Return to the Middle Ages:
A Half-Angle Iteration for the Logarithm of a Unitary Matrix

Sheung Hun Cheng[†], Nicholas J. Higham[‡], Charles S. Kenney[♮], Alan J. Laub[*]

[†]Centre for Novel Computing
Department of Computer Science, University of Manchester
Manchester, M13 9PL, England
scheng@cs.man.ac.uk

[‡]Department of Mathematics, University of Manchester
Manchester, M13 9PL, England
higham@ma.man.ac.uk

[♮]ECE Department, University of California
Santa Barbara, CA 93106-9560, USA
kenney@seidel.ece.ucsb.edu

[*]College of Engineering, University of California
Davis, CA 95616-5294, USA
laub@ucdavis.edu

Keywords: logarithms, unitary matrices, half-angle formula.

Abstract

If A is unitary then $A = e^{iH} = \cos H + i \sin H$ where H is Hermitian. We examine the problem of approximating H. The standard approach to approximating logarithms is to take successive square roots until $A^{1/2^n}$ is close to the identity and then apply a Padé approximation to $\log A = 2^n \log A^{1/2^n}$. For the unitary problem we have $A^{1/2^n} = \cos(H/2^n) + i \sin(H/2^n)$ which suggests the use of half-angle formulas in computing the square roots. In this paper we consider problems associated with incomplete Denman-Beavers square root approximations as applied to the half-angle formulation. Square roots of unitary matrices are again unitary and the desire to have approximate square roots retain this property leads us to a tangent formulation of the half-angle iteration. Numerical tests illustrate this new procedure on a variety of examples.

1 Introduction

Addition formulas for the sine and cosine have been used since antiquity to transform multiplication problems into simpler problems involving addition. For example, suppose that we have a table of cosine values and we want to form the product of $\cos x$ with $\cos y$. We first would look in the table for x and y corresponding to $\cos x$ and $\cos y$. Add and subtract x and y, then find $\cos(x+y)$ and $\cos(x-y)$ in the table of cosines. Sum the two cosine values and divide by 2 to get

$$\cos x \cos y = (\cos(x+y) + \cos(x-y))/2$$

Tables of the form $r \cos x$ allow numbers larger than 1 to be multiplied. If $r = 2 \times 10^k$ then $r \cos x \; r \cos y$ is given by

$$r \cos x \; r \cos y = 10^k (r \cos(x+y) + r \cos(x-y))$$

Tables of cosines could be built up by starting with $\cos x \approx 1 - x^2/2$ for x small and then repeatedly using the doubling formula $\cos 2x = 2 \cos x - 1$. It seems that what was actually used to form products in the Middle Ages was the related formula

$$\sin x \sin y = (\cos(x-y) - \cos(x+y))/2$$

Eventually logarithms supplanted trigonometric addition formulas, but in fact the two are intimately related by Euler's formula $e^{i\theta} = \cos\theta + i \sin\theta$.

In this paper we describe how Euler's formula and half-angle formulas can be used in the approximation of the principal logarithm of a unitary matrix. If A is unitary then [6] $A = e^{iH}$ where H is Hermitian and we may assume that H is positive definite. If, in addition, H has eigenvalues strictly between $-\pi$ and π, then iH is the principal logarithm of A and we write $iH = \log A$.

The standard approach to approximating $\log A$ is the inverse scaling and squaring method [15] which uses the relation

$$\log A = 2^n \log A^{1/2^n}$$

to bring A close to the identity matrix via repeated square roots. Then an estimate of the logarithm of $A^{1/2^n}$ is obtained by using Padé approximants of the function $\log{(I - X)}$ with $X = I - A^{1/2^n}$. See Kenney and Laub [14] for an examination of the error incurred by using the Padé approximations. This method requires the computation of successive matrix square roots; in the procedure presented by Kenney and Laub in [15] this was accomplished by transforming A to upper triangular form, applying the method of Björck and Hammarling [3] to obtain the square roots, and then back transforming after the Padé approximation. (For real logarithms, the square root procedure of Higham [8] allows one to perform essentially the same operations using only real arithmetic.) This approach has the advantage that the square roots are exact in the absence of rounding errors. However, the transformation to upper triangular form may not lend itself to parallel implementations.

