.05170904068177844632921239i$	
$a_3 = 0.04963134399080347125041612 + 0.00584283681423207753349501i$	
$a_5 = 0.02474856149827627051056177 - 0.00610084851840072905292033i$	
$a_4 = 1 - 8a_1 - 4a_2 - 2a_3 - 2a_5$	
$\tilde{Y}_2$, order (6,4), minimizing $\mathcal{O}(\varepsilon^2 h^5)$	
$a_1 = (1 - a_2 - 2a_3)/2$	
$a_2 = 0.47071989362081947165$	
$a_3 = 0.04898669326146179875$	
$\beta = -0.002320917859694561351$	
$\gamma = 0.000032954671822820 3782$	
$\tilde{Y}_2$, order (8,4)	[47 omitted]
$a_1 = 0.360225814638949122073 4647$	
$a_2 = 1 - 2(a_3 + a_1)$	
$a_3 = 0.076610213006929386148 3005$	
$\beta = -0.001036370779182703986912 58$	
$\gamma = 0.000010240482532598594411391$	

$s = 0$. Strang's method can be made into a (6,2) scheme with

$$\tilde{Y}_0 = \mathcal{D}_{h/2}\tilde{\mathcal{B}}_{1,1/24,1/1920}\mathcal{D}_{h/2}. \tag{4.7}$$

We stress that, in principle, a method of order $(2n, 2)$ can be constructed using only a single exponential, however, at the expense of increasingly complicated commutators, $[D, [D, [\ldots, [D, B]] \cdots]$ whose computational complexity cannot be neglected anymore.

$s = 1$. Replacing $\mathcal{B}_{h/2}$ by $\tilde{\mathcal{B}}$ in (4.5), we obtain the (6,4) method

$$\tilde{Y}_1 = \mathcal{D}_{ha_2}\tilde{\mathcal{B}}\mathcal{D}_{ha_1}\tilde{\mathcal{B}}\mathcal{D}_{ha_2}, \tag{4.8}$$

TABLE 3
Theta values for diagonal Padé of order 2m with minimum number of products. The numbers highlighted in boldface correspond to the minimal cost $\pi_{2m} - \log_2(\theta_{2m})$

$u\backslash m$	1	2	3	4	5	6	7	13
$\leq 2^{-53}$	3.65E-8	5.32E-4	1.50E-2	8.54E-2	2.54E-1	5.41E-1	9.50E-1	**5.37**
$\leq$ 1E-10	3.46E-5	1.64E-2	1.47E-1	4.73E-1	9.98E-1	1.69	**2.51**	8.94
$\leq$ 1E-6	3.46E-3	1.64E-1	6.80E-1	1.49	**2.48**	3.58	4.76	1.24E1

where $a_2 = 1/6,\ a_1 = 2/3$ and $\tilde{\mathcal{B}}_{1/2,-1/144,121/311040}$ with unchanged effort $c(\mathcal{B}_{h/2}) + c(\mathcal{B}\mathcal{B})$.

$s = 2$. Using one additional multiplication, we reach $\tilde{Y}_2$, which can be tuned to be of order (8,4) or (6,4) while minimizing the error at $\mathcal{O}(\varepsilon^2 h^5)$, see Table 2.

We have also analyzed other classes of splitting and composition methods. The methods obtained showed a worst performance on the numerical examples tested in this work. The schemes obtained are, however, collected in the appendix for completeness.

5. Error analysis. Our methods have proven successful for a low to medium accuracy since the high-order Padé methods are hard to beat at round-off precision. In a first step, we derive new scaling estimates for Padé methods for lower precision requirements following [9]. Let $\theta_m(u)$ be the largest value of $\|A\|$ s.t. the Padé scheme r_{2m} has precision at least u, i.e.,

$$\forall A, \|A\| \leq \theta_m \ : \ r_{2m}(A) = e^{A+E}, \text{ s.t. } \|E\| \leq u.$$

The new θ_m are given in Table 3. It is clear that the number of necessary scalings for a sought precision is $s = \lceil \log_2(\|A\|/\theta_m) \rceil \in \mathbb{N}_0$ and taking into account the number of multiplications π_m needed with each method, a global minimum $s + \pi_m$ can be found at each precision.

