Linear Algebra and its Applications 539 (2018) 229–250

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Efficient evaluation of matrix polynomials

J. Sastre

Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

ARTICLE INFO

Article history:
Received 31 October 2016
Accepted 9 November 2017
Available online 14 November 2017
Submitted by V. Mehrmann

MSC:
11C08
26C15
41-04
41A10

Keywords:
Matrix
Polynomial
Rational
Mixed rational and polynomial
Approximation
Computation
Matrix function

ABSTRACT

This paper presents a new family of methods for evaluating matrix polynomials more efficiently than the state-of-the-art Paterson–Stockmeyer method. Examples of the application of the methods to the Taylor polynomial approximation of matrix functions like the matrix exponential and matrix cosine are given. Their efficiency is compared with that of the best existing evaluation schemes for general polynomial and rational approximations, and also with a recent method based on mixed rational and polynomial approximants. For many years, the Paterson–Stockmeyer method has been considered the most efficient general method for the evaluation of matrix polynomials. In this paper we show that this statement is no longer true. Moreover, for many years rational approximations have been considered more efficient than polynomial approximations, although recently it has been shown that often this is not the case in the computation of the matrix exponential and matrix cosine. In this paper we show that in fact polynomial approximations provide a higher order of approximation than the state-of-the-art computational methods for rational approximations for the same cost in terms of matrix products.

E-mail address: jsastrem@upv.es.

https://doi.org/10.1016/j.laa.2017.11.010

1. Introduction

In this paper we propose a new family of methods for evaluating matrix polynomials more efficiently than the state-of-the-art Paterson–Stockmeyer method combined with Horner's method [1], [2, Sec. 4.2]. The proposed methods are applied to compute efficiently Taylor polynomial approximations of matrix functions. The computation of matrix functions is a research field with applications in many areas of science and many algorithms for their computation have been proposed [2,3]. Among all matrix functions, the matrix exponential has attracted special attention, see [4–6] and the references therein, and lately the matrix cosine, see [7,8] and the references therein. The main methods for computing matrix functions are those based on rational approximations, like Padé or Chebyshev approximations, polynomial approximations, like Taylor approximation, similarity transformations and matrix iterations [2]. Moreover, a new kind of approximations based on mixed rational and polynomial approximants has been proposed in [9].

Recently, it has been shown that using the combination of Horner and Paterson–Stockmeyer methods [1], [2, Sec. 4.2], polynomial approximations may be more efficient than rational Padé approximations for both the matrix exponential and cosine [6,8]. In this paper we show that using the proposed matrix polynomial evaluation methods, polynomial approximations are more accurate than existing state-of-the-art methods for evaluating both polynomial and rational approximants for the same computing cost. Moreover, we show that the new methods are more efficient than the recent mixed rational and polynomial approximation [9] in some cases, and examples for the computation of the matrix exponential and the matrix cosine are given.

Throughout this paper $\lceil x \rceil$ denotes the lowest integer not less than x, $\lfloor x \rfloor$ denotes the highest integer not exceeding x, $\mathbb{N}$ denotes the set of positive integer numbers, $\mathbb{C}^{n\times n}$ and $\mathbb{R}^{n\times n}$ denote the sets of complex and real matrices of size $n \times n$, respectively, I denotes the identity matrix for both sets, and $\mathcal{R}_{k,m}$ denotes the space of rational functions with numerator and denominator of degrees at most k and m, respectively.

Note that the multiplication by the matrix inverse in matrix rational approximations is calculated as the solution of a multiple right-hand side linear system. Therefore, the cost of evaluating polynomial and rational approximations will be given in terms of the number of matrix products, denoted by M, and the cost of the solution of multiple right-hand side linear systems $AX = B$, where matrices A and B are $n \times n$, denoted by D. From [10, App. C] it follows that, see [9, p. 11940]:

$$D \approx 4/3M. \tag{1}$$

This paper is organized as follows. Section 2 recalls some results for efficient Taylor, Padé, and mixed rational and polynomial approximation of general matrix functions. Section 3 deals with the new matrix polynomial evaluation methods giving examples for the computation of the matrix exponential and the matrix cosine. Section 4 compares the new techniques with efficient state-of-the-art evaluation schemes for polynomial, rational and mixed rational and polynomial approximants. Section 5 gives examples for the matrix exponential computation even more efficient than the ones given in Section 3, suggesting more general formulas for evaluating matrix polynomials. Finally, conclusions are given in Section 6.

2. Polynomial, rational, and mixed rational and polynomial approximants

This section summarizes some results of the computational costs of Taylor, Padé, and the mixed rational and polynomial approximants given in [9].

2.1. Taylor approximation of matrix functions

If $f(A)$ is a matrix function defined by a Taylor series according to Theorem 4.7 of [2, p. 76] where A is a complex square matrix, then we will denote by $T_m(A)$ the matrix polynomial defined by the truncated Taylor series of degree m of $f(A)$. For scalar $x \in \mathbb{C}$ it follows that

$$f(x) - T_m(x) = O(x^{m+1}), \tag{2}$$

about the origin, and, from now on, we will refer to m as the order of the Taylor approximation. The most efficient method in the literature to evaluate a matrix polynomial

$$P_m(A) = \sum_{i=0}^{m} b_i A^i \,, \tag{3}$$

is the combination of Horner and Paterson–Stockmeyer methods [1] given by

$$\begin{aligned} PS_m(A) = \Big(\Big(\cdots \big(&b_m A^s + b_{m-1}A^{s-1} + \ldots + b_{m-s+1}A + b_{m-s}I\big) \\ &\times\ A^s + b_{m-s-1}A^{s-1} + b_{m-s-2}A^{s-2} + \ldots + b_{m-2s+1}A + b_{m-2s}I\big) \\ &\times\ A^s + b_{m-2s-1}A^{s-1} + b_{m-2s-2}A^{s-2} + \ldots + b_{m-3s+1}A + b_{m-3s}I\big) \\ &\quad\vdots \\ &\times\ A^s + b_{s-1}A^{s-1} + b_{s-2}A^{s-2} + \cdots + b_1 A + b_0 I, \end{aligned} \tag{4}$$

where the integer $s > 0$ divides m and the matrix powers $A^2, A^3, \ldots, A^s$, are computed and stored previously.

Table 1 shows the maximum values of m that can be obtained for a given number of matrix products in $T_m(A)$ using Paterson–Stockmeyer method, corresponding to $m = s^2$ and $m = s(s+1)$, for $s \in \mathbb{N}$. The cost of evaluating (4), denoted by C_{PS}, for the values

Table 1
Cost C_{PS} in terms of matrix products for the evaluation of polynomial $P_m(A)$ with Horner and Paterson–Stockmeyer methods for the first eleven values m^* that maximize the polynomial degree obtained for a given cost.

m^*	1	2	4	6	9	12	16	20	25	30	36
C_{PS}	0	1	2	3	4	5	6	7	8	9	10

in m^* is given by [9, Eq. (6)]

$$C_{PS} = (r + s - 2)M, \text{ with } r = m/s, \; m \in m^*. \tag{5}$$

Table 1 presents the cost C_{PS} of evaluating (4) in terms of matrix products for the first eleven values of m^*. For orders $m \notin m^*$ we evaluate $P_m(A) = PS_{m_0}(A)$ using (4) taking $m_0 = \min\{m_1 \in m^*, m_1 > m\}$ and setting the coefficients $b_i = 0$ in (4) for $i = m_0, m_0 - 1, \ldots, m + 1$, at the same cost as evaluating $PS_{m_0}(A)$. Note that because of the way the polynomial is evaluated, the cost of using (4) is lower than that of Paterson–Stockmeyer as implemented in [2, Sec. 4.2] (compare (5) and [2, Eq. (4.3)]).

The matrix exponential is the most studied matrix function [4], [2, Chap. 10]. For $A \in \mathbb{C}^{n\times n}$ the matrix exponential of A can be defined by the Taylor series

$$\exp(A) = \sum_{i\geq 0} \frac{A^i}{i!}. \tag{6}$$

Another matrix function that has received attention recently is the matrix cosine, which can be defined analogously by means of its Taylor series

$$\cos(A) = \sum_{i\geq 0} (-1)^i \frac{A^{2i}}{(2i)!}. \tag{7}$$

Several efficient algorithms based on Taylor approximations have been proposed recently for the computation of the matrix exponential and cosine [6,8].

