A NEW MATRIX SERIES EXPANSIONS FOR THE MATRIX COSINE APPROXIMATION

A NEW MATRIX SERIES EXPANSIONS FOR THE MATRIX COSINE APPROXIMATION

E. Defez[⋆,♮], J.J. Ibáñez[♭,♮], J. M. Alonso[♭], J. Peinado[♢,♮], P. Alonso[†]
⋆ Instituto de Matemática Multidisciplinar
♭ Instituto de Instrumentación para Imagen Molecular
♢ Departamento de Sistemas Informáticos y Computación
† Grupo Interdisciplinar de Computación y Comunicaciones
♮ Group of High Performance Scientific Computing **(HiPerSC)**
Universitat Politècnica de València, Spain
edefez@imm.upv.es, jjibanez@dsic.upv.es, jmalonso@dsic.upv.es,
jpeinado@dsic.upv.es, palonso@dsic.upv.es

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 [1-4] in particular for the matrix cosine function.

References

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

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

[3] N. J. Higham. Functions of matrices: theory and computation (Vol. 104). Siam. 2008.

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

The proposed methods for calculating the matrix cosine can be classified into two classes: Rational approximation methods and Polynomial methods.

References of Rational approximation methods (L_∞, Padé approximation)

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

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

[7] A. H. Al-Mohy, N. J. Higham, 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.

References of Polynomial methods (Taylor and Hermite series)

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

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

[10] E. Defez, J. Ibáñez, J. Peinado, J. Sastre, P. Alonso-Jordá. 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.

The proposed methods for calculating the matrix cosine can be classified into two classes: Rational approximation methods and Polynomial methods.

References of Rational approximation methods (L_∞, Padé approximation)

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

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

[7] A. H. Al-Mohy, N. J. Higham, 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.

References of Polynomial methods (Taylor and Hermite series)

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

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

[10] E. Defez, J. Ibáñez, J. Peinado, J. Sastre, P. Alonso-Jordá. 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.

In general, Polynomial approximations are better than rational approximations

In [10] authors compared the following functions:

- *cosm*. Code based on the Padé rational approximation for the matrix cosine [7].
- *cosmtay*. Code based on Taylor series to compute the matrix cosine [8].
- *cosmtayher*. Code based in Hermite matrix polynomials series [10].

on different classes of matrices, showing that the polynomial methods (Taylor and Hermite) performed better results (in general) than those based on rational methods (Padé).

Table: Relative error comparison [10] between `cosmtayher` (Hermite), `cosmtay` (Taylor) and `cosm` (Padé).

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

Bernoulli polynomials

The Bernoulli polynomials $B_n(x)$ are defined in [11] as the coefficients of the generating function

$$g(x,t) = \frac{te^{tx}}{e^t - 1} = \sum_{n\geq 0} \frac{B_n(x)}{n!} t^n \ , \ |t| < 2\pi. \tag{1}$$

Bernoulli polynomials $B_n(x)$ has the explicit expression

$$B_n(x) = \sum_{k=0}^{n} \binom{n}{k} B_k x^{n-k}. \tag{2}$$

where the Bernoulli numbers are defined by $B_n = B_n(0)$, where

$$B_0 = 1, B_k = -\sum_{i=0}^{k-1} \frac{B_i}{k+1-i}, k \geq 1. \tag{3}$$

References

[11] F. W. Olver, D. W. Lozier, R. F. Boisvert, C. W. Clark.NIST handbook of mathematical functions hardback and CD-ROM, Cambridge University Press, 2010.

Bernoulli matrix polynomials and exponential matrix approximation

For a matrix $A \in \mathbb{C}^{r\times r}$ we define the nth Bernoulli matrix polynomial by the expression

$$B_n(A) = \sum_{k=0}^{n} \binom{n}{k} B_k A^{n-k}. \tag{4}$$

We have the series expansion

$$e^{At} = \left(\frac{e^t - 1}{t}\right) \sum_{n\geq 0} \frac{B_n(A) t^n}{n!} \ , \ |t| < 2\pi. \tag{5}$$

To obtain approximations of the matrix exponential, we let's take s the scaling of the matrix A and take the degree of the approximation N:

$$e^{A2^{-s}} \approx (e-1) \sum_{n=0}^{N} \frac{B_n\left(A2^{-s}\right)}{n!}. \tag{6}$$

Bernoulli matrix polynomials series for the matrix cosine

For a matrix $A \in \mathbb{C}^{r \times r}$, using expression (6) we obtain

$$\boxed{\cos(A) = (\cos(1) - 1)\sum_{n\geq 0} \frac{(-1)^n B_{2n+1}(A)}{(2n+1)!} + \sin(1)\sum_{n\geq 0} \frac{(-1)^n B_{2n}(A)}{(2n)!}} \tag{7}$$

Note that unlike the Taylor and Hermite polynomials that are even or odd, depending on the parity of the polynomial degree n, the Bernoulli polynomials do not verify this property, so in the development of $\cos(A)$ all Bernoulli polynomials are needed (and not just the even-numbered).