We will focus our attention on the medium precision range $u \leq 10^{-6}$, where the 10th order method r_{10} is optimal among the Padé schemes. In analogy to the error control for Padé methods, we discuss the backward error of the previously obtained splitting methods. The BCH formula, in the form (3.1), already gives us a series expansion of the remainder E,

$$E = \sum_{i=p+1} \sum_{j=1} h^i f_{i,j} \mathbf{C}_{i,j}. \tag{5.1}$$

However, the expansion is difficult to compute for $i > 15$ with exponentially growing effort in the symbolic computation. Further complications arise from the nature of the expansion: it involves commutators $\mathbf{C}_{i,j}$ in D, B which we have to estimate. For most cases, the roughest (although sharp) estimate

$$\|[D, B]\| = \|DB - BD\| \leq 2\varepsilon \|D\|^2, \qquad \varepsilon = \|B\|/\|D\|, \tag{5.2}$$

is way to loose to give accurate results. Having in mind matrices with asymmetric spectra, i.e., small positive and large negative eigenvalues, the following estimate is more useful [12, Theorem 4],

$$\|[D, B]\| \leq \|B\|(d^+ - d^-),$$

where the numerical range of D (or easier: the eigenvalues) lies within $[d^-, d^+]$, which corresponds to a factor 2 gain in the estimate. In any case, we can refine the estimate by recycling the calculations for the modified splittings, $[D,[D,B]]$, $[D,[D,[D,[D,B]]]]$ and intermediate steps, $[D,B]$, etc. Then, we estimate the most relevant commutators, recalling the notation $[D^2,B]=[D,[D,B]]$,

$$\begin{aligned}
\|[B,[D,B]]\| &\le 2\|[D,B]\|\|B\|,\\
\|[B,[D,[D,[D,B]]]]\| &\le 2\|[D,B]\|\,\|[D,[D,B]]\|,\\
\|[D,[B,[D,[D,B]]]]\| &\le 2\|[D,B]\|\,\|[D,[D,B]]\|,\\
\|[B,[B,[D,[D,B]]]]\| &\le 4\|B\|^2\,\|[D,[D,B]]\|,\\
\|[D,[D,[D,[D,[D,[D,B]]]]]]\| &\le (d^+-d^-)^2\|[D,[D,[D,[D,B]]]]\|.
\end{aligned}$$

The splitting methods studied in this work can be classified by their order and the leading error commutators are collected in Table 4.

In principle, one could use the error terms at the next larger power in h to estimate the quality of this truncation, but for practical purposes and $h \ll 1$, numerical experiments show that the simpler bounds are sufficient to get a reasonable recommendation for the number of squarings. For illustration, we print the expansion (5.1) for the method (4.7)

$$\begin{aligned}
E^{[6,2]}(h) \le \tilde{E}^{[6,2]} &= 3.11\text{E-}6h^7\|[D^6,B]\| + 8.33\text{E-}2h^3\|[B,[D,B]]\|\\
&\quad + h^5(1.39\text{E-}3\|[B,[D^3,B]]\| + 5.56\text{E-}3\|[[B,D],[D^2,B]]\|)\\
&\quad + h^5(5.56\text{E-}3\|[B^2,[D^2,B]]\| + 2.78\text{E-}3\|[[B,D],[B^2,D]]\|\\
&\quad + \mathcal{O}\left(\varepsilon h^9 + \varepsilon^2 h^7 + \varepsilon^3 h^7\right)
\end{aligned} \tag{5.3}$$

and for method $\tilde{Y}_2$ of order (6,4) from Table 2,

$$\begin{aligned}
E^{[6,4]}(h) \le \tilde{E}^{[6,4]} &= 3.49\text{E-}5h^7\|[D^6,B]\|\\
&\quad + h^5(1.70\text{E-}3\|[B,[D^3,B]]\| + 1.39\text{E-}3\|[[B,D],[D^2,B]]\|)\\
&\quad + h^5(1.39\text{E-}3\|[B^2,[D^2,B]]\| + 4.63\text{E-}4\|[[B,D],[B^2,D]]\|\\
&\quad + \mathcal{O}\left(\varepsilon h^9 + \varepsilon^2 h^7 + \varepsilon^3 h^7\right).
\end{aligned} \tag{5.4}$$

Then, the following algorithm suggests itself: Compute the commutators needed for the modified squarings, estimate their norms and finally evaluate the polynomials $\tilde{E}(h)$ to find an upper bound for h such that the local error remains below given accuracy u. This h translates directly to the number of external squarings $s_2 = \lceil \log_2(h) \rceil$ and now, it only remains to sum the computational cost originating from the number of dense products and exponentials to find the overall most efficient method for a particular set of matrices D, B. In contrast to the static Padé case, where there is a single best method by just fixing the precision, this procedure is more flexible and chooses - at virtually no extra cost - the best method for the given matrix algebra structure.