This concern prompted Cheng et al. [4] to adapt the inverse scaling and squaring approach to the iterative square root method of Denman and Beavers [5]. Given a matrix M with no eigenvalues on the negative real axis, the Denman-Beavers iteration for the square root is

$$\begin{aligned} Y_{n+1} &= \left(Y_n + Z_n^{-1}\right)/2 \\ Z_{n+1} &= \left(Z_n + Y_n^{-1}\right)/2 \end{aligned}$$

where $Y_0 = M$ and $Z_0 = I$. This iteration has been shown by Higham [12] to be stable and converge quadratically with $\lim_{n\to\infty} Y_n = M^{1/2}$ and $\lim_{n\to\infty} Z_n = M^{-1/2}$.

The Denman-Beavers iteration has the advantage of only requiring the matrix operation of inversion — there is no need to transform to upper triangular form.

Cheng et al. [4] were able to exploit this advantage by developing an analysis of the error induced by using the incomplete square root approximations obtained by stopping the Denman-Beavers iteration after only a few iterations.

It is the purpose of this paper to apply and extend the work of [4] to the problem of approximating the logarithm of a unitary matrix. In particular we want to develop an extension that avoids complex arithmetic operations for unitary matrices.

In the Section 2 we look at a direct application of the sine-cosine half-angle formulas to the logarithm problem. After establishing convergence we consider the effects of using incomplete square root approximations. One effect is that even though square roots of unitary matrices are again unitary the approximate square root sequence generally does not share this property. This raises the question of whether at the end of a square root approximation we should try to restore the unitary property by finding the nearest unitary matrix to the approximate square root. The problem of finding the nearest unitary matrix is well-studied and its solution can be effected using the polar decomposition iteration [7]. We present a result showing that the polar decomposition step is equivalent to that of three other iterations for the problem of finding the logarithm (see the Ray Lemma in Section 2).

Unfortunately, the question of error propagation is somewhat difficult in the sine-cosine half-angle formulation. For this reason in Section 3 we rework this approach as a tangent half-angle iteration. This form is completely amenable to a rigorous error analysis and this results in a testable error bound that allows us to specify the desired accuracy of the logarithmic approximation as in the more general approach of Cheng et al. [4].

Section 4 presents results of applying the tangent half-angle approach to a set of numerical examples.

2 Sine and Cosine Half-Angle Iterations

Suppose that A is a unitary matrix of the form $A = e^{iH}$ where H is real and symmetric. Then A satisfies Euler's formula

$$A = \cos H + i \sin H$$

To avoid complex arithmetic in implementing the inverse scaling and squaring method we turn to the half-angle formulas for the sin and cos:

$$\begin{aligned} \cos x &= \left(\frac{1+\cos 2x}{2}\right)^{1/2} \\ \sin x &= \frac{\sin 2x}{2\cos x} \end{aligned}$$

This suggests the following iteration:

Half Angle Iteration

1. Intialize $C_0 = \mathrm{Re}\, A$ and $S_0 = \mathrm{Im}\, A$.
2. Iterate

$$\begin{aligned} C_{k+1} &= \left(\frac{I+C_k}{2}\right)^{1/2} \\ S_{k+1} &= \frac{1}{2}\, S_k C_{k+1}^{-1} \end{aligned}$$

Lemma 1: If $A = e^{iH}$ where $H = H^T$ and $\pi < \lambda(H) < \pi$ then the half-angle iterates satisfy

$$\begin{aligned} C_k &= \cos(H/2^k) \\ S_k &= \sin(H/2^k) \end{aligned}$$

As a consequence $\lim_{k\to\infty} 2^k S_k = H$.

Proof: Since H is symmetric we may diagonalize the problem, in which case the half-angle iteration reduces to a set of scalar iterations. The result then follows from the scalar half-angle formulas.

