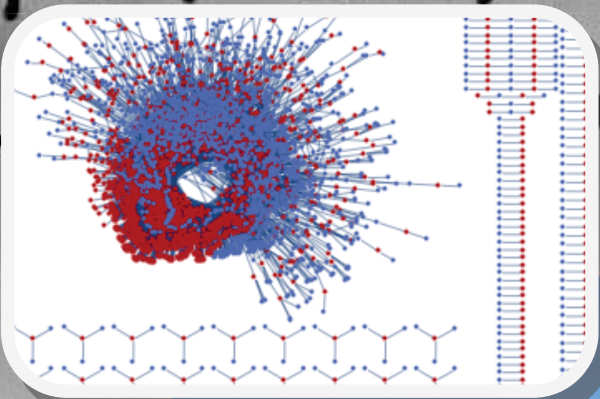
MODELLING FOR ENGINEERING AND HUMAN BEHAVIOUR 2017

Instituto Universitario de Matemática Multidisciplinar



L. Jódar, J. C. Cortés and L. Acedo(Editors)

Instituto Universitario de
Matemática Multidisciplinar

im2 Instituto de Matemática Multidisciplinar

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Ciudad Politécnica de la Innovación

MODELLING FOR ENGINEERING, & HUMAN BEHAVIOUR 2017

Instituto Universitario de Matemática Multidisciplinar
Universitat Politècnica de València
Valencia 46022, SPAIN

Edited by
Lucas Jódar, Juan Carlos Cortés and Luis Acedo
Instituto Universitario de Matemática Multidisciplinar
Universitat Politècnica de València
I.S.B.N.: 978-84-697-8505-8

CONTENTS

Two algorithms for computing the matrix cosine based on new Hermite approximations*

Emilio Defez⋆, Javier Ibáñez♮, Jesús Peinado♮, Jorge Sastre†, Pedro Alonso♮

⋆ Instituto de Matemática Multidisciplinar.

♮ Departamento de Sistemas Informáticos y Computación.

† Instituto de Telecomunicaciones y Aplicaciones Multimedia.

Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain.

edefez@imm.upv.es, {jjibanez, jpeinado}@dsic.upv.es, jorsasma@iteam.upv.es, palonso@dsic.upv.es

November 30, 2017

1 Introduction

We will introduce in this work new rational-polynomial Hermite matrix expansions which permit us obtain accurate and efficient methods for computing the matrix cosine. These proposed methods are compared with the most publicized method based on Padé method for computing the matrix cosine.
The computation of matrix trigonometric functions has received remarkable attention in the last decades due to its usefulness in the solution of systems of second order linear differential equations. Several state-of-the-art algorithms have been provided for computing these matrix functions, see for example [23, 2, 20] and references therein.

The study of orthogonal matrix polynomials are becoming more and more relevant in the last decades. In particular, the matrix Hermite polynomials, introduced and studied in [17, 18] in have received considerable attention for its application in the solution of matrix differential equations, see [4]. The series of Hermite matrix polynomials have been studied for its application in the matrix exponential computation, see [21], matrix cosine approximation, [8, 9, 23] and the hyperbolic sine and cosine computation, see [7, 6], for example.

In the scalar case, Hermite polynomials $H_n(x)$ are widely used in quantum mechanics, mathematical physics, nucleon physics, and quantum optics. Recently, new formulas for series of Hermite scalar polynomials of the type $\sum_{n\geq 0} \frac{H_{2n+l}(x)}{n!}t^n, l = 1, 2, 3, \ldots$ have been obtained in [16] and these formulas have been applied in quantum optics theory. The generalization of this classes of formulae for Hermite matrix polynomials $H_n(x, A)$ can be found in [10].

In this paper we will calculate the exact value of new Hermite matrix polynomial series, in particular

$$\mathcal{A}(x, t; A) := \sum_{n\geq 0} \frac{(-1)^n H_{2n+1}(x, A)}{(2n)!} t^{2n}, \mathcal{B}(x, t; A) := \sum_{n\geq 0} \frac{(-1)^n H_{2n+3}(x, A)}{(2n+1)!} t^{2n}, \tag{1}$$

which are a generalization of formulas [8, p.833]:

$$\sum_{n\geq 0} \frac{(-1)^n H_{2n}(x)}{(2n)!} t^{2n} = e^{t^2} \cos\left(xt\sqrt{A}\right), \sum_{n\geq 0} \frac{(-1)^n H_{2n+1}(x)}{(2n+1)!} t^{2n} = e^{t^2} \sin\left(xt\sqrt{A}\right), x \in \mathbb{R}, |t| < \infty \tag{2}$$

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

(obtained by replacing A by $\sqrt{2A}$ and taking $t = 1/\lambda$, $x = y/t$ in formulas (2.70) – (2.8) of [8]). We use formulas (1) to obtain new rational expansion in Hermite matrix polynomials of the matrix cosine.