Furthermore, we can establish a threshold for the size of the small parameter ε in order to decide when splittings should be preferred over Padé methods. For example, let $u = 10^{-6}(10^{-4})$ be the desired precision, we then know that r_{10} (r_{10}) is optimal and the largest value the norm $\theta = \|A\|$ can take is $\theta_5 = 2.48(\theta_5 = 3.85)$. Given that r_{10} requires three multiplications, we use the splitting method $\tilde{Y}_0$ with three squarings

to yield a method of the same computational cost. In (5.3), this corresponds to taking $h = 2^{-3}$. Applying the roughest possible estimate (5.2) to $\tilde{E}^{[6,2]}(2^{-3})$, we obtain a polynomial in ε which takes values below u for $\varepsilon \leq 0.01(0.05)$. In practice, the norm estimates are sharper since we can use the commutators that have been computed in the algorithm and we expect an even larger threshold for ε.

TABLE 4
Leading error commutators at given order.

order	ε^1	ε^2	ε^3
$(2n,2)$	$[D^{2n},B]$	$[B,[D,B]]$	$[B,[B,[D,[D,B]]]]$
$(2n,4)$	$[D^{2n},B]$	$[B,[D,[D,[D,B]]]],[D,[B,[D,[D,B]]]]$	$[B,[B,[D,[D,B]]]]$

6. Numerical results. In a couple of test scenarios, we attempt to provide an idea about when our new methods are superior to standard Padé methods. In each setting, we define a different matrix D which will be perturbed by a matrix B, s.t.

$$B_{i,j} = k(i-j)/(i+j)$$

and k is chosen to satisfy $\varepsilon = \|B\|_1/\|D\|_1$ for the parameter set $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$. We measure the relative error in the 1-norm, $\|S_p^{[m]} - e^A\|_1/\|e^A\|_1$ for all methods where the exact solution is computed by a high-order Padé method and all splittings use the second-order scheme r_2 to approximate the exponential $\exp(2^{-s}B)$.

6.1. Rotations. Letting

$$D = i\,\mathrm{diag}\{-25, -24.5, \ldots, 24.5, 25\}$$

with $i = \sqrt{-1}$, the performance of Padé methods of order 10 and 26, together with the 16th-order Taylor method using 6 products is studied. Fig. 3 shows the relative error (in logarithmic scale) versus cost (number of matrix–matrix multiplications) for different choices of the scaling parameter, s. The horizontal line shows the tolerance desired for the numerical experiments. It is evident that, as expected, the Padé method r_{10} is the most efficient among these standard schemes and will be used for reference in later experiments. For illustration, Fig. 3 also includes two modified squaring methods without commutators (Y_2, order (6,2) and Y_3, order (6,4) from Table 2), both of which are more efficient than r_{10} in the lower precision range. Notice that, since A isa complex matrix, to use splitting methods with complex coefficients does not increase the cost of the algorithms in this case. Furthermore, the standard methods are insensitive w.r.t. the small parameter ε, whereas the splitting methods improve as ε decreases. In a second experiment in Fig. 4, we use the same matrices as before but choose the most efficient splitting methods with commutators, $\tilde{Y}_0$ and $\tilde{Y}_1$. Using the local error estimates in (5.3) and (5.4), we indicate the point which corresponds to the optimal number of squarings for the splitting methods and compare it with the recommended squaring parameter for Padé r_{10}. For a relatively large parameter ε in the left panel of Fig. 4, the method r_{10} is still superior but is already equaled in terms of computational cost for a smaller perturbation in the center plot, but at higher accuracy. As ε becomes smaller in the right panel, we achieve higher accuracy at lower computational cost, saving one product for $\tilde{Y}_1$ and two products for $\tilde{Y}_2$, respectively.

