COMPUTATIONAL TECHNIQUES FOR REAL LOGARITHMS OF MATRICES *.

by

Luca Dieci, **
Benedetta Morini and Alessandra Papini. ***

AMS Subject Classification: 65F30, 65F35, 65F99
Key Words: real logarithm of a matrix, conditioning, Padé approximants, series expansions, eigendecomposition approaches, error analysis, implementations.

ABSTRACT.

In this work, we consider computing the real logarithm of a real matrix. We pay attention to general conditioning issues, provide careful implementation for several techniques including scaling issues, and finally test and compare the techniques on a number of problems. All things considered, our recommendation for a general purpose method goes to the Schur decomposition approach with eigenvalue grouping, followed by square roots and diagonal Padé approximants of the diagonal blocks. Nonetheless, in some cases, a well implemented series expansion technique outperformed the other methods. We have also analyzed and implemented a novel method to estimate the Frechét derivative of the log, which proved very successful for condition estimation.

Some Notation. $M \in \mathbb{R}^{2n\times 2n}$ is called *Hamiltonian* if $M^TJ + JM = 0$, where $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$. T is called *symplectic* if $T^TJT = J$; equivalently, $T^{-1} = -JT^TJ$. $\Lambda(T) = \{\lambda_i(T),\ i = 1,\dots,n\}$ will indicate the *spectrum* of T, and $\rho(T)$ the spectral radius of T. The notation $\mu = \mathcal{O}(x)$ means that $\frac{\mu}{x} \to c \neq 0$ as $x \to 0$, and c is a constant. $A \otimes B = (a_{ij}B)_{i,j=1}^{n} \in \mathbb{R}^{n^2\times n^2}$ is the Kronecker product of A and B. We write $||A|| = ||A||_2$ for the 2-norm of a matrix A, and $||A||_F$ for its Frobenius norm. Analogously, for a linear operator $L(A) : Z \in \mathbb{R}^{n\times n} \to L(A)Z \in \mathbb{R}^{n\times n}$, we write $||L(A)|| = \max\limits_{||Z||=1} ||L(A)Z||$ for the operator norm induced by the 2-norm of matrices, and $||L(A)||_f = \max\limits_{||Z||_F=1} ||L(A)Z||_F$ for that induced by the Frobenius norm.

1. INTRODUCTION

In this work, we address the issue of finding a real logarithm of a real matrix. This problem has a precise and complete answer from the theoretical point of view, but from the computational point of view much work is

* Work supported in part under NSF Grant DMS-9306412, and MURST and CNR Grants (Italy). The work was initiated while the first author was visiting the Univ. of Florence; the kind hospitality provided by Prof.s Maria Macconi and Aldo Pasquali is gratefully acknowledged.

** School of Math.s, Georgia Tech, Atlanta, GA 30332 U.S.A. (dieci@math.gatech.edu)
*** Dep. Energetica, Univ. of Florence, via C. Lombroso 6-17, 50134 Florence, Italy (ande@vm.idg.fi.cnr.it).

still needed. A main motivation for carrying out the present work has been to provide careful implementation for, and assess performance of, the most promising techniques to compute real logarithms of matrices. We focus on real matrices, but much of what we say in this work can be adapted to the complex arithmetic case.

Undoubtedly, in comparison with other branches of scientific computation, linear algebra software is placed on very solid ground, the LAPACK and LINPACK/EISPACK libraries being the measure of excellence on which to assess quality software. The high quality Matlab system also has in these computational linear algebra components its work-horse. However, there are some linear algebra problems which have not yet found their way into proper implementation and high quality software. We believe that finding the logarithm of a matrix is one of these instances. In fact, more generally, computing functions of a matrix requires more work. (Interestingly, this is one of the very rare instances in which the Matlab implementation can give rather inaccurate answers.) The general lack of good software for functions of a matrix is all the more bothersome since computing functions of a matrix is a common engineering requirement (for the logarithm, see [LS1-2], [SS]). We think that a source of trouble is caused by looking at the computational task as a general task, rather than addressing it in a case by case way, depending on the function at hand. Not surprisingly, the exp-function, which has been singled out for its importance for a long time, enjoys more personalized and robust implementations. We hope that our work will lead towards more robust implementations for the log function.

In the remainder of this Section we briefly review some of the theoretical results we need. In Section 2 we address the sensitivity (or conditioning) issue for the log-function. The key ingredient is naturally the Frechét derivative of the log, and all across this work we try to characterize its norm. In Section 3 we give an algorithmic description of the methods we have chosen to implement, and discuss some of the error's issues for them. In Section 4 we discuss finite precision aspects of the methods, and also the general issue of ameliorating convergence and rescaling. We also present a new technique for estimating the condition number of the log problem, which has proven very reliable, and somewhat efficient. In Section 5 we give details of appropriate implementations for the methods, including cost estimates. Finally, Section 6 contains Examples, and Section 7 Conclusions.

Given a matrix $T \in \mathbb{R}^{n\times n}$, any $n \times n$ matrix X such that $e^X = T$, with e^X the matrix exponential of X, is a *logarithm* of T, and one writes $X = \log(T)$. As it is well known (e.g., see [He] and [Wo]), every invertible matrix has a logarithm (not necessarily real). Amongst the logarithms of T, in this work we are only interested in those which are *primary matrix functions* of T ([HJ], [G], [GvL], [Hi1]). As usual, these can be characterized from the Jordan decomposition of T (e.g., see [GvL, Section 1.11.1-2]).

Of course, to guarantee that $X = \log(T)$ is real (assuming T is), one needs further restriction than mere invertibility. The most complete result is the following.

THEOREM 1.1. ([C], [HJ]). *Let $T \in \mathbb{R}^{n\times n}$ be nonsingular. Then, there exists a real $X = \log(T)$ if and only if T has an even number of Jordan blocks of each size for every negative eigenvalue. If T has any* *eigenvalue on the negative real axis, then no real logarithm of T can be a primary matrix function of T.* □

We will henceforth assume that we have a real logarithm of T, and that it is a primary matrix function of T. Finally, it has to be appreciated that a logarithm can be uniquely characterized once we specify which branch of the log function (acting on complex numbers) we take. For example, there is a unique $X = \log(T)$ such that all of its eigenvalues z satisfy $-\pi < \mathrm{Im}(z) < \pi$: this is known as the *principal logarithm*, and we will restrict to this case from now on.

In many applications, there is extra structure that one is interested in exploiting. For example, different techniques can be devised for the cases when $\Lambda(I-T)$ is inside the unit circle, and/or when $\Re\mathrm{e}(\Lambda(T)) > 0$. Inter alia, the latter case arises for symmetric positive definite T, a situation in which T has a unique symmetric logarithm ([HJ]). Also (see [Si] and [YS]), if T is symplectic (orthogonal), then there exists a real Hamiltonian (skew-symmetric) logarithm. Of course, in these cases we would want approximation techniques which guarantee that we can recover the desired structure. This question was recently addressed in [D]; in the present work, we will use and extend some of the results in [D].

Not much work has been done on computing logarithms of matrices in comparison to its inverse function, computing matrix exponentials. The references [KL1], [KL2], [LS1], [LS2], [V], are a representative sample of works on computation of logarithms of matrices. With the exception of [KL1-2], finite precision issues are not considered in these works. To our knowledge, our work is the first attempt to consider finite precision behavior of several techniques, and to implement and compare them.

2. SENSITIVITY OF THE PROBLEM.

Naturally, before computing the logarithm of a matrix, it is appropriate trying to understand the intrinsic sensitivity of this function. The works of Kenney and Laub ([KL2]) and Mathias ([M]) are important sources of information on the general topic of conditioning of matrix functions. Our presentation is explicitly geared toward the log function, and it is partly different than these works.

Given a matrix function $F(T)$, where $F : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, the basic issue is to understand how the value of the function changes as the argument T does. This leads to relying on the Frechét derivative as a measure of sensitivity. From here on, unless otherwise stated, we use the 2-norm; with minimal changes (if at all), all results hold true for different norms.

DEFINITION 2.1. Given a matrix function $G : T \in \mathbb{R}^{n\times n} \to G(T) \in \mathbb{R}^{n\times n}$, a linear mapping $G'(T) : Z \in \mathbb{R}^{n\times n} \to G'(T)Z \in \mathbb{R}^{n\times n}$ is the Frechét derivative of G at T if for any $Z \in \mathbb{R}^{n\times n}$ we have

$$\lim_{\lambda\to 0} \left\| \frac{G(T+\lambda Z) - G(T)}{\lambda} - G'(T)Z \right\| = 0\,. \tag{2.1}$$

The norm of the Frechét derivative is given by $\|G'(T)\| = \max\limits_{\|Z\|=1} \|G'(T)Z\|$. If G has a Frechét derivative, we say that G is *differentiable*. □

With this definition, one has a general way to assess sensitivity for matrix functions. This is a general procedure, and can be found (essentially identical) in the works [KL2], [Hi1], [MvL], and references there, in special cases.

Let $X \neq 0: \; G(T) = X$, and consider the perturbed input $T + \Delta T$ with corresponding perturbed output $X + \Delta X: \; X + \Delta X = G(T + \Delta T)$. For $G(T) = \log(T)$, from the relation $\Delta X = G(T + \Delta T) - G(T)$, upon using (2.1) we can obtain

$$\frac{||\Delta X||}{||X||} \leq ||G'(T)|| \, \frac{||T||}{||X||} \, \frac{||\Delta T||}{||T||} + \mathcal{O}(||\Delta T||^2) \, . \tag{2.2}$$

The quantity

$$\mathrm{cond}(G(T)) := ||G'(T)|| \, \frac{||T||}{||X||} \tag{2.3}$$

acts as a relative error magnification factor, and it is therefore natural to call it the *condition number* of the matrix function G at T. (Notice that, strictly speaking, we still have to justify the $\mathcal{O}(||\Delta T||^2)$ term in (2.2); this we will do in Section 3.)

REMARKS 2.2.

(i) It is clear that $\mathrm{cond}(G(T))$ depends both on G and T, and on X. A measure of conditioning which neglects any of these components may be faulty.

