Computing matrix functions solving coupled differential models *

Emilio Defez⋆, Jorge Sastre†, Javier Ibáñez♮, Pedro A. Ruiz♮

⋆ Instituto de Matemática Multidisciplinar.
† Instituto de Telecomunicaciones y Aplicaciones Multimedia.
♮ Instituto de Aplicaciones de las Tecnologías de la Información y de las Comunicaciones Avanzadas.

Universidad Politécnica de Valencia,
Camino de Vera s/n, 46022 Valencia, España.
edefez@imm.upv.es, jorsasma@iteam.upv.es, {jjibanez, pruiz }@dsic.upv.es

Abstract

In this paper a modification of the method proposed in [1] for computing matrix sine and cosine based on Hermite matrix polynomial expansions is presented. An algorithm and illustrative examples demonstrate the performance of the new proposed method.

Keywords. Hermite matrix polynomial, Matrix Sine and Matrix Cosine, computation, error bound.

1 Introduction.

It is well known that the wave equation

$$v^2 \frac{\partial^2 \psi}{\partial x^2} = \frac{\partial^2 \psi}{\partial t^2} \ , \tag{1.1}$$

plays an important role in many areas of engineering and applied sciences. The matrix differential problem

$$Y''(t) + AY(t) = 0 \ , \ Y(0) = Y_0 \ , \ Y'(0) = Y_1 \ , \tag{1.2}$$

where A is a matrix and Y_0 and Y_1 are vectors, arises from spatially semi-discretization of the wave equation (1.1), see [2]. Matrix problem (1.2) has the exact solution

$$Y(t) = \cos\left(\sqrt{A}t\right)Y_0 + \left(\sqrt{A}\right)^{-1} \sin\left(\sqrt{A}t\right)Y_1, \tag{1.3}$$

where $\sqrt{A}$ denotes any square root of a non-singular matrix A (see *e.g.* equation 1.2 of [3]). More general problems of type (1.2), 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 [4]. Thus, trigonometric matrix functions play an important role in second order differential systems, similar to matrix exponentials in first order differential problems.

A general algorithm for computing the matrix cosine which uses rational approximations and the double-angle formula $\cos(2A) = 2\cos^2(A) - I$ was proposed by Serbin and Blalock [2]. Higham

*Acknowledgments. This work has been partially supported by the *Generalitat Valenciana* GV/2007/009.

in [3, 5, 6] developed a particular version of this algorithm based on the Padé approximation including truncation and rounding error analysis.

In this paper, that may be regarded as a continuation of [1], we use Hermite matrix polynomial expansions of the matrix cosine and sine in order to perform a very accurate and competitive method for computing them compared to the results given by the function *funm* of MATLAB. The implementations have been tested on an Intel Core 2 Duo T5600 with 2 GB main memory, using 7.5 (R2007b) MATLAB version.

This paper is organized as follows. Section 2 summarizes previous results of Hermite matrix polynomials and includes a new Hermite series expansion of the matrix sine and cosine. Section 3 deals with the Hermite matrix polynomial series expansion of $\cos(At)$ and $\sin(At)$ for an arbitrary matrix as well as with its finite series truncation with a prefixed accuracy in a bounded domain, and an algorithm of the method is given. Section 4 deals with a selection of examples in order to investigate the accuracy of the new method proposed here. Finally, conclusions are presented in section 5.

Throughout this paper, $[x]$ denotes the integer part of x. The matrices I_r and $\theta_{r\times r}$ in $\mathbb{C}^{r\times r}$ denote the matrix identity and the null matrix of order r, respectively. Following [7], for a matrix A in $\mathbb{C}^{r\times r}$, its infinite-norm will be denoted by $\|A\|_\infty$ and its 2-norm will be denoted by $\| A \|_2$. Finally, if $A(k,n)$ are matrices in $\mathbb{C}^{r\times r}$ for $n \geq 0, k \geq 0$, from [1] it follows that

$$\sum_{n\geq 0}\sum_{k\geq 0} A(k,n) \; = \; \sum_{n\geq 0}\sum_{k=0}^{n} A(k,n-k) \; . \tag{1.4}$$

