An efficient method to compute the matrix cosine based on Chebyshev polynomials

Emilio Defez*,[1] Javier Ibáñez* and José M. Alonso♮

(*) Instituto de Matemática Multidisciplinar,
(♮) Instituto de Instrumentación para Imagen Molecular,
Universitat Politècnica de València. Camino de Vera s/n, 46022, Valencia, Spain.

1 Introduction and motivation

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. Recently, several state-of-the-art algorithms have been provided for computing these matrix functions [1, 2, 3, 4] in particular for the matrix cosine function. The proposed methods can be classified into two classes: Rational methods (L_∞, Padé approximation, see references [5, 1, 6]) and Polynomial methods (Taylor, Hermite, Bernoulli and Euler series, see references [7, 8, 9, 10, 11]). In general, polynomial approximations are better than rational approximations. As examples of this assertion, we can show the following three significant examples, where different codes were compared with *cosm*, code based on the Padé rational approximation [6]. In [9], authors compared the following functions:

- *cosmtay*: code based on Taylor series [7],
- *cosmtayher*: code based in Hermite series [9],

with *cosm* on different classes of matrices, see Table 1.

	Test 1	Test 2	Test 3
$E(cosmtayher) < E(cosm)$	92%	81%	77.97%
$E(cosmtayher) < E(cosmtay)$	53%	65%	69.49%

Table 1: Relative error comparison [9] between *cosmtayher* (Hermite), `cosmtay`*cosmtay* (Taylor) and *cosm* (Padé) for different sets of matrices.

Moreover, in [10], authors compared the function *cosmber*_1_4, code based on Bernoulli series [10]. with *cosm* on different classes of matrices, see Table 2. Finally, in reference [11], authors compared the following functions:

- *cosm_euler_at*: code based on Euler series, see [11], (that uses even and odd terms),
- *cosm_euler_et*: code based on Euler series, see [11], (that uses only even terms),

[1] edefez@imm.upv.es

$E(cosm) < E(cosmber\_1\_4)$	13.97%
$E(cosm) > E(cosmber\_1\_4)$	86.03%

Table 2: Relative error comparison between *cosm* versus *cosmber_1_4* for different sets of matrices.

with *cosm* on different classes of matrices, see Table 3. As we can appreciate, Tables 1, 2 and 3 show that the polynomial methods (Taylor, Hermite, Bernoulli, Euler) performed better results (in general) than those based on rational methods (Padé).

	Set 1	Set 2	Set 3
$E(cosm) < E(cosm\_euler\_at)$	3.0%	0.0%	20.00%
$E(cosm) > E(cosm\_euler\_at)$	97.0%	100.0%	80.00%
$E(cosm) < E(cosm\_euler\_et)$	6.0%	26.0%	16.36%
$E(cosm) > E(cosm\_euler\_et)$	94.0%	74.0%	83.64%

Table 3: Improvement percentage among the different codes for the 3 sets of matrices.

2 Scalar Chebyshev polynomials. Extension to the matrix case

Chebyshev polynomials $\{T_n(x)\}_{n\geq 0}$ of the first kind [12]. are defined following explicit expressions:

$$\left.\begin{array}{rcl} T_0(x) & = & 1 \\ T_n(x) & = & \displaystyle\sum_{k=0}^{[\frac{n}{2}]} \binom{n}{2k} \left(x^2-1\right)^k x^{n-2k} \ , \ n \geq 1 \\ & = & n \displaystyle\sum_{k=0}^{n} \frac{\left((-2)^k (k+n-1)!\right)}{(2k)!(n-k)!} (1-x)^k \ \ , \ n \geq 1. \end{array}\right\} \tag{1}$$

Chebyshev polynomials also satisfies the following three-term-recurrence formula:

$$\left.\begin{array}{rcl} T_0(x) & = & 1 \\ T_1(x) & = & x \\ T_{n+1}(x) & = & 2xT_n(x) - T_{n-1}(x), n \geq 1 \end{array}\right\} \tag{2}$$

Some properties of this type of polynomials can be found in references [12, 13].

