Electronic Transactions on Numerical Analysis.
Volume 38, pp. 202-217, 2011.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

EVALUATING THE FRÉCHET DERIVATIVE OF THE MATRIX PTH ROOT*

JOÃO R. CARDOSO †

Abstract. This paper shows that computing the Fréchet derivative of the matrix pth root is equivalent to solve a sequence of p Sylvester equations. This provides the theoretical support to design an algorithm for the effective computation of the Fréchet derivative. The conditioning of the Sylvester sequence is addressed and some numerical experiments are carried out to illustrate the results.

Key words. conditioning, matrix pth root, Sylvester equation, Fréchet derivative, Kronecker products, perturbation theory

1. Introduction. Let $p \geq 2$ be a positive integer. Given a matrix $A \in \mathbb{C}^{n\times n}$ with eigenvalues not belonging to the closed negative real axis, there is a unique matrix X such that $X^p = A$ whose eigenvalues lie on the sector of the complex plane defined by

$$-\frac{\pi}{p} < \arg(z) < \frac{\pi}{p}, \tag{1.1}$$

where $\arg(z)$ denotes the argument of the complex number z. This unique matrix X is called the *principal* pth root of A and is a primary matrix function of A. It is denoted by $A^{1/p}$. We refer the reader to [12] and [14, Ch. 6] for details about the theory of matrix pth roots and primary matrix functions.

The computation of matrix pth roots arises in many technical problems. Due to its closed relation with the matrix sector function, which in turn has applications in Control, many papers have been devoted to finding approximation methods for the matrix pth root. See, for instance, [8, 9, 13, 15, 17, 19, 24]. Applications of the matrix pth rootin other areas such as Finance and Healthcare are pointed out in [13].

It is well-known that the sensitivity of the matrix pth root (and primary matrix functions in general) to small perturbations in the data is measured by a condition number based on the norm of the Fréchet derivative. In this paper we propose a method for evaluating the Fréchet derivative of the matrix pth root which was inspired by the work developed previously by Kenney and Laub [16] for the Fréchet derivatives of the matrix exponential and the matrix logarithm. We first show that the computation of the Fréchet derivative of the power matrix X^p can be reduced to solve a sequence of p Sylvester equations. Then, by reversing the procedure, we are able to conclude that the evaluation of the Fréchet derivative of $A^{1/p}$ is also equivalent to solve p particular Sylvester equations of the form

$$A^{1/p}X - X(\gamma A^{1/p}) = C,$$

where $C \in \mathbb{C}^{n\times n}$ and γ is a complex scalar. Due to the simplified form of these equations, we can save much work by computing first the Schur decomposition of A to obtain Sylvester equations involving only triangular matrices on the left-hand side. The resulting method for the Fréchet derivative involves $O(pn^3)$ operations, which is efficient, at least for p not being a large prime number. In contrast with the method of Kenney and Laub which uses Pade approximants to the function $\tanh(x)/x$, an important feature of our method is that it does not require Pade approximations. As a consequence, our method is free from the truncation errors arising in the Padé approximation.

*Received January 25, 2011. Accepted for publication June 6, 2011. Published online July 4, 2011 Recommended by A. Frommer.

† Coimbra Institute of Engineering, Rua Pedro Nunes, 3030-199 Coimbra – Portugal, and Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra – Portugal (jocar@isec.pt)

ETNA
Kent State University
http://etna.math.kent.edu

The Sylvester equation is a much studied topic, both theoretically and computationally. For some theoretical background, see for instance, [4] and the references therein; for solving the Sylvester equation, one of the most popular methods is due to Bartels and Stewart [3], which is based on the Schur decomposition of matrices. An improvement of this method was proposed in [6]. See also [11] for a study of perturbation of this equation.

In general the methods for approximating the Fréchet derivative do not need to be highly accurate (in many cases an error less than 10^{-1} may be satisfactory). However, this in not the case of our method which performs with very good accuracy. Thus our method is suitable not only to approximate the Frechét derivative, but also for the computation of the matrix pth root, in the spirit of [16]. As far as we know, no numerical scheme has been previously proposed in the literature for the Fréchet derivative of the matrix pth root.

This paper is organized as follows. First, we recall some basic facts about the Fréchet derivatives of the matrix power and matrix pth root functions. Some new bounds are proposed. In Section 3, some lemmas are stated in order to provide the theoretical support of the main result (Theorem 3.4), which enable us to design an algorithm for computing the Fréchet derivative of the matrix pth root. Since this algorithm involves a sequence of particular Sylvester equations, often called a Sylvester cascade, in Section 4 we analyze the propagation of the error along the sequence and propose an expression for the condition number of each equation. Numerical experiments are carried out in Section 5 and some conclusions are drawn in Section 6.

Throughout the paper $\|.\|$ will denote a consistent matrix norm. The Frobenius norm and the 2-norm will be denoted by $\|.\|_F$ and $\|.\|_2$, respectively.

2. Background on the Fréchet derivative. Let $A, E \in \mathbb{C}^{n\times n}$. The Fréchet derivative of a matrix function f at A in the direction of E is a linear operator L_f that maps E to $L_f(A, E)$ such that $f(A + E) - f(A) - L_f(A, E) = O(\|E\|^2)$, for all $E \in \mathbb{C}^{n\times n}$. The Fréchet derivative may not exist, but if it does it is unique. The condition number of f at A is given by

$$\kappa_f(A) = \frac{\|L_f(A)\|\,\|A\|}{\|f(A)\|},$$

where

$$\|L_f(A)\| = \max_{\|E\|=1} \|L_f(A, E)\|;$$

see [12, Ch. 3]. If an approximation to $L_f(A, E)$ is known, then a numerical scheme like the power method on Fréchet derivative [12, Algorithm 3.20] can be used to estimate $\|L_f(A)\|$ and then the condition number $\kappa_f(A)$.

The following results characterize, respectively, the Fréchet derivative of the functions X^p and $X^{1/p}$.