The organization of the paper is as follows: Throughout this paper, we denote by $\mathbb{C}^{r\times r}$ the set of all the complex square matrices of size r. We denote by Θ and I, respectively, the zero and the identity matrix in $\mathbb{C}^{r\times r}$. If $A \in \mathbb{C}^{r\times r}$, we denote by $\sigma(A)$ the set of all the eigenvalues of A. We denote by $\lfloor x \rfloor$ the integer part of x and by $\lceil x \rceil$ the nearest integers to x towards infinity.

If $f(z), g(z)$ are holomorphic functions in an open set Ω of the complex plane, and if $\sigma(A) \subset \Omega$, we denote by $f(A), g(A)$, respectively, the image by the Riesz-Dunford functional calculus of the functions $f(z), g(z)$, respectively, acting on the matrix A, and $f(A)g(A) = g(A)f(A)$, see [12, p.558]. We say that matrix A is positive stable if $Re(z) > 0$ for every eigenvalue $z \in \sigma(A)$. In this case, let us denote $\sqrt{A} = A^{1/2} = \exp\left(\frac{1}{2}\log(A)\right)$ the image of the function $z^{1/2} = \exp\left(\frac{1}{2}\log(z)\right)$ by the Riesz-Dunford functional calculus, acting on the matrix A, where $\log(z)$ denotes the principal branch of the complex logarithm.

In this paper, we use consistent matrix norms. For example, in tests we use the $1-$norm of a matrix $A \in \mathbb{C}^{r\times r}$ defined by $\|A\|_1 = \sup_{x\neq 0} \frac{\|Ax\|_1}{\|x\|_1}$, where $\|\cdot\|_1$ denotes the vector 1-norm defined as $\|y\|_1 = |y_1| + \cdots + |y_r|$, $y \in \mathbb{C}^r$, see chapter 2 from [13]. For a positive stable matrix $A \in \mathbb{C}^{r\times r}$ the $n-$th Hermite matrix polynomial is defined in [17] by:

$$H_n(x,A) = n! \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k \left(\sqrt{2A}\right)^{n-2k}}{k!(n-2k)!} x^{n-2k}, \tag{3}$$

which satisfies the three-term matrix recurrence:

$$H_{-1}(x,A) = \Theta \ , \ H_0(x,A) = I \ , \ H_m(x,A) = x\sqrt{2A}H_{m-1}(x,A) - 2(m-1)H_{m-2}(x,A) \ , \ m \geq 1 \ . \tag{4}$$

The following result about upper bound of Hermite matrix polynomials was demonstrated in [3]:

$$\left.\begin{array}{rcl} \|H_{2n}(x,A)\|_2 & \leq & g_n(x) \ , \ n \geq 1 \\ \|H_{2n+1}(x,A)\|_2 & \leq & |x| \left\| \left(\frac{A}{2}\right)^{-\frac{1}{2}} \right\|_2 \frac{2g_n(x)}{n+1} \ , \ n \geq 0 \end{array}\right\} \ , \ g_n(x) = \frac{(2n+1)!2^{2n}}{n!} \exp\left(\frac{5}{2}\|A\|_2 x^2\right). \tag{5}$$

2 The new formulas

We intend to calculate the exact value of the matrix series $\mathcal{A}(x,t;A)$ and $\mathcal{B}(x,t;A)$ defined by (1). First we will prove that both matrix series are convergent. Taking into account (5) one gets

$$\left\| \frac{(-1)^n H_{2n+1}(x,A)}{(2n)!} t^{2n} \right\| = |x| \left\| \left(\frac{A}{2}\right)^{-\frac{1}{2}} \right\|_2 \frac{2g_n(x)}{(n+1)(2n)!} |t|^{2n}.$$

Since $\sum_{n\geq 0} \frac{g_n(x)}{(n+1)(2n)!}|t|^{2n}$ is convergent for $|t| < \infty$, the matrix series $\mathcal{A}(x,t;A)$ is convergent in any compact real interval. Analogously, taking into account (5) again one gets

$$\left\| \frac{(-1)^n H_{2n+3}(x,A)}{(2n+1)!} t^{2n+1} \right\| = \left\| \frac{(-1)^n H_{2(n+1)+1}(x,A)}{(2n+1)!} t^{2n+1} \right\| = |x| \left\| \left(\frac{A}{2}\right)^{-\frac{1}{2}} \right\|_2 \frac{2g_{n+1}(x)}{(n+2)(2n+1)!} |t|^{2n+1}.$$