2.1 Square Root Computations and the Ray Lemma

In the half-angle iteration we need to compute the square root of $(I + \cos H)/2$. If we use an iterative procedure like the Denman-Beavers method then the result will only be an approximation

$$Y_{approx} \approx ((I + \cos H)/2)^{1/2}$$

This raises several questions. First, if we knew the exact square root

$$Y_{exact} = ((I + \cos H)/2)^{1/2} = \cos(H/2)$$

then to proceed with the half-angle logarithm iteration we need

$$V_{exact} = \sin(H/2) = \sin H(2\cos(H/2))^{-1}$$

which we can form by setting

$$V_{exact} = \sin H(2Y_{exact})^{-1}$$

Since we have Y_{approx} instead of Y_{exact} it would seem that we should use

$$V_{approx} = \sin H(2Y_{approx})^{-1}$$

This turns out to be a bad choice numerically. As we shall see in the Ray Lemma it is better to use

$$V_{approx} = Y_{approx} \sin H(I + \cos H)^{-1}$$

since this gives the correct tangent:

$$\begin{aligned} V_{approx}Y_{approx}^{-1} &= \sin H(I + \cos H)^{-1} \\ &= \sin(H/2)(\cos(H/2))^{-1} \\ &= \tan(H/2) \end{aligned}$$

A second question is raised when we consider that $Y_{approx} + iV_{approx} \approx e^{iH/2}$ should be unitary but generally is not. Should we find the nearest unitary matrix by using, say, Newton's method for the polar decomposition of $Y_{approx} + iV_{approx}$? (See [7],[9], [10], and [11] for background on the polar iteration.)

It was this question that pointed the way to the Ray Lemma: consider the scalar polar iteration for $z = \rho e^{i\theta}$

$$z_{k+1} = \left(z_k + \frac{1}{\bar{z}_k}\right)/2$$

with $z_0 = \rho e^{i\theta}$. Here $\bar{z}_k$ denotes the complex conjugate of z_k.

The iterates z_k stay on the ray $w = re^{i\theta}, r > 0$. This is because $z_k = \rho_k e^{i\theta}$ where the real values ρ_k follow a sign iteration

$$\rho_{k+1} = \left(\rho_k + \frac{1}{\rho_k}\right)/2$$

In addition to this connection between the polar iteration and the sign iteration, the Ray Lemma shows that taking one step of the polar iteration is the same as taking one more step of Newton's method for the square root. That is, Newton's method has the same action as the polar iteration: starting with $m_0 = e^{i2\theta}$ the Newton iteration $m_{k+1} = (m_k + m_0/m_k)/2$ first jumps to the ray $re^{i\theta}$ (because $m_1 = r_1 e^{i\theta}$ with $r_1 = \cos\theta$) and then stays there: $m_k = r_k e^{i\theta}$. As a final connection, the Ray Lemma shows that the real version on Newton's method for the square root of $(I + \cos H)/2$ also generates the same iterate stream.

To be specific let us define these four iterations. Note that we work with Newton's method rather than the Denman-Beavers iteration as a convenience only; in the absence of rounding errors they are equivalent. (See Higham [12].) The first iteration is Newton's method for the square root of $\cos H + i \sin H$.

Complex Newton Square Root Iteration:

$$\begin{aligned} M_0 &= \cos H + i \sin H \\ M_{k+1} &= \frac{M_k + M_0 M_k^{-1}}{2} \end{aligned}$$

The second iteration is Newton's method for the square root of $(I + \cos H)/2$.

Real Newton Square Root Iteration:

$$\begin{aligned} A_0 &= \frac{I + \cos H}{2} \\ A_{k+1} &= \frac{A_k + A_0 A_k^{-1}}{2} \end{aligned}$$

For this iteration we also define the associated sequence

$$B_k = A_k \sin H(I + \cos H)^{-1}$$

The third iteration is the polar decomposition iteration.

Polar Iteration:

$$\begin{aligned} N_0 &= \frac{I + \cos H}{2} + \frac{i \sin H}{2} \\ N_{k+1} &= \frac{N_k + N_k^{-H}}{2} \end{aligned}$$

Here N_k^{-H} denotes the complex conjugate transpose of the inverse of N_k. The fourth iteration is the sign iteration for $\cos(H/2)$.