LEMMA 2.1. *Let $A, E \in \mathbb{C}^{n\times n}$. If $L_{x^p}(A, E)$ denotes the Fréchet derivative of X^p at A in the direction of E, then*

$$L_{x^p}(A, E) = \sum_{j=0}^{p-1} A^{p-1-j} E A^j. \tag{2.1}$$

Proof. See, for instance, [1, Sec. 3]. ∎

LEMMA 2.2. *Let $A, E \in \mathbb{C}^{n\times n}$ and assume that A has no eigenvalues on the closed negative real axis. If $L_{x^{1/p}}(A, E)$ denotes the Fréchet derivative of $X^{1/p}$ at A in the direction*

ETNA
Kent State University
http://etna.math.kent.edu

of E, then $L_{x^{1/p}}(A,E)$ is the unique solution of the generalized Sylvester equation

$$\sum_{j=0}^{p-1} \left(A^{1/p}\right)^{p-1-j} X \left(A^{1/p}\right)^{j} = E. \tag{2.2}$$

Proof. See [12, Problem 7.4] and its solution and [22, Sec. 2.5]. See also [18, Thm. 5.1] for a similar result for the matrix sector function. □

Recently, an iterative method for solving a generalized Sylvester equation including (2.2) as a particular case was proposed in [20]. The method involves $O(pn^3)$ operations and in exact arithmetic the solution is reached after n^2 iterations, for any given initial guess. We have implemented the method but conclude that in finite precision arithmetic it seems to suffer from numerical instability. The convergence is too slow which makes the method impractical for the practical computation of the Fréchet derivative.

Another characterization of the Fréchet derivative $L_{x^{1/p}}(A,E)$ is given by means of the integral [5]:

$$L_{x^{1/p}}(A,E) = \frac{\sin(\pi/p)}{\pi} \int_0^\infty (xI+A)^{-1}E(xI+A)^{-1}x^{1/p}\,dx.$$

A problem that needs to be investigated is the approximation of this integral by numerical integration.

There is a closed expression for the Frobenius norm of the Fréchet derivative:

$$\|L_{x^{1/p}}(A)\|_F = \left\| \left(\sum_{j=0}^{p-1} \left[\left(A^{1/p}\right)^T \right]^j \otimes \left(A^{1/p}\right)^{p-1-j} \right)^{-1} \right\|_2;$$

see Problem 7.4 and its solution in [12]. It is not practical to use this formula directly because it involves Kronecker products and the inverse of an $n^2 \times n^2$ matrix.

If just a rough estimate of the norm of the Fréchet derivative is required, then some bounds are available in the literature. In [13], Higham and Lin have shown that

$$\frac{\|A^{1/p}A^{-1}\|}{p\|I\|} \le \|L_{x^{1/p}}(A)\| \le \frac{1}{p} e^{\|\log(A)/p\|}\|L_{\log}(A)\|,$$

where $\log(A)$ and $L_{\log}(A)$ denote, respectively, the logarithm of A and the Fréchet derivative of the matrix logarithm. More bounds are stated in the following lemmas.

LEMMA 2.3. *Let $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on the closed negative real axis and let $\sigma(A)$ denote the spectrum of A. Then*

$$\|L_{x^{1/p}}(A)\| \ge \frac{1}{\min\limits_{\lambda,\mu\in\sigma(A)} \left| \sum_{j=0}^{p-1} (\lambda^{1/p})^j\,(\mu^{1/p})^{p-1-j} \right|}. \tag{2.3}$$

Proof. Use [12, Th. 3.14] and note that, for $\lambda \neq \mu$,

$$\frac{\lambda^{1/p}-\mu^{1/p}}{\lambda-\mu} = \frac{1}{\sum_{j=0}^{p-1} (\lambda^{1/p})^j\,(\mu^{1/p})^{p-1-j}}.$$ □

From (2.3) the following lower bound for the condition number of the matrix pth root can be derived:

$$\kappa_{x^{1/p}}(A) \geq \frac{1}{\min_{\lambda,\mu\in\sigma(A)} \left| \sum_{j=0}^{p-1} (\lambda^{1/p})^j \, (\mu^{1/p})^{p-1-j} \right|} \frac{\|A\|}{\|A^{1/p}\|}.$$

This lower bound may be viewed as a generalization of (6.3) in [12] and shows that the condition number of the matrix pth root may be large if A has any eigenvalue near zero or close to the negative real axis.

LEMMA 2.4. *Let $A \in \mathbb{C}^{n\times n}$ and let $B := I - A^{-1}$. If $\|B\| < 1$, then*

$$\|L_{x^{1/p}}(A)\| \leq \frac{1}{p}(1 - \|B\|)^{-\frac{p+1}{p}}.$$

Proof. If $|x| < 1$, the function $f(x) = (1-x)^{-1/p}$ has the following Taylor expansion

$$f(x) = \sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{1}{p}\right)_j x^j,$$

where the symbol $(a)_j$ denotes the rising factorial:

$$(a)_0 = 1 \text{ and } (a)_j := a(a+1)\ldots(a+j-1).$$

Let $E \in \mathbb{C}^{n\times n}$. The Fréchet derivative of f can be written as

$$L_f(B,E) = \sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{1}{p}\right)_j \sum_{i=0}^{j-1} B^{j-1-i} E B^i.$$

Since the coefficients of the expansion of $f(x)$ are positive and $\|E\| = 1$, one has

$$\begin{aligned}\|L_f(B)\| &\leq \sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{1}{p}\right)_j j\|B\|^{j-1} \\ &= \frac{1}{p}(1-\|B\|)^{-\frac{p+1}{p}},\end{aligned}$$

and then the result follows. ∎

3. Computation of the Fréchet derivative. First, we shall recall some basic facts about Kronecker products and the vec operator. The Kronecker product of $A, B \in \mathbb{C}^{n\times n}$ is defined by $A \otimes B = [a_{ij}B] \in \mathbb{C}^{n^2\times n^2}$ and the Kronecker sum by $A \oplus B = A \otimes I + I \otimes B$. The notation vec stands for the linear operator that stacks the columns of a matrix into a long vector. If λ_r and μ_s denote the eigenvalues of A and B, respectively, then the eigenvalues of the matrix $\sum_{i,j=0}^{k} \alpha_{ij} A^i \otimes B^j$ are of the form $\sum_{i,j=0}^{k} \alpha_{ij} \lambda_r^i \mu_s^j$, for some $r, s = 1, \ldots, n$. The following properties also hold:

$$(A \otimes B)(C \otimes D) = AC \otimes BD \tag{3.1}$$

$$\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X), \tag{3.2}$$

where $A, B, C, D, X \in \mathbb{C}^{n\times n}$. We refer the reader to [7] and [14] for more details about Kronecker products.