We can also obtain, for $C \in \mathbb{C}^{r \times r}$, the expression:

$$\boxed{\cos(C) = \sin(1)\sum_{n\geq 0} \frac{(-1)^n 2^{2n} B_{2n}\left(\frac{1}{2}(C+I)\right)}{(2n)!}} \tag{8}$$

Bernoulli matrix polynomials series for the matrix cosine

For a matrix $A \in \mathbb{C}^{r\times r}$, using expression (6) we obtain

$$\boxed{\cos(A) = (\cos(1) - 1)\sum_{n\geq 0}\frac{(-1)^n B_{2n+1}(A)}{(2n+1)!} + \sin(1)\sum_{n\geq 0}\frac{(-1)^n B_{2n}(A)}{(2n)!}} \quad (7)$$

Note that unlike the Taylor and Hermite polynomials that are even or odd, depending on the parity of the polynomial degree n, the Bernoulli polynomials do not verify this property, so in the development of $\cos(A)$ all Bernoulli polynomials are needed (and not just the even-numbered).

We can also obtain, for $C \in \mathbb{C}^{r\times r}$, the expression:

$$\boxed{\cos(C) = \sin(1)\sum_{n\geq 0}\frac{(-1)^n 2^{2n} B_{2n}\left(\frac{1}{2}(C+I)\right)}{(2n)!}} \quad (8)$$

The proposed Algorithm *cosmber*

From (7) one gets the approximation

$$\boxed{\cos(A) \approx (\cos(1)-1)\sum_{n=0}^{m}\frac{(-1)^n B_{2n+1}(A)}{(2n+1)!}+\sin(1)\sum_{n=0}^{m}\frac{(-1)^n B_{2n}(A)}{(2n)!}} \tag{9}$$

From (8) one gets the approximation

$$\boxed{\cos(C) \approx \sin(1)\sum_{n=0}^{m}\frac{(-1)^n 2^{2n} B_{2n}\left(\frac{1}{2}(C+I)\right)}{(2n)!}} \tag{10}$$

The MATLAB's implementation *cosmber* uses the new calculation code of m and s with standard estimation and calculation of the matrix powers. The maximum value of m to be used is 36.

Numerical Experiments

Algorithms that have been implemented

- *cosmber* New code based on the new developments of Bernoulli matrix polynomials (formulae (9) and (10)).
- *cosmtay* Code based on the Taylor series for the cosine [12]. It will provide a maximum value of $m = 16$ considering only the even terms, which would be equivalent to $m = 32$ using the even and odd terms.
- *cosmtayher* Code based on the Hermite series for the cosine [13]. It will provide a maximum value of $m = 16$ considering only the even terms, which would be equivalent to $m = 32$ using the even and odd terms.
- *cosm* Code based on the Padé rational approximation for the cosine [14]. It will provide a maximum value of $m = 16$ considering only the even terms, which would be equivalent to $m = 32$ using the even and odd terms.

Implemented on MATLAB R2018b

References

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

[13] E. Defez, J. Ibáñez, J. Peinado, J. Sastre, P. Alonso-Jordá. 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.

[14] A. H. Al-Mohy, N. J. Higham, 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

Numerical test

259 matrices: 100 diagonalizable, 100 non-diagonalizable, 42 from toolbox [15] and 17 from Eigtool [16]. Size 128×128.

References

[15] N.J. Higham. The Test Matrix Toolbox for MATLAB, Numerical Analysis Report No. 237, The University of Manchester, England, 1993.

[16] T.G. Wright. Eigtool, Version 2.1, 16, March 2009. Available online at: http://www.comlab.ox.ac.uk/pseudospectra/eigtool/

The rows of each table show the percentage of cases in which the relative errors of `cosmber` (Bernoulli) are lower, greater or equal than the relative errors of `cosmtay` (Taylor), `cosmtayher` (Hermite) and `cosm` (Padé).