Sign Iteration:

$$\begin{aligned} R_0 &= \cos(H/2) \\ R_{k+1} &= \frac{R_k + R_k^{-1}}{2} \end{aligned}$$

Lemma 2 (Ray Lemma): The four iterations are equivalent in the sense that

$$M_{k+1} = N_k = A_k + iB_k = R_k e^{iH/2}$$

Proof: First note that $M_1 = N_0 = A_0 + iB_0 = R_0 e^{iH/2}$ where $R_0 = \cos(H/2)$. Define $\tilde{R}_k = M_{k+1} e^{-iH/2}$ so that $M_{k+1} = \tilde{R}_k e^{iH/2}$. Substituting this into the iteration for M_k gives $\tilde{R}_{k+1} = (\tilde{R}_k + \tilde{R}_k^{-1})/2$. Thus $\tilde{R}_k$ follows the same sign iteration as R_k. Since they have the same initial value $\tilde{R}_0 = R_0$ we have $M_{k+1} = R_k e^{iH/2}$. Similarly, if we set $\tilde{R}_k = N_k e^{-iH/2}$ then we find that $\tilde{R}_{k+1} = (\tilde{R}_k + \tilde{R}_k^{-H})/2$. However, $\tilde{R}_0 = R_0 = \cos(H/2)$ is real and symmetric. Hence $\tilde{R}_{k+1} = (\tilde{R}_k + \tilde{R}_k^{-1})/2$ and we have again the sign iteration with the same initial matrix R_0; this shows that $N_k = R_k e^{iH/2}$.

Turning our attention to the sequence $A_0 = (I + \cos H)/2$, $A_{k+1} = (A_k + A_0 A_k^{-1})/2$ and $B_k = A_k \sin H (I + \cos H)^{-1}$ we find

$$A_0 A_k^{-1} = A_k (A_k^2 + B_k^2)^{-1}$$

This gives

$$A_{k+1} = A_k (I + (A_k^2 + B_k^2)^{-1})/2$$

Multiply both sides by $\sin H (I + \cos H)^{-1}$ to get

$$B_{k+1} = B_k (I + (A_k^2 + B_k^2)^{-1})/2$$

Combining both of these shows that $A_k + iB_k$ follows the polar iteration

$$A_{k+1} + iB_{k+1} = (A_k + iB_k + (A_k + iB_k)^{-H})/2$$

Since $N_0 = A_0 + iB_0$ we see that $A_k + iB_k$ must be equal to N_k. This completes the proof of the lemma.

Remark: The Ray Lemma shows why the definition $B_k = A_k \sin H (I + \cos H)^{-1}$ is better than the alternative $\tilde{B}_k = \sin H \ (2A_k)^{-1}$. In the limit both yield $\sin(H/2)$. However, since $A_k = R_k \cos(H/2)$ we see that $B_k A_k^{-1} = \tan(H/2)$, i.e., the eigenvalues of $A_k + iB_k$ lie on the rays corresponding to those of $\cos(H/2) + i \sin(H/2)$. In contrast to this, $\tilde{B}_k A_k^{-1} = R_k^{-2} \tan(H/2)$, i.e., the eigenvalues are not on the same rays as those of $\cos(H/2) + i \sin(H/2)$.

3 Tangent Half-Angle Iteration

The Ray Lemma provides connections between seemingly different methods for computing the first square root in the half-angle iteration for the logarithm of a unitary matrix. These connections depend on the fact that the initial values for $\cos H$ and $\sin H$ are exact; after the first square root step we no longer can apply the Ray Lemma since we only have approximations for $\cos(H/2)$ and $\sin(H/2)$. However, by switching to a tangent iteration for $\tan(H/2^k)$ we can bound the build up of error in the final approximation for H.

Consideration of the half-angle tangent iteration was originally motivated by the trigonometic identity

$$\tan(\theta/4) = \frac{0.5 \sin\theta}{\frac{1+\cos\theta}{2} + \sqrt{\frac{1+\cos\theta}{2}}}$$

This means that if we compute the square root of $(I + \cos H)/2$ then we can skip the next square root and jump to $\tan(H/4)$ at the cost of a single matrix inversion. Thereafter we work with $T_k = \tan(H/2^k)$ and use

$$T_{k+1} = T_k \ \left(I + \left(I + T_k^2\right)^{1/2}\right)^{-1}$$

This formula has arisen repeatedly in connection with the natural logarithm. For a nice discussion and links to other references, see Bagby [2].

