Journal of Computational and Applied Mathematics 291 (2016) 370–379

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

High performance computing of the matrix exponential

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

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

ARTICLE INFO

Article history:
Received 15 October 2014
Received in revised form 30 March 2015

Keywords:
Matrix exponential
Scaling and squaring
Taylor series

ABSTRACT

This work presents a new algorithm for matrix exponential computation that significantly simplifies a Taylor scaling and squaring algorithm presented previously by the authors, preserving accuracy. A Matlab version of the new simplified algorithm has been compared with the original algorithm, providing similar results in terms of accuracy, but reducing processing time. It has also been compared with two state-of-the-art implementations based on Padé approximations, one commercial and the other implemented in Matlab, getting better accuracy and processing time results in the majority of cases.

1. Introduction

Matrix function computation has received remarkable attention in the last decades due to its usefulness in a great variety of engineering problems. Especially noteworthy is the matrix exponential, which emerge in the solution of systems of linear differential equations in numerous applications and a large number of methods for its computation have been proposed [1,2]. Moreover, in many cases, the resolution of these systems involve large matrices, so, not only accurate, but also efficient methods are needed. In this sense, the authors presented in [3] two modifications of a Taylor-based scaling and squaring algorithm to reduce computational costs while preserving accuracy.

In [4] the authors presented a scaling and squaring Taylor algorithm based on an improved mixed backward and forward error analysis, which was more accurate than existing state-of-the-art algorithms for matrix exponential such as that in [5], in the majority of test matrices with a lower or similar cost. Subsequently, in [6], the authors gave a new formula for the forward relative error of matrix exponential Taylor approximation and proposed to increase the allowed forward error bound depending on the matrix size and the Taylor approximation order. This algorithm reduces the computational cost in exchange for a small impact in accuracy. In this work, we present a new algorithm that significantly simplifies the one presented in [4] providing a competitive scaling and squaring algorithm for matrix exponential computation in comparison with both previous algorithms and the state-of-the-art implementations based on Padé approximations from [5,7].

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 $\|\cdot\|_1$ and $\|\cdot\|_2$ are the 1-norm and the 2-norm, respectively. The symbols $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the smallest following and the largest previous integer, respectively. This paper is organized as follows: Section 2 presents a general scaling and squaring Taylor algorithm; Section 3 presents the scaling and squaring error analysis; the

* Corresponding author.
E-mail address: pruiz@dsic.upv.es (P. Ruiz).

http://dx.doi.org/10.1016/j.cam.2015.04.001

new algorithm is given in Section 4; finally, Section 5 shows numerical results and Section 6 gives some conclusions. Next Theorem 1 from [6] and the new Theorem 2 will be used in Section 3 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*

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

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

Theorem 2. *Let* $l \in \mathbb{N}$*,* $l \geq 1$*, and let* $q \in \mathbb{N}$ *be the minimum value with* $1 \leq q \leq l$ *such that*

$$\|A^q\|^{\frac{1}{q}} \leq \max\{\|A^k\|^{\frac{1}{k}} : k = l, l+1, \ldots, q_0-1, q_0+1, q_0+2, \ldots, l+q-1\}, \tag{2}$$

where $q_0 \in \mathbb{N}$ *is the multiple of q with* $l \leq q_0 \leq l+q-1$*. Then if*

$$\|A^{k_0}\|^{\frac{1}{k_0}} = \max\{\|A^k\|^{\frac{1}{k}} : k = l, l+1, \ldots, q_0-1, q_0+1, q_0+2, \ldots, l+q-1\} \tag{3}$$

then

$$\max\{\|A^k\|^{\frac{1}{k}} : k \geq l\} = \|A^{k_0}\|^{\frac{1}{k_0}} \tag{4}$$

Proof. Since q_0 is a multiple of q, then $q_0/q \in \mathbb{N}$ and using (2) and (3) one gets

$$\|A^{q_0}\|^{1/q_0} = \|A^{qq_0/q}\|^{1/q_0} \leq \|A^q\|^{q_0/(qq_0)} = \|A^q\|^{1/q} \leq \|A^{k_0}\|^{1/k_0}. \tag{5}$$

For any integer $k \geq l+q$ we can write $k = l+i+jq$ for positive integers i and j with $0 \leq i \leq q-1$ and $j = [k-(l+i)]/q$, and then using (2), (3) and (5) it follows that

$$\|A^k\|^{\frac{1}{k}} \leq \left[\|A^{l+i}\|\|A^q\|^j\right]^{\frac{1}{k}} \leq \left[\|A^{k_0}\|^{\frac{l+i}{k_0}}\|A^{k_0}\|^{\frac{jq}{k_0}}\right]^{\frac{1}{k}} = \|A^{k_0}\|^{\frac{k}{k_0 k}} = \|A^{k_0}\|^{\frac{1}{k_0}}. \quad \square \tag{6}$$

2. Taylor algorithm

Taylor approximation of order m of exponential of matrix $A \in \mathbb{C}^{n\times n}$ can be expressed as the matrix polynomial $T_m(A) = \sum_{k=0}^{m} A^k/k!$. The scaling and squaring algorithms with Taylor approximations are based on the approximation $e^A = \left(e^{2^{-s}A}\right)^{2^s} \approx \left(T_m(2^{-s}A)\right)^{2^s}$ [1], 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 $B = T_{m_k}(A/2^s)$ using (7)
4: **for** $i = 1 : s$ **do**
5: $\quad B = B^2$
6: **end for**
7: Postprocessing of matrix B.

The preprocessing and postprocessing steps (1 and 7) are based on applying transformations to reduce the norm of matrix A, see [2,8], and will not be discussed in this paper.

In Step 2, the optimal order of Taylor approximation $m_k \leqslant m_M$ and the scaling parameter s are chosen. Matrix polynomial $T_m(2^s A)$ can be computed optimally in terms of matrix products using values for m in the set $m_k = \{1, 2, 4, 6, 9, 12, 16, 20, 25, 30, \ldots\}$, $k = 0, 1, \ldots$, respectively, see [2, p. 72–74]. The choice of s is fully described in Section 3.

After that, in Step 3, we compute the matrix exponential approximation of the scaled matrix by using the modified Horner and Paterson–Stockmeyer's method proposed in [4, p. 1836–1837]. Note that this modified method has the same optimal