Table: Using approximations (9) and (10)

$E(cosmber) < E(cosmtay)$	58.30%
$E(cosmber) > E(cosmtay)$	41.70%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	52.90%
$E(cosmber) > E(cosmtayher)$	47.10%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	77.99%
$E(cosmber) > E(cosm)$	22.01%
$E(cosmber) = E(cosm)$	0%

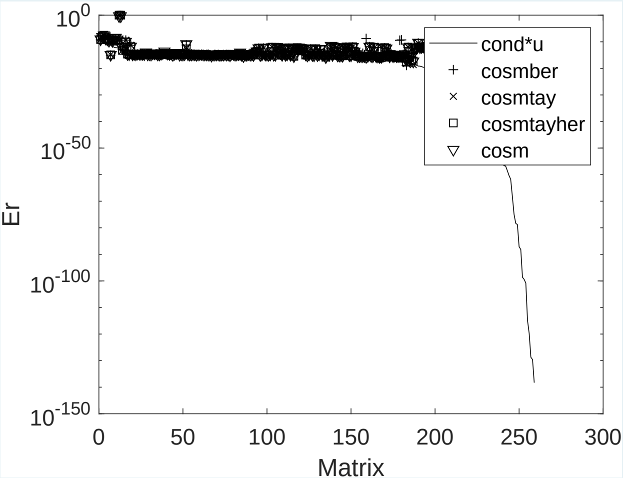
,

$E(cosmber) < E(cosmtay)$	66.02%
$E(cosmber) > E(cosmtay)$	33.98%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	60.23%
$E(cosmber) > E(cosmtayher)$	39.77%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	76.06%
$E(cosmber) > E(cosm)$	23.94%
$E(cosmber) = E(cosm)$	0%

Total number of matrix products: *cosmber*: 3202 *cosmtay*: 2391 *cosmtayher*: 1782, *cosm*: 3016

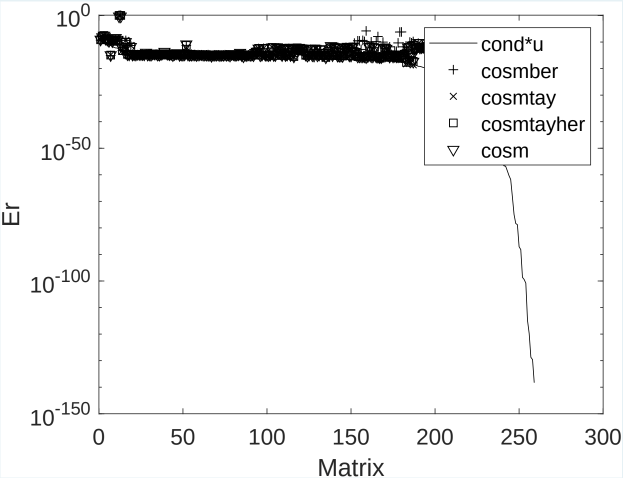
Numerical Test

Normwise relative errors using (9)



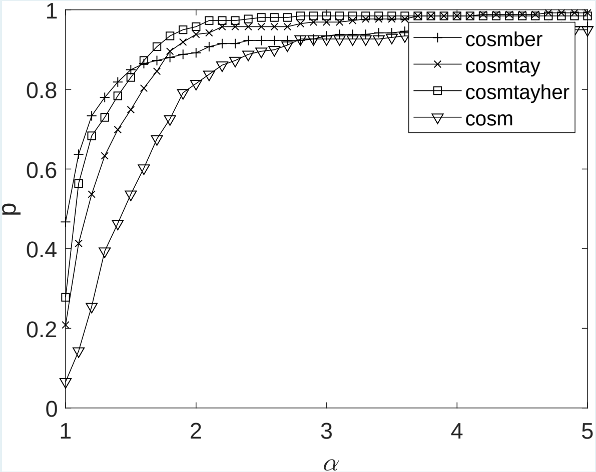
Numerical Test

Normwise relative errors using (10)



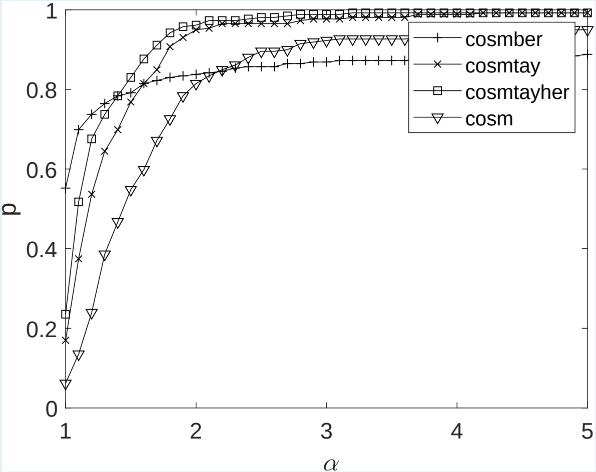
Numerical Test

nprofile relative errors using (9)



Numerical Test

nprofile relative errors using (10)



Conclusions

- In general, the new Bernoulli series implementation based on approximation (10) has been more accurate than that one based on (9), according to the comparison with the approximations based on Taylor (`cosmtay`), Hermite (`cosmtayher`) and Padé (`cosm`). Moreover, the usage of Bernoulli series has obtained relative errors lower than those of the other evaluated methods, and it should be considered from now on when computing the matrix cosine function.