Since $\sum_{n\geq 0} \frac{g_{n+1}(x)}{(n+2)(2n+1)!}|t|^{2n+1}$ is convergent for $|t| < \infty$, the matrix series $\mathcal{B}(x,t;A)$ is convergent in any compact real interval. Using now (4) and (2) one gets

$$\begin{aligned}
\mathcal{A}(x,t;A) &= x\sqrt{2A}\sum_{n\geq 0}\frac{(-1)^n H_{2n}(x,A)}{(2n)!}t^{2n} - 4\sum_{n\geq 1}\frac{(-1)^n n H_{2n-1}(x,A)}{(2n)!}t^{2n}\\
&= H_1(x,A)e^{t^2}\cos\left(xt\sqrt{A}\right) - 2t\sum_{n\geq 1}\frac{(-1)^n H_{2n-1}(x,A)}{(2n-1)!}t^{2n-1}\\
&= H_1(x,A)e^{t^2}\cos\left(xt\sqrt{A}\right) - 2t\sum_{n\geq 0}\frac{(-1)^{n+1} H_{2n+1}(x,A)}{(2n+1)!}t^{2n+1}\\
&= H_1(x,A)e^{t^2}\cos\left(xt\sqrt{A}\right) + 2te^{t^2}\sin\left(xt\sqrt{A}\right).
\end{aligned}$$

Working in a similar form and using (4) and (2), one gets that

$$\begin{aligned}
\mathcal{B}(x,t;A) &= x\sqrt{2A}\sum_{n\geq 0}\frac{(-1)^n H_{2n+2}(x,A)}{(2n+1)!}t^{2n+1} - 2\sum_{n\geq 0}\frac{(-1)^n (2n+2) H_{2n+1}(x,A)}{(2n+1)!}t^{2n+1}\\
&= x\sqrt{2A}\left(x\sqrt{2A}\sum_{n\geq 0}\frac{(-1)^n H_{2n+1}(x,A)}{(2n+1)!}t^{2n+1} - 2\sum_{n\geq 0}\frac{(-1)^n (2n+1) H_{2n}(x,A)}{(2n+1)!}t^{2n+1}\right)\\
&- 2\left(\sum_{n\geq 0}\frac{(-1)^n (2n+1) H_{2n+1}(x,A)}{(2n+1)!}t^{2n+1} + \sum_{n\geq 0}\frac{(-1)^n H_{2n+1}(x,A)}{(2n+1)!}t^{2n+1}\right)\\
&= x\sqrt{2A}\left(x\sqrt{2A}e^{t^2}\sin\left(xt\sqrt{A}\right) - 2t\sum_{n\geq 0}\frac{(-1)^n H_{2n}(x,A)}{(2n)!}t^{2n}\right)\\
&- 2\left(\sum_{n\geq 0}\frac{(-1)^n H_{2n+1}(x,A)}{(2n)!}t^{2n+1} + e^{t^2}\sin\left(xt\sqrt{A}\right)\right)\\
&= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 2xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right) - 2\sum_{n\geq 0}\frac{(-1)^n H_{2n+1}(x,A)}{(2n)!}t^{2n+1}.
\end{aligned}$$

Using again (4) one gets

$$\begin{aligned}
\mathcal{B}(x,t;A) &= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 2xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right)\\
&- 2\left(x\sqrt{2A}\sum_{n\geq 0}\frac{(-1)^n H_{2n}(x,A)}{(2n)!}t^{2n+1} - 2\sum_{n\geq 1}\frac{(-1)^n (2n) H_{2n-1}(x,A)}{(2n)!}t^{2n+1}\right)\\
&= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 2xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right)\\
&- 2\left(xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right) - 2\sum_{n\geq 1}\frac{(-1)^n H_{2n-1}(x,A)}{(2n-1)!}t^{2n+1}\right)\\
&= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 4xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right) + 4t^2\sum_{n\geq 1}\frac{(-1)^n H_{2n-1}(x,A)}{(2n-1)!}t^{2n-1}.
\end{aligned}$$

Rearranging indexes in the last series, we have

$$\begin{aligned}
\mathcal{B}(x,t;A) &= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 4xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right) - 4t^2\sum_{m\geq 0}\frac{(-1)^m H_{2m+1}(x,A)}{(2m+1)!}t^{2m+1}\\
&= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I\right) - 4xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right) - 4t^2e^{t^2}\sin\left(xt\sqrt{A}\right)\\
&= e^{t^2}\sin\left(xt\sqrt{A}\right)\left(2x^2A - 2I - 4t^2I\right) - 4xt\sqrt{2A}e^{t^2}\cos\left(xt\sqrt{A}\right).
\end{aligned}$$

By (3), we have that $H_1(x,A)=\sqrt{2A}x$, $H_2(x,A)=2x^2A-2I$, and we can write the last expression in the form

$$\mathcal{B}(x,t;A):=e^{t^2}\left(H_2(x,A)-4t^2I\right)\sin\left(xt\sqrt{2A}\right)-4tH_1(x,A)e^{t^2}\cos\left(xt\sqrt{2A}\right).$$

Form the previous comments, the following result has been probed:

Lemma 2.1 *Let $A\in\mathbb{C}^{r\times r}$ be a positive stable matrix. Then*

$$\begin{aligned}\mathcal{A}(x,t;A):&=e^{t^2}\left[H_1(x,A)\cos\left(xt\sqrt{2A}\right)+2t\sin\left(xt\sqrt{2A}\right)\right],\\ \mathcal{B}(x,t;A):&=e^{t^2}\left[\left(H_2(x,A)-4t^2I\right)\sin\left(xt\sqrt{2A}\right)-4tH_1(x,A)\cos\left(xt\sqrt{2A}\right)\right].\end{aligned}\tag{6}$$