It is well known that Chebyshev polynomials are orthogonal with respect to the weight function $w(x) = (1-x^2)^{-\frac{1}{2}}$ on the interval $[-1, 1]$, see [14] for example. A function $f(x)$ for $|x| \leq 1$ can be developed in series of Chebyshev polynomials of the form

$$f(x) = \sum_{k\geq 0} a_k T_k(x) \ , a_k = \frac{2}{\pi} \int_{-1}^{1} f(x) T_k(x) (1-x^2)^{-\frac{1}{2}} dx, \tag{3}$$

see the classical references [15, 13, 12] for details. For the particular case that $f(x) = e^x$, we have the following expansion [12] for the scalar exponential function:

$$e^x = J_0(i) + 2 \sum_{k\geq 1} i^k J_k(-i) T_k(x) \ , \ |x| < 1, \tag{4}$$

where i is the imaginary unit and $J_k(x)$ is the Bessel function of the first kind of order k.

During the last years, different extensions of Chebyshev polynomials to matrix framework have been given in conection with matrix differential equations, integral transforms, fast methods for resuming matrix polynomials, etc., see [16, 17, 18, 19, 20, 21, 22]. Given a matrix $X \in \mathbb{C}^{r\times r}$, we define the Chebyshev matrix polynomials $\{T_n(X)\}_{n\geq 0}$ by recurrence as follows:

$$\left.\begin{array}{rcl} T_0(X) & = & I_r, \\ T_1(X) & = & X, \\ T_{n+1}(X) & = & 2XT_n(X) - T_{n-1}(X)\ , n \geq 1. \end{array}\right\} \tag{5}$$

We also have the following explicit formulas for the Chebyshev matrix polynomials:

$$\left.\begin{array}{rcl} T_0(X) & = & I_r \\ T_n(X) & = & \displaystyle\sum_{k=0}^{[\frac{n}{2}]} \binom{n}{2k} \left(X^2 - I_r\right)^k X^{n-2k}\ ,\ n \geq 1 \\ & = & n \displaystyle\sum_{k=0}^{n} \frac{\left((-2)^k (k+n-1)!\right)}{(2k)!(n-k)!} \left(I_r - X\right)^k\ ,\ n \geq 1. \end{array}\right\} \tag{6}$$

Note that $T_n(X) = (-1)^n T_n(-X)$. Formula (4) has been generalized for development of the matrix exponential in a Chebyshev series, [23]:

$$e^A = J_0(i)I_r + 2\sum_{k\geq 1} i^k J_k(-i) T_k(A)\ ,\ A \in \mathbb{C}^{r\times r}\ ,\ \|A\| \leq 1. \tag{7}$$

On the other hand, trigonometric matrix functions sine and cosine have been proven to be especially useful for solving systems of second-order linear differential equations, playing a similar role to that of the exponential matrix for first order systems. Due to the relationship $\sin(A) = \cos\left(A + \frac{\pi}{2}I\right)$, where I is the identity matrix, the matrix sine function can be calculated using the same methods as for the matrix cosine one. Usually, research is concentrated on developing efficient state-of-the-art algorithms to compute the matrix cosine function approximately, see [3] and references therein.

Form formula (7), we obtain

$$\cos(A) = J_0(i)I_r + 2\sum_{k\geq 1} (-1)^k J_{2k}(-i) T_k(iA)\ ,\ \|A\| \leq 1. \tag{8}$$

If we use the Bessel functions $I_k(x)$ $((-1)^k J_{2k}(-i) = I_{2k}(1))$, one gets the equivalent expression

$$\cos(A) = I_0(1)I_r + 2\sum_{k\geq 1} I_{2k}(1) T_k(iA)\ ,\ \|A\| \leq 1. \tag{9}$$

An algorithm called *cosm_chebyshev* has been developed to compute the matrix cosine and implemented on MATLAB Version 2023b. We use an absolute backward error analysis with polynomial orders between 16 and 30 for the estimation of the scaling factor s. We have compared our implementations with the codes *cosmtay*, code based on the Taylor polynomial approximation for the matrix cosine [7], and with *cosm*, code based on the Padé rational approximation for the matrix cosine [24].