2 Hermite matrix polynomials series expansions of matrix sine and matrix cosine.

For the sake of clarity in the presentation of the following results we recall some properties of Hermite matrix polynomials which have been established in [1] and [8]. From (3.4) of [8, p. 25] the nth Hermite matrix polynomial satisfies

$$H_n\left(x, \frac{1}{2}A^2\right) = n! \sum_{k=0}^{[\frac{n}{2}]} \frac{(-1)^k \, (xA)^{n-2k}}{k!(n-2k)!} \; , \tag{2.1}$$

for an arbitrary matrix A in $\mathbb{C}^{r\times r}$. Taking into account the three-term recurrence relationship (3.12) of [8, p. 26], it follows that

$$\left.\begin{array}{c} H_n\left(x, \frac{1}{2}A^2\right) = xAH_{n-1}\left(x, \frac{1}{2}A^2\right) - 2(n-1)H_{n-2}\left(x, \frac{1}{2}A^2\right) \quad , \quad n \geq 1 \\ H_{-1}(x, \frac{1}{2}A^2) = \theta_{r\times r} \quad , \quad H_0(x, \frac{1}{2}A^2) = I_r \end{array}\right] \; , \tag{2.2}$$

and from its generating function in (3.1) and (3.2) [8, p. 24] one gets

$$e^{xtA-t^2 I} = \sum_{n\geq 0} H_n\left(x, \frac{1}{2}A^2\right) t^n/n!, \quad |t| < \infty, \tag{2.3}$$

where $x, t \in \mathbb{C}$. The nth scalar Hermite polynomial is given by [9, p. 60]

$$H_n(x) = n! \sum_{k=0}^{[\frac{n}{2}]} \frac{(-1)^k \, (2x)^{n-2k}}{k!(n-2k)!} \; , \; n \geq 0 \; , \tag{2.4}$$

which coincide with the $n-$th matrix Hermite polynomial (2.1) when $r=1$ and $A=2$.

Taking $y=tx$ and $\mu=1/t$ in (2.3) it follows that

$$e^{Ay}=\mathrm{e}^{\frac{1}{\mu^2}}\sum_{n\geq 0}\frac{1}{\mu^n n!}H_n\left(\mu y,\frac{1}{2}A^2\right),\ \mu\in\mathbb{C},\quad y\in\mathbb{C},\quad A\in\mathbb{C}^{r\times r}\ . \tag{2.5}$$

Now, we look for the Hermite matrix polynomials series expansion of the matrix cosine $\cos(Ax)$. Given an arbitrary matrix $A\in\mathbb{C}^{r\times r}$, with

$$\cos(Ay)=\frac{e^{iAy}+e^{-iAy}}{2}$$

and using (2.5) in combination with [8, p. 25], it follows that

$$H_n(-x,A)=(-1)^n H_n(x,A)\ .$$

Thus, one gets

$$\cos(Ay)\ =e^{\frac{1}{\mu^2}}\sum_{n\geq 0}\frac{1}{\mu^{2n}(2n)!}H_{2n}\left(iy\mu,\frac{1}{2}A^2\right)\ . \tag{2.6}$$

Taking $\lambda=i\mu$ in (2.6), we obtain the looked for expression:

$$\cos(Ay)\ =e^{-\frac{1}{\lambda^2}}\sum_{n\geq 0}\frac{(-1)^n}{\lambda^{2n}(2n)!}H_{2n}\left(y\lambda,\frac{1}{2}A^2\right)\ . \tag{2.7}$$

In a similar form, taking into account that

$$\sin(Ay)=\frac{e^{iAy}-e^{-iAy}}{2i}\ ,$$

it follows that

$$\sin(Ay)\ =e^{-\frac{1}{\lambda^2}}\sum_{n\geq 0}\frac{(-1)^n}{\lambda^{2n+1}(2n+1)!}H_{2n+1}\left(y\lambda,\frac{1}{2}A^2\right). \tag{2.8}$$

Remark 2.1 *Observe that when $\lambda=1$, expressions (2.7) and (2.8) are formulae (19) and (20) of [1, p. 109].*

Denoting by $C_N(A,\lambda)$ the Nth partial sum of series (2.7) for $y=1$, one gets

$$C_N(\lambda,A)=e^{-\frac{1}{\lambda^2}}\sum_{n=0}^{N}\frac{(-1)^n}{\lambda^{2n}(2n)!}H_{2n}\left(\lambda,\frac{1}{2}A^2\right)\approx\cos(A),\ \lambda\in\mathbb{C},\quad A\in\mathbb{C}^{r\times r}. \tag{2.9}$$

Observe that the case $\lambda=1$ corresponds with the matrix cosine approximation $C(A;1;N)$ given in [1]. Denoting by $S_N(A,\lambda)$ the Nth partial sum of series (2.8) for $y=1$, one gets

$$S_N(\lambda,A)=e^{-\frac{1}{\lambda^2}}\sum_{n=0}^{N}\frac{(-1)^n}{\lambda^{2n+1}(2n+1)!}H_{2n+1}\left(\lambda,\frac{1}{2}A^2\right)\approx\sin(A),\ \lambda\in\mathbb{C},\quad A\in\mathbb{C}^{r\times r}. \tag{2.10}$$

In Section 4 we shall see that the introduction of the additional parameter λ will improve the results given in [1].