Applying the vec operator to both sides of equation (2.1) and using (3.2), a Kronecker product-based form of the Fréchet derivative of $L_{x^p}(A, E)$ can be obtained:

$$\mathrm{vec}\,(L_{x^p}(A, E)) = K(A)\ \mathrm{vec}(E),$$

where

$$K(A) = \sum_{j=0}^{p-1} \left(A^T\right)^j \otimes A^{p-1-j} \in \mathbb{C}^{n^2\times n^2}. \tag{3.3}$$

Another expression for $K(A)$ is given in the following lemma. See [16, Sec. 2] and [12, Th. 10.13] for a similar result to the matrix exponential.

LEMMA 3.1. *Assume that $A \in \mathbb{C}^{n\times n}$ is nonsingular and that $K(A)$ is given by (3.3). If $\arg(\lambda) \neq \frac{(2k+1)\pi}{p-1}$, for any $\lambda \in \sigma(A)$ and $k = 0, 1, \ldots, p-2$, then*

$$K(A) = \left((A^{p-1})^T \oplus A^{p-1}\right)\ \phi(A^T \otimes A^{-1}),$$

where

$$\phi(x) = \frac{x^{p-1} + \cdots + x + 1}{x^{p-1} + 1}. \tag{3.4}$$

Proof. The assumption that $\arg(\lambda) \neq \frac{(2k+1)\pi}{p-1}$, for any $\lambda \in \sigma(A)$ and $k = 0, 1, \ldots, p-2$, ensures that $\phi(A^T \otimes A^{-1})$ is well defined. Since

$$\sum_{j=0}^{p-1} x^j = (x^{p-1} + 1)\phi(x),$$

the result follows from the following identities, where basic properties of Kronecker sums and products are used:

$$\begin{aligned} K(A) &= (I \otimes A^{p-1}) \sum_{j=0}^{p-1} (A^T \otimes A^{-1})^j \\ &= (I \otimes A^{p-1}) \left((A^T \otimes A^{-1})^{p-1} + I\right) \phi(A^T \otimes A^{-1}) \\ &= \left((A^T)^{p-1} \otimes I + I \otimes A^{p-1}\right) \phi(A^T \otimes A^{-1}) \\ &= \left((A^T)^{p-1} \oplus A^{p-1}\right) \phi(A^T \otimes A^{-1}). \qquad \square \end{aligned}$$

REMARK 3.2. Note that if $x \neq 1$, $\phi(x)$ can be simplified to $\phi(x) = \frac{x^p - 1}{(x-1)(x^{p-1}+1)}$.

A representation of the function $\phi(x)$ in terms of pth roots of 1 and $(p-1)$th roots of -1 is given in the next lemma.

LEMMA 3.3. *Let $\phi(x)$ be as in (3.4). Then*

$$\phi(x) = \left(\frac{\frac{x}{\alpha_{p-1}} - 1}{\frac{x}{\beta_{p-1}} - 1}\right) \cdots \left(\frac{\frac{x}{\alpha_2} - 1}{\frac{x}{\beta_2} - 1}\right) \left(\frac{\frac{x}{\alpha_1} - 1}{\frac{x}{\beta_1} - 1}\right), \tag{3.5}$$

ETNA
Kent State University
http://etna.math.kent.edu

where

$$\alpha_k = e^{\frac{2k\pi}{p}i}, \quad \beta_k = e^{\frac{(2k-1)\pi}{p-1}i}, \tag{3.6}$$

for $k = 1, \dots, p-1$, are respectively the pth roots of 1 (with the exception of 1) and the $(p-1)$th roots of -1.

Proof. Since $\phi(x) = \frac{x^p-1}{(x-1)(x^{p-1}+1)}$, a simple calculation leads to the result. $\square$

Now we have the theoretical support to show that computing the Fréchet derivative of $L_{x^p}(A, E)$ is equivalent to solving a set of recursive Sylvester equations. Assuming that the conditions of Lemma 3.1 are satisfied, we can write

$$\begin{aligned}\operatorname{vec}\left(L_{x^p}(A,E)\right) &= \left((A^T)^{p-1} \oplus A^{p-1}\right)\, \phi(A^T \otimes A^{-1})\, \operatorname{vec}(E)\\ &= \left((A^{p-1})^T \otimes I + I \otimes A^{p-1}\right)\, \operatorname{vec}(Y),\end{aligned}$$

where Y is an $n \times n$ matrix such that

$$\operatorname{vec}(Y) = \phi(A^T \otimes A^{-1})\, \operatorname{vec}(E). \tag{3.7}$$

Then, by (3.2),

$$L_{x^p}(A,E) = A^{p-1}Y + YA^{p-1}.$$

From (3.5) and (3.7),

$$\begin{aligned}\operatorname{vec}(Y) = &\left(\frac{A^T \otimes A^{-1}}{\beta_{p-1}} - I\right)^{-1} \left(\frac{A^T \otimes A^{-1}}{\alpha_{p-1}} - I\right) \cdots\\ &\cdots \left(\frac{A^T \otimes A^{-1}}{\beta_2} - I\right)^{-1} \left(\frac{A^T \otimes A^{-1}}{\alpha_2} - I\right) \times\\ &\times \left(\frac{A^T \otimes A^{-1}}{\beta_1} - I\right)^{-1} \left(\frac{A^T \otimes A^{-1}}{\alpha_1} - I\right) \operatorname{vec}(E).\end{aligned}$$

Let $X_0 := E$ and X_1 be an $n \times n$ complex matrix such that

$$\operatorname{vec}(X_1) = \left(\frac{A^T \otimes A^{-1}}{\beta_1} - I\right)^{-1} \left(\frac{A^T \otimes A^{-1}}{\alpha_1} - I\right) \operatorname{vec}(X_0). \tag{3.8}$$

Then, by (3.2), equation (3.8) is equivalent to the following Sylvester equation:

$$AX_1 - X_1\frac{A}{\beta_1} = AX_0 - X_0\frac{A}{\alpha_1}.$$

Due to the assumptions on the matrix A, this Sylvester equation has a unique solution because $\sigma(A) \cap \sigma(A/\beta_1) = \emptyset$. Proceeding as above, it follows that the Fréchet derivative of X^p can be expressed as

$$L_{x^p}(A,E) = A^{p-1}X_{p-1} + X_{p-1}A^{p-1}, \tag{3.9}$$

ETNA
Kent State University
http://etna.math.kent.edu

where X_{p-1} arises after solving the following $p-1$ recursive Sylvester equations:

$$\begin{aligned} X_0 &= E \\ AX_1 - X_1\frac{A}{\beta_1} &= AX_0 - X_0\frac{A}{\alpha_1} \\ AX_2 - X_2\frac{A}{\beta_2} &= AX_1 - X_1\frac{A}{\alpha_2} \\ \cdots \quad & \quad \cdots \\ AX_{p-1} - X_{p-1}\frac{A}{\beta_{p-1}} &= AX_{p-2} - X_{p-2}\frac{A}{\alpha_{p-2}}. \end{aligned}$$