(ii) Of course, different functions G might allow for more specialized ways to characterize $\mathrm{cond}(G(T))$, as it is clearly evidenced in the work on the matrix exponential (see [vL], [MvL]). One of our tasks in the remainder of this work is to better characterize the Frechét derivative of the log function, hence $\mathrm{cond}(\log(T))$.

(iii) If $X \approx 0$, it is of course more sensible to assess absolute errors, and thus to replace (2.2) with

$$||\Delta X|| \leq ||G'(T)|| \, ||\Delta T|| \, + \, \mathcal{O}(||\Delta T||^2) \, .$$

We begin with the following elementary result, already in [KL2, Lemma B2], which is just the Chain Rule.

LEMMA 2.3. *Let F and G be matrix functions such that $G(T)$ is in the domain of F. Consider the composite function $H(T) := F(G(T))$. Let $G'(T)$ and $F'(G(T))$ be the Frechét derivatives of the functions G and F, at T and $G(T)$ respectively. Then, the Frechét derivative of the composite function is characterized as the linear mapping*

$$H'(T): \; Z \in \mathbb{R}^{n \times n} \rightarrow F'(G(T))G'(T)Z \in \mathbb{R}^{n \times n} \, . \qquad \square$$

As a consequence of Lemma 2.3, we have (essentially, [KL2, Lemma B1])

COROLLARY 2.4. *Let F and G be inverse functions of each other, that is $F(G(T)) = T$, $\forall T$ in the domain of G, and $G(T)$ in the domain of F, and let F and G be differentiable, as in Lemma 2.3. Also, let $F'(G(T))$ be invertible. Then we have*

$$G'(T)Z = (F'(G(T)))^{-1} Z \, , \tag{2.4}$$

and therefore also

$$||G'(T)|| = ||(F'(G(T)))^{-1}||\,. \tag{2.5}$$

Proof. Apply the chain rule of Lemma 2.3 to the relation $F(G(T)) = T$. □

LEMMA 2.5. *Let $G(T) = \log(T)$ and $F(Y) = e^Y$. Then we have*

$$||G'(T)|| \geq ||T^{-1}||\,, \tag{2.6}$$

and therefore

$$\text{cond}(G(T)) \geq \frac{\text{cond}(T)}{||\log(T)||}\,, \quad \text{where} \quad \text{cond}(T) = ||T||\,||T^{-1}||\,.$$

Proof. From [vL, formula (1.3) and p. 972] we have

$$F'(Y)Z = \int_0^1 e^{Y(1-s)} Z e^{Ys} ds\,, \tag{2.7}$$

and therefore with $Y = \log(T)$ from Corollary 2.4 we have (take $Z = I$ below)

$$||G'(T)|| = \max_{||Z||=1} ||(\int_0^1 e^{Y(1-s)} Z e^{Ys} ds)^{-1}|| \geq$$
$$||(\int_0^1 e^Y ds)^{-1}|| = ||e^{-Y}|| = ||T^{-1}||\,,$$

where we have used the identity $-\log(T) = \log(T^{-1})$ (see [HJ]). □

REMARKS 2.6.

(i) From (2.7) we can get implicit representations for $G'(T)$ in the case $G(T) = \log(T)$, and $F(Y) = e^Y$; for example,

$$Z = \int_0^1 T^{1-s} G'(T) Z T^s ds\,.$$

(ii) In Section 3, we prove that, for positive definite matrices, in (2.6) we have equality.

In [KL2], Kenney and Laub consider matrix functions admitting a series representation such as

$$F(X) := \sum_{n=0}^{\infty} a_n X^n\,, \tag{2.8}$$

with associated scalar series absolutely convergent. In this case, they can represent the Frechét derivative as the infinite series

$$F'(X) :\ Z \to \sum_{n=1}^{\infty} a_n \sum_{k=0}^{n-1} X^k Z X^{n-k-1}\,. \tag{2.9}$$

Next, they unroll the Frechét derivative by column ordering, call $D(X) \in \mathbb{R}^{n^2 \times n^2}$ the resulting matrix acting on the unrolled Z:

$$D(X) = \sum_{n=1}^{\infty} a_n \sum_{k=0}^{n-1} (X^T)^{n-1-k} \otimes X^k\,, \tag{2.10}$$

and then focus on the 2-norm of $D(X)$: $||D(X)||_2$. To proceed with their analysis, one must realize that $||D(X)||_2$ is the same as $||F'(X)||_f$ (see the Notation at the beginning of this work). They have some general results giving a lower bound for this norm, and then show that this lower bound is achieved when X is normal. We highly recommend careful reading of their work for details. Notice, however, that the assumption on being able to represent $F(X)$ as the series (2.8) rules out a direct application of their theory to the log function. To deal with the Frechét derivative of the function $G(T) = \log(T)$, they rely on (2.5), and are thus able to estimate $||G'(T)||_f$ via estimates on the norm of the inverse of the Frechét derivative of the exponential function. Their approach can be profitably used to get some more information on the norm of the Frechét derivative of the log. Although what follows is not explicitly given in [KL2], it can be deduced from their approach.

Consider the case of $G(T)$ and $F(Y)$ inverse functions of each other, so that $F(G(T)) = T$. Moreover, let $F(Y)$ be a matrix function for which (2.8)-(2.10) hold. For example, this is true for $G(T) = \log(T)$, and $F(Y) = e^Y$. Let $F'(G(T))$ be invertible, and let $D(G(T))$ be the unrolled Frechét derivative of $F(Y)$ at $G(T)$. Then, we have

$$||G'(T)||_f \;=\; ||(F'(G(T)))^{-1}||_f \;=\; ||(D(G(T)))^{-1}||_2\,.$$

Let λ_i be the eigenvalues of $D(G(T))$, and let $|\lambda_1| \geq \ldots \geq |\lambda_{n^2}|$. One always has the inequality

$$\frac{1}{|\lambda_1|} \leq ||(D(G(T)))^{-1}||_2\,,$$

and if we assume that $D(G(T))$ be diagonalizable by the matrix S : $S^{-1}D(G(T))S = \mathrm{diag}(\lambda_i)$, then also the following inequality is well known

$$||(D(G(T)))^{-1}||_2 \leq \mathrm{cond}_2(S)\frac{1}{|\lambda_1|}.$$

Now, let T be diagonalizable by V, $T = V\Lambda V^{-1}$, and so also $G(T) = VG(\Lambda)V^{-1}$. Hence for $D(G(T))$ one has (use [HJ, Problem 3 p.249])

$$D(G(T)) = (V^{-T}\otimes V)(\sum_{n=1}^{\infty} a_n \sum_{k=0}^{n-1}(G(\Lambda))^{n-1-k}\otimes G(\Lambda)^k)(V^{-T}\otimes V)^{-1}.$$

With $S = V^{-T}\otimes V$, putting it all together, we get that for diagonalizable matrices T the following holds:

$$\frac{1}{|\lambda_1|} \leq ||(D(G(T)))^{-1}||_2 \;=\; ||G'(T)||_f \;\leq \mathrm{cond}_2(S)\frac{1}{|\lambda_1|}. \tag{2.11}$$

To complete this discussion, we now recall that normal matrices can be brought to diagonal form (almost diagonal, i.e., diagonal with possibly 2 × 2 blocks along the diagonal to allow for complex conjugate pairs of eigenvalues, if we insist on real arithmetic) with a unitary (orthogonal) matrix. So, let V be unitary above. Moreover, if T is normal then so is $G(T)$ ([HJ, Problem 2 p.439]). Finally, if V is unitary, so is V^{-T} and so also $S = V^{-T}\otimes V$ is unitary [HJ, p.249]. So, for normal matrices, one has the precise characterization

$$||G'(T)||_f \;=\; \frac{1}{\min\limits_{1\leq i\leq n^2}|\lambda_i(D(G(T)))|}. \tag{2.12}$$

In fact, to have $\mathrm{cond}_2(S) = 1$ in (2.11) we must have all singular values of S equal 1, and thus (2.12) holds, for the class of diagonalizable matrices, only if T is normal.

REMARKS 2.7.

(i) In particular, all of the above holds for the function $G(T) = \log(T)$. But the above reasoning also holds for many other matrix functions $G(T)$ not satisfying (2.8), but for which their inverse function satisfies (2.8); amongst others, $G(T) = T^{1/p}$, $p = 2, \ldots$.

(ii) Characterization of the eigenvalues of $D(G(T))$ in terms of those of $G(T)$ is done in [KL2, Lemma 2.1].

To obtain a relation between $\|G'(T)\|_f$ and the operator norm $\|G'(T)\|$, reason as follows. Let $G'(T)Z = B(Z) \in \mathbb{R}^{n\times n}$, and let $\sigma_i(Z)$, $\sigma_i(B(Z))$, be the (ordered) singular values of Z, $B(Z)$, respectively. Then

$$\|G'(T)\| = \max_{\sigma_1(Z)=1} \sigma_1(B(Z))\,, \quad \|G'(T)\|_f = \max_{\sigma_1^2(Z)+\ldots+\sigma_n^2(Z)=1} (\sigma_1^2(B(Z)) + \ldots + \sigma_n^2(B(Z)))^{1/2}\,,$$

and the following inequalities are then simple to obtain:

$$\|G'(T)\| \leq \|G'(T))\|_f \leq \sqrt{n}\ \|G'(T)\|\,. \tag{2.13}$$

Notice that (2.13) are the usual inequalities between the Frobenius and spectral norms of matrices.

Of course, in order for a measure of conditioning of the $G(T)$ problem to be an effective computational tool, one should be able to estimate $\|G'(T)\|$ (perhaps in some norm other than the 2-norm) without drastically increasing the expense needed for the computation of $G(T)$. This seems to be a tall task. Nonetheless, some interesting ideas are in [KL2] and [M], and some other possibilities are discussed in the next two Sections.

3. SOME METHODS. MORE ON CONDITIONING.

Here we present some methods: (i) two series expansion techniques ([GvL], [LS1-2], [D]), (ii) Padé approximation methods ([KL1-2], [D]), (iii) the Schur decomposition approach ([GvL], [Matlab]), and a (iv) ODE reformulation approach.