3 Accurate and error bounds for cosine and sine approximation. Algorithm.

By (2.1) and (2.4), it follows that

$$\left\| H_n\left(x, \frac{1}{2}A^2\right)\right\|_2 \leq \sum_{k=0}^{[n/2]} \frac{n!\,(|x|\,\|A\|_2)^{n-2k}}{k!(n-2k)!} \ , \tag{3.1}$$

and thus

$$\left\| H_{2n}\left(\lambda, \frac{1}{2}A^2\right)\right\|_2 \leq \sum_{k=0}^{n} \frac{(2n)!\,(\lambda\,\|A\|_2)^{2(n-k)}}{k!(2(n-k))!} \ . \tag{3.2}$$

Using (1.4), the following expression holds

$$\sum_{n\geq 0}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!\,(2(n-k))!} \ = \ \cosh\left(\|A\|_2\right)e^{\frac{1}{\lambda^2}} \ . \tag{3.3}$$

Taking the approximate value $C_N(\lambda, A)$ given by (2.9) and taking into account (3.2), it follows that

$$\begin{aligned}
\|\cos(A) - C_N(\lambda, A)\|_2 \ &\leq \ \mathrm{e}^{-\frac{1}{\lambda^2}} \sum_{n\geq N+1} \frac{1}{\lambda^{2n}(2n)!}\left\| H_{2n}\left(\lambda, \frac{1}{2}A^2\right)\right\|_2 \\
&\leq \ \mathrm{e}^{-\frac{1}{\lambda^2}} \sum_{n\geq N+1}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!(2(n-k))!} \\
&= \ \mathrm{e}^{-\frac{1}{\lambda^2}} \left[\sum_{n\geq 0}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!(2(n-k))!} - \sum_{n=0}^{N}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!(2(n-k))!}\right] \ .
\end{aligned}$$

Considering the previous expression, one gets an error bound for approximation (2.9):

$$\|\cos(A) - C_N(\lambda, A)\|_2 \leq \mathrm{e}^{-\frac{1}{\lambda^2}} \left[\cosh\left(\|A\|_2\right)e^{\frac{1}{\lambda^2}} - \sum_{n=0}^{N}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!(2(n-k))!}\right] \ . \tag{3.4}$$

Now, let $\varepsilon > 0$ be an *a priori* error bound. Using (3.4), if N is the first positive integer so that

$$\sum_{n=0}^{N}\sum_{k=0}^{n} \frac{\|A\|_2^{2(n-k)}}{\lambda^{2k}k!(2(n-k))!} \geq \ \cosh\left(\|A\|_2\right)e^{\frac{1}{\lambda^2}} - \varepsilon\,\mathrm{e}^{\frac{1}{\lambda^2}} \ , \tag{3.5}$$

from (3.4) and (3.5) one gets,

$$\|\cos(A) - C_N(\lambda, A)\|_2 \ \leq \ \varepsilon \ .$$

Summarizing, the next result, similar to theorem 3.1 of [1], has been proved:

Theorem 3.1 *Let A be a matrix in $\mathbb{C}^{r\times r}$ and let $\lambda > 0$. Let $\varepsilon > 0$. If N is the first positive integer so that inequality (3.5) holds. Then*

$$\|\cos(A) - C_N(\lambda, A)\|_2 \ \leq \ \varepsilon \ . \tag{3.6}$$

Furthermore, using that relation $\sin(A) = \cos\left(A - \frac{\pi}{2}I\right)$, it is possible avoid the computation of the matrix sine. On the other hand, we can obtain a similar result to theorem 3.1 for the case of the matrix sine:

Theorem 3.2 *Let A be a matrix in $\mathbb{C}^{r\times r}$ and let $\lambda > 0$. Let $\varepsilon > 0$. If N is the first positive integer so that inequality*

$$\sum_{n=0}^{N}\sum_{k=0}^{n}\frac{\|A\|_2^{2(n-k)+1}}{\lambda^{2k}k!(2(n-k)+1)!} \geq \sinh\left(\|A\|_2\right)e^{\frac{1}{\lambda^2}} - \varepsilon\, \mathrm{e}^{\frac{1}{\lambda^2}} \ ,$$

holds. Then, approximation $S_N(\lambda, A)$ given by (2.10) satisfies

$$\left\|\sin\left(A\right) - S_N(\lambda, A)\right\|_2 \ \leq \ \varepsilon \ . \tag{3.7}$$

Starting with expressions (2.9) and (2.10), it is possible to simultaneously compute the matrix cosine and sine using the following algorithm 1.