2.2. Padé approximations of matrix functions

The rational scalar function $r_{km}(x) = p_{km}(x)/q_{km}(x)$ is a $[k/m]$ Padé approximant of the scalar function $f(x)$ if $r_{k,m} \in \mathcal{R}_{k,m}$, $q_{km}(0) = 1$, and

$$f(x) - r_{km}(x) = O(x^{k+m+1}). \tag{8}$$

From now on, d_R will denote the degree of the last term of the Taylor series of f about the origin that $r_{km}(x)$ agrees with, i.e. $d_R = k + m$, and we will refer to d_R as the order of the Padé approximation. Table 2 (see [9, Table 2]) shows the maximum values of m that can be obtained for a given number of matrix products in $r_{mm}(A)$, denoted by the

Table 2
Cost C_R in terms of matrix products for diagonal rational approximation $r_{mm}(A)$ taking $D = 4/3M$. Approximation order d_R if r_{mm} is a Padé approximant of a given function f.

m^+	1	2	3	4	6	8	10	12	15	18	21
C_R	1.33	2.33	3.33	4.33	5.33	6.33	7.33	8.33	9.33	10.33	11.33
d_R	2	4	6	8	12	16	20	24	30	36	42

set m^+, and the corresponding computing cost, denoted by C_R given by

$$C_R = (2r + s - 3)M + D \approx (2r + s - 1 - 2/3)M, \quad r = m/s, \tag{9}$$

where s takes whichever value $s = \lceil\sqrt{2m}\,\rceil$ or $s = \lfloor\sqrt{2m}\rfloor$ that divides m and gives the smaller C_R. Table 2 also gives the corresponding order d_R of the approximation $r_{mm}(x)$ if it is a Padé approximant of a given function $f(x)$, i.e. $d_R = 2m$.

Finally, it is important to note that for a given f, k and m, a $[k/m]$ Padé approximant might not exist. Moreover, when computing rational approximations r_{km} of a function f for a given square matrix A, we must verify that the matrix $q_{km}(A)$ is nonsingular, and, for an accurate computation, that it is well conditioned. This is not the case for polynomial approximations, since they do not require matrix inversions.

2.3. Mixed rational and polynomial approximants

For a square matrix A the method proposed in [9] is based on using aggregations of mixed rational and polynomial approximants of the type

$$\begin{aligned} t_{ijs}(A) = \Big(\Big(\cdots\Big(&u_s^{(i)}(A)(v_s^{(i)}(A))^{-1} + u_s^{(i-1)}(A)\Big)(v_s^{(i-1)}(A))^{-1} + u_s^{(i-2)}(A)\Big) \\ &(v_s^{(i-2)}(A))^{-1} + \cdots + u_s^{(1)}(A)\Big)(v_s^{(1)}(A))^{-1} + w_{js}(A), \end{aligned} \tag{10}$$

where $v_s^{(k)}(A)$, $u_s^{(k)}(A)$, $k = 1, 2, \ldots, i$, are polynomials of A of degrees at most s, $w_{js}(A)$ is a polynomial of A with degree at most js, and $i \geq 0$, $s \geq 0$ and $j \geq 0$. Note that if $i = 0$ we consider that $t_{ijs}(A) = w_{js}(A)$, having no rational part. In [9, Sec. 4] a method to obtain t_{ijs} from rational approximations is given. Similarly to rational approximations, each multiplication by a matrix inverse is calculated as the solution of a multiple right-hand side linear system. Therefore, when computing $t_{ijs}(A)$ it is important to verify that the matrices $v_s^{(1)}(A)$, $v_s^{(2)}(A), \ldots,$ $v_s^{(i)}(A)$ are nonsingular and well conditioned. The total cost for computing (10), denoted by C_{RP}, is given by, see [9, Sec. 5]

$$C_{RP} = (s + j - 2)M + iD \approx (s + j - 2 + 4i/3)M, \quad j > 0,\, s > 0,\, i \geq 0. \tag{11}$$

Note that for the case where approximation (10) is intended to reproduce the first terms of the Taylor series of a given function f, it is equivalent to a $[(i + j)s/is]$ Padé approx-

Table 3
Approximation order d_{RP} if the mixed rational and polynomial approximation $t_{ijs}(A)$ from Section 2.3 reproduces the d_{RP} first terms of the Taylor series of a given function f, cost in terms of matrix products C_{RP} for the mixed rational and polynomial approximation $t_{ijs}(A)$, taking $D = 4/3M$, and values of i, j and s, that maximize d_{RP} for a given cost.

d_{RP}	1	2	3	4	6	9	10	12	15	16	20	21	25	28	30	35	36	42	45	49	54	55	56	63
i	0	0	1	0	1	1	2	1	2	1	2	3	2	3	2	3	4	3	4	3	4	5	3	4
j	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
s	1	2	1	2	2	3	2	4	3	4	4	3	5	4	6	5	4	6	5	7	6	5	8	7
C_{RP}	0	1	1.33	2	2.33	3.33	3.67	4.33	4.67	5.33	5.67	6	6.67	7	7.67	8	8.33	9	9.33	10	10.33	10.67	11	11.33

imant, and then, whenever it exists, t_{ijs} for scalar $x \in \mathbb{C}$ satisfies

$$f(x) - t_{ijs}(x) = O(x^{(2i+j)s+1}). \tag{12}$$

In that case we denote by d_{RP} the order of the mixed rational and polynomial approximation

$$d_{RP} = (2i + j)s. \tag{13}$$

Table 3 (see [9, Table 3]) shows for $t_{ijs}(A)$ the approximation order d_{RP} if t_{ijs} reproduces the first terms of the Taylor series of a given function f, and the cost C_{RP} in terms of matrix products for the values of i, j, s that maximize d_{RP} for a given cost. See [9] for a complete description.

3. On the evaluation of matrix polynomials. Application to the approximation of matrix functions

This section gives new general methods for evaluating matrix polynomials in a more efficient way than the combination of Horner and Paterson–Stockmeyer methods. Examples for computing the Taylor matrix polynomial approximation of degree m of the matrix exponential and the matrix cosine are given. These examples allow us to compute both approximations at a lower cost than Horner and Paterson–Stockmeyer methods. Note that in this section we used MATLAB R2017a for all the computations.

Example 3.1. Let

$$y_{02}(A) = A^2(c_4A^2 + c_3A), \tag{14}$$

$$\begin{aligned} y_{12}(A) = {} & (y_{02}(A) + d_2A^2 + d_1A)(y_{02}(A) + e_2A^2) \\ & + e_0y_{02}(A) + f_2A^2 + f_1A + f_0I, \end{aligned} \tag{15}$$

where c_4, c_3, d_2, d_1, e_2, e_0, f_2, f_1 and f_0 are scalar coefficients. In order to evaluate a matrix polynomial (3) of degree $m = 8$, taking $y_{12}(A) = P_m(A)$ and equating the coefficients of the matrix powers A^i, $i = 8, 7, \dots, 0$, the following system of equations arises

$$c_4 c_4 A^8 = b_8 A^8, \tag{16}$$

$$2c_3 c_4 A^7 = b_7 A^7, \tag{17}$$

$$(c_4(d_2 + e_2) + c_3 c_3)A^6 = b_6 A^6, \tag{18}$$

$$(c_4 d_1 + c_3(d_2 + e_2))A^5 = b_5 A^5, \tag{19}$$

$$(d_2 e_2 + c_3 d_1 + c_4 e_0)A^4 = b_4 A^4, \tag{20}$$

$$(d_1 e_2 + c_3 e_0)A^3 = b_3 A^3, \tag{21}$$

$$f_2 A^2 = b_2 A^2, \tag{22}$$

$$f_1 A = b_1 A, \tag{23}$$

$$f_0 I = b_0 I. \tag{24}$$

Note that for clarity the coefficient indices were chosen so that the sum of the indices is equal to the exponent of the power of A that coefficient is multiplying. For instance, for (16) one gets $4 + 4 = 8$, for (17) one gets $3 + 4 = 7$, for (18) one gets $4 + 2 = 6$ and $3 + 3 = 6$, and so on.