Series Expansions. Under appropriate restrictions on the spectrum of T, the principal logarithm of T can be expressed as a series. In particular, two such series have frequently appeared in the literature. Computational procedures arise upon truncating these series.

Series 1. Let $A = I - T$, and assume $\rho(A) < 1$. Then,

$$G(T) := \log(T) = \log(I - A) = -\sum_{k=1}^{\infty} \frac{A^k}{k}\,. \tag{3.1}$$

Subject to obvious restrictions on spectral radii, from (3.1) we get

$$\log(T + Y) = \log(T) + (\log(T))'Y + E(Y)\,,$$

and $\|E(Y)\| \leq \mathcal{O}(\|Y\|^2)$. From this, we obtain an expression for the Frechét derivative:

$$G'(T) : Y \to \sum_{n=1}^{\infty} \frac{1}{n} \sum_{k=0}^{n-1} A^k Y A^{n-1-k}\,, \ \ A = I - T\,, \tag{3.2}$$

and if $||A|| < 1$:

$$||G'(T)|| \leq \sum_{n=1}^{\infty} \frac{1}{n} \sum_{k=0}^{n-1} ||A||^{n-1} = \frac{1}{1-||A||} = \frac{1}{1-||I-T||}.$$

From the above, we get that for positive definite matrices $||G'(T)|| \leq \frac{1}{\min\limits_{\lambda \in \Lambda(T)} |\lambda|}$, that is $||G'(T)|| \leq ||T^{-1}||$, which justifies Remark 2.6(ii), for positive definite matrices for which (3.1) holds.

Series 2. This is obtained from the series expansion (3.1) for $\log(I+X)-\log(I-X) = \log((I+X)(I-X)^{-1})$, via the conformal transformation $T = (X-I)(X+I)^{-1}$, thereby obtaining

$$\log(T) = 2\sum_{k=0}^{\infty} \frac{1}{2k+1}[(T-I)(T+I)^{-1}]^{2k+1}. \tag{3.3}$$

Notice that the restriction $\rho(A) < 1$ needed for (3.1) now has become $\Re\mathrm{e}(\Lambda(T)) > 0$. Reasoning as before, if also $\Re\mathrm{e}(\Lambda(T+Y)) > 0$, we obtain another expression for the Frechét derivative of $\log(T)$:

$$(\log(T))' : Y \rightarrow 2\sum_{k=0}^{\infty} \frac{1}{2k+1} \sum_{j=0}^{2k} B^j(2CYC)B^{2k-j}\,,\; C := (X+I)^{-1}\,,\; B := (X-I)C\,. \tag{3.4}$$

Padé Approximants. Under the assumption $\rho(I-T) < 1$, these consist in approximating the function $\log(I-A)$, $A = I-T$ with the rational matrix polynomial $R_{n,m}(A) = P_n(A)(Q_m(A))^{-1}$, where $P_n(A)$ and $Q_m(A)$ are polynomials in A of degree n and m respectively, in such a way that $R_{n,m}(A)$ agrees with $n+m$ terms in the series expansion (3.1) of $\log(I-A)$. This is a universal and powerful tool [BG-M], well examined in the context of $\log(T)$ in the works [KL1-2]. It is easy, with the help of tools such as *Maple*, to obtain the coefficients of the matrix polynomials $P_n(A)$ and $Q_m(A)$. Based on the error estimates in [KL1], we have only considered diagonal Padé approximants.

To assess the conditioning of the Padé approximants, we can reason as follows. For given n, m, let $R(A) = R_{n,m}(A) = P(A)(Q(A))^{-1} = \sum_{k=0}^{n} a_k A^k (\sum_{k=0}^{m} b_k A^k)^{-1}$. Suppose that rather than T we have $T+Y$, that is $A-Y$, instead of A, and $||Y|| \ll 1$. Then it is easy to obtain

$$R(A-Y) - R(A) = -(E(Y) - R(A)F(Y))(Q(A))^{-1} + H(Y)\,, \tag{3.5}$$

where $||H(Y)|| \leq \mathcal{O}(||Y||^2)$, and $E(Y) = \sum_{k=1}^{n} a_k \sum_{j=0}^{k-1} A^j Y A^{k-1-j}$, and $F(Y) = \sum_{k=1}^{m} b_k \sum_{j=0}^{k-1} A^j Y A^{k-1-j}$, are the first order perturbation terms for $P(A)$ and $Q(A)$. From (3.5) we obtain

$$||R(A-Y) - R(A)|| \leq (||E(Y)|| + ||F(Y)||\; ||R(A)||)||(Q(A))^{-1}|| + \mathcal{O}(||Y||^2)\,,$$

or in a relative error sense (if $||R(A)|| \neq 0$)

$$\frac{||R(A-Y) - R(A)||}{||R(A)||} \leq (\frac{||E(Y)||}{||R(A)||} + ||F(Y)||)\frac{\mathrm{cond}(Q(A))}{||Q(A)||} + \mathcal{O}(||Y||^2)\,. \tag{3.6}$$

Therefore, we see that for the conditioning of the Padé problem the most important factor is the conditioning of the denominator problem. In [KL1], this issue is investigated in the case $||A|| < 1$, in particular see [KL1, Lemma 3].

To understand better the term $(E(Y)-R(A)F(Y))(Q(A))^{-1}$ in (3.5), we can use first order perturbation arguments for the matrix function $R(A)$, to obtain

$$(E(Y)-R(A)F(Y))(Q(A))^{-1} \;=\; R'(A)Y\,.$$

We also have the following general result

LEMMA 3.1. *Let* $F(A)=\sum_{k=0}^{\infty}c_kA^k$, *and let* $R(A)$ *be a Padé approximant agreeing with the series of* $F(A)$ *up to the power* A^{n+m} *included. Then* $R'(A)Y$ *agrees with* $F'(A)Y$ *up to the term* $\sum_{j=1}^{n+m}c_j\sum_{l=0}^{j-1}A^lYA^{j-1-l}$.

Proof. Write $F(A)=R(A)+M(A)$, so that $M(A)$ has a power series with terms beginning with A^{k+m+1}. Now, since $F'(A)Y=R'(A)Y+M'(A)Y$, the result follows. □

REMARK 3.2. For the case of the log, since $F(A-Y)=\log(T+Y)$, Lemma 3.1 tells us that the conditioning of the Padé problem (hence also of the truncated series (3.1)), is close to the conditioning of $\log(T)$ (essentially the same if $\|A\|<1$, for high enough $n+m$). No extra pathological behavior is introduced.

Schur Decomposition Approach. When properly implemented, this is an extremely effective and reliable technique. The basic principles of the technique are general (see [GvL]), but our adaptation to $\log(T)$ seems to be new. Let Q be orthogonal such that $QTQ^T := R$ be in real Schur form (upper quasi-triangular). Moreover, let R be partitioned as $R := \begin{pmatrix} R_{11} & \cdots & R_{1m} \\ & \ddots & \vdots \\ 0 & & R_{mm} \end{pmatrix}$, where we assume that $\Lambda(R_{ii})\cap\Lambda(R_{jj})=\emptyset$, $i\neq j$ (this can be done in standard ways). To obtain $L:=\log(R)$, one realizes that L has the same block-structure as R, and (see [GvL, Section 11.1]) can get L from the relation $LR=RL$. The following recursion can be used to get L ([P]):

```
For $i = 1, 2, \ldots, m$
                    $L_{ii} = \log(R_{ii})$                                        (3.7)
Endfor $i$
For $p = 1, 2, \ldots, m-1$
    For $i = 1, 2, \ldots, m-p$, with $j = i+p$, solve for the $L_{ij}$:
        $L_{ij}R_{jj} - R_{ii}L_{ij} = R_{ij}L_{jj} - L_{ii}R_{ij} + \sum_{k=i+1}^{j-1}(R_{ik}L_{kj} - L_{ik}R_{kj})$        (3.8)
    Endfor $i$
Endfor $p$.
```

In general, the R_{ii} can be the 1×1 or 2×2 blocks of eigenvalues, or also much larger quasi-triangular blocks. If T is normal, then Q brings T to block diagonal form with either $(1,1)$ or $(2,2)$ diagonal blocks, and only (3.7) is required. Otherwise, to solve the Sylvester equation (3.8) is standard (see [GvL, [.387], and notice that (3.8) is uniquely solvable, since $\Lambda(R_{ii}) \cap \Lambda(R_{jj}) = \emptyset$). To obtain L_{ii} from (3.7) is just a function call if R_{ii} is (1×1), and also if $R_{ii} \in \mathbb{R}^{2\times 2}$ with complex conjugate eigenvalues a direct evaluation is possible (see Lemma 3.3 below), while in all other cases we need some approximation method, e.g. by truncating the previous series or using Padé approximants (if applicable).

LEMMA 3.3. *Let* $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *with complex conjugate eigenvalues* $\theta \pm i\mu$ $(\mu \neq 0)$. *Then*

$$\log(A) = \alpha I - \beta \frac{2\mu}{4bc + (a-d)^2} \begin{pmatrix} a-d & 2b \\ 2c & -a+d \end{pmatrix},$$

where $\alpha = \log(\rho)$, $\rho^2 = \theta^2 + \mu^2$, *and* $\beta = \cos^{-1}(\frac{\theta}{\rho})$, $0 \leq \beta < \pi$.

Proof. The proof is just a simple calculation. □

COROLLARY 3.4. *Let* $B \in \mathbb{R}^{2\times 2}$ *be normal, that is* $B = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$. *With notation of Lemma 3.3, we have* $\log(B) = \begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}$. *Moreover, if* B *is orthogonal, then* $\alpha = 0$. □

REMARKS 3.5.

(i) Corollary 3.4, coupled with prior real Schur reduction, guarantees that the computation of a real logarithm of a normal matrix T can be done in such a way that the end result is a real, normal, matrix. In particular, this fact makes such an algorithm interesting for computing the skew-symmetric logarithm of an orthogonal matrix, an approach not considered in [D].

(ii) Of course, $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ can be identified with the complex number $z = a + ib$, which makes Corollary 3.4 obvious ($\log z = \log|z| + i \arg z$). This observation renders more transparent also the first part of Lemma 3.8 below.