Obviously, the explicit formula (3.9) is not recommended for computational purposes because there are more attractive formulae, for instance,

$$L_{x^p}(A,E) = Z_p\ ,$$

where Z_p is obtained by the recurrence

$$\begin{aligned} Y_1 &= AE,\ Z_1 = E \\ Y_{j+1} &= AY_j \\ Z_{j+1} &= Z_jA + Y_j, \end{aligned}$$

$j = 1, \ldots, p-1$, which can be obtained from (2.1); see also [1, (3.4)] for a similar formula. Our interest in deriving (3.9) is that the sequence of Sylvester equations involved can be reversed. This enables us to show that computing the Fréchet derivative of the matrix pth root is equivalent to solving a set of p recursive Sylvester equations. This is the main result of the paper and is stated in the next theorem.

THEOREM 3.4. *Let $A, E \in \mathbb{C}^{n\times n}$ and assume that A has no eigenvalue on the closed negative real axis. If α_k and β_k are as in (3.6) and if $B := A^{1/p}$, then*

$$L_{x^{1/p}}(A,E) = X_0, \tag{3.10}$$

where X_0 results from the following sequence of p Sylvester equations:

$$\begin{aligned} B^{p-1}X_{p-1} + X_{p-1}B^{p-1} &= E \\ BX_{p-2} - X_{p-2}\frac{B}{\alpha_{p-1}} &= BX_{p-1} - X_{p-1}\frac{B}{\beta_1} \\ \cdots \quad & \quad \cdots \\ BX_1 - X_1\frac{B}{\alpha_2} &= BX_2 - X_2\frac{B}{\beta_2} \\ BX_0 - X_0\frac{B}{\alpha_1} &= BX_1 - X_1\frac{B}{\beta_1}. \end{aligned} \tag{3.11}$$

Proof. Since A has no eigenvalues lying on the closed negative real axis, the eigenvalues of B satisfy the condition $-\pi/p < \arg(\lambda) < \pi/p$, and then the assumptions of Lemma 3.1 hold. This also guarantees that all the Sylvester equations involved in (3.11) have a unique solution. From Lemma 2.2 and (3.9) the result follows. ∎

The strategy of reversing a sequence of Sylvester equations to obtain the derivative of the inverse function has also been put forward in [16] for the matrix exponential and the

ETNA
Kent State University
http://etna.math.kent.edu

matrix logarithm. The main difference is that the sequence of Sylvester equations (3.11) is derived from the exact rational function $\phi(x)$ defined in (3.4) while Kenney and Laub used sequences of Sylvester equations arising from the $(8, 8)$ Padé approximants of $\tanh(x)/x$. Some advantages of our method are: the matrix A does not have to be close to the identity (no square rooting or squaring is necessary) and we do not have to deal with the truncation errors arising from the Padé approximation.

The method of Theorem 3.4 is summarized in the following algorithm. Since $A^{1/p}$ is required, it is recommended to combine the algorithm with a method for computing the principal pth root based on the Schur decomposition (for instance the methods of Smith [23] or Guo and Higham [8]). Once the Schur decomposition of A is known, no more Schur decompositions need to be evaluated in the algorithm.

ALGORITHM 3.5. *Let $A \in \mathbb{C}^{n\times n}$ with no eigenvalue on the closed negative real axis and let α_k and β_k be given as in (3.6). Assume in addition that the matrices U and $T^{1/p}$ in the decomposition $A^{1/p} = UT^{1/p}U^\star$, with U unitary and T upper triangular, are known. This algorithm computes the Fréchet derivative $L_{x^{1/p}}(A, E)$.*

1. *Set $B := T^{1/p}$ and $\tilde{E} = U^\star EU$;*
2. *Compute $\tilde{B} := B^{p-1}$ by solving the triangular matrix equation $BX = T$;*
3. *Find Y in the Sylvester equation $\tilde{B}Y + Y\tilde{B} = \tilde{E}$;*
4. *Set $Y_{p-1} := Y$;*
5. *for $k = p-1, \dots, 2, 1$, find Y_{k-1} in the Sylvester equation*

$$BY_{k-1} - Y_{k-1}\frac{B}{\alpha_k} = BY_k - Y_k\frac{B}{\beta_k};$$

6. *$L_{x^{1/p}}(A, E) = UY_0U^\star$; see (3.12) below.*

Cost. $(4p + 19/3)n^3$

To derive the above expression for estimating the cost of Algorithm 3.5, we have based the flop counts on the tables given in [12, p. 336-337]. We note that the effective cost may be higher than that estimate because the algorithm involves complex arithmetic. Step 6 is based on the following identity involving the Fréchet derivative and the Schur decomposition:

$$L_{x^{1/p}}(A, E) = UL_{x^{1/p}}(T, U^\star EU)U^\star, \tag{3.12}$$

which can be easily derived from (2.2) ; see also [12, Problem 3.2].

A drawback of Algorithm 3.5 is that the number of Sylvester equations involved increases with p which makes it quite expensive for large values of both n and p, especially when p is a large prime number. However, if p is large but is composite, say $p = p_1p_2$, then only $p_1 + p_2$ Sylvester equations are involved because

$$L_{x^{1/p}}(A, E) = L_{x^{1/p_1}}(A, E)A^{1/p_2} + A^{1/p_1}L_{x^{1/p_2}}(A, E)\;; \tag{3.13}$$

see [12, Th. 3.3]. At first glance it seems that for p large, finding (m, m) diagonal Padé approximants to $\phi(x)$, with $m \ll p$, would avoid the use of a large number of Sylvester equations. It happens that this strategy does not work because Padé approximants to $\phi(x)$ of some orders may not exist or may coincide. This phenomenon seems to be typical for Padé approximants to rational functions, as analyzed in detail throughout [2, Ch. 2]. See in particular the so-called Gragg example on page 13 and its Padé table on page 23. By virtue of [2, Th. 2.2], we also note that, for all $\ell, m \geq p-1$, Padé approximants to $\phi(x)$ of order (ℓ, m) coincide with $\phi(x)$.