In actual computation we don't compute the square root of $I + T_k^2$ exactly. To analyze the error that this introduces we first consider the scalar case.

3.1 Scalar Tangent Half-Angle Iteration

Let $t_k = \tan\theta_k$.

Lemma 3: If

$$t_{k+1} = \frac{t_k}{1 + \rho_{k+1}\sqrt{1 + t_k^2}}$$

then

$$t_{k+1} = \tan\left(\frac{\theta_k}{2} + e_{k+1}\right)$$

where

$$\tan e_{k+1} = \tan\left(\frac{\theta_k}{2}\right) \frac{1 - \rho_{k+1}}{1 + \rho_{k+1}}$$

Consequently if $\theta_k \in (-\pi/2, \pi/2)$ and $|e_{k+1}| < \pi/2$ then

$$|e_{k+1}| \leq \frac{|1 - \rho_{k+1}|}{|1 + \rho_{k+1}|}$$

Proof: Let $t_{k+1} = \tan\theta_{k+1}$ so that $e_k = \theta_{k+1} - \theta_k/2$. Now use $\tan(a + b) = (\tan a + \tan b)/(1 - \tan a \tan b)$ to write

$$\begin{aligned}
\tan e_{k+1} \quad &= \quad \frac{\tan(\theta_{k+1}) - \tan(\theta_k/2)}{1 + \tan(\theta_{k+1})\tan(\theta_k/2)} \\
&= \quad \frac{\frac{t_k}{1+\rho_{k+1}\sqrt{1+t_k^2}} - \frac{t_k}{1+\sqrt{1+t_k^2}}}{1 + \frac{t_k}{1+\rho_{k+1}\sqrt{1+t_k^2}} \ \frac{t_k}{1+\sqrt{1+t_k^2}}} \\
&= \quad \frac{t_k}{1 + \sqrt{1 + t_k^2}} \ \frac{1 - \rho_{k+1}}{1 + \rho_{k+1}} \\
&= \quad \tan(\theta_k/2) \ \frac{1 - \rho_{k+1}}{1 + \rho_{k+1}}
\end{aligned}$$

To complete the proof suppose that $\theta_k \in (-\pi/2, \pi/2)$ and $|e_{k+1}| < \pi/2$. Then $|\tan(\theta_k/2)| < 1$ and

$$|e_{k+1}| \quad \leq \quad |\tan e_{k+1}|$$

$$
\begin{aligned}
&= \quad |\tan(\theta_k/2)| \; \frac{|1-\rho_{k+1}|}{|1+\rho_{k+1}|} \\
&\leq \quad \frac{|1-\rho_{k+1}|}{|1+\rho_{k+1}|}
\end{aligned}
$$

Corollary 1: Let $T_1 = \sin H(I + \cos H)^{-1}$ and

$$T_{k+1} = T_k(I + R_{k+1}(I + T_k^2)^{1/2})^{-1}$$

where R_k commutes with H and $\|R_k - I\| < 0.5$. Then $T_k = \tan H_k$ where

$$2^k H_k = H + 2E_1 + 2^2 E_2 + \cdots + 2^k E_k$$

and

$$\|E_k\| \leq \|I - R_k\|/(2 - \|I - R_k\|)$$

In addition, if we assume that

$$\|R_k - I\| \leq \frac{2\delta}{4^k}$$

then

$$\|2^k H_k - H\| \leq \delta$$

Remark: In the above and throughout the paper $\|\cdot\|$ refers to the 2-norm; we work with the 2-norm because of the property that for any symmetric matrix M the 2-norm M is equal to the spectral radius of M

$$\|M\| = \max_i |\lambda_i(M)|$$

This raises the question of how to ensure that the bound $\|R_k - I\| \leq 2\delta/4^k$ is satisfied.