3 On new rational-polynomial Hermite matrix expansions for the matrix cosine

Let $A\in\mathbb{C}^{r\times r}$ be a positive stable matrix, then the matrix polynomial $H_1(x,A)=\sqrt{2A}x$ is invertible. Substituting $\sin\left(xt\sqrt{2A}\right)$ given in (2) into the expression of $\mathcal{A}(x,t;A)$ given in (6) we obtain a new rational expression for the matrix cosine in terms of Hermite matrix polynomials:

$$\begin{aligned}\cos\left(xt\sqrt{2A}\right)&=e^{-t^2}\left(\sum_{n\geq 0}\frac{(-1)^nH_{2n+1}(x,A)}{(2n)!}\left(1-\frac{2t^2}{2n+1}\right)t^{2n}\right)\left[H_1(x,A)\right]^{-1},\\ &\quad x\in\mathbb{R},|t|<+\infty.\end{aligned}\tag{7}$$

Substituting $\cos\left(xt\sqrt{2A}\right)$ given by (2) into the expression of $\mathcal{B}(x,t;A)$ given in (6), the expression obtained is reduced to that given in (2). On the other hand, replacing the expression of $\sin\left(xt\sqrt{2A}\right)$ given in (2) into $\mathcal{B}(x,t;A)$, we have another new rational expression for the matrix cosine in terms of Hermite matrix polynomials:

$$\begin{aligned}&\cos\left(xt\sqrt{2A}\right)\\ &=\frac{-e^{-t^2}}{4}\left[\sum_{n\geq 0}\frac{(-1)^nH_{2n+3}(x,A)}{(2n+1)!}t^{2n}-\underbrace{\left(H_2(x,A)-4t^2I\right)\left(\sum_{n\geq 0}\frac{(-1)^nH_{2n+1}(x,A)}{(2n+1)!}t^{2n}\right)}_{\star}\right]\left[H_1(x,A)\right]^{-1},\\ &\quad x\in\mathbb{R},|t|<+\infty.\end{aligned}\tag{8}$$

We always have one more product of matrices in formula (8), the matrix product remark by ($\star$). Due to the importance of reducing the number of matrix products, see [22] for more details, we will focus on the expansion (7).

Substituting in (7) the matrix A by matrix $A^2/2$ we avoid the square roots of matrices. In addition, from (3), it follows that

$$A^{-1}H_{2n+1}\left(x,\frac{1}{2}A^2\right)=(2n+1)!\sum_{k=0}^{n}\frac{(-1)^kx^{2(n-k)}A^{2(n-k)}}{k!(2(n-k)+1)!}=\widetilde{H}_{2n+1}\left(x,\frac{1}{2}A^2\right),\tag{9}$$

so the right side of (9) is still defined in the case where the matrix A is singular. In this way, we can re-write the relation (7) in terms of the matrix polynomial $\widetilde{H}_{2n+1}\left(x,\frac{1}{2}A^2\right)$, and taking $x=\lambda,\lambda\neq 0,t=1/\lambda$, we obtain

Table 1: Values of z_m.

m	2	4	6	9	12	16
z_m	$1.4440e-5$	$7.70884e-3$	$1.3286e-1$	$1.3292e0$	5.2844	$1.7679e1$

$$\cos(A) = \frac{e^{-\frac{1}{\lambda^2}}}{\lambda}\sum_{n\geq 0}\frac{(-1)^n\widetilde{H}_{2n+1}\left(\lambda,\frac{1}{2}A^2\right)}{(2n)!\lambda^{2n}}\left(1-\frac{2}{(2n+1)\lambda^2}\right). \tag{10}$$

Note that expansion given in (10) is really a polynomial series in matrix A. Truncating the given series (10) until order m, we obtain the approximation $C_m(\lambda, A) \approx \cos(A)$ defined by:

$$C_m(\lambda, A) = \frac{e^{-\frac{1}{\lambda^2}}}{\lambda}\sum_{n=0}^{m}\frac{(-1)^n\widetilde{H}_{2n+1}\left(\lambda,\frac{1}{2}A^2\right)}{(2n)!\lambda^{2n}}\left(1-\frac{2}{(2n+1)\lambda^2}\right) \approx \cos(A), 0<|\lambda|<+\infty. \tag{11}$$

for any matrix $A \in \mathbb{C}^{r\times r}$.