We can solve the previous system using the equations (16)–(24) from top to bottom. Using (16)–(19), one gets

$$c_4 = \pm\sqrt{b_8}, \tag{25}$$

$$c_3 = b_7/(2c_4), \tag{26}$$

$$d_2 + e_2 = (b_6 - c_3^2)/c_4, \tag{27}$$

$$d_1 = (b_5 - c_3(d_2 + e_2))/c_4. \tag{28}$$

If $b_8 \neq 0$ then $c_4 \neq 0$ and therefore c_4, c_3, the sum $d_2 + e_2$ and d_1 can be obtained explicitly. From now on we will denote $\underline{de}_2 = d_2 + e_2$ to simplify the notation and to remark that this quantity can be computed explicitly. Using (20) it follows that

$$e_0 = (b_4 - c_3 d_1 - \underline{de}_2 e_2 + e_2^2)/c_4, \tag{29}$$

where using (25)–(28) e_0 is a polynomial of second order in the variable e_2. Hence, using (21) and (29) one gets

$$d_1 e_2 + c_3 e_0 = b_3 \;\Rightarrow\; -b_3 + d_1 e_2 + c_3(b_4 - c_3 d_1 - \underline{de}_2 e_2 + e_2^2)/c_4 = 0 \tag{30}$$

Table 4
One possible choice for the coefficients in (14) and (15) for Taylor approximation of exponential and cosine of order $m = 8$.

	exp	cos
c_4	$4.980119205559973 \times 10^{-3}$	$2.186201576339059 \times 10^{-7}$
c_3	$1.992047682223989 \times 10^{-2}$	$-2.623441891606870 \times 10^{-5}$
d_2	$7.665265321119147 \times 10^{-2}$	$6.257028774393310 \times 10^{-3}$
d_1	$8.765009801785554 \times 10^{-1}$	$-4.923675742167775 \times 10^{-1}$
e_2	$1.225521150112075 \times 10^{-1}$	$1.441694411274536 \times 10^{-4}$
e_0	$2.974307204847627 \times 10^{0}$	$5.023570505224926 \times 10^{1}$

which is an equation of second order in the variable e_2, and therefore, using (25)–(28), the equation on the right-hand side of (30) has the solutions

$$e_2 = \frac{\frac{c_3}{c_4}\underline{de}_2 - d_1 \pm \sqrt{\left(d_1 - \frac{c_3}{c_4}\underline{de}_2\right)^2 + 4\frac{c_3}{c_4}\left(b_3 + \frac{c_3^2}{c_4}d_1 - \frac{c_3}{c_4}b_4\right)}}{2c_3/c_4}, \tag{31}$$

i.e., two solutions if we take $c_4 = \sqrt{b_8}$ from (25), and other two solutions if we take $c_4 = -\sqrt{b_8}$. Substituting the four solutions of e_2 in (27) and (29), four solutions are obtained for $d_2 = \underline{de}_2 - e_2$ and e_0, respectively, and from (22)–(24) it follows that

$$f_2 = b_2,\ f_1 = b_1,\ f_0 = b_0. \tag{32}$$

The cost of evaluating (15) is $3M$, i.e. one matrix product to compute and store A^2, and then two matrix products to compute (14) and (15), being $y_{12}(A)$ a polynomial of degree 8. From Table 1, the polynomial of maximum degree that can be computed with Horner and Paterson–Stockmeyer methods and cost $3M$ is the lower value $d_{PS} = 6$.

Table 4 shows one of the four solutions in IEEE double precision arithmetic for the coefficients of the Taylor approximation of the exponential and cosine, where $b_i = 1/i!$, and $b_i = (-1)^i/(2i)!$, respectively, for $i = 0, 1, \ldots, 8$. Note that all the four solutions are real, avoiding complex arithmetic if $A \in \mathbb{R}^{n \times n}$. In order to check the stability of the double precision arithmetic solutions c_i, d_i and e_i from Table 4, they were substituted in equations (16)–(21) to compute the relative error for each coefficient b_i, for $i = 3, 4, \ldots, 8$. For instance, from (21) it follows that the relative error for b_3 is $|b_3 - (d_1 e_2 + c_3 e_0)|/|b_3|$. We checked that all the relative errors for all b_i, for $i = 3, 4, \ldots, 8$, were below the unit roundoff in IEEE double precision arithmetic, i.e. $u = 2^{-53} \approx 1.11$.

Note that if we take

$$y_{12}(A) = (y_{02}(A) + d_2 A^2 + d_1 A)(y_{02}(A) + e_2 A^2 + e_1 A) + f_2 A^2 + f_1 A + f_0 I, \tag{33}$$

instead of (15), the four solutions for the corresponding coefficients for the exponential and cosine Taylor approximations of order $m = 8$ are complex. Therefore, if A is real, using (33) instead of (15) is not efficient for the computation of either matrix function since it is necessary to use complex arithmetic for evaluating (33).

Following Example 3.1 we can take in general

$$y_{0s}(A) = A^s \sum_{i=1}^{s} c_{s+i} A^i, \tag{34}$$

$$\begin{aligned} y_{1s}(A) = {} & \left(y_{0s}(A) + \sum_{i=1}^{s} d_i A^i \right) \left(y_{0s}(A) + \sum_{i=2}^{s} e_i A^i \right) \\ & + e_0 y_{0s}(A) + \sum_{i=0}^{s} f_i A^i, \end{aligned} \tag{35}$$

where A^i, $i = 2, 3, \dots, s$, can be computed once and stored to be reused in all the computations, and, then, $y_{1s}(A)$ is a matrix polynomial of degree, denoted by $d_{y_{1s}}$, and computing cost, denoted by $C_{y_{1s}}$

$$d_{y_{1s}} = 4s, \quad C_{y_{1s}} = s + 1, \quad s = 2, 3, \dots. \tag{36}$$

Note that (14) and (15) are a particular case of (34) and (35) where $s = 2$. Again, in order to evaluate a matrix polynomial $P_m(A)$ of degree $m = 4s$, we take $y_1(A) = P_m(A)$, and equate the coefficients of the matrix powers A^i, $i = m, m-1, \dots, 0$, from $y_{1s}(A)$ and $P_m(A)$. The solution for the coefficients taking $s = 2$ is given in Example 3.1, where the substitution of variables gives a polynomial equation in $e_s = e_2$ of degree 2 with the exact solution given by (31). In the following a general solution is given for $s > 2$. The s equations corresponding to the coefficients of the powers A^{4s-k}, for $k = 0, 1, \dots, s-1$ are, respectively

$$\sum_{i=0}^{k} c_{2s-i} c_{2s+i-k} = b_{4s-k}, \; k = 0, 1, \dots, s-1. \tag{37}$$

Since (37), is a triangular system, if $b_{4s} \neq 0$ then $c_{2s} \neq 0$ and it follows that:

$$\begin{aligned} c_{2s} &= \pm\sqrt{b_{4s}} \\ c_{2s-1} &= b_{4s-1}/(2c_{2s}), \\ c_{2s-k} &= (b_{4s-k} - \sum_{i=1}^{k-1} c_{2s-i} c_{2s+i-k})/(2c_{2s}), \; k = 2, 3, \dots, s-1. \end{aligned} \tag{38}$$

Note that if $b_{4s} < 0$, to prevent c_{2s} from being complex we can compute $y_{1s}(A) = -P_m(A)$ using (35), where $c_{2s} = -b_{4s} > 0$ which gives $P_m(A) = -y_{1s}(A)$.

Taking again $\underline{de}_i = d_i + e_i$ for abbreviation, and $\underline{de}_1 = d_1$, since there is no coefficient e_1 in (35), the equations corresponding to the coefficients of powers A^{3s-k}, for $k = 0, 1, \dots, s-1$, are, respectively

$$\sum_{j=s-k}^{s} c_{3s-k-j}\underline{de}_j + \sum_{i=1}^{s-k-1} c_{2s-k-i}c_{s+i} = b_{3s-k}, \quad k = 0, 1, \dots, s-2, \tag{39}$$
$$\sum_{j=s-k}^{s} c_{3s-k-j}\underline{de}_j = b_{3s-k}, \quad k = s-1,$$

and using (38) it follows that

$$\begin{aligned} \underline{de}_s &= (b_{3s} - \sum_{i=1}^{s-1} c_{2s-i}c_{s+i})/c_{2s}, \\ \underline{de}_{s-k} &= (b_{3s-k} - \sum_{j=s+1-k}^{s} c_{3s-k-j}\underline{de}_j - \sum_{i=1}^{s-1-k} c_{2s-k-i}c_{s+i})/c_{2s}, \\ &\qquad k = 1, 2, \dots, s-2, \\ d_1 &= (b_{2s+1} - \sum_{j=2}^{s} c_{2s+1-j}\underline{de}_j)/c_{2s}, \end{aligned} \tag{40}$$

where, if $c_{2s} \neq 0$, each sum $\underline{de}_i = d_i + e_i$, $i = s, s-1, \dots, 2$, and the coefficient d_1 can be obtained explicitly using the coefficients c_i, $i = s+1, s+2, \dots, 2s$ obtained from (38).