Algorithm 1 computes sine and cosine of a matrix by means of Hermite approximants.

Function $[C, S] = sincosher(A, N, \lambda)$
Inputs: Matrix $A \in \mathbb{R}^{r\times r}$; $2N+1$ is the order of the Hermite approximation ($N \in \mathbb{N}$) of sine/cosine function; parameter $\lambda \in \mathbb{R}$
Output: Matrices $C = \cos(A) \in \mathbb{R}^{r\times r}$ and $S = \sin(A) \in \mathbb{R}^{r\times r}$

$H_0 = I_r$
$H_1 = \lambda A$
$C = H_0$
$S = H_1/\lambda$
$aux = 1/\lambda$
for $n = 2 : 2N+1$ **do**
    $H = \lambda A H_1 - 2(n-1)H_0$
    $H_0 = H_1$;
    $H_1 = H$
    $aux = aux/(\lambda n)$
    if $\bmod\ (n, 4) < 2$ **then**
        if $\bmod\ (n, 2) == 0$ **then**
            $C = C + auxH$;
        else
            $C = C - auxH$;
        end if
    else
        if $\bmod\ (n, 2) == 0$ **then**
            $S = S + auxH$;
        else
            $S = S - auxH$;
        end if
    end if
end for
$C = e^{-1/l^2}C$
$S = e^{-1/l^2}S$

---

4 Numerical examples.

In this section we provide results for numerical experimentation of the computational method based on expansion (2.7) compared with the results given by the function *funm* of MATLAB. This function allows to compute general matrix functions by the Schur-Parlett algorithm, [10], and it is the only function that MATLAB has to compute matrix sine and cosine. The implementations have been tested on an Intel Core 2 Duo T5600 with 2 GB main memory, using 7.5 (R2007b) MATLAB version.

In the first example, we apply the computation of the matrix cosine of a matrix A treated in [1] using the expansion (2.7). Note that there are different possible choices for the parameter λ.

Example 4.1 *Let A be a matrix defined by*

$$A = \begin{pmatrix} 3 & -1 & 1 \\ 2 & 0 & 1 \\ 1 & -1 & 2 \end{pmatrix}, \tag{4.1}$$

with $\sigma(A) = \{1, 2\}$. Matrix A is non-diagonalizable. Using the minimal theorem [11, p. 571], see also [1], the exact value of $\cos(A)$ is

$$\begin{aligned}\cos(A) &= \begin{pmatrix} \cos(2) - \sin(2) & \sin(2) & -\sin(2) \\ -\cos(1) + \cos(2) - \sin(2) & \cos(1) + \sin(2) & -\sin(2) \\ -\cos(1) + \cos(2) & \cos(1) - \cos(2) & \cos(2) \end{pmatrix} \\ &= \begin{pmatrix} -1.325444263372824 & 0.909297426825682 & -0.909297426825682 \\ -1.865746569240964 & 1.449599732693821 & -0.909297426825682 \\ -0.956449142415282 & 0.956449142415282 & -0.4161468365471424 \end{pmatrix}.\end{aligned}$$

In [1], for an admissible error $\varepsilon = 10^{-5}$, we need $N = 15$ to provide the required accuracy. In practice, the number of terms required to obtain a prefixed accuracy uses to be smaller than the one provided by Theorem 3.1 of [1]. So for instance, taking $N = 9$ one gets:

$$C_9(1, A) = \begin{pmatrix} -1.3254444650245485 & 0.9092974459509594 & -0.9092974459509594 \\ -1.8657468968644513 & 1.4495998777908623 & -0.9092974459509594 \\ -0.9564494509134919 & 0.9564494509134919 & -0.4161470190735891 \end{pmatrix},$$

and

$$\|\cos(A) - C_9(1, A)\|_2 = 7.995228661905607 \times 10^{-7}.$$

We will compare these results obtained letting $\lambda = 1$ in Theorem 3.1 of [1] with the new Theorem 3.1. Taking $\lambda = 2000$, using Theorem 3.1 we need $N = 10$ to obtain the same prefixed accuracy. Again, the number of terms required to obtain a prefixed accuracy uses to be smaller than the one provided by (3.6). For instance, taking $N = 7$ one gets