A NEW MATRIX SERIES EXPANSIONS FOR THE MATRIX COSINE APPROXIMATION

E. Defez[⋆,♮], J.J. Ibáñez[♭,♮], J. M. Alonso[♭], J. Peinado[♢,♮], P. Alonso[†]
⋆ Instituto de Matemática Multidisciplinar
♭ Instituto de Instrumentación para Imagen Molecular
♢ Departamento de Sistemas Informáticos y Computación
† Grupo Interdisciplinar de Computación y Comunicaciones
♮ Group of High Performance Scientific Computing **(HiPerSC)**
Universitat Politècnica de València, Spain
edefez@imm.upv.es, jjibanez@dsic.upv.es, jmalonso@dsic.upv.es, jpeinado@dsic.upv.es, palonso@dsic.upv.es

Test 1

100 test diagonalizable matrices 128×128. The matrices are obtained as $A = V \cdot D \cdot V^T$ where D is a diagonal matrix (with complex or real values) and matrix V is an orthogonal matrix $V = H/16$, where H is a Hadamard matrix. We have $2.18 \leq \|A\|_1 \leq 207.52$. The matrix cosine is exactly calculated as

$$\cos(A) = V \cdot \cos(D) \cdot V^T$$

The rows of each table show the percentage of cases in which the relative errors of `cosmber` (Bernoulli) are lower, greater or equal than the relative errors of `cosmtay` (Taylor), `cosmtayher` (Hermite) and `cosm` (Padé).

Table: Using approximations (9) and (10)

$E(cosmber) < E(cosmtay)$	43.00%
$E(cosmber) > E(cosmtay)$	57.00%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	37.00%
$E(cosmber) > E(cosmtayher)$	63.00%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	84.00%
$E(cosmber) > E(cosm)$	16.00%
$E(cosmber) = E(cosm)$	0%

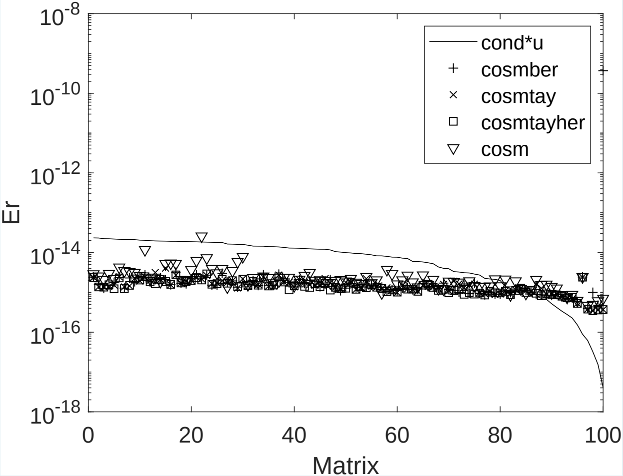
,

$E(cosmber) < E(cosmtay)$	67.00%
$E(cosmber) > E(cosmtay)$	33.00%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	60.00%
$E(cosmber) > E(cosmtayher)$	40.00%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	85.00%
$E(cosmber) > E(cosm)$	15.00%
$E(cosmber) = E(cosm)$	0%

Total number of matrix products: *cosmber*: 1279 *cosmtay*: 933 *cosmtayher*: 667, *cosm*: 1129

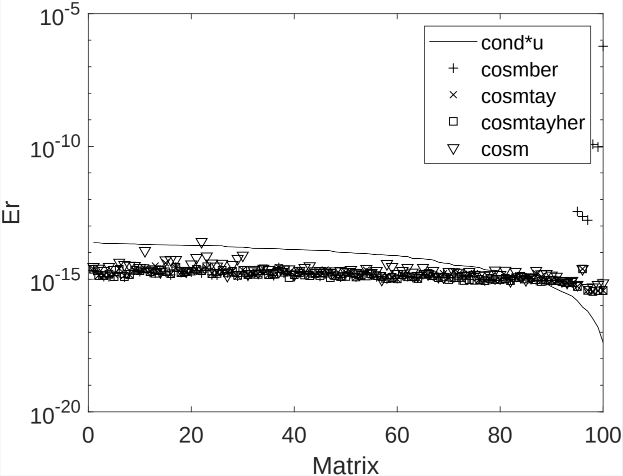
Test 1

Normwise relative errors using (9)