The equations corresponding to the coefficients of powers A^{2s-k}, for $k = 0, 1, \dots, s-1$, are

$$\sum_{i=0}^{k} d_{s-i}e_{s-k+i} + g_k + e_0c_{2s-k} = b_{2s-k}, \qquad k = 0, 1 \dots, s-1, \tag{41}$$

where

$$g_k = \sum_{i=1}^{s-1-k} c_{s+i}\underline{de}_{s-i-k}, \qquad k = 0, 1, \dots, s-2, \ \ g_{s-1} = 0, \tag{42}$$

and the coefficients g_k can be computed explicitly using (38) and (40).

Using (41) with $k = 0$ it follows that

$$e_s\underline{de}_s - e_s^2 + g_0 + e_0c_{2s} = b_{2s} \ \Leftrightarrow \ e_0 = (b_{2s} - g_0 - e_s\underline{de}_s + e_s^2)/c_{2s}, \tag{43}$$

provided that $c_{2s} \neq 0$. Hence, since $\underline{de}_s$, g_0 and c_{2s} can be computed using (38) and (40), the coefficient e_0 is a polynomial of second order in the variable e_s. Using now (41) with $k = 1$ one gets

$$e_{s-1}(\underline{de}_s - 2e_s) + e_s\underline{de}_{s-1} + g_1 + e_0c_{2s-1} = b_{2s-1}, \tag{44}$$

and then if $d_s \neq e_s$ it follows that $\underline{de}_s - 2e_s = d_s - e_s \neq 0$ and

$$e_{s-1} = (b_{2s-1} - g_1 - e_0 c_{2s-1} - e_s \underline{de}_{s-1})/(\underline{de}_s - 2e_s), \tag{45}$$

where e_{s-1} is a rational function of e_s, since by (43) e_0 is a polynomial of e_s of second order, and all the remaining quantities can be computed using (38), (40) and (42). Note that analogously, using (41) with $k = 2$ it follows that

$$e_{s-2}(\underline{de}_s - 2e_s) + e_s \underline{de}_{s-2} + e_{s-1}\underline{de}_{s-1} - e_{s-1}^2 + g_2 + e_0 c_{2s-2} = b_{2s-2}, \tag{46}$$

and then, again if $d_s \neq e_s$ it follows that

$$e_{s-2} = (b_{2s-2} - g_2 - e_0 c_{2s-2} - e_s \underline{de}_{s-2} - e_{s-1}\underline{de}_{s-1} + e_{s-1}^2)/(\underline{de}_s - 2e_s), \tag{47}$$

where similarly e_{s-2} is also a rational function of e_s since by (43) and (45) one gets that e_0 is a polynomial of e_s, and e_{s-1} is a rational function of e_s, and all the remaining quantities can be computed using (38), (40) and (42). Note that from (45) and (47) it follows that the rational function e_{s-2} has denominator $(\underline{de}_s - 2e_s)^3$.

Analogously, it is easy to show that

$$\begin{aligned} e_{s-k} = \Big(&b_{2s-k} - g_k - e_0 c_{2s-k} - e_s \underline{de}_{s-k} \\ &- \sum_{i=1}^{\lceil k/2\rceil -1} \big(e_{s-i}\underline{de}_{s-k+i} - e_{s-k+i}(\underline{de}_{s-i} - 2e_{s-i})\big) && (48) \\ &+ \begin{cases} 0\Big)/(\underline{de}_s - 2e_s), & \text{odd } k,\ 2 < k \le s-2, \\ - e_{s-k/2}\underline{de}_{s-k/2} - e_{s-k/2}^2\Big)/(\underline{de}_s - 2e_s), & \text{even } k,\ 2 < k \le s-2, \end{cases} && (49) \end{aligned}$$

where e_{s-k} is also a rational function of e_s with denominator $(\underline{de}_s - 2e_s)^{i_{k,s}}$ where $i_{k,s} > 0$ is an integer number depending on k and s.

The last equation of this group is

$$\begin{aligned} 0 = -b_{s+1} + e_0 c_{s+1} + e_s d_1 + \sum_{i=1}^{\lceil s/2\rceil -1} \big(e_{s-i}\underline{de}_{1+i} - e_{1+i}(\underline{de}_{s-i} - 2e_{s-i})\big) \\ + \begin{cases} 0, & \text{even } s > 2, \\ -e_{\frac{s+1}{2}}\underline{de}_{\frac{s+1}{2}} - e_{\frac{s+1}{2}}^2, & \text{odd } s > 2, \end{cases} \end{aligned} \tag{50}$$

Using the expressions (45), (47) and (48) obtained for e_{s-k}, for $k = 1, 2, \dots, s-2$, as rational functions of e_s and e_0 in (43) as a polynomial of e_s, it follows that expression (50) is a rational function of e_s, and multiplying it by $(\underline{de}_s - 2e_s)^{i_s}$, where i_s is an integer number depending on s, expression (50) can be written as a polynomial of e_s, provided that $\underline{de}_s - 2e_s = d_s - e_s \neq 0$. Hence, it has as many solutions as the resulting polynomial degree. Substituting these solutions in the expressions (45), (47) and (48) obtained for e_{s-k}, $k = 1, 2, \dots, s-2$, and e_0 from (43) the coefficients e_0 and e_{s-k},

Table 5
Order of the approximation d_{PS} that can be achieved using Horner and Paterson–Stockmeyer methods and order $d_{y_{1s}}$ using method given by (34) and (35) for a given cost C in terms of matrix products.

$C(M)$	3	4	5	6	7	8	9	10	11	12
d_{PS}	6	9	12	16	20	25	30	36	42	49
$d_{y_{1s}}$	8	12	16	20	24	28	32	36	40	44

$k = 1, 2, \dots, s-2$, can be obtained. The coefficients d_i, for $i = 1, 2, \dots, s$, can be obtained using the coefficients e_i, for $i = 0, 2, 3, \dots, s$, and (40). The solution for the coefficients with $s = 3$ and $s = 4$ gives polynomial equations in the variable e_s of degrees 4 and 6, respectively, and for $s \geq 5$ larger degree polynomials are obtained, and then, there are even more solutions for e_s.

Finally, from the equations involving A^i, for $i = s, s-1, \dots, 0$, it is easy to show that

$$f_{s-k} = b_{s-k} - \sum_{i=1}^{s-k-2} d_i e_{s-k-i} \tag{51}$$
$$f_i = b_i, \; i = 2, 1, 0.$$

Using (36) and Table 1, Table 5 shows the maximum orders that can be achieved for a given cost $C(M)$ in terms of matrix products with Horner and Paterson–Stockmeyer methods and the method given by $y_{1s}(A)$ using (34) and (35). Note that $y_{1s}(A)$ allows to evaluate a polynomial of degree greater than Horner and Paterson–Stockmeyer methods for a cost from $3M$ to $9M$, i.e. polynomial degrees from $d_{y_{1s}} = 8$ to 32 corresponding to $s = 2, 3, \dots, 8$, in $y_{1s}(A)$. We checked that there were at least 4 real solutions for all the coefficients in (34) and (35) when $y_{1s}(A)$ was equal to the exponential and cosine Taylor approximations of the corresponding degrees $d_{y_{1s}}$, avoiding complex arithmetic if A is a real square matrix.