3 Experimental results

We have considered the following three groups of matrices to perform the experiments.

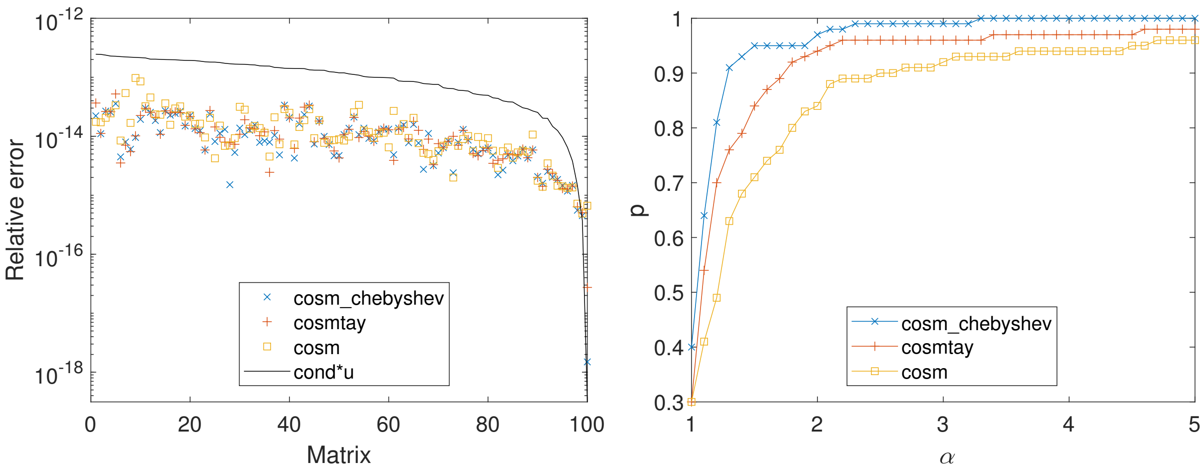
TEST 1. 100 diagonalizable 128×128-sized matrices. They have been obtained as the result of $A = V \cdot D \cdot V^{-1}$, where D is a diagonal matrix (with real and complex eigenvalues) and matrix V is an orthogonal matrix being $V = H/\sqrt{n}$, where H is a Hadamard matrix and n its number of rows or columns. As 2-norm, we have that $0.1 \leq \|A\|_2 \leq 350$.

In Table 4, the percentage of cases in which the relative error commited by *cosm_chebyshev* is better or worse than *cosmtay* and *cosm* is respectively shown.

$E(cosm\_chebyshev) < E(cosmtay)$	55.00%
$E(cosm\_chebyshev) > E(cosmtay)$	45.00%
$E(cosm\_chebyshev) = E(cosmtay)$	0.00%
$E(cosm\_chebyshev) < E(cosm)$	65.00%
$E(cosm\_chebyshev) > E(cosm)$	35.00%
$E(cosm\_chebyshev) = E(cosm)$	0.00%

Table 4: Relative error comparison between *cosm_chebyshev*, *cosmtay* and *cosm*.

The total number of matrix product was 1431 (*cosm_chebyshev*), 1205 (*cosmtay*) and 1409 (*cosm*), while the total time (in seconds) was 1.50 (*cosm_chebyshev*), 1.02 (*cosmtay*) and 2.09 (*cosm*). Figure (1a) depicts the normwise relative error caused by each method in the computation of matrices in our test with respect to the exact solution. Figure (1b) plots the performance profiles. Each point (α, p) on the picture represents the percentage p of matrices, expressed as a percentage of one, for which the error that takes places in calculating their cosine by each method in particular is less than or equal to α times the smallest relative error made by all the codes to be compared.