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 3.5 is also suitable for real matrices, though it involves complex arithmetic because the α_k and β_k are not real. A possible strategy to overcome this problem is to write $\phi(x)$ in (3.4) as a product of rational functions involving real quadratic polynomials in the numerator and in the denominator. This holds only for p odd, which is the case that matters by virtue of (3.13). Thus, assuming that p is odd, we can rearrange the factors in (3.5) and write $\phi(x)$ as

$$\begin{aligned}\phi(x) &= \left(\frac{\frac{x}{\alpha_1}-1}{\frac{x}{\beta_1}-1}\right)\left(\frac{\frac{x}{\alpha_{p-1}}-1}{\frac{x}{\beta_{p-1}}-1}\right)\cdots\left(\frac{\frac{x}{\alpha_{(p-1)/2}}-1}{\frac{x}{\beta_{(p-1)/2}}-1}\right)\left(\frac{\frac{x}{\alpha_{(p+1)/2}}-1}{\frac{x}{\beta_{(p+1)/2-1}}}\right)\\ &= \left(\frac{x^2-(\alpha_1+\alpha_{p-1})x+1}{x^2-(\beta_1+\beta_{p-1})x+1}\right)\cdots\left(\frac{x^2-(\alpha_{(p-1)/2}+\alpha_{(p+1)/2})x+1}{x^2-(\beta_{(p-1)/2}+\beta_{(p+1)/2})x+1}\right).\end{aligned}$$

Proceeding as before, some calculation enables us to conclude that computing $L_{x^p}(A,E)$ is equivalent to solving $(p-1)/2$ recursive Sylvester equations of the form

$$X_kA^2-(\beta_k+\beta_{p-k})AX_kA+A^2X_k = X_{k-1}A^2-(\alpha_k+\alpha_{p-k})AX_{k-1}A+A^2X_{k-1}, \tag{3.14}$$

with $k=1,\ldots,(p-1)/2$. Since $\alpha_k+\alpha_{p-k}$ and $\beta_k+\beta_{p-k}$ are real, the problem of computing $L_{x^p}(A,E)$ can be reduced to solving $(p-1)/2$ real quadratic Sylvester equations. By reversing the procedure, a similar conclusion can be drawn for the Fréchet derivative of the matrix pth root. This is a topic that needs further research because it is not clear how to solve efficiently an equation of the form (3.14).

We end this section by noting that Algorithm 3.5 with minor changes is appropriate for solving a generalized Sylvester equation of the form

$$\sum_{j=0}^{p-1} B^{p-1-j}XB^j = C,$$

with B having eigenvalues satisfying $-\pi/p < \arg(\lambda) < \pi/p$ and $C \in \mathbb{C}^{n\times n}$; see [5].

4. Perturbation analysis. We have seen that the computation of the Fréchet derivative involves a sequence of $(p-1)$ Sylvester equations of the form

$$BX_{k-1} - X_{k-1}\frac{B}{\alpha_k} = BX_k - X_k\frac{B}{\beta_k},$$

where B, α_k, β_k are as in Theorem 3.4. The right-hand side of this equation is known and we need to solve the equation in order to find X_{k-1}. Since X_k results from solving the previous Sylvester equation, it may be affected by an error that propagates through the Sylvester cascade. We would like to know how this error affects the solution X_{k-1}. To simplify the notation, we work instead with the equation

$$BX - X\frac{B}{\alpha_k} = BY - Y\frac{B}{\beta_k}, \tag{4.1}$$

where we assume that Y is known while X has to be found. Consider the following perturbed version of equation (4.1):

$$\begin{aligned}(B+\Delta B)(X+\Delta X) - (X+\Delta X)\left(\tfrac{1}{\alpha_k}(B+\Delta B)\right) &=\\ = (B+\Delta B)(Y+\Delta Y) - (Y+\Delta Y)\left(\tfrac{1}{\beta_k}(B+\Delta B)\right).&\end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

Ignoring second order terms, we obtain

$$B\Delta X - \frac{1}{\alpha_k}\Delta X\, B = \Delta B(Y-X) + \left(\frac{1}{\alpha_k}X - \frac{1}{\beta_k}Y\right)\Delta B + B\Delta Y - \frac{1}{\beta_k}\Delta Y\, B.$$

Applying the vec operator and using its properties,

$$\begin{aligned}&\left(I\otimes B - \tfrac{B^T}{\alpha_k}\otimes I\right)\mathrm{vec}(\Delta X) = \\ &\qquad = \left[I\otimes\left(\tfrac{X}{\alpha_k}-\tfrac{Y}{\beta_k}\right) - (X-Y)^T\otimes I, \quad I\otimes B - \tfrac{B^T}{\beta_k}\otimes I\right]\begin{bmatrix}\mathrm{vec}(\Delta B)\\ \mathrm{vec}(\Delta Y)\end{bmatrix},\end{aligned}$$

which can be written in the form

$$\mathrm{vec}(\Delta X) = M^{-1}[N_1\ N_2]\begin{bmatrix}\mathrm{vec}(\Delta B)\\ \mathrm{vec}(\Delta Y)\end{bmatrix},$$

where

$$\begin{aligned}M &= I\otimes B - \frac{B^T}{\alpha_k}\otimes I,\\ N_1 &= I\otimes\left(\frac{X}{\alpha_k}-\frac{Y}{\beta_k}\right) - (X-Y)^T\otimes I,\\ N_2 &= I\otimes B - \frac{B^T}{\beta_k}\otimes I.\end{aligned}$$

Since

$$\mathrm{vec}(\Delta X) = M^{-1}\Big[\,\|B\|_F\,N_1 \quad \|Y\|_F\,N_2\,\Big]\begin{bmatrix}\mathrm{vec}(\Delta B)/\|B\|_F\\ \mathrm{vec}(\Delta Y)/\|Y\|_F\end{bmatrix},$$

it follows that

$$\|\Delta X\|_F \le \sqrt{2}\left\|M^{-1}\Big[\,\|B\|_F\,N_1 \quad \|Y\|_F\,N_2\,\Big]\right\|_2 \max\left\{\frac{\|\Delta B\|_F}{\|B\|_F}, \frac{\|\Delta Y\|_F}{\|Y\|_F}\right\}, \tag{4.2}$$

and therefore

$$\frac{\|\Delta X\|_F}{\|X\|_F} \le \sqrt{2}\,\delta\,\Phi, \tag{4.3}$$

where

$$\begin{aligned}\delta &= \max\left\{\frac{\|\Delta B\|_F}{\|B\|_F}, \frac{\|\Delta Y\|_F}{\|Y\|_F}\right\},\\ \Phi &= \frac{\left\|M^{-1}\Big[\,\|B\|_F N_1 \quad \|Y\|_F N_2\,\Big]\right\|_2}{\|X\|_F}.\end{aligned}$$

Inequality (4.3) gives a bound for the relative error of the solution X of the Sylvester equation (4.1) in terms of the relative errors affecting B and Y. According to [11, Sec. 4], where a similar perturbation analysis was performed, this bound is sharp (to first order in δ) and Φ can be seen as the condition number for the Sylvester equation (4.1). Thus, if Φ is small an accurate result is expected after solving the Sylvester cascade.