3.1. Combination of $y_{1s}(A)$ with Horner and Paterson–Stockmeyer methods

The following proposition combines Horner and Paterson–Stockmeyer evaluation formula (4) with (35) to increase the degree of the resulting polynomial to be evaluated:

Proposition 1. *Let $z_{1ps}(x)$ be*

$$\begin{aligned} z_{1ps}(x) = \big(\big(\cdots \big(y_{1s}(x)x^s + a_{p-1}x^{s-1} + a_{p-2}x^{s-2} + \ldots + a_{p-s+1}x + a_{p-s}\big) \\ \times\ x^s + a_{p-s-1}x^{s-1} + a_{p-s-2}x^{s-2} + \ldots + a_{p-2s+1}x + a_{p-2s}\big) \\ \times\ x^s + a_{p-2s-1}x^{s-1} + a_{p-2s-2}x^{s-2} + \ldots + a_{p-3s+1}x + a_{p-3s}\big) \\ \vdots \\ \times\ x^s + a_{s-1}x^{s-1} + a_{s-2}x^{s-2} + \cdots + a_1 x + a_0, \end{aligned} \tag{52}$$

Table 6
Parameters s and p for $z_{1ps}(x)$ from (52) to obtain the same approximation order m as Horner and Paterson–Stockmeyer methods with a saving of 1 matrix product, where C_{PS} is the cost for evaluating (4) and $C_{z_{1ps}}$ is the cost for computing $z_{1ps}(x)$, both costs in terms of matrix products. The first row shows the maximum values of m obtained in $z_{1ps}(x)$ for a given number of matrix products.

m	8	12	16	20	20	25	30	30	36	42	42	49	56	56	$\cdots$
s	2	3	4	4	5	5	5	6	6	6	7	7	7	8	$\cdots$
p	0	0	0	4	0	5	10	6	12	18	14	21	28	24	$\cdots$
$C_{PS}(M)$	4	5	6	7	7	8	9	9	10	11	11	12	13	13	$\cdots$
$C_{z_{1ps}}(M)$	3	4	5	6	6	7	8	8	9	10	10	11	12	12	$\cdots$

where p is a multiple of s and $y_{1s}(x)$ is computed with (34) and (35). Then the degree of $z_{1ps}(x)$ and its computational cost for $x = A \in \mathbb{C}^{n\times n}$ are

$$d_{z_{1ps}} = 4s + p, \quad C_{z_{1ps}} = (1 + s + p/s)M. \tag{53}$$

Proof. The value of $d_{z_{1ps}}$ follows from (36) and (52). For the value of $C_{z_{1ps}}$ note that the matrix powers A^i, $i = 2, 3, \dots, s$, to be evaluated for Horner and Paterson–Stockmeyer evaluation formulas can be reused to compute $y_{1s}(A)$, and note also that one matrix product is needed to compute $y_{1s}(A)A^s$ in (52). Then, if p is a multiple of s, using (36) and (52) it follows the value of $C_{z_{1ps}}$ in (53). □

If we apply the evaluation formula (52) to evaluate a polynomial of degree $m+p$, i.e. $P_{m+p}(A)$, it follows that

$$z_{1ps}(A) = y_{1s}(A)A^p + \sum_{i=0}^{p-1} a_i A^i = P_{m+p}(A) = \sum_{i=0}^{m+p} b_i A^i. \tag{54}$$

Therefore, the coefficients a_i, $i = 0, 1, \dots, p-1$, are directly the corresponding coefficients b_i, $i = 0, 1, \dots, p-1$, from (54), and the coefficients from $y_{1s}(A)$ can be obtained changing b_i to b_{i+p} in (38), (40), (43), (45), (47), (48), (50), (51).

Using (53) Table 6 shows the parameters s and p to evaluate a polynomial of maximum degree m for a given cost using $z_{1ps}(A)$ from (52), and it is compared to the cost of Paterson–Stockmeyer method for the same values of m. Except for $m = 8$, all the values are in the set m^* from Table 1, and for all of them one matrix product is saved with respect to using only the Paterson–Stockmeyer method. The evaluation scheme $z_{1ps}(A)$ allows to evaluate polynomials of higher degree than that of the Paterson–Stockmeyer method for a cost greater than or equal to $3M$. Note that for a cost lower than or equal to $5M$ the maximum degree is obtained using

$$z_{1,p=0,\,s}(A) = y_{1s}(A), \tag{55}$$

from (35). Therefore, $z_{1ps}(A)$ can be considered as a generalization of $y_{1s}(A)$.

In order to evaluate polynomials of degrees different from those given in Table 6 other combinations $z_{1ps}(A)$ of the new method with the Paterson–Stockmeyer method can be

Table 7
One real solution for coefficients from (34) and (35) for computing Taylor approximation of the exponential of order $m = 30$ with (52) taking $s = 5$ and $p = 10$. Note that in this case coefficients in (54) are $b_i = 1/i!$, $i = 0, 1, \dots, 30$.

c_{10}	$-6.140022498994532 \times 10^{-17}$	e_4	$-2.785084196756015 \times 10^{-9}$
c_9	$-9.210033748491798 \times 10^{-16}$	e_3	$-4.032817333361947 \times 10^{-8}$
c_8	$-1.980157255925737 \times 10^{-14}$	e_2	$-5.100472475630675 \times 10^{-7}$
c_7	$-4.508311519886735 \times 10^{-13}$	e_0	$-1.023463999572971 \times 10^{-3}$
c_6	$-1.023660713518307 \times 10^{-11}$	f_5	$4.024189993755686 \times 10^{-13}$
d_5	$-1.227011356117036 \times 10^{-10}$	f_4	$7.556768134694921 \times 10^{-12}$
d_4	$-6.770221628797445 \times 10^{-9}$	f_3	$1.305311326377090 \times 10^{-10}$
d_3	$-1.502070379373464 \times 10^{-7}$	f_2	$2.087675698786810 \times 10^{-9}$
d_2	$-3.013961104055248 \times 10^{-6}$	f_1	$2.505210838544172 \times 10^{-8}$
d_1	$-5.893435534477677 \times 10^{-5}$	f_0	$2.755731922398589 \times 10^{-7}$
e_5	$-3.294026127901678 \times 10^{-10}$		

used, where p is not a multiple of s. For instance, a polynomial of degree $m = 23$ can be written as

$$P_{23}(x) = z_{1,7,4}(A) = (y_{1,4}(x)x^3 + a_6x^2 + a_5x + a_4)x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, \quad (56)$$

where the coefficients of $y_{1,4}(x)$ can be obtained similarly to those of $y_{1s}(x)$ in (54).

Example 3.2. Table 7 presents one solution for the coefficients for an example of $z_{1ps}(x)$ from (52) combining (34) and (35) with Horner and Paterson–Stockmeyer methods with $p = 10$ and $s = 5$ to compute Taylor approximation of the matrix exponential of order $m = 30$.

From (53) the cost of computing $z_{1,10,5}(A)$ is $C_{z_{1,10,5}} = 8M$, 1 matrix product less than using Horner and Paterson–Stockmeyer methods, see Table 6.

Analogously, using $z_{1ps}(x)$ from (52) with (34) and (35), we computed the coefficients from (34) and (35) for computing Taylor exponential and cosine approximation polynomials for all the approximation orders m in Table 6 up to approximation order $m = 81$. This process gave always several real solutions for all the coefficients involved. The maximum degree used in the Taylor approximation of the matrix exponential in double precision arithmetic from [6] is $m = 30$, and in the matrix cosine in [8] is $m = 16$. Note that the values from Table 7 can be directly used to evaluate Taylor approximation of order $m = 30$ in the algorithm from [6]. We also checked that using $z_{1,p=0,\,s}(A) = y_{1s}(A)$ from (35) gave also real coefficients for computing Taylor exponential and cosine approximation polynomials with $s = 2, 3, 4$. Hence, if A is a real square matrix, using $z_{1ps}(A)$ we can compute the exponential and cosine approximations using real arithmetic saving $1M$ with respect to the algorithms in [6,8] for Taylor polynomial degrees $m \in m^*$ from Table 1, $m \geq 12$.

Finally, similarly to Example 3.1 we checked the stability of the solutions of the coefficients in IEEE double precision arithmetic from Table 7, substituting them in the system of equations (37), (39) taking $\underline{de}_i = d_i + e_i$ where d_i and e_i are the values from Table 7, (41) and (51). Analogously, in all cases the relative error $|b_i - 1/i!|i!$, $i = p, p+1, \ldots, m+p$, see (54), was lower than the unit roundoff u.