(a) Normwise relative error. (b) Performance profile.

Figure 1: Experimental results for Group 1.

TEST 2. 100 Non-diagonalizable 128×128-sized matrices. They have been obtained as $A = V \cdot J \cdot V^{-1}$, where J is a Jordan matrix with complex eigenvalues whose modules are less than

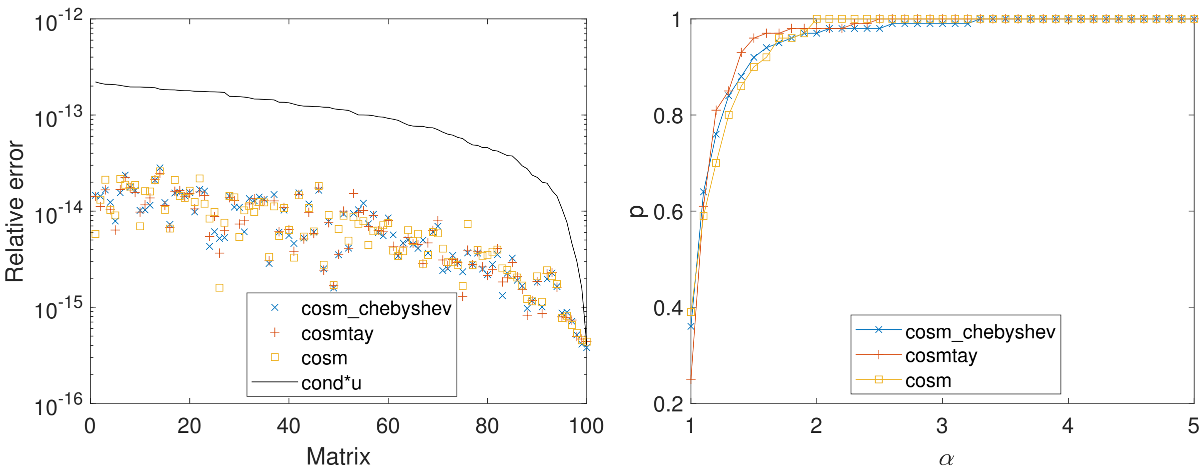
5 and the algebraic multiplicity is randomly generated between 1 and 4. V is an orthogonal random matrix with elements in the interval $[-0.5, 0.5]$. As 2-norm, we have obtained that $3.76056 \leq \|A\|_2 \leq 337.715$.

In Table 5, the percentage of cases in which the relative error commited by *cosm_chebyshev* is better or worse than *cosmtay* and *cosm* is respectively shown.

$E(cosm\_chebyshev) < E(cosmtay)$	48.00%
$E(cosm\_chebyshev) > E(cosmtay)$	52.00%
$E(cosm\_chebyshev) = E(cosmtay)$	0%
$E(cosm\_chebyshev) < E(cosm)$	54.00%
$E(cosm\_chebyshev) > E(cosm)$	46.00%
$E(cosm\_chebyshev) = E(cosm)$	0.00%

Table 5: Relative error comparison between *cosm_chebyshev*, *cosmtay* and *cosm*.

The total number of matrix product was 1427 (*cosm_chebyshev*), 1203 (*cosmtay*) and 1406 (*cosm*), while the total time (in seconds) was 1.60 (*cosm_chebyshev*), 1.09 (*cosmtay*) and 2.19 (*cosm*). Figure (2a) depicts the normwise relative error caused by each method in the computation of matrices in our test with respect to the exact solution. Figure (2b) plots the performance profiles.



(a) Normwise relative error.

(b) Performance profile.

Figure 2: Experimental results for Group 2.