$$C_7(2000, A) = \begin{pmatrix} -1.3254442633775207 & 0.9092974268299509 & -0.9092974268299509 \\ -1.8657465692456603 & 1.4495997326980907 & -0.9092974268299509 \\ -0.9564491424157093 & 0.9564491424157093 & -0.41614683654756973 \end{pmatrix},$$

and

$$\|\cos(A) - C_7(2000, A)\|_2 = 7.717270333884585 \times 10^{-8}.$$

The choice of parameter λ can still be refined. For example, taking $\lambda = 4.1$ one gets

$$\|\cos(A) - C_7(4.1, A)\|_2 = 7.098351906265066 \times 10^{-10}.$$

Figure 1 presents the error 2-norm of approximation (2.7) for $N = 8$ fixed and $\lambda \in]0, 25]$. This figure illustrates how the error norm depends on the varying parameter λ and it becomes evident that an adequate choice of λ may provide results with higher accuracy.

Figure 2 shows the 2−norm error bound of $C_N(\lambda, A)$ for the fixed value of $\lambda = 4.1$ varying N. For $N = 10$, we obtain

$$\|\cos(A) - C_{10}(4.1, A)\|_2 = 1.7763568394002505 \times 10^{-15}.$$

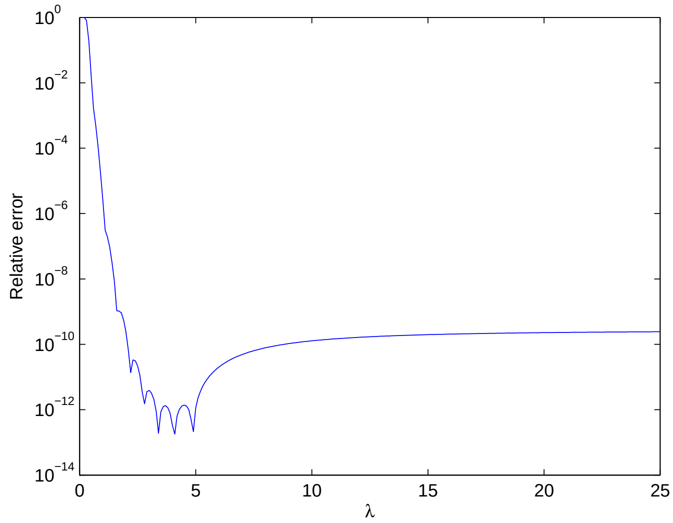


Figure 1: For $N = 8$ fixed and varying λ.

Example 4.2 *In this experiment we consider* 100 *random matrices of the form*

$$A = PDP^{-1}, \tag{4.2}$$

where D *is a diagonal matrix with uniform random values in the interval* $[-5, 5]$ *and* P *is a matrix with uniform random values in the same interval. The dimensions of all matrices are* 100×100*. We have computated the approximation of the matrix cosine* $C_N(\lambda, A)$ *and sine* $S_N(\lambda, A)$ *with* $N = 20$ *and the experimental value of* λ *was* $\lambda = 0.7936$*.*

It is well known that the exact solutions are

$$\cos(A) = P\cos(D)P^{-1} \quad , \quad \sin(A) = P\sin(D)P^{-1} \ .$$

In the experiment each exact solution has been obtained at $256-$*digit precision using MATLAB's Symbolic Math Toolbox.*

Figure 3 shows the comparison between the relative errors of function `funm` *of MATLAB and series (2.7) with* $\lambda = 0.7936$ *using the infinite norm:*

$$E_r(x^\star) = \frac{\|x - x^\star\|_\infty}{\|x\|_\infty} \ . \tag{4.3}$$

The mean processing time for `funm` *was* 0.114550 *seconds and the mean processing time for the Hermite approximation was* 0.023535 *seconds. The first average time corresponds only to the computation of* $\cos(A)$ *using the function* `funm`*. The second value corresponds to the computation of* $\cos(A)$ *and* $\sin(A)$ *using Hermite expansion. Our proposed implementation was* 4.8672 *times faster. In the computation of* $\cos(A)$*, the Hermite method gave a smaller error than* `funm` *in* 70% *of the test cases. In the computation of* $\sin(A)$*, the Hermite method gave a smaller error than* `funm` *in* 67% *of the test cases.*