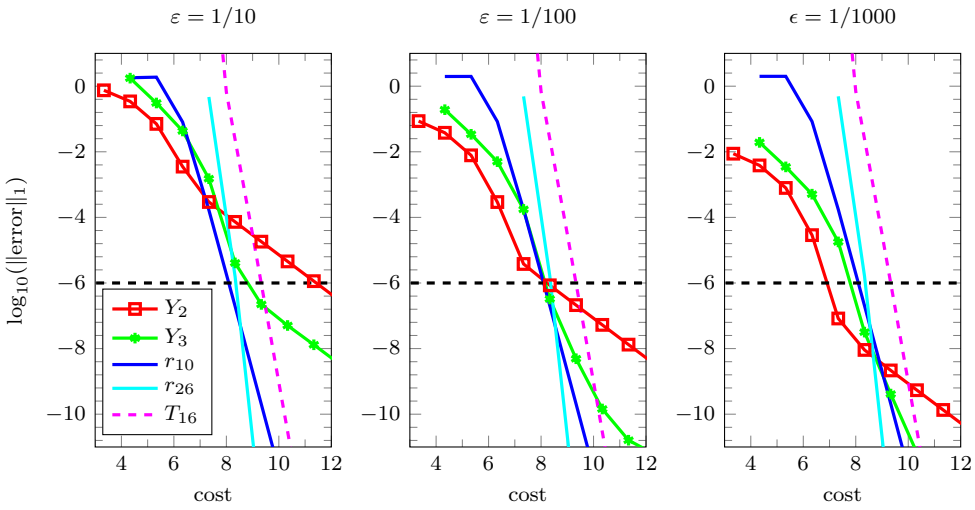

FIG. 3. *Relative error (in logarithmic scale) versus computational cost given by the number of dense matrix-matrix products for the standard Padé and Taylor methods r_{10}, r_{26}, T_{16}, and the splitting methods Y_2 and Y_3 of order (6,2) and (6,4), respectively, without commutators from Table 2.*

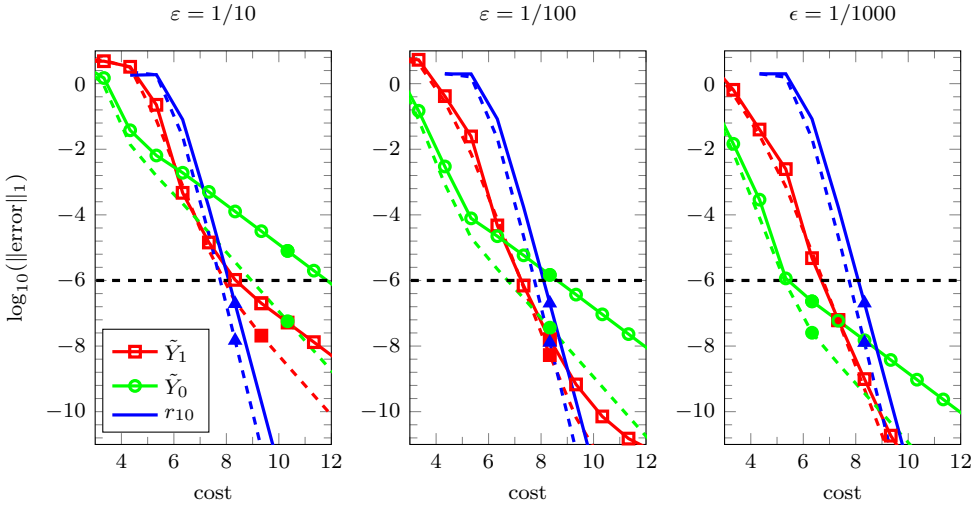

FIG. 4. *The solid lines show the relative global error e^A after squaring versus the overall computational cost and the dashed curves depict the relative local error in $e^{2^{-s}A}$ (before squaring) which is used for the error estimate, both for Padé and the splittings. The filled markers indicate the position of the recommended (automatic) algorithm.*

In the next plot, Fig. 5, we increase the norm of the matrix and set $D_2 = 100D$, and B is scaled accordingly to maintain the quotient $\|B\|_1/\|D_2\|_1 = \varepsilon$. The implications are a substantial increase in the number of necessary squarings with prior scaling and corresponds to a long-time integration in which we observe the favorable behavior expected from Fig. 1. The gain with respect to Padé's method is striking as ε decreases.