ODE Approach. This will be a very useful tool to better characterize both $\log(T)$ and its Frechét derivative. The starting point is to embed the problem into a continuous model, similar in spirit to a "homotopy" path.

Let the time dependent matrix $X(t)$ be implicitly defined as

$$X(t): \; e^{X(t)} = (T-I)t + I, \quad 0 \leq t \leq 1. \tag{3.9}$$

Notice that $X(1)$ defines $\log(T)$, and that $X(t)$ is well defined, and real, $\forall t \in [0,1]$, because for $(T-I)t+I$ Theorem 1.1 holds, since it holds for T. Since $Te^{X(t)} = e^{X(t)}T$, then we also have that $X(t)$ satisfies the ODE

$$\begin{aligned} &\dot{X} = (T-I)e^{-X(t)}, \; 0 \leq t \leq 1, \\ &X(0) = 0. \end{aligned} \tag{3.10}$$

By construction, (3.10) defines the principal log of $(T-I)t+I$. Upon using (3.9), we have the explicit solution of (3.10)

$$X(t) = \int_0^t (T-I)((T-I)s+I)^{-1} ds, \; 0 \leq t \leq 1, \tag{3.11}$$

and therefore we find this expression for $\log(T)$

$$\log(T) = X(1) = \int_0^1 (T-I)((T-I)t+I)^{-1}dt\,. \tag{3.12}$$

REMARKS 3.6.

(i) Formula (3.12) is also derived in the works by Helton and Wouk ([He], [Wo]). Their interest was in showing that every invertible matrix had a logarithm.

(ii) Computational procedures for $\log(T)$ can be obtained by using integration formulas for the ODE (3.10), or quadrature rules on (3.12). We have experimented with explicit Runge-Kutta integrators for the ODE (3.10), and several quadrature rules for (3.12). We found that quadrature rules were consistently less costly. Notice that the midpoint rule on (3.12) gives the $(1,1)$ Padé approximant; see also Theorem 4.3.

Formula (3.12) can also be used to obtain a new formula for the Frechét derivative of $G(T) = \log(T)$. In fact, upon considering (3.12) for $\log(T+Z)$, using first order perturbation arguments, and some algebra, yields the following:

$$G'(T)Z = \int_0^1 ((T-I)t+I)^{-1}\, Z\, ((T-I)t+I)^{-1}\, dt\ . \tag{3.13}$$

We also notice that using (3.12) for $\log(T+\Delta T)$, and expanding the inverse there in powers of ΔT, justifies the $\mathcal{O}(\|\Delta T\|^2)$ term in (2.2).

Now, from (3.13) with $Z = I$, since

$$\int_0^1 ((T-I)t+I)^{-2}dt = -(T-I)^{-1}[((T-I)t+I)^{-1}]_0^1 = T^{-1}\,,$$

we obtain $\|T^{-1}\| \le \|G'(T)\|$, and so:

$$\|T^{-1}\| \le \|G'(T)\| \le \int_0^1 \|((T-I)t+I)^{-1}\|^2 dt\,. \tag{3.14}$$

Moreover, (3.13) and (3.14) can be profitably exploited to gain further insight into $\|G'(T)\|$.

LEMMA 3.7. *If T is positive definite, then*

$$\|G'(T)\| = \|T^{-1}\|\,.$$

Proof. Diagonalize T with orthogonal Q on the right-hand side of (3.14), and perform the integration. □

LEMMA 3.8. *If $T \in \mathbb{R}^{2\times 2}$ is normal with complex conjugate eigenvalues $a \pm ib$, then*

$$\|G'(T)\| = \frac{1}{\rho}\frac{\theta}{\sin(\theta)}$$

where $-\pi < \theta < \pi$ is the argument of the eigenvalues of T, and ρ their modulus. If T is normal of dimension n, then

$$\|G'(T)\| \ge \max_k \frac{1}{\rho_k}\frac{\theta_k}{\sin(\theta_k)} \ge \|T^{-1}\|\,, \tag{3.15}$$

where θ_k's are the arguments of the eigenvalues of T (if $\theta_k = 0$, replace $\frac{\theta_k}{\sin(\theta_k)}$ by 1), and ρ_k's their modulus.

Proof. In the (2×2) case T is of the form $T=\begin{pmatrix} a & b \\ -b & a\end{pmatrix}$ with complex conjugate eigenvalues $a\pm ib$ (and let $b\neq 0$, otherwise T is positive definite). Then $\lambda(t)=((a\pm ib-1)t+1)^{-1}$ are the eigenvalues of $((T-I)t+I)^{-1}$. Now, if we take $Z=\begin{pmatrix} 0 & 1 \\ 1 & 0\end{pmatrix}$ in (3.13), we get that

$$||G'(T)||\geq \int_0^1 |\lambda(t)|^2 dt\,.$$

For $a\neq 0$, with some algebra, this integral equals $\frac{1}{b}\tan^{-1}\frac{b}{a}=\frac{1}{\rho}\frac{\theta}{\sin(\theta)}$, where θ belongs to $(0,\pi/2)$, $(-\pi,-\pi/2)$, $(-\pi/2,0)$, $(\pi/2,\pi)$ depending on whether $b/a>0$, and $b>0$ or $b<0$, or $b/a<0$, and $b<0$ or $b>0$. Now, one always has $||G'(T)||\leq \int_0^1 ||((T-I)t+I)^{-1}||^2 dt$, and, because of normality, the norm of $((T-I)t+I)^{-1}$ equals the square root of $|\lambda(t)|$. Therefore, as before, we get the reverse inequality

$$||G'(T)||\leq \frac{1}{\rho}\frac{\theta}{\sin(\theta)}$$

subject to same restriction on the argument. Therefore, the result for T normal and (2×2) follows. If $a=0$, one gets simply $||G'(T)||=\frac{1}{\rho}\frac{\pi}{2}$.

For general $T\in \mathbb{R}^{n\times n}$, normal, let Q bring T to the almost diagonal form QTQ^T. Next, consider all matrices Z given by all zeros, except that on the diagonal they have just one 1 or one $\begin{pmatrix} 0 & 1 \\ 1 & 0\end{pmatrix}$ block according to the eigenvalue structure of QTQ^T, and then (3.15) follows from the previous (2×2) case. □

REMARK 3.9. The bound (3.15) indicates that there are two key factors determining the condition of the log problem: one is, as usual, nearness to singularity, as evidenced by the $\frac{1}{\rho}$ factor, the other is nearness to the negative real axis, as evidenced by the $\sin(\theta)$ factor in the denominator. This second fact detects ill conditioning based on the restrictions imposed by the choice of real arithmetic.

Finally, (3.13) can also be used to estimate $||G'(T)||_f$ **directly**. We reason similarly to [KL2], but stress that (3.13) is a representation for $G'(T)Z$ which does not need a power series representation, nor to go through the inverse function (the exponential). We have

THEOREM 3.10. *Let $A(t):=((T-I)t+I)^{-1}$, and let $D(T):=\int_0^1 (A^T(t)\otimes A(t))dt$. Then, we have*

$$||G'(T)||_f=||D(T)||_2\,. \tag{3.16}$$

Proof. Let $\mathrm{vec}(Z)$ be the vector obtained by writing the columns of Z one after another, and so $||G'(T)||_f=\max\limits_{||\mathrm{vec}(Z)||_2=1}||\mathrm{vec}(G'(T)Z)||_2$. But by (3.13)

$$\begin{aligned}\mathrm{vec}(G'(T)Z)&=\int_0^1 \mathrm{vec}(A(t)ZA(t))dt=\int_0^1 (A^T(t)\otimes A(t))\mathrm{vec}(Z)dt\\ &=\int_0^1 (A^T(t)\otimes A(t))dt\,\mathrm{vec}Z\,,\end{aligned}$$

and the result follows. □

REMARK 3.11. The above result can be used, in the same spirit as in [KL2, p.192], as a starting point for a procedure to estimate $||G'(T)||_f$. In fact, since $||D(T)||_2 = (\lambda_{\max}(D^T(T)D(T)))^{1/2}$, a power method approach to get the dominant eigenvalue is suitable. By noticing that $D^T(T) = D(T^T)$, with $A(t)$ given in Theorem 3.10, a cycle of this power method can be compactly written as: " Given $Z_0 : ||Z_0||_F = 1$, let $Z_1 = \int_0^1 A(t)Z_0A(t)dt$, and then $Z_2 = \int_0^1 A^T(t)Z_1A^T(t)dt$, so that $(||Z_2||_F)^{1/2}$ is an estimate for $||G'(T)||_f$. If more accuracy is required, repeat this cycle with $Z_0 := \frac{Z_2}{||Z_2||_F}$". In practice, of course, the integral has to be replaced by a quadrature rule, and we experimented with composite trapezoidal and Simpson rules, and Gauss-Legendre rules. For the initial Z_0, we used what we would have got after one cycle of the procedure had we started with $\frac{1}{\sqrt{n}}I$; that is, one first would get $Z_1 = \frac{1}{\sqrt{n}}T^{-1}$, and then a quadrature rule for the next integral would give some Z_2 (e.g., $Z_2 = \frac{1}{6\sqrt{n}}(T^{-T}T^{-1}T^{-T} + 16(T-I)^{-T}T^{-1}(T-I)^{-T} + T^{-1})$, if we use Simpson rule). Thus, we used $Z_0 := Z_2/||Z_2||$. This choice of Z_0 gave consistently better results than starting with a random matrix. We have experimented with this way to estimate $||G'(T)||_f$, by using at most 10 equally spaced subdivisions for the quadrature rules. This approach was very inexpensive, of course, but not entirely reliable. Often, it overestimated the true value (interestingly, almost never underestimated it); so, it revealed itself as a good indicator of ill-conditioning, but did not a give a good measure of achieved accuracy. On the other hand, we cannot expect that for arbitrary T, hence $A(t)$ in Theorem 3.10, a quadrature rule with few points will be accurate; naturally, when we raised the number of quadrature points, the estimate got better, but this became too expensive. For these reasons, we turned our attention to a different technique, explained in the next Section.