In a similar way we also checked the stability for the computation of the exponential Taylor polynomial approximation for all the degrees m from Table 6 up to $m = 81$ obtaining the following results:

- There were 4 real solutions for all orders except for $m = 25$, with 12 real solutions, $m = 49$, 64, and 56 (with parameters $s = 8$, $p = 24$) with 8 real solutions, and $m = 42$ (with $p = 14$, $s = 7$) with 20 real solutions.
- The solutions for e_s were in decreasing module from $m = 12$ with $|e_s|$ of order 10^{-2} to $m = 81$ with $|e_s|$ of order 10^{-44}.
- In the case $m = 42$ (with $p = 14$, $s = 7$) the 20 solutions had all positive values $e_s \in \left[2.23 \times 10^{-16}, 8.07 \times 10^{-16}\right]$. Taking the solutions e_s in double precision arithmetic, from the 20 solutions there were 12 solutions that gave a maximum relative error for all coefficients b_i less than $3u$, being stable. However, 8 solutions showed certain signs of instability, giving a maximum relative error for coefficients b_i between 5.04×10^{-12} and $2.99 \times 10^{-10} > u$. Therefore, it is important to select a solution for e_s in double precision arithmetic that gives relative errors for all coefficients b_i of order u.

  We checked also the stability for the Taylor approximation of the matrix exponential in all the cases from Table 5 and found that the worst case was $m = 28$ with $s = 7$. This is not a case of practical use since, from Table 5 it has a cost $8M$, and from Table 6, using $z_{1ps}(A)$ with $p = 10$ and $s = 5$ gives the greater order $m = 30$ for the same cost, and that option was checked above to be stable. However, we checked its stability as a worst case study. This case gave 3 real solutions, where one of them had multiplicity 10. For the coefficients using the two solutions e_s with multiplicity 1 the maximum relative errors for all coefficients b_i where of order $10^{-15} > u$. We also checked the scalar case $A = 1$, giving relative errors $|\exp(1) - y_{1,s=7}(1)|/\exp(1) = 4.36 \times 10^{-16}$ and 3.70×10^{-15}, respectively. However, using the solution with multiplicity 10 gave a maximum relative error $10.75 \ggg u$ for coefficient b_8. For the rest of coefficients the maximum relative error was 1.49×10^{-14}, and for $|\exp(1) - y_{1,s=7}(1)|/\exp(1) = 9.81 \times 10^{-5}$, so the accuracy was much lower when using the solution of e_s with multiplicity 10.

  Therefore, it is necessary to check the stability of the solutions for e_s before using the method to evaluate a given polynomial. In general, we propose to select the solution for e_s in double precision arithmetic that gives the lowest maximum relative error for all coefficients b_i. If there is no solution giving relative errors of order u for a given polynomial with degree m, a different parameter selection from Tables 6 and 5 should be tested, since in Table 5 for $m > 16$ there are two possibilities for p and s that gives each value of m.

Table 8
Maximum approximation orders if any of the approximations reproduce the first terms of the Taylor series of a given function f for a given cost C for polynomial approximations, C_R for rational approximations and C_{RP} for mixed rational and polynomial approximants, where rational approximations are computed as in Section 2.2 and mixed rational and polynomial approximants are evaluated as in Section 2.3. The polynomial approximations considered are Horner and Paterson–Stockmeyer $PS_m(A)$ from Section 2.1, and $y_{1s}(A)$ and $z_{1ps}(A)$ from Section 3. Bold style is applied to the maximum degrees over all polynomial approximations, and to $t_{ijs}(A)$ when it provides the maximum degree over all approximations with an integer cost.

$C(M)$	$PS_m(A)$	$y_{1s}(A)$	$z_{1ps}(A)$	$C_R(M)$	$r_{mm}(A)$	$C_{RP}(M)$	$t_{ijs}(A)$
3	6	**8**	**8**	3.33	6	3.33	9
4	9	**12**	**12**	4.33	8	4.33	12
5	12	**16**	**16**	5.33	12	5.33	16
6	16	**20**	**20**	6.33	16	6	**21**
7	20	24	**25**	7.33	20	7	**28**
8	25	28	**30**	8.33	24	8	**35**
9	30	32	**36**	9.33	30	9	**42**
10	36	36	**42**	10.33	36	10	**49**
11	42	40	**49**	11.33	42	11	**56**

4. Comparison with existing methods

Using (36), (53) and Tables 1, 2 and 3, 5 and Table 6, it follows Table 8 that shows the approximation orders that can be obtained with Taylor polynomial approximations evaluated using Horner and Paterson–Stockmeyer methods $PS_m(A)$, $y_{1s}(x)$ from (35), $z_{1ps}(A)$ from (52), Padé rational approximation from Section 2.2, and the mixed rational and polynomial approximation from Section 2.3, for a given cost in terms of matrix products, if each approximation reproduces the first terms of the Taylor series of a given function f, whenever all the approximations exist. Note that the cost of solving the multiple right-hand side linear system in rational approximations was taken as $4/3M$.

Table 8 shows that the polynomial approximation that allows for the highest approximation order is $y_{1s}(A)$ for a cost $C \leq 6M$ and $z_{1ps}(A)$ for $C \geq 3M$. Note that in Section 3.1 for $C \leq 5M$ we took $z_{1ps}(A) = z_{1,p=0,s}(A) = y_{1s}(A)$, see (55). Hence, the approximation orders allowed by $z_{1ps}(A)$ for $C \geq 3M$ are higher than the approximation orders available with both Paterson–Stockmeyer and rational Padé method. The highest order for $C \geq 6M$ is given by the mixed rational and polynomial approximation $t_{ijs}(A)$ (10). In the following section particular examples are given in order to increase the efficiency of polynomial approximations even more.

5. General expressions

This section gives examples that suggest new general expressions for evaluating matrix polynomials more efficiently than the evaluation schemes given in Section 3.

Example 5.1. Consider

$$y_{02}(A) = A^2(c_{16}A^2 + c_{15}A), \tag{57}$$

$$y_{12}(A) = (y_{02}(A) + c_{14}A^2 + c_{13}A)(y_{02}(A) + c_{12}A^2 + c_{11}I) + c_{10}y_{02}(A), \tag{58}$$

Table 9
Coefficients of y_{02}, y_{12}, y_{22} from (57)–(59) for computing the matrix exponential Taylor approximation of order $m = 15$.

c_{16}	$4.018761610201036 \times 10^{-4}$	c_8	$2.116367017255747 \times 10^{0}$
c_{15}	$2.945531440279683 \times 10^{-3}$	c_7	$-5.792361707073261 \times 10^{0}$
c_{14}	$8.712167566050691 \times 10^{-2}$	c_6	$-1.491449188999246 \times 10^{-1}$
c_{13}	$4.017568440673568 \times 10^{-1}$	c_5	$1.040801735231354 \times 10^{1}$
c_{12}	$-6.352311335612147 \times 10^{-2}$	c_4	$-6.331712455883370 \times 10^{1}$
c_{11}	$2.684264296504340 \times 10^{-1}$	c_3	$3.484665863364574 \times 10^{-1}$
c_{10}	$1.857143141426026 \times 10^{1}$	c_2	$-1.224230230553340 \times 10^{-1}$
c_9	$2.381070373870987 \times 10^{-1}$	c_1	1

$$\begin{aligned} y_{22}(A) = {} & (y_{12}(A) + c_9 A^2 + c_8 A)(y_{12}(A) + c_7 y_{02}(A) + c_6 A) \\ & + c_5 y_{12}(A) + c_4 y_{02}(A) + c_3 A^2 + c_2 A + c_1 I, \end{aligned} \tag{59}$$

where the coefficients are numbered correlatively and A^2 is computed once and stored to be reused in all the computations. It is easy to show that the degree of polynomial $y_{22}(A)$ is $m = 16$ and it can be evaluated with a cost $C_{y_{22}} = 4M$.

Using function `solve` from MATLAB Symbolic Math Toolbox, Table 9 gives one solution for the coefficients to compute the exponential Taylor approximation $P_m(A)$ of order $m = 15$, i.e. $b_i = 1/i!$, $i = 0, 1, \ldots, 15$. For the solution given in Table 9 if we write $y_{22}(A)$ as a polynomial $P_m(A)$ of degree $m = 16$ the relative error for b_{16} with respect to the corresponding Taylor polynomial coefficient is

$$(b_{16} - 1/16!)16! = -0.454, \tag{60}$$

showing three significant digits.