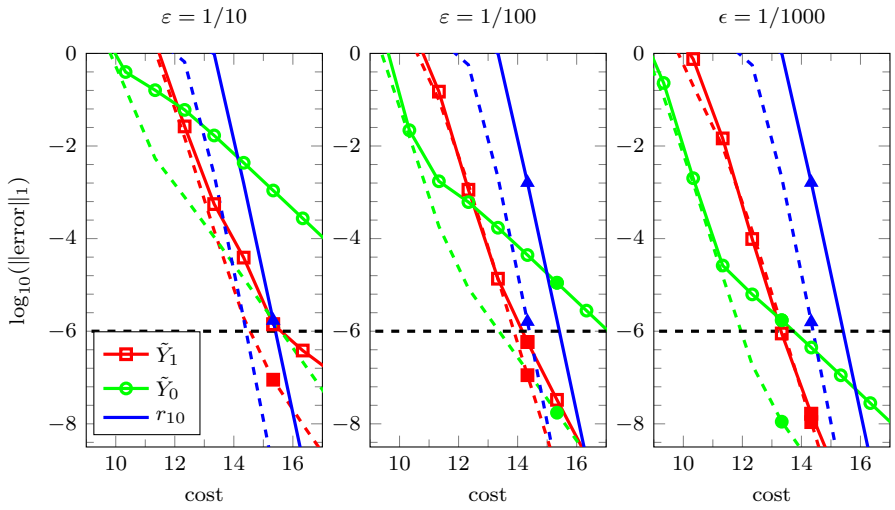

FIG. 5. *Same as Fig. 4 for an exponential of a large norm matrix, with diagonal part* $D_2 = 100D$.

6.2. Dissipation. A less favorable problem for our algorithm is given using a stiff matrix with large positive and negative eigenvalues,

$$D = \text{diag}\{15, 14.5, \ldots, -14.5, -15\}.$$

The perturbation B is scaled as before to $\|B\|/\|D\| = \varepsilon$. Fig. 3 shos the results obtained. Again, our methods perform well for low accuracies for not too large perturbations and improve as ε becomes smaller.

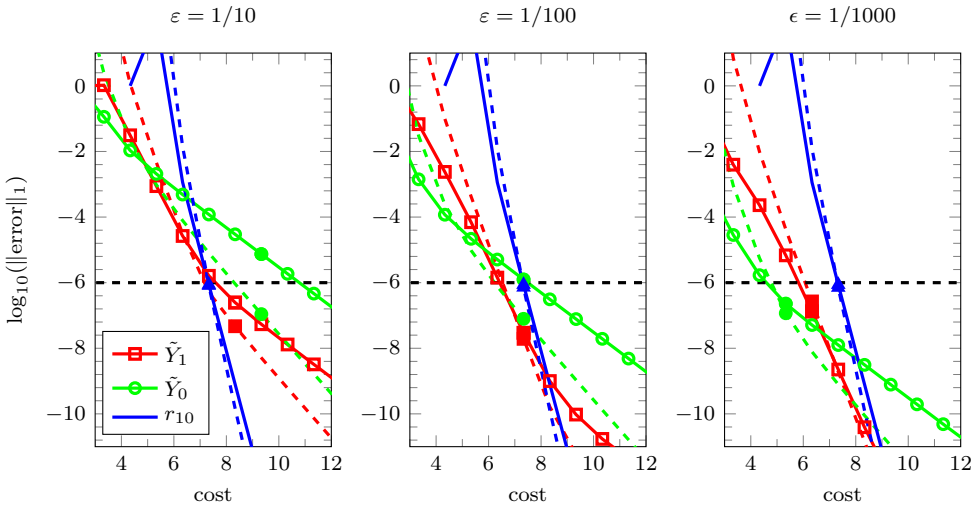

FIG. 6. *Same as Fig. 4 but for the stiff matrix case* $D = \text{diag}\{15, 14.5, \ldots, -14.5, -15\}$.

7. Conclusions. We have proposed a new recursive algorithm based on splitting methods for the computation of the exponential of perturbed matrices which can be

written as the sum $A = D + \varepsilon B$ of a sparse and efficiently exponentiable matrix D with sparse exponential e^D and a dense matrix εB which is of small norm in comparison with D. We have considered the scaling and squaring technique but replacing the Padé or Taylor methods to compute the exponential of the scaled matrix by an appropriate splitting methods tailored for this class of matrices. We have proposed a recursive algorithm which allows to save computational cost and still leaves some free parameters for optimization. An important feature of splitting methods for perturbed problems is that the error is a sum of a local error of order $\mathcal{O}(\varepsilon)$ plus a global error of order $\mathcal{O}(\varepsilon^2)$ and this allows to build new methods with high performance when low to medium accuracy is desired. The new schemes are built taking into account that the dominant computational cost arises from the computation of dense matrix products and we present a modified squaring which takes advantage of the smallness of the perturbed matrix B in order to reduce the number of squarings necessary. The recursive character of the modified squarings implies only light memory requirements. Theoretical results on local error and error propagation for splitting methods are complemented with numerical experiments and show a clear improvement over existing and highly optimized Padé methods when low to medium precision is sought.