Example 4.3 *We consider 100 randomly matrices in the same conditions as in experiment 4.2. We have computed the approximation of the matrix cosine* $C_N(\lambda, A)$ *and sine* $S_N(\lambda, A)$ *with* $N = 25$*.*

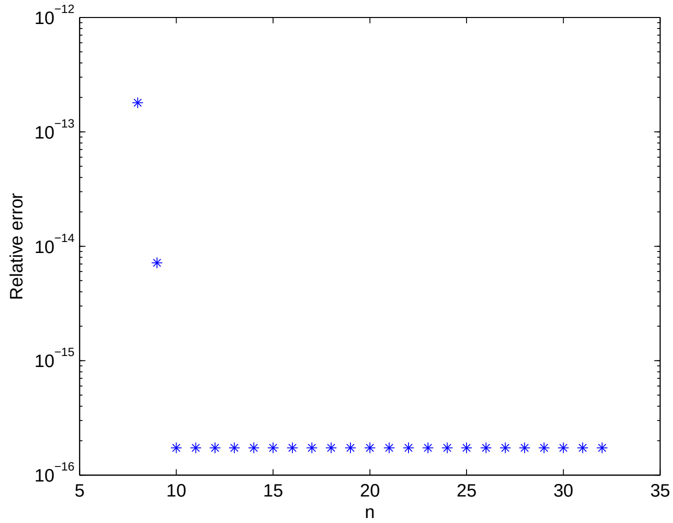


Figure 2: Relative error of Hermite series (2.7) for example 4.1 for $\lambda = 4.1$.

We choose in this new experiment $\lambda = 0.6175$.

Figure 4 shows the comparison between the relative errors of function `funm` *of MATLAB and series (2.7) with* $\lambda = 0.6175$ *using infinite norm (4.3).*

Now, the mean processing time for `funm` *was* 0.113997 *seconds and the mean processing time for the Hermite approximation was* 0.027875 *seconds. The first average time corresponds only to the computation of* $\cos(A)$ *using the function* `funm`. *The second value corresponds to the computation of* $\cos(A)$ *and* $\sin(A)$ *using Hermite expansion. Our proposed implementation was* 4.0896 *times faster. In the computation of* $\cos(A)$, *the Hermite method gave a smaller error than* `funm` *in* 74% *of the test cases. In the computation of* $\sin(A)$, *the Hermite method gave a smaller error than* `funm` *in* 74% *of the test cases.*

5 Conclusions.

In this paper a modification of the method proposed in [1] for computing matrix cosine and sine based on Hermite matrix polynomial expansion is presented. Numerical tests and an algorithm are given. The described method allows the simultaneous evaluation of the matrix sine and cosine and it has been compared with the function `funm` of MATLAB. The method depends on the parameter λ, whose impact on the numerical efficiency is currently studied. Furthermore, pending work focuses on the optimal scaling of the matrix and the study of the evaluation [12] of the approximations (2.9) and (2.10). To do parallel implementation of the algorithms presented in this work in a distributed memory platform, using the message passing paradigm, MPI and BLACS for communications, and PBLAS and ScaLAPACK [13] for computations.

References

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

[2] S. Serbin, S. Blalock, An algorithm for computing the matrix cosine, SIAM Journal on Scientific and Statistical Computing 1 (2) (1980) 198–204.

[3] G. I. Hargreaves, N. J. Higham, Efficient algorithms for the matrix cosine and sine, Numerical Algorithms 40 (2005) 383–400.

[4] S. Serbin, Rational approximations of trigonometric matrices with application to second-order systems of differential equations, Applied Mathematics and Computation 5 (1) (1979) 75–92.

[5] N. Higham, M. Smith, Computing the matrix cosine, Numerical Algorithms 34 (1) (2003) 13–26.

[6] N. J. Higham, Functions of Matrices: Theory and Computation, 2008.

[7] G. H. Golub, C. F. V. Loan, Matrix computations, 2nd Edition, The Johns Hopkins University Press, Baltimore, MD, USA, 1989.

[8] L. Jódar, R. Company, Hermite matrix polynomials and second order matrix differential equations, Journal Approximation Theory Application 12 (2) (1996) 20–30.