We selected different possibilities for a new coefficient c_0 added in (57)–(59), trying compute the matrix exponential and the matrix cosine Taylor approximations of order 16, for instance changing (58) for

$$y_{12}(A) = (y_{02}(A) + c_{14}A^2 + c_{13}A + c_0 I)(y_{02}(A) + c_{12}A^2 + c_{11}I) + c_{10}y_{02}(A), \tag{61}$$

and other options. However, sometimes MATLAB could not find an explicit solution for the coefficients, and the other times MATLAB gave solutions with numeric instability.

Note that in Example 5.1 the degree of $y_{k,2}(A)$, $k = 1, 2$, is twice the degree of the polynomial $y_{k-1,2}(A)$, increasing the cost by just $1M$ when computing $y_{k,2}(A)$ using $y_{k-1,2}(A)$. Therefore, the polynomial degree increases exponentially while the cost increases linearly. Following this idea Proposition 2 gives expressions $y_{ks}(A)$, $k \geq 1$ more general than (34) and (35) where the degree of the polynomial $y_{ks}(A)$ is twice the degree of the polynomial $y_{k-1,s}(A)$, $k \geq 1$, while the cost increases by $1M$ when computing $y_{ks}(A)$ using $y_{k-1,s}(A)$:

Proposition 2. *Let*

$$y_{0s}(x) = x^s \sum_{i=1}^{s} c_i^{(0,1)} x^i + \sum_{i=0}^{s} c_i^{(0,2)} x^i, \tag{62}$$

$$\begin{aligned} y_{1s}(x) &= \left(\sum_{i=0}^{0} c_i^{(1,1)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(1,2)} x^i\right)\left(\sum_{i=0}^{0} c_i^{(1,3)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(1,4)} x^i\right) \\ &\quad + \sum_{i=0}^{0} c_i^{(1,5)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(1,6)} x^i, \end{aligned} \tag{63}$$

$$\begin{aligned} y_{2s}(x) &= \left(\sum_{i=0}^{1} c_i^{(2,1)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(2,2)} x^i\right)\left(\sum_{i=0}^{1} c_i^{(2,3)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(2,4)} x^i\right) \\ &\quad + \sum_{i=0}^{1} c_i^{(2,5)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(2,6)} x^i, \end{aligned} \tag{64}$$

$$\vdots$$

$$\begin{aligned} y_{ks}(x) &= \left(\sum_{i=0}^{k-1} c_i^{(k,1)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(k,2)} x^i\right)\left(\sum_{i=0}^{k-1} c_i^{(k,3)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(k,4)} x^i\right) \\ &\quad + \sum_{i=0}^{k-1} c_i^{(k,5)} y_{is}(x) + \sum_{i=0}^{s} c_i^{(k,6)} x^i, \end{aligned} \tag{65}$$

where $y_{ks}(x)$ is a polynomial of x. Then, the maximum polynomial degree, denoted by $d_{y_{ks}}$, and the computing cost if $x = A$, $A \in \mathbb{C}^{n\times n}$ in terms of matrix products, denoted by $C_{y_{ks}}$ are given by

$$d_{y_{ks}} = 2^{k+1}s, \quad C_{y_{ks}} = (s+k)M. \tag{66}$$

Proof. From (62), the maximum degree of the polynomial $y_{0s}(x)$ is $2s$. Then using (62)–(65) the maximum degree of $y_{is}(x)$, $i \le k$ is $2^{i+1}s$.

If $x = A$, $A \in \mathbb{C}^{n\times n}$, then the cost of computing $y_{ks}(A)$ is $s-1$ matrix products for computing A^i, for $i = 2, 3, \ldots, s$, and one matrix product in each iteration from (62)–(65), i.e. $k+1$. Therefore, $C_{y_{ks}} = (s+k)M$. □

Note that (34) and (35) are particular cases of Proposition 2 where $k = 1$ and some coefficients $c_i^{(l,j)}$, $l = 0, 1$, in (62) and (63) are zero. Similarly, (57)–(59) are particular cases of (62)–(64) where $k = 2$, $s = 2$ and some coefficients $c_i^{(l,j)}$, $l = 0, 1, 2$, are also zero.

If we write (65) in powers of x as

$$y_{ks}(x) = \sum_{i=0}^{m} a_i x^i, \tag{67}$$

then a_i, $i = 0, 1, \dots, m$, are functions of the coefficients $c_i^{(l,j)}$, for all i, j, l in (62)–(65). Hence, it is possible to evaluate matrix polynomial $P_m(A)$ using (62)–(65) if the system of equations

$$\begin{aligned} a_m(c_i^{(l,j)}) &= b_m, \\ a_{m-1}(c_i^{(l,j)}) &= b_{m-1}, \\ &\vdots \\ a_0(c_i^{(l,j)}) &= b_0, \end{aligned} \tag{68}$$

for all coefficients $c_i^{(l,j)}$ from (62)–(65) involved in each coefficient a_i, $i = 0, 1, \dots, m$, has at least one solution, where b_i are the polynomial coefficients of $P_m(A)$. We have obtained a general solution for evaluating polynomials using (34) and (35) corresponding to particular cases of (62) and (63). And we obtained one solution for computing the exponential Taylor approximation of order 15 with (57)–(59). Future work is addressed to obtain general solutions for evaluating matrix polynomials of different degrees using (62)–(65), and to study if at least there are particular solutions for evaluating polynomials such that the Taylor polynomial approximation of certain degrees for different matrix functions. That is the case of Example 5.1 which provides formulas for computing the exponential Taylor approximation polynomial of order $m = 15$ with a cost $C = 4M$. From Table 8 it follows that with a cost of $4M$ Paterson–Stockmeyer method allows to compute the matrix exponential Taylor approximation polynomial of order only $m = 9$, Padé rational method $r_{mm}(A)$ allows an order less than 8, the mixed rational and polynomial approximation $t_{ijs}(A)$ allows an order less than 12, and the new method based on (34) and (35) allows an order $m = 12$.

In the following example we consider the computation of the Taylor exponential approximation of order 16 by using the product of two polynomials of degree 8, both evaluated using (14) and (15).

Example 5.2. Let

$$h_{2m_1}(A) = P_{m_1}(A)P'_{m_1}(A) + \beta_0 = \sum_{i=0}^{m_1} b_i A^i \sum_{i=0}^{m_1} b'_i A^i + \beta_0, \tag{69}$$

where we took $m_1 = 8$, $b'_8 = b_8$, $b'_0 = 0$ and $h_{2m_1}(0) = \beta_0$, and, therefore, $P_{m_1}(A)$ and $P'_{m_1}(A)$ are both polynomials as (3) of degree 8, and $h_{2m_1}(A)$ can be written as a polynomial of degree 16 with 17 coefficients, i.e. b_i, $i = 0, 1, \dots, 8$, b'_i, $i = 1, \dots, 7$ and β_0. Using the MATLAB Symbolic Math Toolbox `solve` function, Table 10 presents one solution for the coefficients of an example where $h_{2m_1}(A) = \sum_{i=0}^{16} A^i/i!$, i.e. the exponential Taylor polynomial approximation of degree $m = 16$.