Appendix. Further approaches. In this subsection, we collect results on approaches that are successful in the context of splittings for ordinary differential equations, however, have been found less efficient on the numerical experiments than the methods presented before.

A.1. On processing. A basic property of the adjoint action,

$$e^P Y e^{-P} = e^{\mathrm{ad}_P} Y = Y + [P, Y] + \frac{1}{2}[P, [P, Y]] + \cdots$$

together with the cheap computability of the commutator $[D, B] = DB - BD$ motivates the use of *processing techniques*, well-known for the numerical integration of differential equations, to eliminate error terms. The idea is now based on the observation that $(XYX^{-1})^N = XY^NX^{-1}$ and essentially corresponds to a change of basis in which the error propagation (recall that large s can be regarded as a (long-) time integration using a small time-step $h = 1/2^s$) is expected to be less severe.

The modified Strang algorithm (4.7) has leading error proportional to

$$[B, [D, B]], \quad [B, [D, [D, [D, B]]]], \quad [D, [D, [B, [D, B]]]].$$

The first two of which can be eliminated using a processor with $P = \alpha[D, B] + \beta[D, [D, [D, B]]]$, thus motivating the ansatz

$$e^{\alpha\varepsilon h^2[D,B]+\beta\varepsilon h^4[D,[D,[D,B]]]}\tilde{Y}_s e^{-\alpha\varepsilon h^2[D,B]-\beta\varepsilon h^4[D,[D,[D,B]]]}.$$

The norm of the outer exponents is small and a low order Padé approximation, say $r_2(P)$, usually provides sufficient accuracy. Therefore, at the expense of one exponential, one multiplication and one inversion (which is performed together with the multiplication, as for the Padé methods, $(\mathcal{B}\mathcal{D})\tilde{\mathcal{B}}^{-1}$), we get two free parameters, α, β. Using the kernel $\tilde{Y}_0$, we reach order (6,4), whereas $\tilde{Y}_1$ is sufficient for order (10,4) and (6,6,4), see Table 5.

A.2. More exponentials. For problems where complex coefficients a_j lead to a substantial increase in computational complexity (e.g., when $A, B \in \mathbb{R}^{n\times n}$) or matrix commutators are not desirable, it could be advantageous to allow negative values for some b_j.

A first example is the four-stage method

$$S_4^{[4]} = \mathcal{D}_{ha_1}\mathcal{B}_{hb_1}\mathcal{D}_{ha_2}\mathcal{B}_{hb_2}\mathcal{D}_{ha_2}\mathcal{B}_{hb_1}\mathcal{D}_{ha_1}. \tag{A.1}$$

This scheme requires two exponentials, two products and has two free parameters which can produce a fourth-order method with real coefficients a_j, b_j, known as triple jump [6, 19, 20], see Table 5.

Another product is necessary to compute the six-stage composition

$$S_{(6,4)}^{[6]} = \mathcal{D}_{ha_1}(\mathcal{B}_{hb_1}\mathcal{D}_{ha_2}\mathcal{B}_{hb_1})\mathcal{D}_{ha_3}\mathcal{B}_{hb_2}\mathcal{D}_{ha_3}(\mathcal{B}_{hb_1}\mathcal{D}_{ha_2}\mathcal{B}_{hb_1})\mathcal{D}_{ha_1}.$$

Three free parameters are sufficient to construct (6,4) methods, however, with complex time-steps. The real-valued fourth-order method minimizing the error at $\mathcal{O}(\varepsilon h^5)$ can be found in Table 5. An additional stage with a grouping similar to the modified splittings,

$$S_{(6,4)}^{[7]} = \mathcal{D}_{ha_1}(\mathcal{B}_{hb_1}\mathcal{D}_{ha_2}\mathcal{B}_{hb_2}\mathcal{D}_{ha_2}\mathcal{B}_{hb_1})\mathcal{D}_{ha_3}(\mathcal{B}_{hb_1}\mathcal{D}_{ha_2}\mathcal{B}_{hb_2}\mathcal{D}_{ha_2}\mathcal{B}_{hb_1})\mathcal{D}_{ha_1},$$

requires the same number of products but has real solutions of order (6,4). Among the four real-valued solutions, the one minimizing the error at $\mathcal{O}(\varepsilon h^7)$ is printed in Table 5. We have found that supposedly clever re-utilization of exponentials by setting b_j to be a rational multiple of an already computed exponent b_k are not competitive since - at its very best - one can save the computation of an exponential at the cost of an inversion ($b_j = -b_k$) or a matrix product ($b_j = 2b_k$), however, the direct use of the sufficiently accurate r_2 Padé method needs only one inversion.