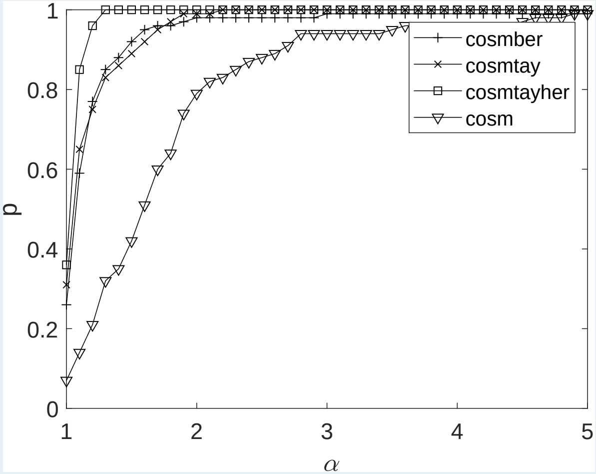
Test 1

Normwise relative errors using (10)



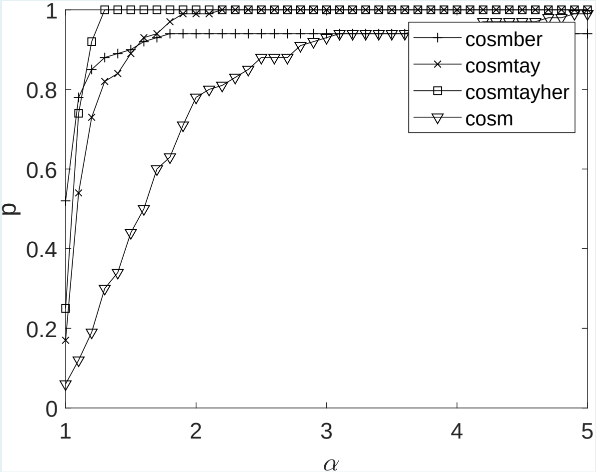
Test 1

nprofile using (9)



Test 1

nprofile using (10)



Test 2

100 test non-diagonalizable matrices 128×128. The matrices are obtained as $A = V \cdot J \cdot V^{-1}$ where J is a Jordan matrix with complex eigenvalues with module less than 10 and random algebraic multiplicity between 1 and 5, and matrix V is a random matrix with elements in the interval $[-0.5, 0.5]$. We have $1279.16 \leq \|A\|_1 \leq 87886.4$. The matrix cosine is exactly calculated as

$$\cos(A) = V \cdot \cos(J) \cdot V^{-1}$$

The rows of each table show the percentage of cases in which the relative errors of `cosmber` (Bernoulli) are lower, greater or equal than the relative errors of `cosmtay` (Taylor), `cosmtayher` (Hermite) and `cosm` (Padé).

Table: Using approximations (9) and (10)

$E(cosmber) < E(cosmtay)$	84.00%
$E(cosmber) > E(cosmtay)$	16.00%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	85.00%
$E(cosmber) > E(cosmtayher)$	15.00%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	86.00%
$E(cosmber) > E(cosm)$	14.00%
$E(cosmber) = E(cosm)$	0%

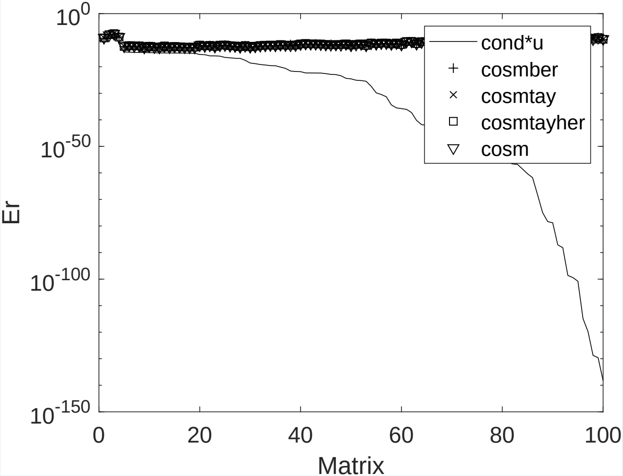
,

$E(cosmber) < E(cosmtay)$	84.00%
$E(cosmber) > E(cosmtay)$	16.00%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	85.00%
$E(cosmber) > E(cosmtayher)$	15.00%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	86.00%
$E(cosmber) > E(cosm)$	14.00%
$E(cosmber) = E(cosm)$	0%