Note that one can evaluate both polynomials $P_{m_1}(A)$ and $P'_{m_1}(A)$ using an evaluation scheme (14) and (15), see Example 3.1. Finally, from (69) it follows that $\beta_0 = 1$ so that

Table 10
Coefficients from (69) for computing the matrix exponential Taylor approximation of order $m = 16$ where coefficient $b'_8 = b_8$ and $b'_0 = 0$.

b_8	$2.186201576339059 \times 10^{-7}$	b'_8	$2.186201576339059 \times 10^{-7}$
b_7	$9.839057366529322 \times 10^{-7}$	b'_7	$2.514016785489562 \times 10^{-6}$
b_6	$1.058964584814256 \times 10^{-5}$	b'_6	$3.056479369585950 \times 10^{-5}$
b_5	$1.554700173279057 \times 10^{-4}$	b'_5	$3.197607034851565 \times 10^{-4}$
b_4	$2.256892506343887 \times 10^{-3}$	b'_4	$2.585006547542889 \times 10^{-3}$
b_3	$2.358987357109499 \times 10^{-2}$	b'_3	$1.619043970183846 \times 10^{-2}$
b_2	$1.673139636901279 \times 10^{-1}$	b'_2	$8.092036376147299 \times 10^{-2}$
b_1	$7.723603212944010 \times 10^{-1}$	b'_1	$3.229486011362677 \times 10^{-1}$
b_0	$3.096467971936040 \times 10^{0}$	β_0	1

Table 11
Coefficients from system (16)–(24) for evaluating polynomials $y_1(A) = P_{m_1}(A)$ and $y'_1(x) = P'_{m_1}(A)$ from (69) with coefficients given by Table 10. Note that $f'_0 = 0$ since $y'_1(0) = b'_0 = 0$.

c_4	$4.675683454147702 \times 10^{-4}$	c'_4	$4.675683454147702 \times 10^{-4}$
c_3	$1.052151783051235 \times 10^{-3}$	c'_3	$2.688394980266927 \times 10^{-3}$
d_2	$-3.289442879547955 \times 10^{-2}$	d'_2	$2.219811707032801 \times 10^{-2}$
d_1	$2.868706220817633 \times 10^{-1}$	d'_1	$3.968985915411500 \times 10^{-1}$
e_2	$5.317514832355802 \times 10^{-2}$	e'_2	$2.771400028062960 \times 10^{-2}$
e_0	$7.922322450524197 \times 10^{0}$	e'_0	$1.930814505527068 \times 10^{0}$
f_2	$1.673139636901279 \times 10^{-1}$	f'_2	$8.092036376147299 \times 10^{-2}$
f_1	$7.723603212944010 \times 10^{-1}$	f'_1	$1.614743005681339 \times 10^{-1}$
f_0	$3.096467971936040 \times 10^{0}$	f'_0	0

$h_{2m_1}(0) = \exp(0) = 1$. Table 11 shows one solution for the coefficients from (16)–(24) using (25)–(32) taking $y_{1s}(A) = P_{m_1}(A)$, and the coefficients taking $y'_{1s}(A) = P'_{m_1}(A)$, corresponding to c'_4, c'_3, d'_2, d'_1, e'_2, e'_0, f'_2, f'_1 and f'_0.

In general, if we evaluate both polynomials $P_{m_1}(A)$ and $P'_{m_1}(A)$ by using (34) and (35) with $m_1 = 4s$, if there exist a solution for the coefficients b_i and b'_i for $P_{m_1}(A)$ and $P'_{m_1}(A)$, using (36) the degree of the matrix polynomial $h_{2m_1}(A)$ and its computing cost are

$$d_{h_{2m_1}} = 8s, \quad C_{h_{2m_1}} = (s + 4)M. \tag{70}$$

Table 12 shows the comparison of the polynomial degrees that can be obtained by Horner and Paterson–Stockmeyer methods, $z_{1ps}(A)$ from (52) and $h_{2m_1}(A)$ given by (69) varying m_1, for a given cost, whenever a solution for all the coefficients involved in $h_{2m_1}(A)$ exists. Since for $C > 6M$ they would be more efficient than Paterson–Stockmeyer method and for $C > 7M$ they would be more efficient than the method given by (52), it is worth studying if there exist evaluation schemes like (69) in general, or if at least they exist for the polynomial approximation of specific matrix functions or for the evaluation of matrix polynomials in the applications. Moreover, in order to obtain a polynomial degree equal to $2m_1$, note that one can think of other possibilities

Table 12
Order of the approximation d_{PS} that can be obtained using Horner and Paterson–Stockmeyer methods, order $d_{z_{1ps}}$ that can be obtained using $z_{1ps}(A)$ from (52), and order $d_{h_{m_1}}$ that can be obtained using method given by $h_{m_1}(A)$ from (69), using (34) and (35) for evaluating the polynomials therein, for a given cost C in terms of matrix products, whenever the solutions for the coefficients from (69), (34) and (35) exist.

$C(M)$	6	7	8	9	10	11	12
d_{PS}	16	20	25	30	36	42	49
$d_{z_{1ps}}$	20	25	30	36	42	49	56
$d_{h_{m_1}}$	16	24	32	40	48	56	64

to have $2m_1 + 1$ coefficients in $h_{2m_1}(A)$ different from selecting $b_{m_1} = b'_{m_1}$ and b'_0 as in Example 5.2.

Note that similarly to Section 3.1 Paterson–Stockmeyer method can be combined with any other method proposed above. And analogously to Example 5.2, we can also obtain new methods for evaluating matrix polynomials and matrix polynomial approximations using products of the evaluation schemes proposed above whenever a solution for the all the coefficients involved exists. The same powers A^i, $i = 1, 2, \ldots, s$, should be used in each evaluation scheme involved, so that they can be reused in all the computations. It is important to note that even in the case of the well known Padé approximations, for a given function f, k and m, a $[k/m]$ Padé approximant $r_{k,m}$ might not exist, see Section 2.2. Therefore, the existence of particular cases of the methods proposed in this section for computing matrix functions arising often in the applications is useful if they are more efficient than the existing methods in those concrete cases. That is the case of Example 5.1 with the matrix exponential Taylor approximation of order 15 which can be computed with just $4M$.

6. Conclusions

This paper proposes the new general evaluation schemes for matrix polynomials given by $y_{0s}(A)$ (34), $y_{1s}(A)$ (35) and $z_{1ps}(A)$ (52), and a method to check their stability was given. It was shown that these evaluation schemes allow to evaluate polynomials of degree higher than that of the Paterson–Stockmeyer method for the same cost. It was also shown that they provide a greater Taylor approximation order than diagonal Padé approximation for the same cost. Moreover, the new evaluation schemes are more efficient than the recent mixed rational and polynomial approximation from [9] for several orders of approximation.

Through Examples 5.1 and 5.2, we suggest the study of more general polynomial evaluation schemes that can be even more efficient, and applications to the Taylor approximation of matrix functions were given.

With the proposed methods we can state that the combination of Horner and Paterson–Stockmeyer methods is no longer the most efficient general method for evaluating matrix polynomials, and that Padé approximations are no longer more accurate than polynomial approximations for the same cost either.

Future work is:

- To determine if it is possible to find general solutions for evaluating matrix polynomials using (62)–(65) with $s \geq 2$ and $k \geq 2$, or at least particular solutions for cases of interest as in Example 5.1.
- To study if there are general solutions, or at least particular solutions for the matrix polynomial evaluation using products of the new proposed matrix polynomial evaluation schemes, similarly to Example 5.2.

Acknowledgements

This work has been supported by Spanish Ministerio de Economía y Competitividad and European Regional Development Fund (ERDF) grant TIN2014-59294-P. We thank the anonymous referee who revised this paper so thoroughly and carefully.

References

[1] M.S. Paterson, L.J. Stockmeyer, On the number of nonscalar multiplications necessary to evaluate polynomials, SIAM J. Comput. 2 (1) (1973) 60–66.
[2] N.J. Higham, Functions of Matrices: Theory and Computation, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.
[3] G.H. Golub, C.V. Loan, Matrix Computations, 3rd edition, Johns Hopkins Stud. Math. Sci., The Johns Hopkins University Press, 1996.
[4] C.B. Moler, C.V. Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (2003) 3–49.
[5] A.H. Al-Mohy, N.J. Higham, A new scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl. 31 (3) (2009) 970–989.
[6] P. Ruiz, J. Sastre, J. Ibáñez, E. Defez, High performance computing of the matrix exponential, J. Comput. Appl. Math. 291 (2016) 370–379.
[7] A.H. Al-Mohy, N.J. Higham, S. Relton, New algorithms for computing the matrix sine and cosine separately or simultaneously, SIAM J. Sci. Comput. 37 (1) (2015) A456–A487.
[8] J. Sastre, J. Ibáñez, P. Alonso, J. Peinado, E. Defez, Two algorithms for computing the matrix cosine function, Appl. Math. Comput. 312 (2017) 66–77.
[9] J. Sastre, Efficient mixed rational and polynomial approximation of matrix functions, Appl. Math. Comput. 218 (24) (2012) 11938–11946.
[10] S. Blackford, J. Dongarra, Installation Guide for LAPACK, LAPACK Working Note 41, Department of Computer Science University of Tennessee, 1999.