Journal of Computational and Applied Mathematics 309 (2017) 325–332

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Efficient and accurate algorithms for computing matrix trigonometric functions[☆]

Pedro Alonso[a], Javier Ibáñez[b], Jorge Sastre[c], Jesús Peinado[b,*], Emilio Defez[d]

[a] Department of Information Systems and Computation, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain
[b] Instituto de Instrumentación para Imagen Molecular, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain
[c] Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain
[d] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain

ARTICLE INFO

Article history:
Received 27 November 2015

Keywords:
Matrix cosine
Matrix sine
Scaling and squaring method
Taylor series
Backward error
Parallel implementation

ABSTRACT

Trigonometric matrix functions play a fundamental role in second order differential equations. This work presents an algorithm based on Taylor series for computing the matrix cosine. It uses a backward error analysis with improved bounds. Numerical experiments show that MATLAB implementations of this algorithm has higher accuracy than other MATLAB implementations of the state of the art in the majority of tests. Furthermore, we have implemented the designed algorithm in language C for general purpose processors, and in CUDA for one and two NVIDIA GPUs. We obtained a very good performance from these implementations thanks to the high computational power of these hardware accelerators and our effort driven to avoid as much communications as possible. All the implemented programs are accessible through the MATLAB environment.

1. Introduction

Many engineering processes are described by second order differential equations, whose exact solution is given in terms of trigonometric matrix functions sine and cosine. For example, the wave problem

$$v^2\frac{\partial^2\psi}{\partial x^2}=\frac{\partial^2\psi}{\partial t^2}, \tag{1}$$

plays an important role in many areas of engineering and applied sciences. If the spatially semi-discretization method is used to solve (1), we obtain the matrix differential problem

$$X''(t)+AX(t)=0,\qquad X(0)=X_0,\qquad X'(0)=X_1, \tag{2}$$

where A is a square matrix and X_0 and X_1 are vectors. The solution of (2) is

$$X(t)=\cos\left(\sqrt{A}t\right)X_0+\left(\sqrt{A}\right)^{-1}\sin\left(\sqrt{A}t\right)X_1, \tag{3}$$

☆ This work has been supported by Spanish Ministerio de Economía y Competitividad and European Regional Development Fund (ERDF) grant TIN2014-59294-P.
* Corresponding author.
E-mail addresses: palonso@dsic.upv.es (P. Alonso), jjibanez@dsic.upv.es (J. Ibáñez), jorsasma@iteam.upv.es (J. Sastre), jpeinado@dsic.upv.es (J. Peinado), edefez@imm.upv.es (E. Defez).

http://dx.doi.org/10.1016/j.cam.2016.05.015
0377-0427/

where $\sqrt{A}$ denotes any square root of a non-singular matrix A [1, p. 36]. More general problems of type (2), with a forcing term $F(t)$ on the right-hand side arise from mechanical systems without damping, and their solutions can be expressed in terms of integrals involving the matrix sine and cosine [2].

Numerous methods have been proposed for computing $f(A)$, where $f(.)$ is a scalar function defined on the spectrum of the matrix $A \in \mathbb{C}^{n\times n}$. Many of them have a dubious numerical stability [3]. A complete theoretical study of matrix functions and their computational methods and algorithms can be found in [1], in particular the computation of matrix trigonometric functions. The main methods are based on matrix decompositions and on polynomial and rational approximations. Since polynomial and rational approximations are accurate only near the origin, scaling and recovering techniques [4–6] are usually used. Moreover, to reduce computational costs Paterson–Stockmeyer method [7] is used for evaluating the polynomials which appear in these approximations.

In this work we present sequential and parallel algorithms based on Taylor series that use Theorem 1 from [8] for computing matrix trigonometric functions.

Throughout this paper $\mathbb{C}^{n\times n}$ denotes the set of complex matrices of size $n \times n$, I the identity matrix for this set, $\rho(X)$ the spectral radius of matrix X, and $\mathbb{N}$ the set of positive integers. In this paper we use the 1-norm to compute the actual norms. Sections 2 and 3 present sequential and parallel Taylor algorithms for computing matrix trigonometric functions, respectively. Section 4 deals with numerical tests and finally in Section 5 the conclusions are presented.