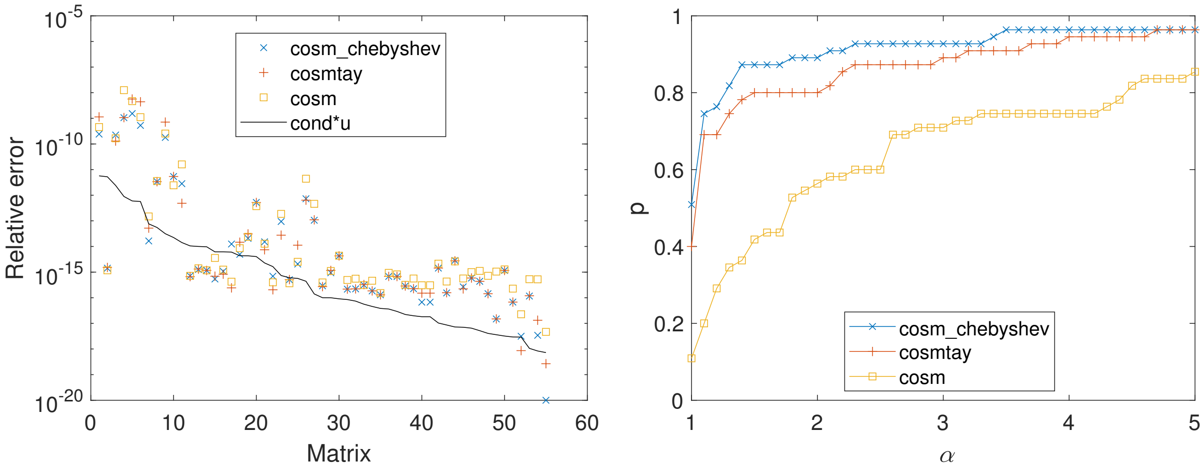
TEST 3. Matrices from the Matrix Computation Toolbox [25] and from Eigtool MATLAB Package [26]. For that matrices, we have obtained $1 \leq \|A\|_2 \leq 398423$.

Table 6 shows the relative error comparison between *cosm_chebyshev*, *cosmtay* and *cosm*.

The total number of matrix product was 536 (*cosm_chebyshev*), 482 (*cosmtay*) and 604 (*cosm*), while the total time (in seconds) was 0.24 (*cosm_chebyshev*), 0.19 (*cosmtay*) and 0.43 (*cosm*). Figure (3a) depicts the normwise relative error caused by each method in the computation of matrices in our test with respect to the exact solution. Figure (3b) plots the performance profiles.

$E(cosm\_chebyshev) < E(cosmtay)$	56.36%
$E(cosm\_chebyshev) > E(cosmtay)$	41.82%
$E(cosm\_chebyshev) = E(cosmtay)$	1.82%
$E(cosm\_chebyshev) < E(cosm)$	81.82%
$E(cosm\_chebyshev) > E(cosm)$	18.18%
$E(cosm\_chebyshev) = E(cosm)$	0.00%

Table 6: Relative error comparison between *cosm_chebyshev*, *cosmtay* and *cosm*.



(a) Normwise relative error.

(b) Performance profile.

Figure 3: Experimental results for Group 3.

4 Conclusions

The implementation based on the Chebyshev series (algorithm *cosm_chebyshev*) is more accurate comparing with the most publicized (based on the Padé rational approximation) algorithm *cosm*. The *cosm_chebyshev* algorithm is slightly superior to *cosmtay* in terms of accuracy. Algorithm *cosm_chebyshev* is quite similar in computational cost to *cosm*, but is more expensive than *cosmtay*.

Acknowledgments

This work has been supported by the Vicerrectorado de Investigación de la Universitat Politècnica de València (PAID-11-22 and PAID-11-23).

References

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

[2] Dehghan M., Hajarian M., Computing matrix functions using mixed interpolation methods. *Mathematical and Computer Modelling*, 52(5-6):826–836, 2010.

[3] Higham, N. J., Functions of Matrices: Theory and Computation. SIAM Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[4] Alonso-Jordá P., Peinado J., Ibáñez J., Sastre J., Defez E., Computing Matrix Trigonometric Functions with GPUs through Matlab, *The Journal of Supercomputing*, 75:1227–1240, 2019.