ETNA
Kent State University
http://etna.math.kent.edu

To obtain a better understanding into how the error propagates through the Sylvester equations, we simplify slightly the problem by assuming that B is known exactly, that is, $\Delta B = 0$. Then (4.2) becomes

$$\|\Delta X\|_F \leq \left\|M^{-1}N\right\|_2 \|\Delta Y\|_F,$$

with

$$M = I \otimes B - \frac{B^T}{\alpha_k} \otimes I, \quad N = I \otimes B - \frac{B^T}{\beta_k} \otimes I, \tag{4.4}$$

and (4.3) simplifies to

$$\frac{\|\Delta X\|_F}{\|X\|_F} \leq \Phi \frac{\|\Delta Y\|_F}{\|Y\|_F}, \tag{4.5}$$

with

$$\Phi = \frac{\left\|M^{-1}N\right\|_2 \|Y\|_F}{\|X\|_F}.$$

We have computed the value of Φ in (4.5) for several matrices B with eigenvalues satisfying (1.1) and have observed that for several examples Φ is small (more precisely, $1 \leq \Phi \leq 2$), but it can be large (see Section 5). To understand why, let us assume in addition that B is normal. Then the matrices M and N given in (4.4) are also normal. Moreover, they commute and can be written as

$$M = \left(-\frac{B^T}{\alpha_k}\right) \oplus B,$$
$$N = \left(-\frac{B^T}{\beta_k}\right) \oplus B.$$

The product $M^{-1}N$ is also normal and hence

$$\|M^{-1}N\|_2 = \max_{\lambda \in \sigma(M^{-1}N)} |\lambda|.$$

Since the eigenvalues of $M^{-1}N$ are of the form

$$\lambda = \frac{\beta_k \lambda_i - \lambda_j}{\alpha_k \lambda_r - \lambda_s},$$

for some $\lambda_i, \lambda_j, \lambda_r, \lambda_s \in \sigma(B)$, we can see that if the eigenvalues of B are close to zero or have arguments close to $\pm\pi/p$, then $\left\|M^{-1}N\right\|_2$ may be large, and then a large condition number Φ is expected.

5. Numerical experiments. Algorithm 3.5 has been implemented in MATLAB, with unit roundoff $u \approx 1.1 \times 10^{-16}$. For the numerical experiments we first consider the following 8 pairs (A, E) of real and complex matrices combined with $p = 5$, $p = 19$ and $p = 53$:

- Pair 1: $\mathtt{A} = \mathtt{hilb}(8)$, $\mathtt{E} = \mathtt{rand}(8)$; A is an 8×8 Hilbert matrix which is almost singular and E is a randomized matrix of the same size, with uniformly distributed pseudorandom entries on the open interval $]0, 1[$;
- Pair 2: $\mathtt{A} = \mathtt{gallery}('\mathtt{frank}', 8)$, $\mathtt{E} = \mathtt{rand}(8)$; A is a Frank 8×8 matrix taken from the MATLAB gallery which is very ill conditioned;

ETNA
Kent State University
http://etna.math.kent.edu

- Pair 3: $\texttt{A} = \texttt{gallery}(3)$, $\texttt{E} = \texttt{rand}(3)$; A is an 3×3 badly conditioned matrix;
- Pair 4: $A = SQS^{-1}$, where

$$Q = \begin{bmatrix} e^5 & 0 & 0 & 0 \\ 0 & e^{-5} & 0 & 0 \\ 0 & 0 & \cos(3.14) & -\sin(3.14) \\ 0 & 0 & \sin(3.14) & \cos(3.14) \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 0 & 0 & 9 & 10 \\ 0 & 0 & 11 & 12 \end{bmatrix},$$

  and $\texttt{E} = \texttt{rand}(4)$; This matrix has eigenvalues very close to the negative real axis and was taken from [16, Ex. 6];
- Pairs 5 to 8: $\texttt{A} = \texttt{expm}(2 * \texttt{randn}(10) + 2 * \texttt{i} * \texttt{randn}(10))$, $\texttt{E} = \texttt{rand}(10) + \texttt{i} * \texttt{rand}(10)$; Both A and E are complex with size 10×10.

To study the quality of the computed Fréchet derivative $\tilde{L} \approx L_{x^{1/p}}(A, E)$, we evaluated the relative residual of the $n^2 \times n^2$ linear system that results from applying the vec operator to the generalized Sylvester equation (2.2):

$$\rho(A, E) = \frac{\|M \operatorname{vec}(\tilde{L}) - \operatorname{vec}(E)\|_F}{\|M\|_F \|\operatorname{vec}(\tilde{L})\|_F}. \tag{5.1}$$

The results are displayed in Figure 5.1, where we can observe that the computed Fréchet derivative is very satisfactory in the sense that $\rho(A, E) \lesssim u$ for each pair and each p considered. The Sylvester equations in Algorithm 3.5 have been solved by the Bartels and Stewart method [3], whose codes are available in the Matrix Function Toolbox [10]. For the computation of the principal matrix pth root $T^{1/p}$ in step 1, we have modified the Schur-Newton Algorithm 3.3 in [8] in order to run it with complex arithmetic.