Total number of matrix products: *cosmber*: 1200 *cosmtay*: 900 *cosmtayher*: 702, *cosm*: 1197

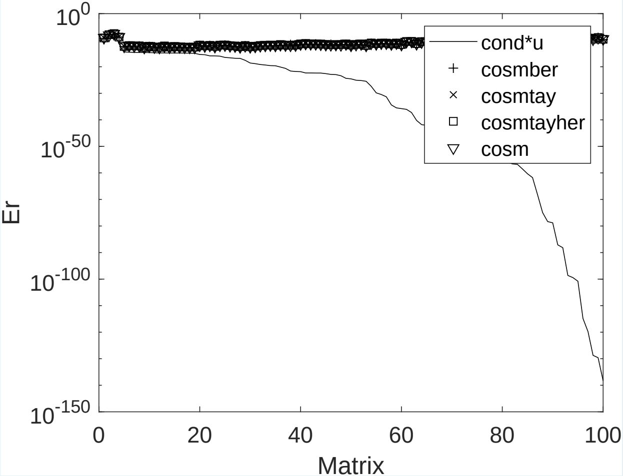
Test 2

Normwise relative errors using (9)



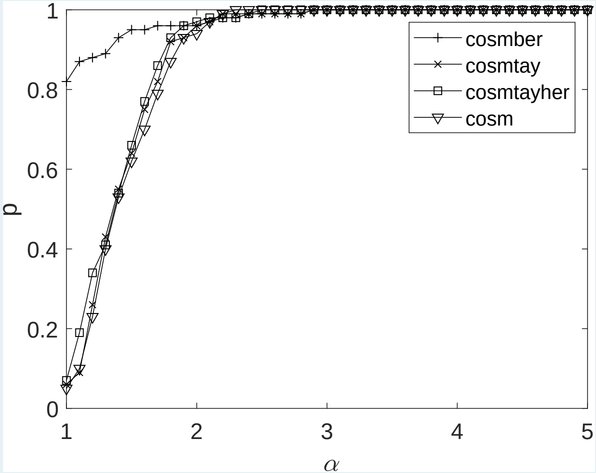
Test 2

Normwise relative errors using (10)



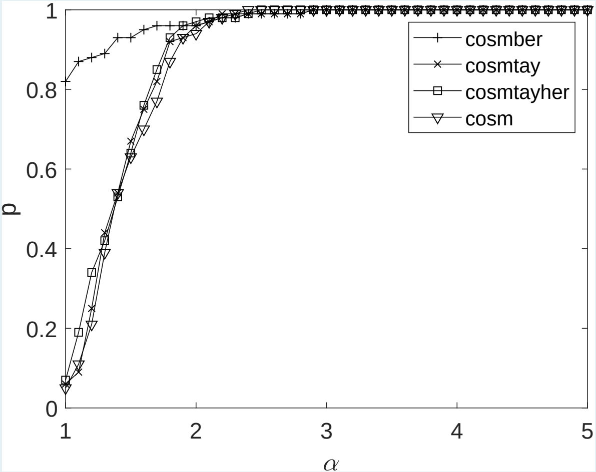
Test 2

nprofile using (9)



Test 2

nprofile using (10)



Test 3

42 matrices from the Matrix Computation Toolbox [15] with size 128×128.

References

[15] N.J. Higham. The Test Matrix Toolbox for MATLAB, Numerical Analysis Report No. 237, The University of Manchester, England, 1993.

The rows of each table show the percentage of cases in which the relative errors of `cosmber` (Bernoulli) are lower, greater or equal than the relative errors of `cosmtay` (Taylor), `cosmtayher` (Hermite) and `cosm` (Padé).