A.3. Splitting for low-order Padé. Technically, the stated splitting orders assume the exact computation of all exponentials, but in practice, the cheap underlying Padé scheme r_2 has accuracy limit $\mathcal{O}(\varepsilon^3 h^3)$. Since we assumed ε to be a small parameter, comparable to h^2, it could be regarded as $\mathcal{O}(\varepsilon h^7)$. Instead of switching to the more precise r_4 method ($\mathcal{O}(\varepsilon^5 h^5)$) for the exponential $\mathcal{B}$, (using r_2 for the processor has error $\mathcal{O}(h^6\varepsilon^3)$ and is therefore sufficient), we attempt to use a free parameter to decrease the r_2-related error in $\mathcal{B}$ to $h^5\varepsilon^5$.

The procedure is based on the observation that the approximant $r_2(h\varepsilon B)$ can be expressed as a single exponential

$$r_2(h\varepsilon B) = e^{h\varepsilon B + h^3\varepsilon^3 C + \mathcal{O}(h^5\varepsilon^5)}$$

for some matrix C. Notice that the exponent can be expanded in odd powers of h since r_2 is symmetric. Now, we simply add the (unknown) matrix C to the algebra and in addition to the previous order conditions, we have to solve $\sum_{i=1}^m b_i^3 = 0$. It is clear that condition $b_i = 1/m$ has to be dropped and at least three exponentials $\mathcal{B}_{b_j h}$ are necessary. We embark by modifying (A.1) to

$$\Psi^{[4,mod]} = \mathcal{D}_{ha_1}\tilde{\mathcal{B}}_1\mathcal{D}_{ha_2}\tilde{\mathcal{B}}_2\mathcal{D}_{ha_2}\tilde{\mathcal{B}}_1\mathcal{D}_{ha_1}. \tag{A.2}$$

Using two exponentials (inversions) and two multiplications, we have six free parameters and only one additional equation. The freedom in the parameters allows to construct real-coefficient methods of order (10,4) and alternatively, at order (8,4), a method minimizing the squared error polynomials $e_{5,2}$ at $\varepsilon^2 h^5$, see Table 5.

TABLE 5
Further splitting methods, including several exponentials and processing techniques.

$S^{[4]}$, 4 stages, order 4	2 exp, 2 prod
$a_1 = \frac{1}{6}(2 + 1/2^{1/3} + 2^{1/3})$, $b_1 = \frac{1}{3}(2 + 1/2^{1/3} + 2^{1/3})$	2 complex sol. omitted
$S^{[6]}$, 6 stages, order 4	2 exp, 3 prod
$a_1 = 0.1973110756624279163 1$, $a_2 = 0.3825264659473131295 5$, $a_3 = (1 - 2a_1 - 2a_2) = -0.0798375416097410458 62$, $b_1 = 0.4251934190991034507 1$, $b_2 = 1 - 4b_1 = -0.7007736763964138028 4$.	[minimizes $\mathcal{O}(\varepsilon h^5)$]
$S^{[7]}$, 7 stages, order (6,4)	2 exp, 3 prod
$a_1 = 0.3593752962197870894 1$, $a_2 = -0.0983792310552348355 26$, $a_3 = (1 - 2a_1 - 4a_2) = 0.6747663317813651644 8$, $b_1 = 0.6770296354476050058 6$, $b_2 = 1/2 - 2b_1 = -0.8540592708952100117 3$.	[minimizes $\mathcal{O}(\varepsilon h^7)$]
Processed $e^{xh^2[D,B]+yh^4[D,[D,[D,B]]]}\tilde{Y}_0 e^{-xh^2[D,B]-yh^4[D,[D,[D,B]]]}$	2 exp, 1 prod, 1 inv
Order (6,4)	
$a_1 = 1/2$, $\beta = -1/24$, $\gamma = 31/5760$ $x = -1/12$, $y = 1/120$.	
Processed $e^{xh^2[D,B]+yh^4[D,[D,[D,B]]]}\tilde{Y}_1 e^{-xh^2[D,B]-yh^4[D,[D,[D,B]]]}$	2 exp, 1 prod, 1 inv
Order (6,6,4)	
$a_2 = 0.2587977340833403434530275$, $\beta = -0.0052276833645836254216539 25$, $\gamma = 0.0000329546718228203782$, $x = -0.0230327668541684191965902 2$, $y = 0.0007499977372301362425777840$.	
Order (10,4)	
$a_2 = 0.2502255012888943852139 24$, $\beta = -0.0052083460460411565905784$, $\gamma = 0.0000329546718228203782$, $x = -0.0208897086555569296368143$, $y = 0.0000573371861339342917744$	
$\mathcal{D}_{ha_1}\tilde{\mathcal{B}}_1\mathcal{D}_{ha_2}\tilde{\mathcal{B}}_2\mathcal{D}_{ha_2}\tilde{\mathcal{B}}_1\mathcal{D}_{ha_1}$	2 exp, 2 prod
real (10,4) based on r_2, $d_2 = -0.0017987433839305087766$, $c_2 = -0.14389703981903926044$, $d_1 = 0.0000393451173268162726 08$, $c_1 = -0.0079989398412468330564$, $b_2 = -0.5826865215312073584 8$, $a_2 = 0.5046861998972319219 1$	
(8,4) minimizing $e_{5,2}$, $d_2 = 0.0094609567584454808 26$, $c_2 = -0.0378019688845376510 8$, $d_1 = 0.0011653151315644152329$, $c_1 = -0.0610464753084976377 33$, $b_2 = -0.5826865215312073584 8$, $a_2 = 0.5046861998972319219 1$	