4. FINITE PRECISION, RESCALING, DISCRETIZATIONS.

Finite Precision. For the *two series* (3.1) and (3.3), the asymptotic rates of convergence are determined by $\rho(A)$, $A := I - T$, and $\rho(B)$, $B := (I-T)(I+T)^{-1}$, respectively. However, the finite precision behavior of a truncated expansion is influenced by progressively taking powers: A^k for (3.1), and B^{2k+1} for (3.3). Moreover, for (3.3) there is also the inverse of $I + T$ to contend with. A worst case analysis tells that roundoff might be magnified by powers of $||A||$ or $||B||$, respectively. If $||A|| < 1$, then (3.1) leads to a safe computation. Also, when $||A|| < 1$, for (3.3) we would have $||B|| < ||(I+T)^{-1}||$, and this can be easily bounded since $I + T = 2(I - \frac{I-T}{2})$, and so $(I+T)^{-1} = \frac{1}{2}(I - A/2)^{-1}$. Then,

$$||(I+T)^{-1}|| \leq \frac{1}{2}\,\frac{1}{1-||A||/2} < 1\,.$$

So, under the assumption $||A|| = ||I - T|| < 1$, the two series (3.1) and (3.3) lead to a safe computation. Also for the *Padé approximants*, the assumption on $||A|| < 1$ seems essential in order to make progress. Under this assumption, the finite precision behavior of Padé approximants is well analyzed in [KL1]. In particular, see Lemma 3 of [KL1].

Because the transformation to Schur form is a stable process, the finite precision behavior of the *Schur method* is chiefly determined by two factors: finding $L_{ii} = \log(R_{ii})$ in (3.7) in case in which Lemma 3.3 does not apply, and solving (3.8). The former factor is the usual one. The second factor is carefully analyzed in [Hi2]. One has to solve the following Sylvester equation for Z:

$$R_{ii}Z - ZR_{jj} = C \ ,$$

where the spectra of R_{ii} and R_{jj} are disjoint. Ideally, we would like to select the block partitioning of the matrix R in such a way that all Sylvester equations to be solved are well conditioned, so that no eventual loss of precision in the computation is introduced. But, of course, to assess the conditioning of a Sylvester equation requires the equation and its solution, whereas –for efficiency sake– we would like to have a criterion to determine the partitioning of R before hand. We reasoned as follows. If we call ϕ the Sylvester equation operator, $\phi : Z \to R_{ii}Z - ZR_{jj}$, then $\|\phi^{-1}\|$ is an upper bound for a relative error magnification factor (see [Hi2]). It is also known that $\|\phi^{-1}\| \geq \frac{1}{\min|\lambda-\mu|}$, where $\lambda \in \Lambda(R_{ii})$, $\mu \in \Lambda(R_{jj})$ (see [GvL, p.389]), and this lower bound we can easily control, by making sure that $\Lambda(R_{ii})$ and $\Lambda(R_{jj})$ are sufficiently separated. Of course, this does not suffice to make the Sylvester equation well conditioned. Still, after extensive computational experiments, we decided to cluster the eigenvalues so that $|\Lambda(R_{ii}) - \Lambda(R_{jj})| \geq 1/10$, and we have **never** encountered a problem where a system (3.8) was ill conditioned, but the log was well conditioned. For this reason, we think the method should be regarded as stable.

For the *ODE approach*, a quadrature rule must replace the integral in (3.12). That is,

$$\log(T) = \int_0^1 (T-I)((T-I)t+I)^{-1}dt := \int_0^1 F(t)dt, \tag{4.1}$$

must be approximated by a rule of the type

$$Q := \sum_{k=1}^{N} c_k F(t_k). \tag{4.2}$$

For example, consider a composite Simpson rule (identical reasoning applies to different quadratures) to approximate (4.1). Let $F(t) = (T-I)((T-I)t+I)^{-1} =: (T-I)A(t)$, with $A(t) = ((T-I)t+I)^{-1}$. The composite Simpson rule with equal spacing $h = 1/N$ (N even) is

$$CS := \frac{h}{3}(F(0) + 4(h) + 2F(2h) + 4F(3h) + \cdots + 2F((N-2)h) + 4F((N-1)h) + F(1)).$$

It is easy to bound the error as:

$$\|\log(T) - CS\| \leq \frac{nh^4}{180} \max_{0\leq t\leq 1} \|F^{iv}(t)\|. \tag{4.3}$$

We can verify that $F^{(k)}(t) = (-1)^k k!((T-I)A(t))^{k+1}$, from which

$$F^{iv}(t) = 24[(T-I)A(t)]^5 \,, \tag{4.4}$$

which can be used in (4.3) to get error estimates. In case $||I-T|| = \omega < 1$, the error bound can be sharpened. In fact, we easily get $||A(t)|| \le \frac{1}{1-\omega t}$, so that $||F^{iv}(t)|| \le 24(\frac{\omega}{1-\omega})^5$, and therefore

$$||\log(T) - CS|| \le \frac{4n}{45}\, h^4 \left(\frac{\omega}{1-\omega}\right)^5 . \tag{4.5}$$

REMARK 4.1. A direct computational procedure based on a composite quadrature rule discretization of (3.12) can eventually be very accurate, but in general it will be expensive, unless T is not far from the identity. Still, for low accuracy, a formula like (4.2) can be profitably used. For example, a modification of the above proved very useful to estimate the norm of the Frechét derivative of $\log(T)$, see later.

To complete the discussion on quadrature rules, we now give a new equivalence result about Gauss-Legendre quadratures on (4.1), and diagonal Padé approximants. Aside from its theoretical interest, this fact allows for a new representation of the error for diagonal Padé approximants.

LEMMA 4.2. *Any quadrature rule of the type (4.2) is equivalent to a rational approximation of* $\log(T)$*.*

Proof. We have $Q := \sum_{k=1}^{N} c_k F(t_k)$, and $F(t) = (T-I)((T-I)t+I)^{-1}$. Since $F(t_i)F(t_j) = F(t_j)F(t_i)\,, \forall i, j$, then we can rewrite Q as

$$Q = (T-I)[\sum_{k=1}^{N} c_k \prod_{i=1, i\neq k}^{N} ((T-I)t_i + I)]\,[\prod_{i=1}^{N}((T-I)t_i + I)]^{-1} ,$$

from which the claim follows. □

THEOREM 4.3. *Let $\rho(I-T) < 1$, and let Q in (4.2) be the N-point Gauss-Legendre quadrature rule for* $\log(T)$*. Then, Q is the (N, N) diagonal Padé approximant to* $\log(T)$*.*

Proof. With previous notation, and under the stated assumptions, we have

$$F(t) = (T-I)\sum_{k=0}^{\infty}(-1)^k (T-I)^k t^k ,$$

where the series converges. Therefore,

$$\log(T) = \sum_{k=1}^{\infty}(T-I)\int_0^1 (-1)^k (T-I)^k t^k dt .$$

Since N-points Gauss-Legendre rules are exact for polynomials of degree up to t^{2N-1}, we immediately realize that Q agrees with $\log(T)$ up to the term $(T-I)^{2N+1}$ excluded. From Lemma 4.2, Q is a rational approximation to $\log(T)$, and thus it must be the (N, N) diagonal Padé approximant. □

COROLLARY 4.4. *Under the assumptions of Theorem 4.3, we have the following error estimate for the (N, N) diagonal Padé approximants Q to* $\log(T)$*:*

$$\log(T) - Q = \frac{(N!)^4}{(2N+1)((2N)!)^3}\sum_{k=0}^{\infty}(2N+k)\cdots(k+1)A^{2N+k+1}\eta^k ,$$

where $0 \le \eta \le 1$, and $A = I - T$.

Proof. From standard quadrature errors for Gauss-Legendre rules (e.g., see [AS]), and differentiating under the series of Theorem 4.3, the result follows at once. □

REMARK 4.5. The previous results hint that a possible way to use quadrature rules is to first pass to their rational form equivalent. On the other hand, for diagonal Padé approximants, it might be instead more desirable to pass to their quadrature formula equivalent (4.2), to avoid ill-conditioning in the denominator of the rational function. Moreover, from Theorem 4.3 we see that Gauss formulas are an excellent candidate for a parallel implementation of Padé approximants.

From the preceding discussion, it has become clear that it would be generally desirable to have T close to I. This would make the finite precision behavior of the above techniques much better.

Scaling. An ideal scaling strategy, in the context of computing $\log(T)$, is to precondition the problem so that (for a modified matrix T) $T \approx I$. In any case, a reasonable scaling ought to give a T for which $\|I - T\| < 1$.

One approach is to find, inexpensively, some X_1 approximating $\log(T)$ such that $X_1 T = T X_1$, and then consider $e^{-X_1}T$, find its logarithm, and finally recover $\log(T) = X_1 + \log(e^{-X_1}T)$. Some ideas on this are in [D]. Also (3.12) can be used in this light, since any quadrature rule of the type (4.2) gives $X_1 : X_1 T = T X_1$.

A more systematic approach results from the *inverse scaling and squaring* procedure of Kenney and Laub [KL2]. The basic idea of this approach is to "flatten out" the matrix T. It is based upon the identity $\log(T) = \log((T^{1/2^k})^{2^k}) = 2^k \log(T^{1/2^k})$, and the realization that, eventually, $T^{1/2^k} \to I$. With respect to this scaling procedure, we need to consider some aspects: (i) how to take square roots and which square roots should we take, (ii) when should we take square roots, (iii) what is the conditioning of the overall procedure, and (iv) if there are risks involved with this scaling strategy.

With respect to the first issue, we have adopted the choice made by Higham (see [Hi1], and also [BH]), thereby relying on a real Schur approach. Under the assumptions of Theorem 1.1, there are many square roots of T, see [Hi1, Theorems 5 and 7]. However, in our context, to eventually find the principal branch of $\log(T)$, there is only one choice. We **must** select the square root(s) according to the Lemma below (see also [KL2, Lemma A1]).

LEMMA 4.6. *Let $B \in \mathbb{R}^{m\times m}$ be invertible with no eigenvalues on the negative real axis. Then, B has a unique 2^k-th root S, i.e. $S^{2^k} = B$, which is a primary matrix function of B, and such that if $\nu \in \Lambda(S)$, then*

(a) $\frac{-\pi}{2^k} < \arg(\nu) < \frac{\pi}{2^k}$, and

(b) $\Re e(\nu) > 0$, for $k = 1, 2, \ldots$.