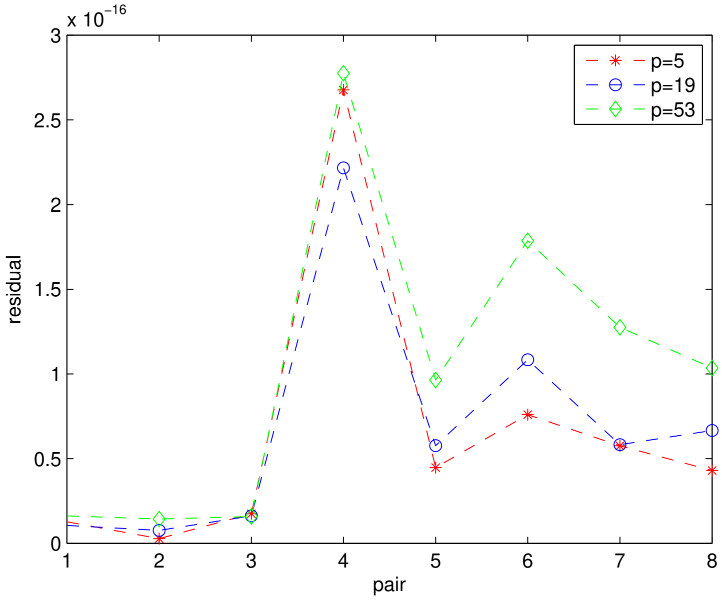

FIG. 5.1. *Values of $\rho(A, E)$ for the 8 pairs of matrices (A, E) combined with the values $p = 5, 19, 53$.*

Given an algorithm for computing a matrix function f, the Fréchet derivative can be obtained using the following relation ([21, Thm. 2.1], [12, Eq. (3.16)])

$$f\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}\right) = \begin{bmatrix} f(A) & L_f(A,E) \\ 0 & f(A) \end{bmatrix} \tag{5.2}$$

by reading off the $(1,2)$ block of the matrix in the right-hand side in a single invocation of any algorithm that computes f. Formula (5.2) has the advantage of providing a very simple algorithm for the Fréchet derivative. A drawback of (5.2) is that it involves an $2n \times 2n$ matrix, and then the cost of evaluating $L_f(A,E)$ is about 8 times the cost of $f(A)$, unless the particular block structure is exploited. Moreover, as pointed out in [1, Sec. 6], this formula is not suitable to be combined with techniques widely used for some matrix functions such as scaling and squaring the matrix exponential, inverse scaling and squaring the matrix logarithm, and square rooting and squaring the matrix pth root. The main reason is that when $\|E\| \gg \|A\|$, algorithms based on (5.2) may require the computation of unnecessary scalings (respectively, square rootings) to bring the matrix in the left-hand side close to zero (resp., close to the identity). This unpleasant situation has been addressed in many papers ; see for instance [16] and the references therein. Since $L_f(A, \alpha E) = \alpha L_f(A,E)$, an algorithm for computing $L_f(A,E)$ should not be influenced by the norm of E.

We have carried out some numerical experiments with (5.2) in order to compare it with Algorithm 3.5. For comparison purposes, we have assumed that the computation of a Schur decomposition and one pth root is included in Algorithm 3.5. For the computation of the matrix pth root we have considered the algorithms of Smith [23, Algorithm 4.3] and of Guo and Higham [8, Algorithm 3.3]. Both algorithms are based on the Schur decomposition and are available in [10]. Due to the reasons mentioned above, we have not considered the algorithm of Higham and Lin [13] which involves square rooting and squaring.

We first compare the cost. Since the Schur decomposition of an $n \times n$ matrix involves about $25n^3$ flops, much work can be saved if the block structure of the matrix in the left-hand side of (5.2) is respected:

$$\begin{bmatrix} A & E \\ 0 & A \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} T & U^\star E U \\ 0 & T \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix}^\star, \tag{5.3}$$

where $A = UTU^\star$, with T upper triangular and U unitary. Using (5.3), the Schur decomposition of the $2n \times 2n$ matrix in (5.2) can be computed through $29n^3$ flops, instead of $200n^3$ if computed directly. Combining the algorithms of Smith and Guo and Higham with (5.3) we can say that these algorithms in general requires slightly fewer flops than Algorithm 3.5. However, this does not mean that algorithms of Smith and Guo and Higham are faster than Algorithm 3.5. Indeed, in tests carried out with p a prime number between 2 and 100 and several matrices with sizes $10 \leq n \leq 100$ we have measured the CPU times (in seconds) of the three algorithms and noticed that Algorithm 3.5 is the fastest. The differences become more significative when we fix p and increase n. Table 5.1 displays the average CPU times required for computing the Fréchet derivative $L_{x^{1/p}}(A,E)$, for 4 pairs of matrices (A,E) (pairs 9 to 12) obtained using `A = rand(n)^2` and `E = randn(n)`, for $n = 10, 50, 80, 100$ and $p = 19$. It is clear that the algorithm of Smith is much slower than the others and that Algorithm 3.5 performs much faster than the algorithm of Guo and Higham. One of the reasons for this may be related with storage. While Algorithm 3.5 requires the storage of $n \times n$ matrices, the other algorithms deal with $2n \times 2n$ matrices that may require larger storage. However, we believe that using formula (5.2) to derive new algorithms for the Fréchet derivative by exploiting the block structure may be a promising topic for further research. We recall that some work has

Pair	size	Algorithm 3.5	Smith	Guo and Higham
9	10×10	0.0059	0.3897	0.0255
10	50×50	0.1027	9.5072	0.9854
11	80×80	0.3682	22.1088	2.9212
12	100×100	0.6439	39.2173	4.1325

TABLE 5.1
CPU times (in seconds) required by Algorithm 3.5 and algorithms of Smith and Guo and Higham, with $p = 19$.

already been done for the Fréchet derivative of the matrix square root [1, Sec. 2] and for the matrix sector function [18, Thm. 5.3], though implementation issues have not been discussed.

We have also analysed the residuals (5.1) produced by the two algorithms based on (5.2) with several pairs of matrices, including pairs 1 to 12 mentioned above. Although in some tests Algorithm 3.5 produced slightly smaller residuals, the residuals were in general of the same order.

We saw above that the Sylvester equations in the sequence (3.11) may be ill conditioned when the eigenvalues of A are close to the negative real axis. To illustrate this phenomenon, we have considered the matrix $A = SQ(\theta)S^{-1}$, where

$$Q(\theta) = \begin{bmatrix} e^2 & 0 & 0 & 0 \\ 0 & e^{-2} & 0 & 0 \\ 0 & 0 & \cos(\theta) & -\sin(\theta) \\ 0 & 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad \mathtt{S} = \mathtt{randn}(4),$$

and $\mathtt{E} = \mathtt{rand}(4)$. Figure 5.2 displays the values of the condition number Φ given by (4.3) for each of the equations involved in the Sylvester sequence (3.11), for $p = 19$. There are exactly 19 Sylvester equations involved in this sequence. Both graphics use a linear scale in the x axis and a log-scale in the y axis. The plot in the left-hand side concerns to the value of $\theta = 3.14$ in A and the one in the right-hand side to $\theta = \pi/2$. As expected, in the extreme case of A having eigenvalues very close to the negative real axis ($\theta = 3.14$) the Sylvester equations may be badly conditioned. However, this does not happen with other values of θ, as illustrated in the right-hand side picture.