We can treat this generally by considering the Denman-Beavers iteration for the square root of a matrix M. In this iteration $Y_{n+1} = (Y_n + Z_n^{-1})/2$ and $Z_{n+1} = (Z_n + Y_n^{-1})/2$ with $Y_0 = M$ and $Z_0 = I$. In the limit, Y_n tends to $M^{1/2}$ and Z_n tends to $M^{-1/2}$. If we write $Y_n = R_n M^{1/2}$ then the relation $Y_n = MZ_n$ implies that $Z_n = R_n M^{-1/2}$. From this we have

$$
\begin{aligned}
(Y_n - Y_{n-1})(Z_n - Z_{n-1}) &= Y_n Z_n - I \\
&= R_n^2 - I \\
&= (R_n - I)(R_n + I)
\end{aligned}
$$

Taking norms and simplifying gives

$$\|R_n - I\| \leq \frac{\|Y_n - Y_{n-1}\| \, \|Z_n - Z_{n-1}\|}{2 - \|R_n - I\|}$$

In particular if $\|R_n - I\| \leq 0.5$ then

$$\|R_n - I\| \leq \|Y_n - Y_{n-1}\| \, \|Z_n - Z_{n-1}\|$$

This error bound is convenient since it only requires monitoring the difference in the iterates of the Denman-Beavers algorithm. Combining this result with that of Corollary 1 we have the following bound.

Corollary 2: Using the notation of Corollary 1, let $T_1, \ldots, T_k$ be the tangent half-angle iterates and for $1 \leq i \leq k$ let $Y_i^{(n)}$ and $Z_i^{(n)}$ be the nth iterates of the Denman-Beavers square root algorithm with $Y_i^{(0)} = I + T_i^2$ and $Z_i^{(0)} = I$. If we require that for each i, $n = n(i)$ be large enough so that

$$\|Y_i^{(n)} - Y_i^{(n-1)}\| \, \|Z_i^{(n)} - Z_i^{(n-1)}\| \leq \frac{2\delta}{4^i}$$

then

$$\|2^k H_k - H\| \leq \delta$$

3.2 Padé Approximation of the Arctangent

It remains to approximate $H_k = \arctan T_k$. We do this using the Padé approximants of the inverse tangent function. Following Baker [1] the principal Padé approximants of $\arctan x$ can be recovered from the continued fraction expansion

$$\arctan x = \cfrac{x}{1 + \cfrac{x^2/1\cdot 3}{1 + \cfrac{2^2x^2/3\cdot 5}{1 + \cfrac{3^2x^2/5\cdot 7}{1 + \cdots}}}}$$

The mth order principal Padé approximant is the rational function $R_m = P_m/Q_m$ where the polynomials P_m and Q_m are determined by the triple recursion

$$
\begin{aligned}
P_{m+1} &= P_m + m^2x^2 P_{m-1}/(4m^2 - 1) \\
Q_{m+1} &= Q_m + m^2x^2 Q_{m-1}/(4m^2 - 1)
\end{aligned}
$$

with $P_0 = 0, P_1 = x, Q_0 = 1$ and $Q_1 = 1$. For example the first few approximants are

$$
\begin{aligned}
R_1 &= x \\
R_2 &= \frac{3x}{3 + x^2} \\
R_3 &= \frac{15x + 4x^3}{15 + 9x^2} \\
R_4 &= \frac{105x + 55x^3}{105 + 90x^2 + 9x^4}
\end{aligned}
$$

The principal Padé approximants $R_m(x)$ converge to $\arctan x$ as $m \to \infty$ if $|x| < 1$. See [1].

Returning to the notation of the tangent half-angle iteration we note that since T_k is symmetric we can bound the Padé matrix error by the scalar Padé error: if $\|T_k\| < 1$ then

$$\|\arctan T_k - R_m(T_k)\| \leq |\arctan \|T_k\| - R_m(\|T_k\|)|$$

We may combine this with Corollary 2 by noting that $\arctan T_k = H_k$.

