Applied Mathematics and Computation 340 (2019) 206–220

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

# Boosting the computation of the matrix exponential☆

J. Sastre[a,*], J. Ibáñez[b], E. Defez[c]

[a] Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, Spain
[b] Instituto de Instrumentación para Imagen Molecular, Universitat Politècnica de València, Spain
[c] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Spain

ARTICLE INFO

*Keywords:*
Matrix exponential
Scaling and squaring
Taylor series
Efficient matrix polynomial evaluation

ABSTRACT

This paper presents new Taylor algorithms for the computation of the matrix exponential based on recent new matrix polynomial evaluation methods. Those methods are more efficient than the well known Paterson–Stockmeyer method. The cost of the proposed algorithms is reduced with respect to previous algorithms based on Taylor approximations. Tests have been performed to compare the MATLAB implementations of the new algorithms to a state-of-the-art Padé algorithm for the computation of the matrix exponential, providing higher accuracy and cost performances.



## 1. Introduction

The computation of matrix functions has received remarkable attention in the last decades because of its numerous applications in science and engineering [1]. From all the matrix functions the matrix exponential has been the most studied function, and a large number of methods for its computation have been proposed [1,2].

In 2009 the authors submitted their first work with Taylor based algorithms for computing the matrix exponential [3]. Until then, Padé approximants for the matrix exponential were preferred to Taylor approximations because Padé algorithms were more efficient than the existing Taylor algorithms, for similar accuracy [1]. Applying and improving the algorithms for Padé approximants from [15] to Taylor approximations, the Taylor algorithms from [3] showed to be generally more accurate than the Padé algorithm from [15] in tests, with a slightly higher cost.

In [4] the authors presented a scaling and squaring Taylor algorithm for computing the matrix exponential based on an improved mixed backward and forward error analysis. It was more accurate than the state-of-the-art Padé algorithm from [14] in the majority of tests, with a slightly higher cost. Subsequently, Sastre et al. [5] provided a formula for the forward relative error of the matrix exponential Taylor approximation, and proposed to increase the allowed error bounds depending on the matrix size and the Taylor approximation order. This algorithm reduced the computational cost in exchange for a small impact in accuracy. The method proposed in [6] simplified the algorithm of [4], preserving accuracy, and showing to be more accurate than the Padé algorithm from [14] in the majority of tests, being also more efficient in some cases. Finally, Defez et al. [8] used Taylor approximations combined with spline techniques to increase accuracy, also increasing the cost. In this work, we present new Taylor algorithms based on the efficient matrix polynomial evaluation methods from [9],

☆ This work has been supported by Spanish Ministerio de Economía y Competitividad and European Regional Development Fund (ERDF) grant TIN2014-59294-P.

* Corresponding author at: Instituto de Telecomunicaciones y Aplicaciones Multimedia, Spain.
*E-mail address:* jsastrem@upv.es (J. Sastre).

https://doi.org/10.1016/j.amc.2018.08.017
0096-3003/

increasing significantly the efficiency of the previous Taylor methods. We will show that the new algorithms are generally both more accurate and efficient than the state-of-the-art Padé algorithm from [14].

Throughout this paper $\mathbb{C}^{n\times n}$ denotes the set of complex matrices of size $n\times n$, $I$ denotes the identity matrix for this set, $\rho(A)$ is the spectral radius of matrix $A$, and $\mathbb{N}$ denotes the set of positive integers. The matrix norm $\|\cdot\|$ denotes any subordinate matrix norm; in particular $\|\cdot\|_1$ is the 1-norm. The symbols $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the smallest following and the largest previous integer, respectively. The cost of the Taylor algorithms will be given in terms of the number of evaluations of matrix products, denoting the cost of one matrix product by $M$. Note that the multiplication by the matrix inverse in Padé approximations is calculated as the solution of a multiple right-hand side linear system. The cost of the solution of multiple right-hand side linear systems $AX=B$, where matrices $A$ and $B$ are $n\times n$ will be denoted by $D$. Taking into account that, see [10, Appendix C]

$$D\approx 4/3M, \tag{1}$$

the cost of evaluating rational approximations will be also given in terms of $M$. All the given algorithms are intended for IEEE double precision arithmetic. Their extension to different precision arithmetics is straightforward.