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, (2009), pp. 970–989.

[2] F. CASTELLA, P. CHARTIER, S. DESCOMBES, AND G. VILMART, *Splitting methods with complex times for parabolic equations*, BIT Numer. Math. 49 (2009), pp. 487–508.

[3] F. CASAS AND A. MURUA *An efficient algorithm for computing the BakerCampbellHausdorff series and some of its applications*, J. Math. Phys. 50 (2009), pp. 033513-1–033513-23 (2009), pages

[4] E. CELLEDONI AND A. ISERLES, Approximating the exponential from a Lie algebra to a Lie

group, Math. Comput., 69 (2000), pp. 1457–1480.
[5] E. Celledoni and A. Iserles, Methods for the Approximation of the Matrix Exponential in a Lie-Algebraic Setting, IMA J. Numer. Anal., 21 (2001), pp. 463–488.
[6] M. Creutz and A. Gocksch, *Higher-order hybrid Monte Carlo algorithms* Phys. Rev. Lett., 63 (1989), pp. 9–12.
[7] N. J. Higham, *The Scaling and Squaring Method for the Matrix Exponential*, SIAM J. Matrix Anal. Appl., 26, (2005), pp. 1179–1193.
[8] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2008).
[9] N. J. Higham, *The Scaling and Squaring Method for the Matrix Exponential Revisited*, SIAM Review, 51, (2009), pp. 747–764.
[10] N. J. Higham and A. H. Al-Mohy, *Computing Matrix Functions*, Acta Numerica, 51, (2010), pp. 159–208.
[11] A. Iserles, H. Z. Munthe-Kaas, S.P. Nørsett and A. Zanna, *Lie group methods*, Acta Numerica, 9, (2000), pp. 215–365.
[12] F. Kittaneh, *Norm Inequalities for Commutators of Normal Operators*, In: Inequalities and Applications, Int. Series of Num. Math, Birkhäuser Basel
[13] J. Laskar and P. Robutel. *High order symplectic integrators for perturbed Hamiltonian systems*, Celest. Mech. and Dyn. Astro., 80 (2001), pp. 39–62.
[14] R. I. McLachlan. *Composition methods in the presence of small parameters*, BIT, 35 (1995), pp. 258–268.
[15] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review 45 (2003), pp. 3–49.
[16] M.S. Paterson and L.J. Stockmeyer , *On the number of nonscalar multiplications necessary to evaluate polynomials*, SIAM J. Comput. 2 (1973), pp. 60–66.
[17] J. Sastre, J. Ibáñez, P. Ruiz, and E. Defez, *Accurate and efficient matrix exponential computation*, Int. J. Comput. Math., 91, (2014), pp. 97–112.
[18] R. B. Sidje, *Expokit: a software package for computing matrix exponentials*, ACM Trans. Math. Software 24 (1998), pp. 130–156.
[19] M. Suzuki, *Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations* Phys. Lett. A, 146 (1990), pp. 319–323.
[20] H. Yoshida, *Construction of higher order symplectic integrators* Phys. Lett. A, 150 (1990), pp. 262–268.

View publication stats