6. Conclusions. The Fréchet derivative is the key to understand the effects of perturbations of first order in primary matrix functions. For the particular case of the matrix pth root we have derived an effective method for the computation of its Fréchet derivative, which involves $O(pn^3)$ operations and is based on the solution of a certain sequence of Sylvester equations. Both theory and computation of Sylvester equations are well understood. Numerical experiments we have carried out showed that the proposed method is faster than methods based on (5.2) and has relative residuals close to the unit roundoff. Some issues that need further research, as for instance the restriction of Algorithm 3.5 to the real case involving only real arithmetic, were pointed out.

Acknowledgments. The author thanks the referees for their valuable and insightful comments and suggestions.

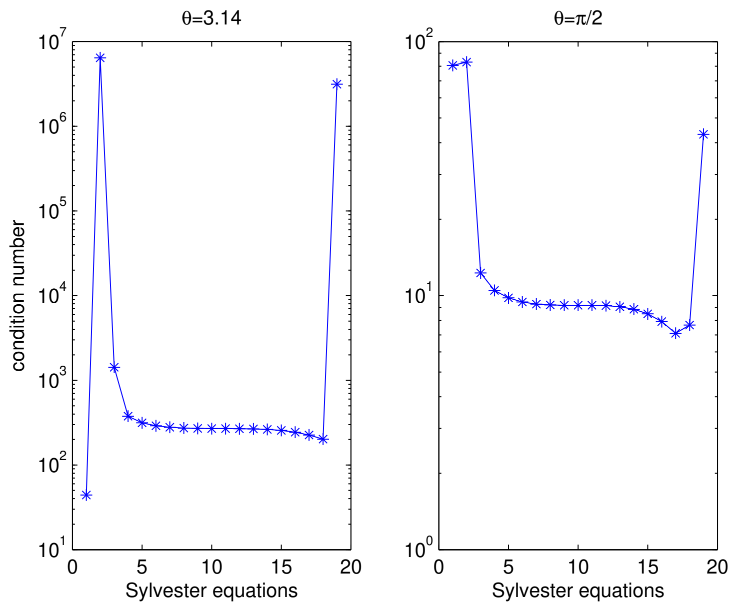

FIG. 5.2. *The condition number Φ in (4.3) of each equation involved in the Sylvester sequence (3.11), with $p = 19$.*

REFERENCES

[1] A. AL-MOHY AND N. J. HIGHAM, *Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimate*, SIAM J. Matrix Anal. Appl., 30 (2009), pp. 1639–1657.

[2] G. A. BAKER, *Essentials of Padé approximants*, Academic Press, New York, USA, 1975.

[3] R. H. BARTELS AND G. W. STEWART, *Algorithm 432: Solution of the matrix equation AX+XB=C*, Comm. ACM, 15 (1972), pp. 820–826.

[4] R. BHATIA AND P. ROSENTHAL, *How and why to solve the operator equation $AX - XB = Y$*, Bull. London Math. Soc., 29 (1997), pp. 1–21.

[5] R. BHATIA AND M. UCHIYAMA, *The operator equation $\sum_{i=0}^{n} A^{n-i}XB^i = Y$*, Expo. Math., 27 (2009), pp. 251–255.

[6] G. H. GOLUB, S. NASH, AND C. F. VAN LOAN, *A Hessenberg-Schur method for the problem $AX + XB = C$*, IEEE Trans. Automat. Control, 24 (1979), pp. 909–913.

[7] A. GRAHAM, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester, UK, 1981.

[8] C. -H. GUO AND N.J. HIGHAM, *A Schur-Newton method for the matrix pth root and its inverse*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 788–804.

[9] M.A. HASAN, A.A. HASAN AND K.B. EJAZ, *Computation of matrix nth roots and the matrix sector function*, in: Proc. of the 40th IEEE Conf. on Decision and Control, Orlando, (2001), pp. 4057–4062.

[10] N. J. HIGHAM, *The Matrix Function Toolbox*, http://www.ma.man.ac.uk/$\sim$higham/mftoolbox.

[11] ———, *Perturbation theory and backward error for $AX - XB = C$*, BIT, 33 (1992), pp. 124–136.

[12] ———, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, 2008.

[13] N. J. HIGHAM AND L. LIN, *A Schur–Padé algorithm for fractional powers of a matrix*, MIMS EPrint 2010.91, Manchester Institute for Mathematical Sciences, University of Manchester, UK, 2010.

[14] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge Univ. Press, Cambridge, UK, 1994.

[15] B. IANNAZZO, *On the Newton method for the matrix pth root*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 503–523.

[16] C. KENNEY AND A. J. LAUB, *A Schur-Frechet algorithm for computing the logarithm and exponential of a matrix*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 640–663.

[17] C. K. Koç and B. Bakkaloǧlu, *Halley method for the matrix sector function*, IEEE Trans. Automat. Control, 40 (1995), pp. 944–949.
[18] B. Laszkiewicz and K. Ziętak, *Algorithms for the matrix sector function*, Electron. Trans. Numer. Anal., 31 (2008), pp. 358–383. Available at: http://etna.mcs.kent.edu/vol.31.2008/pp358-383.dir/pp358-383.html
[19] ———, *A Padé family of iterations for the matrix sector function and the matrix pth root*, Num. Lin. Algebra Appl., 16 (2009), pp. 951–970.
[20] Z.-Y. Li, B. Zhou, Y. Wang, and G.-R. Duan, *Numerical solution to linear matrix equation by finite steps iteration*, IET Control Theory Appl., 4 (2010), pp. 1245–1253.
[21] R. Mathias, *A chain rule for matrix functions and applications*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 610–620.
[22] M. I. Smith, *Numerical Computation of Matrix Functions*, PhD thesis, Department of Mathematics, University of Manchester, Manchester, UK, September 2002.
[23] ———, *A Schur algorithm for computing matrix pth roots*, *SIAM J. Matrix Anal. Appl.*, 25 (2003), pp. 971–989.
[24] J.S.H. Tsai, L.S. Shieh and R.E. Yates, *Fast and stable algorithms for computing the principal nth root of a complex matrix and the matrix sector function*, Comput. Math. Appl., 15 (1988), pp. 903–913.