Proof. A constructive proof can be based upon the method of Higham (see [Hi1, p.417] for details). □

When to take square roots? Ultimately, it all depends on what algorithm we use to approximate $\log(T)$. For algorithms fully based on truncated series or Padé approximants, then square roots of the full matrix T have to be taken in order to ensure numerical stability and rapid convergence. When using a

Schur decomposition approach, the procedure is only needed to obtain L_{ii} in (3.7), in those cases for which approximation techniques are required for the L_{ii}. One thing to keep in mind is that, asymptotically, taking square roots gives a decrease in norm by a factor of 2. Therefore, how many square roots to take depends on which algorithm we eventually use for computing the log of the scaled matrix.

To examine the conditioning of the inverse scaling and squaring procedure, we must look at the Frechét derivative of $M(T) := 2^k \log(T^{1/2^k})$. Let $T_j = T^{1/2^j}$, $j = 0, 1, \ldots, k$ (so $T_0 = T$), and let G and F be the log and square root functions, respectively. Then, upon repeated use of Lemma 2.3, we have

$$M'(T)Z = 2^k G'(T_k) F'(T_{k-1}) \cdots F'(T_0) Z \, .$$

In other words, unavoidably, the better value for the norm of the Frechét derivative of the log (because $T_k \approx I$) is being paid by the Frechét derivatives of the square roots. The problem of estimating the Frechét derivative of the square root function can be based on Corollary 2.4, by considering $S(X) := X^2$, and the identity $S(F(T)) = T$. Therefore, we have the equalities

$$F'(T_0)Z = (S'(F(T_0)))^{-1}Z, \quad F'(T_1)(F'(T_0)Z) = (S'(F(T_1)))^{-1}F'(T_0)Z \, , \ldots ,$$

$$F'(T_{k-1})(F'(T_{k-2}) \cdots F'(T_0)Z) = (S'(F(T_{k-1})))^{-1}(S'(F(T_{k-2})))^{-1} \cdots (S'(F(T_0)))^{-1}Z \, ,$$

and thus we have

$$G'(T)Z = 2^k G'(T_k) \left\{ (S'(F(T_{k-1})))^{-1} \cdots (S'(F(T_0)))^{-1} Z \right\} \, . \tag{4.6}$$

Formula (4.6) forms the basis of the following algorithm to estimate $\|G'(T)Z_0\|$, for a given Z_0, and hence to estimate $\text{cond}(G(T))$. This procedure gave us much better results (both in terms of accuracy and expense) than one directly based on Theorem 3.10.

Let $T_0 = T$, $T_j = T^{1/2^j}$, $j = 1, \ldots, k$ where the index k must be chosen so that $\|I - T_k\| = \omega < 1$, and let Z_0 be given.

(a) Solve

$$F(T_j)Z_{j+1} + Z_{j+1}F(T_j) = Z_j \, , \quad j = 0, 1, \ldots, k-1 \tag{4.7}$$

(notice that the $F(T_j)$ stay quasi-triangular if T_0 is such; also, one might already have the T_j from scaling via taking square roots, but only if square roots of all of T had been taken);

(b) since $G'(T)Z_0 = 2^k G'(T_k)Z_k$, we approximate $G'(T_k)Z_k$ by using a quadrature rule on (3.13).

It is obvious that the algorithm is well defined, since the Sylvester equations (4.7) are uniquely solvable. In terms of computational cost, by using a composite quadrature rule with N points, at leading order one needs $\frac{1}{6}(k+N)n^3$ flops, plus the cost of computing the T_j's if they are not available, which might amount to another $\frac{1}{6}kn^3$ flops, plus the initial cost of the Schur reduction of T.

Next, we show that the above eventually provides a good estimate of $\|G'(T)Z_0\|$. We show this for the composite Simpson rule, but the reasoning applies to any other quadrature rule.

THEOREM 4.7. *Let $T \in \mathbb{R}^{n\times n}$ be given such that $\|I-T\| = \omega < 1$, and let $G(T) = \log(T)$. Let Z be given, and let $G'(T)Z$ be given by (3.13). Let CS be the composite Simpson rule with N points (N even) approximating (3.13), so that $h = 1/N$ below. Then we have*

$$\|G'(T)Z - CS\| \le \frac{2nh^4}{3}\frac{\omega^4}{(1-\omega)^6}\|Z\|\,. \tag{4.8}$$

Proof. We have

$$G'(T)Z = \int_0^1 A(t)ZA(t)dt = \int_0^1 F(t,Z)dt\,,$$

where we have set $A(t) = ((T-I)t+I)^{-1}$, and $F(t,Z) = A(t)ZA(t)$. From standard quadrature errors, we have

$$\|G'(T)Z - CS\| \le \frac{nh^4}{180}\max_{0\le t\le 1}\|F^{(iv)}(t,Z)\|.$$

Now, we can verify that $A^{(j)}(t) = (-1)^j j! A(t)((T-I)A(t))^j$, and that

$$F^{(k)}(t,Z) = \sum_{j=0}^{k}\binom{k}{j}A^{(k-j)}(t)ZA^{(j)}(t) = (-1)^k k!\sum_{j=0}^{k}A(t)((T-I)A(t))^{k-j}ZA(t)((T-I)A(t))^j\,,$$

from which it is easy to get

$$\|F^{iv}(t,Z)\| \le 120\frac{\omega^4}{(1-\omega)^6}\|Z\|\,,$$

and the result follows. □

THEOREM 4.8. *Let $T \in \mathbb{R}^{n\times n}$, $G(T) = \log(T)$, and $E(T) = e^T$. Let Z_0 of norm 1 be given, and let k be such that $\|I - T_k\| = \omega < 1$, with $T_k := T^{1/2^k}$. Let Z_k be obtained from (4.7), so that $G'(T)Z_0 = 2^k G'(T_k)Z_k$. Let CS be the composite Simpson rule with N points (N even) approximating $G'(T_k)Z_k$ from (3.13), so that $h = 1/N$ below, and let $G'(T_k)$ be invertible. Then, we have*

$$\frac{\|G'(T)Z - 2^k CS\|}{\|G'(T)Z_0\|} \le \frac{nh^4}{9}\frac{\omega^4(1+\omega)}{(1-\omega)^6}\,. \tag{4.9}$$

Proof. From Theorem 4.7, we have

$$\|G'(T)Z_0 - 2^k CS\| \le \frac{2nh^4}{3}\frac{\omega^4}{(1-\omega)^6}\|2^k Z_k\|\,.$$

On the other hand, from $G'(T)Z_0 = 2^k G'(T_k)Z_k$, we also have $\|2^k Z_k\| \le \|G'(T)Z_0\|\,\|(G'(T_k))^{-1}\|$, and from Corollary 2.4 we get $\|(G'(T_k))^{-1}\| = \|E'(G(T_k))\|$. Therefore, we have

$$\frac{\|G'(T)Z_0 - 2^k CS\|}{\|G'(T)Z_0\|} \le \frac{2nh^4}{3}\frac{\omega^4}{(1-\omega)^6}\|E'(G(T_k))\|\,. \tag{4.10}$$

Now, using (2.7) we have

$$\|E'(G(t_k))\| = \max_{Y:\|Y\|=1}\|\int_0^1 T_k(1-s)YT_k\,s\,ds\| \le \frac{1}{6}\|T_k\| \le \frac{1+\omega}{6}\,.$$

Using this in (4.10) gives the result. □

EXAMPLE 4.9. If $h^{-1} \approx (n)^{1/4}$, then $\omega = .25$ gives 3 digits accuracy , and $\omega = .35$ gives 2 digits. More than acceptable for condition estimation. □

REMARK 4.10. Use of (4.9) to achieve a good estimate of $||G'(T)||$ requires an appropriate choice of Z_0. We have found that selecting Z_0 according to Remark 3.11 always gave excellent results, and no need arose to further iterate the process. For our experiments in Section 6, we always used this choice of Z_0 along with (4.9), to estimate cond$(G(T))$. This strategy seems to be both very reliable and efficient in comparison with existing alternatives ([KL2]).

To complete this Section, we ought to warn against some possible risks involved with the "inverse scaling and squaring" procedure. Its main limitation is exactly its power: one progressively flattens out the spectrum of the matrices $T_j = T^{1/2^j}$. This may lead to unwanted loss of numerical significance in those cases in which the original T has close eigenvalues (but not identical) and several square roots are required in order to obtain a $T_j : ||I - T_j|| < 1$. The risk is that, after many square roots, all eigenvalues have numerically converged to 1, and are no longer distinct. Our experience has shown that this might occasionally happen, but only for ill conditioned problems, for which $||T^{1/2^j}||$ increases with j, before decreasing.

5. IMPLEMENTATION & EXPENSE.

In our implementations to approximate $\log(T)$, we have always first reduced the matrix T to ordered quasi-triangular form via a real Schur reduction. The ordered Schur reduction is standard, and we used routines from EISPACK and from [St], thereby ordering eigenvalues according to their modulus. Unless more information is available on T, we always recommend a Schur reduction prior an approximation technique; inter alia, it allows for an immediate solution of the problem if T is normal (see Corollary 3.4), and it renders transparent whether or not some methods are suitable for the given problem. In what follows, we will therefore assume that T is quasi-triangular, and not normal. In tune with our discussion on scaling, we will also assume hereafter that T has been scaled so that $||A|| < 1$, where $A = I - T$. Typically, this has been achieved by progressively taking square roots of T. To assess the computational expense, we give the leading order flops' count of the algorithms; a flop is the combined expense of one floating point multiplication and one floating point addition.

Both for truncated expansions of the two series, and for diagonal Padé approximants, one needs to evaluate matrix polynomials. Ignoring finite precision considerations, let us first discuss what degree is needed in order to obtain a desired accuracy, for a given $||A||$. We fixed the accuracy to 10^{-18}.