Working analogously to the proof of the formula (3.3) of [6], you have

$$\left\|\widetilde{H}_{2n+1}\left(x,\frac{1}{2}A^2\right)\right\|_2 \leq (2n+1)!\frac{e\sinh\left(x\left\|A^2\right\|_2^{1/2}\right)}{\left\|A^2\right\|_2^{1/2}}. \tag{12}$$

We can perform the following approximation of the approximation error:

$$\begin{aligned}\left\|\cos(A)-C_m(\lambda,A)\right\|_2 &\leq \frac{e^{-\frac{1}{\lambda^2}}}{\lambda}\sum_{n\geq m+1}\frac{\left\|\widetilde{H}_{2n+1}\left(\lambda,\frac{1}{2}A^2\right)\right\|_2}{(2n)!\lambda^{2n}}\left|1-\frac{2}{(2n+1)\lambda^2}\right| \\ &\leq \frac{e^{1-\frac{1}{\lambda^2}}\sinh\left(\lambda\left\|A^2\right\|_2^{1/2}\right)}{\lambda\left\|A^2\right\|_2^{1/2}}\sum_{n\geq m+1}\frac{2n+1}{\lambda^{2n}}\left|1-\frac{2}{(2n+1)\lambda^2}\right|.\end{aligned} \tag{13}$$

Taking $\lambda > \sqrt{2}$ it is follows that $\frac{2}{(2n+1)\lambda^2} < 1$, and one gets

$$\sum_{n\geq m+1}\frac{2n+1}{\lambda^{2n}}\left(1-\frac{2}{(2n+1)\lambda^2}\right) = \frac{2+(2m+3)\lambda^2(\lambda^2-1)}{\lambda^{2m+2}\left(\lambda^2-1\right)^2},$$

thus from (13) we finally obtain:

$$\left\|\cos(A)-C_m(\lambda,A)\right\|_2 \leq \frac{e^{1-\frac{1}{\lambda^2}}\sinh\left(\lambda\left\|A^2\right\|_2^{1/2}\right)\left(2+(2m+3)\lambda^2(\lambda^2-1)\right)}{\left\|A^2\right\|_2^{1/2}\lambda^{2m+3}\left(\lambda^2-1\right)^2}. \tag{14}$$

From this expression (14) we derived the optimal values $(\lambda_m; z_m)$ such that

$$z_m = \max\left\{z=\left\|A^2\right\|_2;\ \frac{e^{1-\frac{1}{\lambda^2}}\sinh\left(\lambda z^{1/2}\right)\left(2+(2m+3)\lambda^2(\lambda^2-1)\right)}{z^{1/2}\lambda^{2m+3}\left(\lambda^2-1\right)^2} < u\right\}$$

where u is the unit roundoff in IEEE double precision arithmetic, $u = 2^{-53}$. The optimal values of m, z and λ have been obtained through a MATLAB program. The results are given in the table 1.

4 The proposed MATLAB implementations

The matrix cosine can be computed for $A \in \mathbb{C}^{n\times n}$ by the expression

$$P_m(B) = \sum_{i=0}^{m} p_i B^i, \tag{15}$$

where $B = A^2$, and p_i is the coefficient polynomial of Hermite expression (11), or $p_i = \frac{(-1)^i}{(2i)!}$, if the Taylor approximation is used. Since Hermite and Taylor series are accurate only near the origin, in algorithms that use these approximations the norm of matrix B must be reduced by scaling the matrix. Once the cosine of scaled matrix has been computed, the approximation of $\cos(A)$ is recovered by means of the double angle formula $\cos(2X) = 2\cos^2(X) - I$. Algorithm 1 shows a general algorithm for computing the matrix cosine based on Taylor approximation. By using the fact that $\sin(A) = \cos(A - \frac{\pi}{2}I)$, Algorithm 1 also can be easily used to compute the matrix sine.

Algorithm 1 Given a matrix $A \in \mathbb{C}^{n\times n}$, this algorithm computes $C = \cos(A)$ by Taylor/Hermites series.

1: Select adequate values of m and s ▷ Phase I
2: $B = 4^{-s}A^2$
3: $C = P_m(B)$ ▷ Phase II: Compute Taylor/Hermite approximation
4: **for** $i = 1 : s$ **do** ▷ Phase III: Recovering $\cos(A)$
5: $\quad C = 2C^2 - I$
6: **end for**

In Phase I of Algorithm 1, m and s can be calculated so that the Hermite or Taylor approximations of the scaled matrix is computed accurately and efficiently. In this phase some powers B^i, $i \geq 2$, are usually computed for estimating m and s and if so they are used in Phase II.
Phase II consists of computing the approximations (11) or (15). Taylor matrix polynomial approximation (15), 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 the Paterson-Stockmeyer's method [19] choosing m from the set