This paper is organized as follows: Section 2 presents a general scaling and squaring Taylor algorithm. Section 3 introduces efficient evaluation formulas for the Taylor matrix polynomial approximation of the matrix exponential based on [9]. Section 4 presents the scaling and squaring error analysis. The new algorithm is given in Section 5. Section 6 shows numerical results and Section 7 gives some conclusions. Next theorem from [5] will be used in Section 4 to bound the norm of matrix power series.

**Theorem 1.** *Let $h_l(x)=\sum_{k\geq l} b_k x^k$ be a power series with radius of convergence $R$, and let $\tilde{h}_l(x)=\sum_{k\geq l}|b_k|x^k$. For any matrix $A\in\mathbb{C}^{n\times n}$ with $\rho(A)<R$, if $a_k$ is an upper bound for $||A^k||$ $(||A^k||\leq a_k)$, $p\in\mathbb{N}$, $1\leq p\leq l$, $p_0\in\mathbb{N}$ is the multiple of $p$ with $l\leq p_0\leq l+p-1$, and*

$$\alpha_p=\max\{a_k^{\frac{1}{k}}: k=p,l,l+1,l+2,\dots,p_0-1,p_0+1,p_0+2,\dots,l+p-1\}, \tag{2}$$

*then $||h_l(A)||\leq\tilde{h}_l(\alpha_p)$.*

## 2. General Taylor algorithm

The Taylor approximation of order $m$ of the matrix exponential of $A\in\mathbb{C}^{n\times n}$, denoted by $T_m(A)$, is defined by the expression

$$T_m(A)=\sum_{k=0}^{m}\frac{A^k}{k!}. \tag{3}$$

The scaling and squaring algorithms with Taylor approximation (3) are based on the approximation $e^A=\left(e^{2^{-s}A}\right)^{2^s}\approx(T_m(2^{-s}A))^{2^s}$ [2], where the nonnegative integers $m$ and $s$ are chosen to achieve full machine accuracy at a minimum cost.

A general scaling and squaring Taylor algorithm for computing the matrix exponential is presented in Algorithm 1, where $m_M$ is the maximum allowed value of $m$.

**Algorithm 1** General scaling and squaring Taylor algorithm for computing $B=e^A$, where $A\in\mathbb{C}^{n\times n}$ and $m_M$ is the maximum approximation order allowed.

1: Preprocessing of matrix $A$.
2: Choose $m_k\leqslant m_M$, and an adequate scaling parameter $s\in\mathbb{N}\cup\{0\}$ for the Taylor approximation with scaling.
3: Compute the matrix polynomial $B=T_{m_k}(A/2^s)$
4: **for** $i=1:s$ **do**
5: $\quad B=B^2$
6: **end for**
7: Postprocessing of matrix $B$.

In this paper the evaluation of the Taylor matrix polynomial of Step 3 is improved. The preprocessing and postprocessing steps (1 and 7) are based on applying transformations to reduce the norm of matrix $A$, see [1], and will not be discussed in this paper. In Step 2, the optimal order of Taylor approximation $m_k\leq m_M$ and the scaling parameter $s$ will be chosen improving the algorithm from [6].

In [6] the matrix polynomial $T_m(2^{-s}A)$ was evaluated using the Paterson–Stockmeyer method evaluation formula (7) of [6], see [11]. The optimal Taylor orders $m$ for that method were in the set $m_k=\{1,2,4,6,9,12,16,20,25,30,\dots\}$, $k=0,1,\dots$, respectively, where the matrix powers $A^2,A^3,\dots,A^q$ were evaluated and stored to be used in all the computations. Table 1, see [6, Table 1], shows some optimal values of $q$, denoted by $q_k$, used in [6] for orders $m_k$, $k=0,1,2,\dots,M$, and $m_M=20$, 25 or 30. In this work $T_m(2^{-s}A)$ will be computed using new evaluation methods based on [9], more efficient than Paterson–Stockmeyer method.