Figure 1 is a graph showing which degrees q are needed as functions of $||A||$, in order to be guaranteed an absolute error less than 10^{-18}, for approximation resulting from:

(i) truncating the series (3.1)

$$S_1 := \sum_{k=1}^{q} \frac{A^k}{k} ; \tag{5.1}$$

(ii) truncating the series (3.3)

$$S_2 := 2\sum_{k=0}^{m} \frac{B^{2k+1}}{2k+1}\,, \quad B = (T-I)(T+I)^{-1}, \quad q = 2m+1\,; \tag{5.2}$$

(iii) considering the diagonal Padé approximant $R_{q,q}(A)$.

To obtain the degrees q, we have made sure that the remainders contributed less than the desired accuracy. This is easy enough to do for (5.1) and (5.2), and for the Padé approximants we used the explicit form of the remainder from [KL1, Theorem 5].

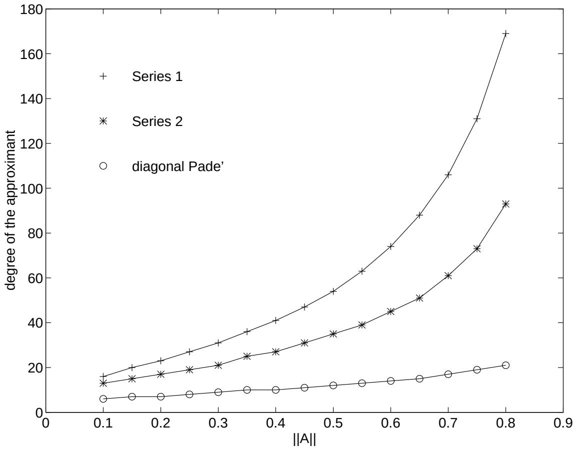

Figure 1.

As an example, for $||A|| \le 0.35$, 0.3, we need $q = 36$, 31, for S_1, $q = 25$, 21, for S_2, and $q = 10$, 9, for $R_{q,q}$. (If $||A|| = 0.35$, the $(9,9)$ Pade' guarantees an error of 1.152×10^{-18}.)

Naturally, for Padé one also needs to be aware of the condition number of the denominator $Q(A)$, since this matrix needs to be inverted. Borrowing from [KL1, Lemma 3], an upper bound on $\text{cond}(Q(A))$ is given by $Q(-||A||)/Q(||A||)$. Figure 2 shows this upper bound on $\text{cond}(Q(A))$ for the case of $q = 9$, for $||A|| \in (0,1)$. For example, for $||A|| = 0.35$, 0.3, one has that $\text{cond}(Q(A)) \le 25.34$, 15.66.

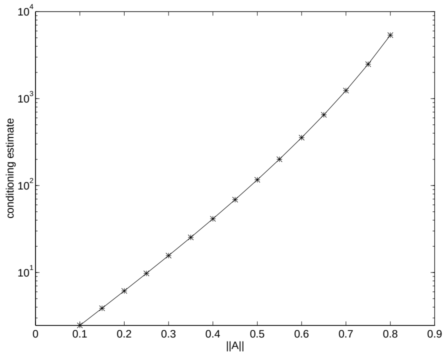

Figure 2.

Next, we need to consider the expense associated with evaluating polynomials of degree q and the $q \times q$ diagonal Padé. As usual, let T be quasi-triangular of dimension n. The algorithm we used to evaluate the polynomials is taken from [GvL, Section 11.2], and it requires the explicit computation of $A^2, A^3, \ldots, A^s$, where s is a given integer satisfying $1 \leq s \leq \sqrt{q}$. Let $r = \lfloor q/s \rfloor$; then, following [GvL], it is easy to show that, at leading order, the evaluation of S_1 requires $(r+s-2)\frac{1}{6}n^3$ flops if $sr = q$, and $(r+s-1)\frac{1}{6}n^3$ flops otherwise. The choice $s = \lfloor \sqrt{q} \rfloor$ ensures the minimal flop count.

The cost associated with S_2 can be obtained in a similar way, taking into account the cost of the evaluation of $B = (T-I)(T+I)^{-1}$ (about $\frac{1}{6}n^3$ flops) and observing that only odd powers of B are required. With $q = 2m+1$ now we have $s = \lfloor \sqrt{m} \rfloor$, $r = \lfloor m/s \rfloor$ and a leading cost of $(r+s+1)\frac{1}{6}n^3$ flops if $sr = m$, and $(r+s+2)\frac{1}{6}n^3$ flops otherwise.

Finally, the cost associated with $R_{q,q}(A)$ can be obtained observing that $A^2, A^3, \ldots, A^s$ must be computed only once for the two polynomials $P(A)$ and $Q(A)$, and adding the cost of the evaluation of $P(A)(Q(A))^{-1}$. With the above notation, we have a leading cost of $(2r+s-2)\frac{1}{6}n^3$ flops if $sr = q$, and $(2r+s)\frac{1}{6}n^3$ flops otherwise. In this case, a better compromise for s is $s = \lceil \sqrt{q} \rceil$, which permit to gain something in the flop count, with respect to taking $s = \lfloor \sqrt{q} \rfloor$.

Figure 3 shows the asymptotic cost associated with S_1, S_2 and $R_{q,q}(A)$ to have an error less than 10^{-18} in function of $||A||$. For example, if $||A|| \leq 0.35$, 0.3, S_1 requires about $10\,\frac{1}{6}n^3$ flops, S_2 needs $8\,\frac{1}{6}n^3$ flops, and $R_{q,q}(A)$ needs $q = 10$ and $8\frac{1}{6}n^3$ flops for $||A|| = 0.35$, whereas $q = 9$ and $7\frac{1}{6}n^3$ flops suffice when $||A|| = 0.3$. It is interesting to observe that also using a $(12,12)$ Padé gives leading flop count of about $8\frac{1}{6}n^3$ flops.

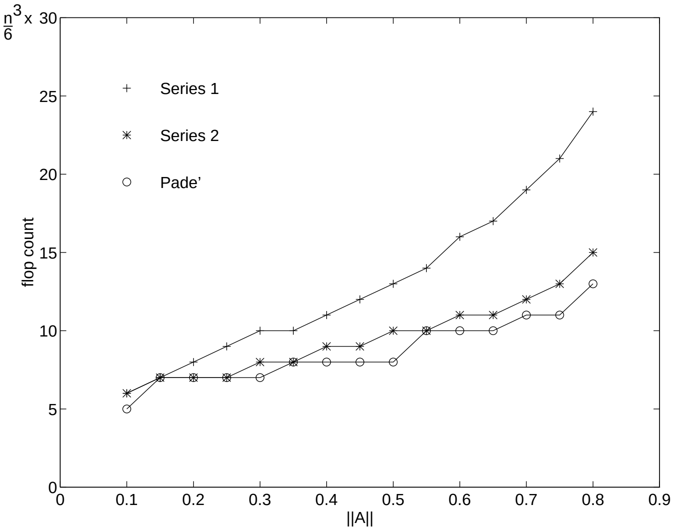

Figure 3.

Finally, there is to consider the cost of the real Schur decomposition, and of taking square roots. The cost of solving (3.8) is a complicated function of the block sizes; for distinct eigenvalues, i.e., the triangular case, it amounts to $\frac{1}{3}n^3$ flops. In any case, the bulk of the expense is the ordered real Schur decomposition, which costs about $15n^3$ flops. Then, one square root costs about $\frac{1}{6}n^3$ flops (see [Hi1]). Since taking square roots, asymptotically, decreases the norm by $1/2$, then we see that it makes better sense, from the point of view of the cost, to take square roots rather than to use a high degree approximant. We found that a good compromise is to take square roots up to having $\|A\| \leq 0.35$, followed by the $(9,9)$ Padé or S_2.

6. EXAMPLES.

In this Section we report on some of the problems we have solved numerically. All computations have been done on a Sparc10 in double precision ($EPS \approx 2.2 * 10^{-16}$).

We mainly report on results obtained by the methods which have proven robust enough to handle the largest portion of all problems considered; for example, we do not report on results obtained by using (5.1), nor by using the ODE approach in either formulation (3.10) or (3.12) (but see Theorem 4.3). Thus, unless otherwise noted, all problems below have been solved by the following general strategy:

(i) Schur reduction with eigenvalues' clustering according to increasing modulus. We have used the software in [St] (with minimal modifications) to do this step. The tolerance for the QR algorithm was set to $2 * EPS$.

(ii-a) Scaling of diagonal blocks by taking square roots, up to obtaining $\|A\| \leq 0.35$, followed by the 9×9 diagonal Padé approximant for these blocks, inverse scaling, and use of (3.8). Diagonal blocks in the real Schur form have been considered distinct if the minimum distance between their eigenvalues was greater than $1/10$. Needless to say, if –after grouping– all diagonal blocks are either 1×1 or 2×2 of complex conjugate eigenvalues, then we used Lemma 3.3 instead of scaling and Padé approximants.

(ii-b) Truncated expansion (5.2) on the whole matrix in lieu of scaling by square roots and Padé approximants, if convergence criteria for such series were met.

(iii) Back transformation.

As measure of accuracy for the computed logarithms, we have considered err $:= \frac{\|e^{\log_c(T)} - T\|}{\|T\|}$, where $\log_c(T)$ is our computed approximation to the log. This essentially boils down to assessing the absolute error in the log itself. To approximate the exponential function, we have used both Matlab functions *expm* and *expm2*, that is a Schur based technique and a series expansion technique. Typically, *expm* performed better, but on occasions *expm2* was needed. We have also used our own implementation of the method of scaling and squaring followed by a diagonal Padé approximant to the exponential, following [GvL, Algorithm 11.3.1 p. 558]. In the examples below, we also report the estimates "cond" of the condition number (3.2). This is done according to Theorem 4.8.

Many tests have been done on random matrices. These were generated by exponentiating the matrices obtained with the *randn* function of Matlab, which returns entries in $[-1,1]$ according to the normal distribution. If a particular structure was desired (e.g., orthogonal) these random matrices were further manipulated (e.g., taking their QR factorization).