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

where the elements of $\mathbb{M}$ are denoted as $m_1, m_2, m_3, \dots$ The algorithm computes first the powers B^i, $2 \leq i \leq q$ not computed in the previous phase, being $q = \lceil \sqrt{m_k} \rceil$ or $q = \lfloor \sqrt{m_k} \rfloor$ an integer divisor of m_k, $k \geq 1$, both values giving the same cost in terms of matrix products. Therefore, (15) can be computed efficiently as

$$\begin{aligned} P_{m_k}(B) \;=& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (16)\\ (((p_{m_k}B^q \;+&\; p_{m_k-1}B^{q-1} + p_{m_k-2}B^{q-2} + \cdots + p_{m_k-q+1}B + p_{m_k-q}I)B^q \\ +&\; p_{m_k-q-1}B^{q-1} + p_{m_k-q-2}B^{q-2} + \cdots + p_{m_k-2q+1}B + p_{m_k-2q}I)B^q \\ +&\; p_{m_k-2q-1}B^{q-1} + p_{m_k-2q-2}B^{q-2} + \cdots + p_{m_k-3q+1}B + p_{m_k-3q}I)B^q \\ \dots& \\ +&\; p_{q-1}B^{q-1} + p_{q-2}B^{q-2} + \cdots + p_1 B + p_0 I. \end{aligned}$$

We made two implementations based on the algorithms `cosher` [5] and `cosmtay` [20]. The MATLAB implementation `cosherm` is a modification of the MATLAB's code `cosher`, taking the values of m, z and λ of Section 3 instead of these values from [5]. The MATLAB's implementation `cosmtayher` is a modification of the MATLAB's code `cosmtay`, taking the values of z_m instead of Θ_m to obtain the approximation degree m and the scaling factor s of the matrix A, and replacing the Taylor polynomial of `cosmtay` by the "*Hermite*" matrix polynomial obtained from λ_m.

5 Numerical experiments

The following MATLAB implementations (version: 9.1.0.441655, R2016b) have been compared:

- `cosm`. Code based on the Padé rational approximation for the matrix cosine [1].

- `cosmtay`. Code based on the Taylor series for the matrix cosine [20].
- `cosherm`. New code for computing the matrix cosine based on the new developments of Hermites matrix polynomials (11) and the algorithm `cosmtayher` from [5].
- `cosmtayher`. New code for computing the matrix cosine based on the new developments of Hermites matrix polynomials (11) and the algorithm `cosmtayher` from [20].

In tests, fifty eight 128×128 matrices from Toolbox [14] and Eigtool [24] packages have been used. These matrices have been chosen because they have more varied and significant characteristics. For the accurate of MATLAB codes we have calculated the relative errors as

$$\frac{\left\|\tilde{X} - \cos(A)\right\|_1}{\|\cos(A)\|_1},$$

where $\tilde{X}$ is the computed matrix cosine.

We only show the accurate and computational costs of `cosmtayher` compared to the accurate and computational costs of the other implementations, since `cosmtayher` is the implementation that presents the best performances. To compare the computational costs, the number of products of each code has been calculated, since this operation has the highest computational cost. The resolution of linear systems that appears in the code based on Padé approximation `cosm` has been calculated as $4/3$ products, because from a computational point of view, the cost of that operation compared to the product of matrices is approximately equal to $4/3$, see Table C.1 from [15, p. 336].

Table 2 shows the percentage of cases in which the relative errors of MATLAB codes `cosm`, `cosmtay` and `cosherm` are, respectively, lower and greater than the relative errors of `cosmtayher`. Table 3 show the matrix products of the four MATLAB codes.

Table 2: Error comparative between `cosmtayher` and the other MATLAB implementations.

	E(Ex)$<$E(`cosmtayher`)	E(Ex)$>$E(`cosmtayher`)
Ex$\equiv$`cosm`	21.05%	78.95%
Ex$\equiv$`cosmtay`	33.33%	66.67%
Ex$\equiv$`cosherm`	47.37%	52.63%

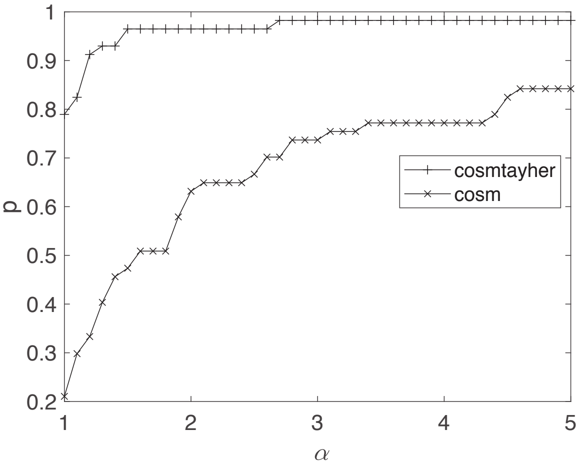
Table 3: Matrix products of the four MATLAB implementations.