Table: Using approximations (9) and (10)

$E(cosmber) < E(cosmtay)$	42.86%
$E(cosmber) > E(cosmtay)$	57.14%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	28.57%
$E(cosmber) > E(cosmtayher)$	71.43%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	61.90%
$E(cosmber) > E(cosm)$	38.10%
$E(cosmber) = E(cosm)$	0%

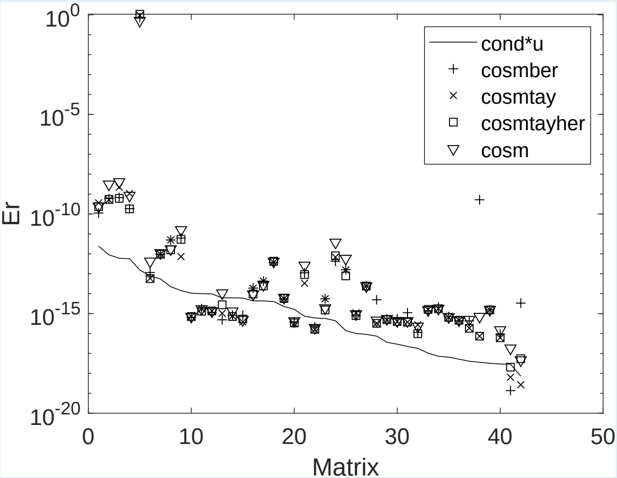
,

$E(cosmber) < E(cosmtay)$	35.71%
$E(cosmber) > E(cosmtay)$	64.29%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	19.05%
$E(cosmber) > E(cosmtayher)$	80.95%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	50.00%
$E(cosmber) > E(cosm)$	50.00%
$E(cosmber) = E(cosm)$	0%

Total number of matrix products: *cosmber*: 512 *cosmtay*: 389 *cosmtayher*: 282, *cosm*: 480

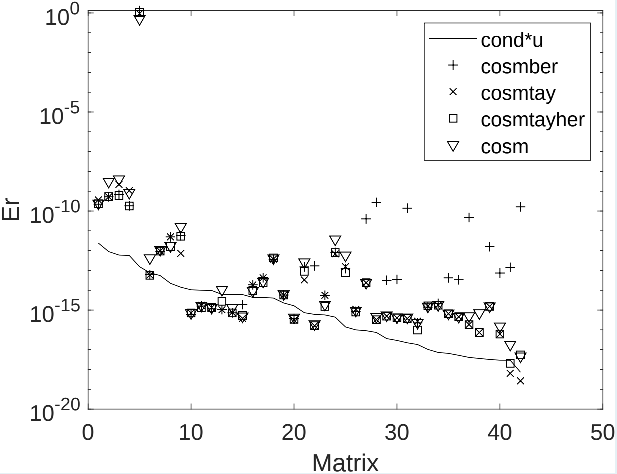
Test 3

Normwise relative errors using (9)



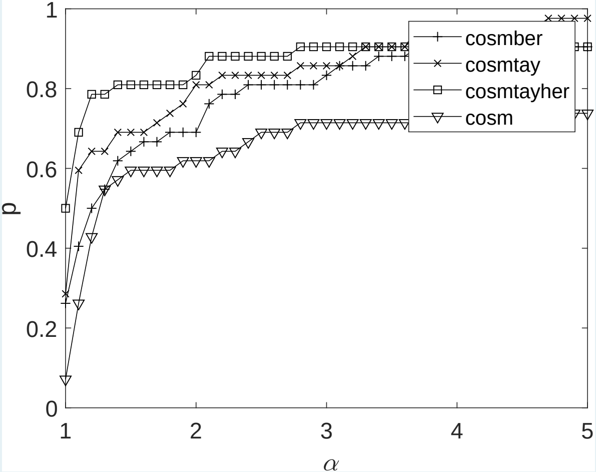
Test 3

Normwise relative errors using (10)



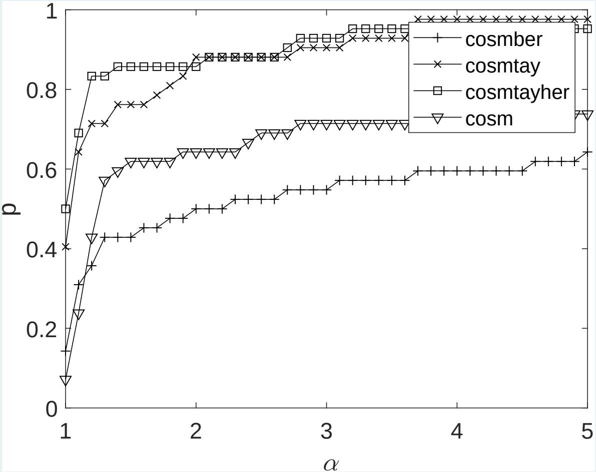
Test 3

nprofile using (9)



Test 3

nprofile using (10)



Test 4

17 matrices from the Eigtool Matlab package [16] with size 128 $\times$ 128.