[5] Tsitouras C., Katsikis V. N., Bounds for variable degree rational L_∞ approximations to the matrix cosine, *Computer Physics Communications*, 185(11): 2834–2840, 2014.

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

[7] Sastre J., Ibáñez J., Alonso-Jordá P., Peinado J., Defez E., Two algorithms for computing the matrix cosine function, *Applied Mathematics and Computation*, 312 (1): 66–77, 2017.

[8] Sastre J., Ibáñez J., Alonso-Jordá P. , Peinado J., Defez E., Fast Taylor polynomial evaluation for the computation of the matrix cosine, *Journal of Computational and Applied Mathematics*, 354:641–650, 2019.

[9] Defez E., Ibáñez J., Peinado J., Sastre J., Alonso-Jordá P., An efficient and accurate algorithm for computing the matrix cosine based on new Hermite approximations, *Journal of Computational and Applied Mathematics*, 348(1):1–13, 2019.

[10] Defez E., Ibáñez J., Alonso J.M., Alonso-Jordá P., On Bernoulli series approximation for the matrix cosine, *Mathematical Methods in the Applied Sciences*, 45(6): 3239–3253, 2022.

[11] Alonso J.M., Defez E., Ibáñez J., Sastre J., On the use of Euler polynomials to approximate the matrix cosine. Modelling for engneering and human behavior 2023, 427–433, 2023.

[12] Mason, J. C. and Handscomb, D. C., Chebyshev Polynomials. Chapman and Hall/CRC, 2002.

[13] Rivlin, T. J., Chebyshev Polynomials. From Approximation Theory to Algebra and Number Theory. Second Edition. John Wiley & Sons, IN, 1990.

[14] Chihara T. S., An introduction to orthogonal polynomials. Courier Corporation, 2011.

[15] Rivlin T. J., Chebyshev polynomials, Courier Dover Publications, 1974.

[16] Defez E., Jódar L., Chebyshev matrix polynomials and second order matrix differential equations. *Utilitas Mathematica*, 61, 2002.

[17] Liang W., Baer R., Saravanan C., Shao Y., Bell A.T., Head-Gordon M., Fast methods for resuming matrix polynomials and Chebyshev matrix polynomials. *Journal of Computational Physics*, 194(2):575–587, 2004.

[18] Altin A., Cekim B., Generating matrix functions for Chebyshev matrix polynomials of the second kind. *Hacettepe Journal of Mathematics and Statistics*, 41(1): 25–32, 2012.

[19] Metwally M.S., Mohamed M.T., Shehata A., On Chebyshev matrix polynomials, matrix differential equations and their properties. *Afrika Matematika*, 26: 1037–1047, 2015.

[20] Metwally M.S., Mohamed M.T., Shehata A., Associated matrix polynomials with the second kind Chebyshev matrix polynomials. *Applications and Applied Mathematics: An International Journal*, 14(1): 24, 2019.

[21] Kargin L., Veli K., Chebyshev-type matrix polynomials and integral transforms. *Hacettepe Journal of Mathematics and Statistics*, 44(2): 341–350, 2015.

[22] Liang W., Saravanan C., Shao Y., Baer R., Bell A.T., Head-Gordon M., Improved fermi operator expansion methods for fast electronic structure calculations. *The Journal of chemical physics*, 119(8): 4117–4125, 2003.

[23] Defez E., Ibáñez J., Alonso J.M., Peinado J., An efficient method to compute the matrix exponential based on Chebyshev polynomials. Modelling for engineering and human behavior 2023, 434–441, 2023.

[24] Al-Mohy A. H., Higham N. J., A new scaling and squaring algorithm for the matrix exponential, *SIAM Journal on Matrix Analisys*, 31(3): 970–989, 2009.

[25] Higham N. J., The matrix computation toolbox. 2002.

[26] Wright T.G., Eigtool, version 2.1. 2009.