2. Sequential algorithms for computing matrix cosine and sine

The matrix cosine can be defined for all $A \in \mathbb{C}^{n\times n}$ by

$$\cos(A) = \sum_{i=0}^{\infty} \frac{(-1)^i A^{2i}}{(2i)!},$$

and let

$$T_{2m}(A) = \sum_{i=0}^{m} \frac{(-1)^i B^i}{(2i)!} \equiv P_m(B), \tag{4}$$

be the Taylor approximation of order $2m$ of $\cos(A)$, where $B = A^2$. Since Taylor series are accurate only near the origin, in algorithms that use this approximation the norm of matrix B is reduced by scaling the matrix. Then, a Taylor or Padé approximation is computed, and finally the approximation of $\cos(A)$ is recovered by means of the double angle formula $\cos(2X) = 2\cos^2(X) - I$.

Using the same notation as in [5], we have that Taylor matrix polynomial approximation (4), expressed as $P_m(B) = \sum_{i=0}^{m} p_i B^i$, $B \in \mathbb{C}^{n\times n}$, can be computed with optimal cost by Paterson–Stockmeyer's method [7] choosing m from the set

$$\mathbb{M} = \{1, 2, 4, 6, 9, 12, 16, 20, 25, 30, 36, 42, \ldots\}, \tag{5}$$

where the elements of $\mathbb{M}$ are denoted as $m_1, m_2, m_3, \ldots$ (see [1, pp. 72–74] for a complete description). The algorithm computes firstly the matrix powers $B^2, B^3, \ldots, B^q$ being $q = \lceil \sqrt{m_k} \rceil$ or $q = \lfloor \sqrt{m_k} \rfloor$, and integer divisor of m_k. As stated in [1, p. 74] using those values for q results in the same cost. Thus, the evaluation formula (23) from [9, p. 6455] is computed as

$$\begin{aligned} P_{m_k}(B) = {} & (((p_{m_k} B^q + p_{m_k-1} B^{q-1} + p_{m_k-2} B^{q-2} + \cdots + p_{m_k-q+1} B + p_{m_k-q} I) \cdot B^q \\ & + p_{m_k-q-1} B^{q-1} + p_{m_k-q-2} B^{q-2} + \cdots + p_{m_k-2q+1} B + p_{m_k-2q} I) \cdot B^q \\ & + p_{m_k-2q-1} B^{q-1} + p_{m_k-2q-2} B^{q-2} + \cdots + p_{m_k-3q+1} B + p_{m_k-3q} I) \cdot B^q \\ & \ldots \\ & + p_{q-1} B^{q-1} + p_{q-2} B^{q-2} + \cdots + p_1 B + p_0 I. \end{aligned} \tag{6}$$

We define the boxing size as the largest polynomial degree which appear in (6), i.e. the value q. Table 1 shows the values of q for different values of m. Taking into account Table 4.1 from [1, p. 74], then the cost of evaluating (4) with (6) in terms of matrix products, denoted by Π_{m_k}, for $k = 1, 2, \ldots,$ is

$$\Pi_{m_k} = k. \tag{7}$$

The difficulty of the algorithms based on Taylor series is to find appropriate values m_k and the scaling factor s such that $\cos(A)$ is computed accurately and with minimal computational cost.

Next theorem will be used to bound the norm of the matrix Taylor series.

Theorem 1 ([5]). *Let $h_l(x) = \sum_{i=l}^{\infty} p_i x^i$ be a power series with radius of convergence w, $\tilde{h}_l(x) = \sum_{i=l}^{\infty} |p_i| x^i$, $B \in \mathbb{C}^{n\times n}$ with $\rho(B) < w$, $l \in \mathbb{N}$ and $t \in \mathbb{N}$ with $1 \leqslant t \leqslant l$. If t_0 is the multiple of t such that $l \leqslant t_0 \leqslant l + t - 1$ and*

$$\beta_t = \max\{b_j^{1/j} : j = t, l, l+1, \ldots, t_0 - 1, t_0 + 1, t_0 + 2, \ldots, l + t - 1\}, \tag{8}$$