`cosm`	`cosmtay`	`cosherm`	`cosmtayher`
621	508	541	355

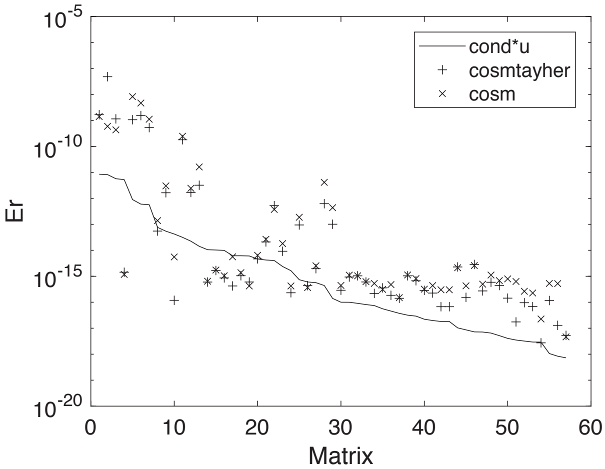
Figure 1 shows the normwise relative error and the performance profiles graphics [11]. The solid line of Subfigures 1a, 1c and 1e is the function $k_{\cos}u$, where $k_{\cos}$ is the condition number of matrix cosine function [15, Chapter 3] and $u = 2^{-53}$ is the unit roundoff in the double precision floating-point arithmetic.

According to the results obtained, we can outline the following conclusions:

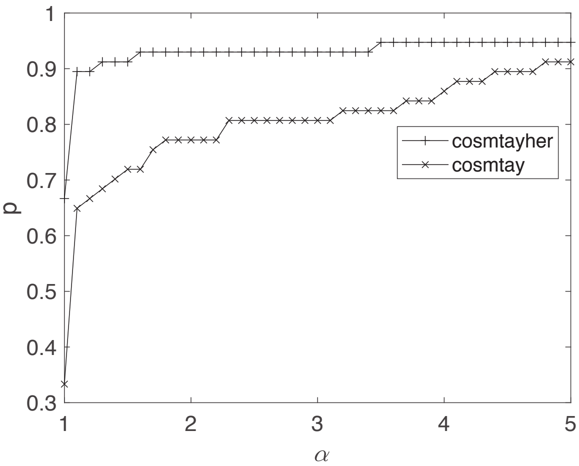
- Subfigures 1b, 1d and 1f show that the four implementations have a similar numerical stability.
- The MATLAB code `cosmtayher` has a lower computational cost than the other MATLAB codes (see Table 3).
- In general, MATLAB code `cosmtayher` is more accurate than the other MATLAB codes (see Table 2 and Subfigures 1a, 1c and 1e).



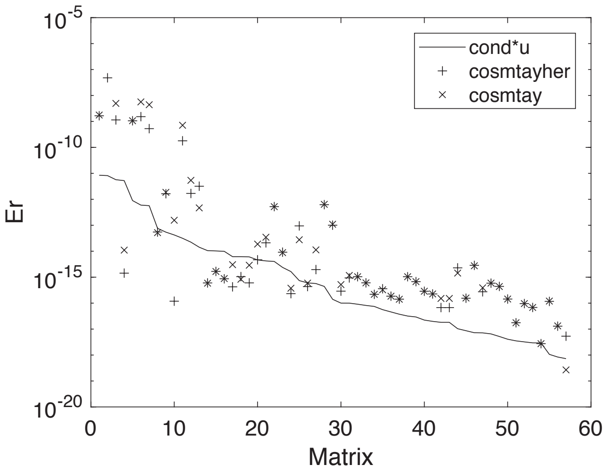
(a) Normwise relative errors of `comtayher-cosm`.



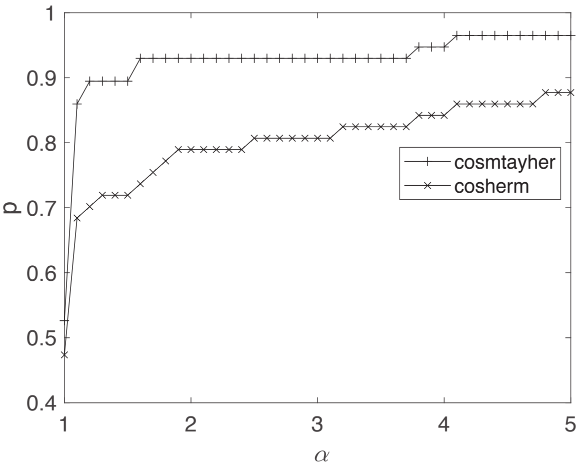
(b) Perfomance profile of `comtayher-cosm`.



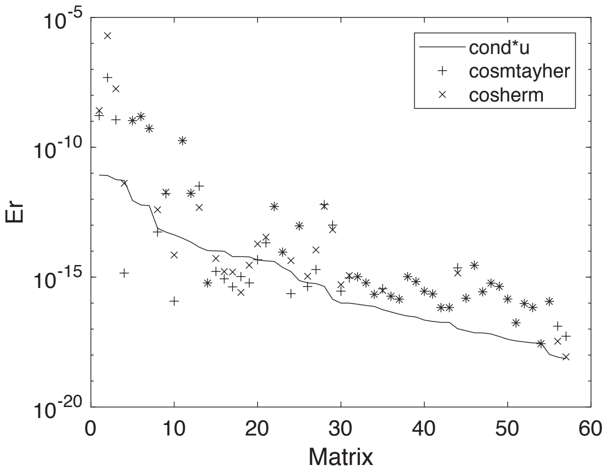
(c) Normwise relative errors of `comtayher-cosmtay`.