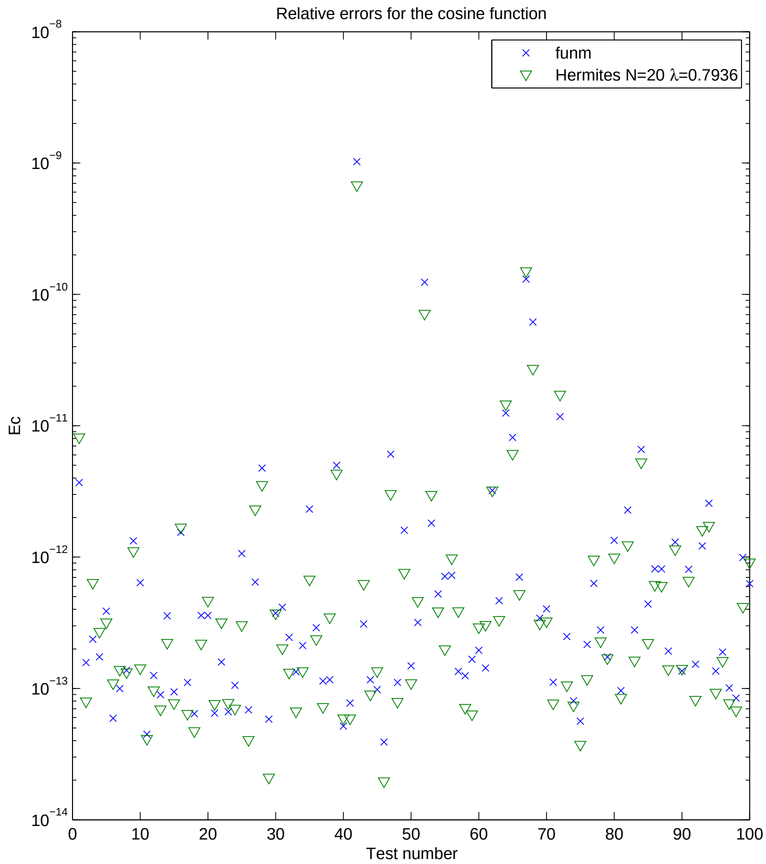
[9] N. N. Lebedev, Special Functions and Their Applications, New York, 1972.

[10] P. I. Davies, N. J. Higham, A schur-parlett algorithm for computing matrix functions, SIAM Journal on Matrix Analysis and Applications 25 (2) (2003) 464–485.

[11] N. Dunford, J. Schwartz, Linear Operators, Part I., New York, 1957.

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

[13] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, ScaLAPACK Users' Guide, SIAM, 1997.



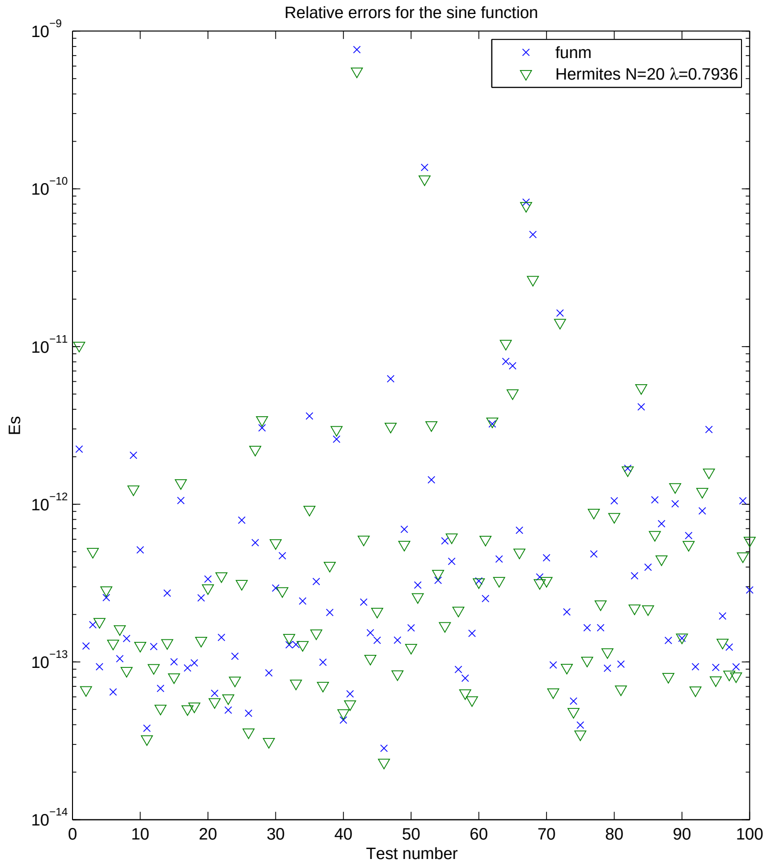
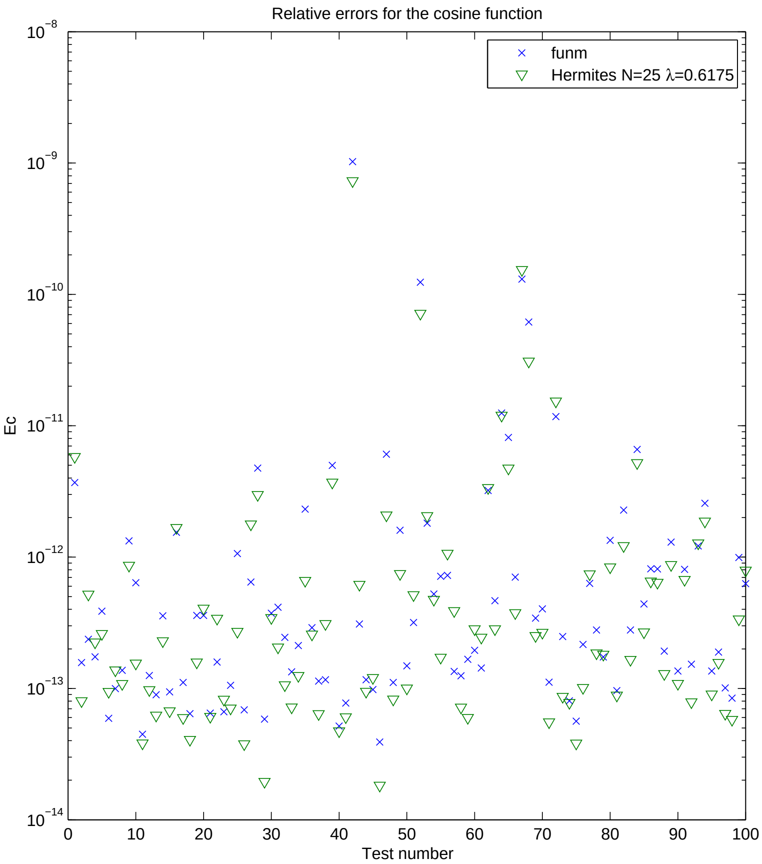