References

[16] T.G. Wright. Eigtool, Version 2.1, 16, March 2009. Available online at:
http://www.comlab.ox.ac.uk/pseudospectra/eigtool/

The rows of each table show the percentage of cases in which the relative errors of `cosmber` (Bernoulli) are lower, greater or equal than the relative errors of `cosmtay` (Taylor), `cosmtayher` (Hermite) and `cosm` (Padé).

Table: Using approximations (9) and (10)

$E(cosmber) < E(cosmtay)$	35.29%
$E(cosmber) > E(cosmtay)$	64.71%
$E(cosmber) = E(cosmtay)$	0%
$E(cosmber) < E(cosmtayher)$	17.65%
$E(cosmber) > E(cosmtayher)$	82.35%
$E(cosmber) = E(cosmtayher)$	0%
$E(cosmber) < E(cosm)$	35.29%
$E(cosmber) > E(cosm)$	64.71%
$E(cosmber) = E(cosm)$	0%

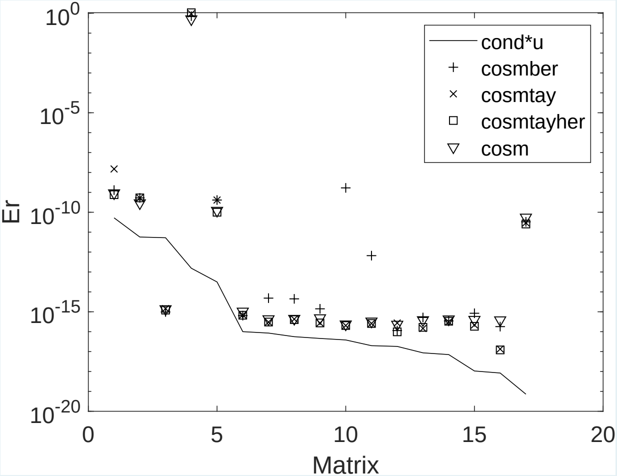
,

$E(cosmber) < E(cosmtay)$	16.67%
$E(cosmber) > E(cosmtay)$	77.78%
$E(cosmber) = E(cosmtay)$	5.56%
$E(cosmber) < E(cosmtayher)$	5.56%
$E(cosmber) > E(cosmtayher)$	88.89%
$E(cosmber) = E(cosmtayher)$	5.56%
$E(cosmber) < E(cosm)$	27.78%
$E(cosmber) > E(cosm)$	66.67%
$E(cosmber) = E(cosm)$	5.56%

Total number of matrix products: *cosmber*: 211 *cosmtay*: 169 *cosmtayher*: 131, *cosm*: 210

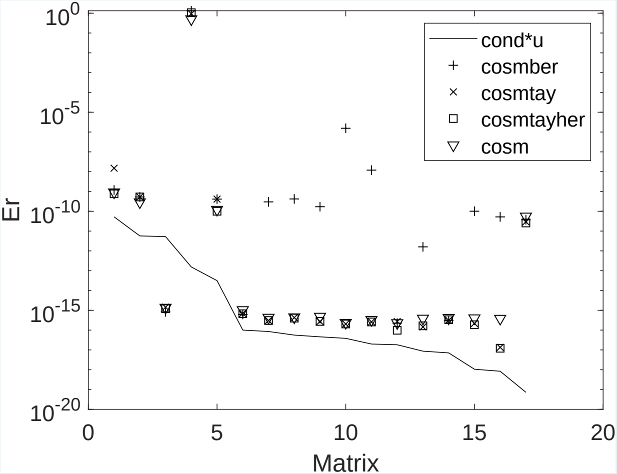
Test 4

Normwise relative errors using (9)



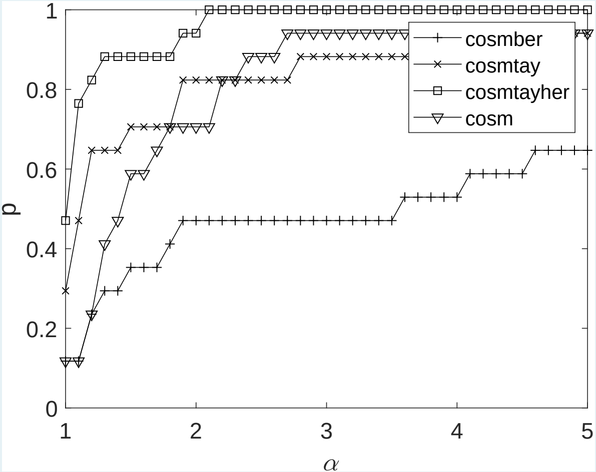
Test 4

Normwise relative errors using (10)



Test 4

nprofile using (9)



Test 4

nprofile using (10)