(d) Perfomance profile of `comtayher-cosmtay`.



(e) Normwise relative errors of `comtayher-cosherm`.



(f) Perfomance profile of `comtayher-cosherm`.

Figure 1: Normwise relative errors and the performance profiles.

6 Conclusions

In this work we have developed more accurate MATLAB implementations (`cosherm`,`cosmtayherm`) based on the new Hermite series (11) that improve others from the state of the art for the computation of the matrix cosine function (`cosm`,`cosmtay`). Among the new implementations it is worth highlighting the MATLAB implementation `cosmtayherm`, because it has a lower computational cost than the other MATLAB codes.

References

[1] A. H. Al-Mohy, N. J. Higham, and S. D. Relton. New algorithms for computing the matrix sine and cosine separately or simultaneously. *SIAM Journal on Scientific Computing*, 37(1):A456–A487, 2015.

[2] P. Alonso, J. Ibáñez, J. Sastre, J. Peinado, and E. Defez. Efficient and accurate algorithms for computing matrix trigonometric functions. *Journal of Computational and Applied Mathematics*, 309:325–332, 2017.

[3] E. Defez, A. Hervás, L. Jódar, and A. Law. Bounding Hermite matrix polynomials. *Mathematical and Computer Modelling*, 40(1):117–125, 2004.

[4] E. Defez and L. Jódar. Some applications of the Hermite matrix polynomials series expansions. *Journal of Computational and Applied Mathematics*, 99(1):105–117, 1998.

[5] E. Defez, J. Sastre, J. Ibáñez, and P. Ruiz. Computing matrix functions arising in engineering models with orthogonal matrix polynomials. *Mathematical and Computer Modelling*, pages 1738–1743, 2011. . doi:10.1016/j.mcm.2011.11.022.

[6] E. Defez, J. Sastre, J. Ibáñez, and J. Peinado. Solving engineering models using hyperbolic matrix functions. *Applied Mathematical Modelling*, 40(4):2837–2844, 2016.

[7] E. Defez, J. Sastre, J. Ibáñez, J. Peinado, and M. M. Tung. A method to approximate the hyperbolic sine of a matrix. *International Journal of Complex Systems in Science*, 4(1):41–45, 2014.

[8] E. Defez, J. Sastre, J. Ibáñez, and P. Ruiz. Computing matrix functions solving coupled differential models. *Mathematical and Computer Modelling*, 50(5):831–839, 2009.

[9] E. Defez, J. Sastre, J. Ibáñez, and P. Ruiz. Computing matrix functions arising in engineering models with orthogonal matrix polynomials. *Mathematical and Computer Modelling*, 57(7):1738–1743, 2013.

[10] E. Defez and M. M. Tung. A new type of hermite matrix polynomial series. *Quaestiones Mathematicae*, pages 1–8, 2017.

[11] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[12] N. Dunford and J. T. Schwartz. Linear operators, vol. I. *Interscience, New York*, 1958.

[13] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins University Press, 1996.

[14] N. J. Higham. *The test matrix toolbox for MATLAB (Version 3.0).* University of Manchester Manchester, 1995.

[15] N. J. Higham. *Functions of Matrices: Theory and Computation.* SIAM, Philadelphia, PA, USA, 2008.

[16] F. Hong-Yi and Z. De-Hui. New generating function formulae of even-and odd-Hermite polynomials obtained and applied in the context of quantum optics. *Chinese Physics B*, 23(6):060301, 2014.

[17] J. Jódar and R. Company. Hermite matrix polynomials and second order matrix differential equations. *Approximation Theory and its Applications*, 12(2):20–30, 1996.

[18] L. Jódar and E. Defez. On Hermite matrix polynomials and Hermite matrix functions. *Approximation Theory and its Applications*, 14(1):36–48, 1998.

[19] M. S. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM Journal on Computing*, 2(1):60–66, 1973.

[20] J. Sastre, J. Ibáñez, P. Alonso, J. Peinado, and E. Defez. Two algorithms for computing the matrix cosine function. *Applied Mathematics and Computation*, 312:66–77, 2017.

[21] J. Sastre, J. Ibáñez, E. Defez, and P. Ruiz. Efficient orthogonal matrix polynomial based method for computing matrix exponential. *Applied Mathematics and Computation*, 217(14):6451–6463, 2011.

[22] J. Sastre, J. Ibáñez, E. Defez, and P. Ruiz. New scaling-squaring Taylor algorithms for computing the matrix exponential. *SIAM Journal on Scientific Computing*, 37(1):A439–A455, 2015.

[23] J. Sastre, J. Ibáñez, P. Ruiz, and E. Defez. Efficient computation of the matrix cosine. *Applied Mathematics and Computation*, 219(14):7575–7585, 2013.

[24] TG Wright. Eigtool, version 2.1. *URL: web. comlab. ox. ac. uk/pseudospectra/eigtool*, 2009.