Figure 3: Comparison between the relative errors for cosine and sine computation with $N = 20$ and $\lambda = 0.7936$.



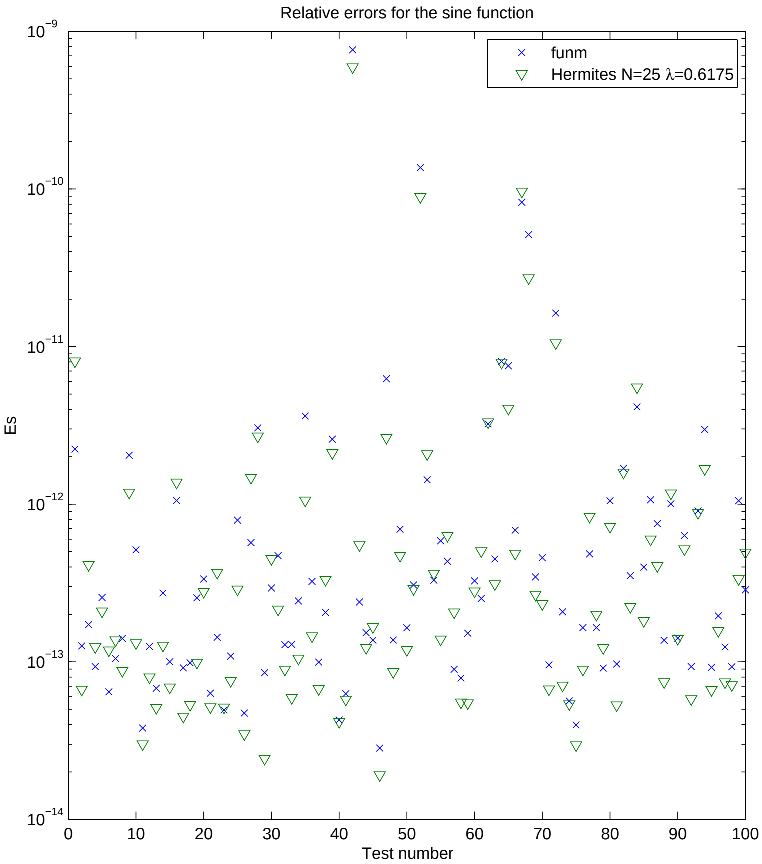


Figure 4: Comparison between the relative errors for cosine and sine computation with $N = 25$ and $\lambda = 0.6175$.