In the tables below, for the computed logarithm $\log_c T$, we report: L= $\|\log_c T\|$, cond, nbl/nrad (the number of diagonal blocks, and the most square roots taken on any of these blocks), err, q (the number of terms taken for (5.2) directly on T, if applicable), err_2 (the error for (5.2)), and err_m (the error obtained by using the Matlab function *logm* to approximate $\log T$). Exponential notation is used throughout; e.g., 2.3×10^7 is written as 2.3E7. All results are given for the Frobenius norm, to conform to previously published results.

EXAMPLE 6.1. *"Easy" Problems.* A set of randomly generated positive definite and orthogonal matrices was considered just to test the technique based on Corollary 3.4. In all cases, accuracy to machine precision was obtained. We also generated more than 60 general random matrices, of dimension between 5 and 100. Also in these cases we obtained accuracy to full machine precision.

EXAMPLE 6.2. *Symplectic T.* We generated a dozen random symplectic matrices by exponentiating (via diagonal Padé approximants) randomly generated Hamiltonian matrices. For some of these matrices we got a very large condition number (3.2). Nonetheless, we obtained very accurate answers for the computed logarithms. However, the end result was often far from being a Hamiltonian matrix, that is the relevant structure got lost. For these problems, when applicable, using (5.2) directly was also an effective way to proceed; even though some of the linear algebra (such as matrix inversion) was done by non-symplectic methods, the end result was much more nearly a Hamiltonian matrix than with the Schur method (see [D]).

EXAMPLE 6.3. *"Harder" Problems.* These problems have been chosen to illustrate some of the dangers in using the *logm* function of Matlab. In Table 1, Tests 1-3 refer to a triangular matrix of dimension 20, with all 1's above the diagonal, and $1/4, 1$, and 4 on the diagonal, respectively. Of course, for these matrices, no Schur reduction or grouping occurred. Test 4, instead, has been chosen to illustrate the potential danger of taking too many square roots. It is the matrix $\begin{pmatrix} 1+10^{-7} & 10^5 & 10^4 \\ 0 & 1 & 10^5 \\ 0 & 0 & 1 \end{pmatrix}$. In this case, (5.2) has clearly to be preferred.

Table 1.

Test	L	cond	nrad	err	q	err_2	err_m
1	6.98E7	4.75E10	28	1.2E-8	238	1.1E-8	82.54
2	5.32	5.0865	4	0	19	0	9E-3
3	6.56	0.9511	5	2.7E-15	129	3.7E-16	1.7E-2
4	5E9	5.67E14	34	5.9E-4	2	5E-13	1.4E15

EXAMPLE 6.4. *Examples from the literature.* These problems have previously appeared in the literature, see [Wa] and [KL2]. We tested our method to independently confirm the results of [KL2] about conditioning. In Table 2, Tests 1-6 refer to the Examples 1-6 of [KL2]. We notice that our estimates for cond are in perfect agreement with the results in [KL2]. For Tests 1,2, and 3, we also used scaling by square roots and the 9×9 diagonal Padé approximant on the whole matrix; this required 5, 8, and 11 square roots, respectively, for the same accuracy.

Table 2.

Test	L	cond	nbl/nrad	err	q	err$_2$	err$_m$
1	7.48	5.08	2/4	3.5E-15	7949	1.7E-13	1.1E-15
2	53.85	9E6	3/0	9.E-15	–	–	7.1E-15
3	575.95	6.44E9	3/0	6.2E-14	–	–	5.2E-13
4	2.9997	3.76	1/4	2.5E-15	19	1.5E-16	6.7E-9
5	1E6	3.33E11	1/22	0	1	0	6.2E-6
6	172.68	5.94E6	1/9	3.7E-13	229	2.3E-10	6.4E-10

7. CONCLUSIONS.

In this work, we have provided analysis and implementation of techniques for computing the principal branch of a real logarithm of a matrix T, $\log(T)$. Some of the techniques considered had been around for a while, like Padé approximants and series expansion. Some other techniques had not been previously analyzed or even introduced. In particular, the Schur method with eigenvalue grouping followed by a back recursion, and integral based representations for both the logarithm and its Frechét derivative. This latter aspect is related to the conditioning of the problem, an issue we have addressed in details, and on which we have given many new results that better characterize it. In fact, from the theoretical point of view, our main contributions are the results about conditioning, and those related to the integral representation of $\log(T)$.

From the computational point of view, all things considered, we think that the most reliable and efficient general-purpose method is one based on the real Schur decomposition with eigenvalues' grouping, scaling of the diagonal blocks via square roots, and diagonal Padé approximants. Also using S_2 (see (5.2)), instead of the Padé approximant, is a sound choice. Moreover, using S_2 was definitely the most appealing choice for poorly conditioned problems. Although all of the programs we have written are of an experimental nature, we believe they are robust enough to be indicative of the typical behavior. We hope that our work will prove valuable to people interested in mathematical software, the more so since the only existing software tool which computes the logarithm of a matrix ([Matlab]) does not use a foolproof algorithm to do so. Moreover, the implementation of Matlab nearly always produces complex matrices for answers, because it uses unitary reduction to complex Schur form.

The problem of reliably estimating the Frechét derivative of $\log(T)$, at a fraction of the cost of computing $\log(T)$, or at least without a drastic increase in cost, is truly an outstanding difficulty. None of the methods of which we are aware succeeds in this. One technique we have considered, based on Theorem 3.10 and Remark 3.11, is usually very inexpensive, but not always reliable. The other technique we introduced, based on Theorem 4.8, has at least proven very reliable, but, in general, it is at least as expensive as computing the log itself.

Finally, in this work we have focused on the problem of computing **one** logarithm of **one** matrix. Different conclusions are reached if one is interested in computing a branch of logarithms of slowly varying matrices. In such cases, of course, one should favor an approach which uses the previously computed logarithms, and thus more carefully consider iterative techniques and different scaling strategies. We anticipate some work in this direction.

Acknowledgments. We thank N. Higham, A. Iserles, and C. Kenney for insightful comments on this work. Also, thanks are due to N. Higham for pointing out to us references [C], [He], [SS], and [Wo].

8. REFERENCES.

[AS] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 10th edition, J. Wiley & Sons (1972).

[BG-M] G.A. Baker, P. Graves-Morris, *Padé Approximants, Part I & Part II*, Encyclopedia of Mathematics vol.s 13-14, Addison-Wesley (1981).

[BH] A. Bjorck, S. Hammarling, "A Schur Method for the Square Root of a Matrix", *Linear Algebra & Its Applic.*, **52/53** (1983), pp. 127-140.

[C] W.J. Culver, "On the Existence and Uniqueness of the Real Logarithm of Matrix", *Proc. Amer. Math. Soc.*, **17** (1966), pp. 1146-1151.

[D] L. Dieci,"Considerations on Computing Real Logarithms of Matrices, Hamiltonian Logarithms, and Skew-Symmetric Logarithms," to appear in *Linear Algebra & Its Applic.* (1994).

[G] F.R. Gantmacher, *Théorie des Matrices* **1&2**, Dunod, Paris (1966).

[GvL] G.H. Golub, C.F. Van Loan, *Matrix Computations*, 2nd edition, The Johns Hopkins University Press (1989).

[He] B.W. Helton, "Logarithms of matrices", sl Proc. Amer. Math. Soc., **19** (1968), pp. 733-738.

[Hi1] N.J. Higham, "Computing real Square Roots of a Real Matrix", *Linear Alg. and Its Applic.*, **88/89** (1987), pp. 405-430.

[Hi2] N.J. Higham, "Perturbation Theory and backward Error for $AX - XB = C$", *BIT*, **33** (1993), pp. 124-136.

[HJ] R.A. Horn, C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press (1991).

[KL1] C. Kenney and A.J. Laub, "Padé Error Estimates for the Logarithm of a Matrix", *International Journal of Control*, **50-3** (1989), pp. 707-730.

[KL2] C. Kenney and A.J. Laub, "Condition Estimates for Matrix Functions", *SIAM J. Matrix Analysis & Applications*, **10** (1989), pp. 191-209.

[LS1] G.J. Lastman and N.K. Sinha, "Transformation Algorithm for Identification of Continuous-Time Multivariable Systems from Discrete Data", *Electronics Letters*, **17**, (1981), pp. 779-780.

[LS2] G.J. Lastman and N.K. Sinha, "Infinite Series for Logarithm of Matrix, Applied to Identification of Linear Continuous-Time Multivariable Systems from Discrete-Time Models", *Electronics Letters*, **27-16**, (1991), pp. 1468-1470.

[M] R. Mathias, "Condition Estimation for Matrix Functions via the Schur Decomposition", *SIAM J. Matrix Anal. and Applic.*, **16-2** (1995), pp. 565-578.

[Matlab] *Matlab Reference Guide*, The MathWorks, Inc. (1992).

[MvL] C.B. Moler and C. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *Siam Review*, **20** (1978), pp. 801-836.

[P] B.N. Parlett, "A Recurrence Among the Elements of Functions of Triangular Matrices", *Linear Alg. and Its Applic.*, **14** (1976), pp. 117-121.

[Si] Y. Sibuya, "Note on Real Matrices and Linear Dynamical Systems with Periodic Coefficients", *J. Mathematical Analysis and Applications*, **1** (1960), pp. 363-372.

[SS] B. Singer and S. Spilerman, "The Representation of Social Processes by Markov Models", *Amer. J. Sociology*, **82-1** (1976), pp. 1-54.

[St] G.W. Stewart, "HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix", *ACM Trans. Math. Software* **2** (1970), pp. 275-280.

[vL] C. Van Loan, "The sensitivity of the matrix exponential", *Siam J. Numer. Analysis*, **14** (1977), pp. 971-981.

[V] E.I. Verriest, "The Matrix Logarithm and the Continuization of a Discrete Process", *Proc.s 1991 American Control Conference* (1991), pp.184-189.

[YS] V.A. Yakubovich and V.M. Starzhinskii, *Linear Differential Equations with Periodic Coefficients* **1&2**, John-Wiley, New York (1975).

[Wa] R. C. Ward, "Numerical Computation of the Matrix Exponential with Accuracy Estimates", *SIAM J. Numer. Anal.* **14** (1977), pp. 600-610.

[Wo] A. Wouk, "Integral Representation of the Logarithm of Matrices and Operators", **11** (1965), pp. 131-138.