Corollary 3: Using the notation of Corollaries 1 and 2, let $T_1, \ldots, T_k$ be the tangent half-angle iterates and assume that for $1 \le i \le k$

$$\|Y_i^{(n)} - Y_i^{(n-1)}\| \, \|Z_i^{(n)} - Z_i^{(n-1)}\| \le \frac{2\delta}{4^i}$$

Let m be large enough so that

$$|\arctan \|T_k\| - R_m(\|T_k\|)| \le \delta/2^k$$

then $\tilde{H}$ defined by

$$\tilde{H} = 2^k R_m(T_k)$$

satisfies

$$\|\tilde{H} - H\| \le 2\delta$$

3.3 The Half-Angle Tangent Algorithm

We summarize Corollaries 1-3 as an algorithm. We assume that we are given a matrix $A = e^{iH}$ where H is symmetric with eigenvalues in $(-\pi, \pi)$. In the absence of rounding errors, this algorithm produces an approximation $\tilde{H}$ to H that satisfies $\|\tilde{H} - H\| \le 2\delta$

1. Let $T_1 = \sin H (I + \cos H)^{-1}$
2. For $2 \le i \le k$,
   (a) Let $Y_i^{(0)} = I + T_{i-1}^2$ and $Z_i^{(0)} = I$
   (b) Form the D-B iterates
   $$\begin{aligned} Y_i^{(n+1)} &= \left(Y_i^{(n)} + \left(Z_i^{(n)} \right)^{-1} \right)/2 \\ Z_i^{(n+1)} &= \left(Z_i^{(n)} + \left(Y_i^{(n)} \right)^{-1} \right)/2 \end{aligned}$$
   (c) Let $n = n(i)$ be large enough so that
   $$\|Y_i^{(n)} - Y_i^{(n-1)}\| \, \|Z_i^{(n)} - Z_i^{(n-1)}\| \le \frac{2\delta}{4^i}$$
   (d) Set $T_i = Y_i^{(n(i))}$
3. Let m be large enough so that the Padé arctangent approximant R_m satisfies
   $$|\arctan \|T_k\| - R_m(\|T_k\|)| \le \delta/2^k$$
4. Set $\tilde{H} = 2^k R_m(T_k)$

4 Numerical Considerations

In this section we consider problems of implementation and efficiency for the tangent half-angle (THA) algorithm and compare it with some alternative procedures.

The first issue to be dealt with is the selection of the number of square root stages, k, for THA. In the related problem of selecting the number of square root stages for approximating the logarithm of a general matrix, the strategy of Cheng et al. [4] is to estimate, at the end of each square root stage, the effort needed to finish the computation using a Padé approximation. This estimated effort is compared with an estimate of the effort required to perform one more square root stage and then use a Padé approximation. If the current stage leads to a smaller estimated effort then the current value of k is selected.

We may estimate the Padé effort by considering the scalar problem of approximating $\arctan \|T_k\|$. That is, we increase the order m of the approximation until the difference $|\arctan \|T_k\| - R_m(\|T_k\|)|$ is less than the required tolerance. To estimate the effort needed for an additional square root stage we can use the effort of the current square root approximation. The effort needed to approximate the arctangent at the next stage is estimated from the scalar problem of approximating $\arctan \|T_k\|/2$ since $\|T_{k+1}\| \approx \|T_k\|/2$. When combined with the tangent half-angle iteration this approach will be referred to as the basic THA method. Note that any errors in estimating computational effort do not affect the accuracy of the computed solution.

In implementing the basic THA method there are a couple of ways to reduce the computational burden. First, the triple recursion formulas for the polynomials P_m and Q_m in the Padé approximations of the arctangent require twice as many matrix multiplications as the equivalent evaluation by forming powers of $M = X^2$ and then using explicit coefficients for the two polynomials. A related discussion of evaluating Padé approximations for the matrix logarithm is given by Higham in [13].

Second, as shown by Higham [12], Newton's method for the square root of a matrix is stable if the condition number of the matrix is less than 3. Newton's method requires only one matrix inversion per step whereas the Denman-Beavers method uses two matrix inversions per step. In the absence of rounding errors both iterations are equivalent in that they produce identical matices $Y_i^{(n)}$. Since the matrices $I + T_k^2$ have condition numbers bounded by $1 + \tan^2(\pi/2^k)$, we may safely switch to Newton's method after the first square root stage.

4.1 Quick Approximations

We anticipate that most applications for the THA method will be in areas needing only low to middle level accuracy. For this reason we have constructed for comparison some low accuracy methods that require very little computational effort.

The first such method results from taking one half-angle step with one Denman-Beavers square root step together with the first order Padé approximation for $\arctan H/4$; this yields an approximation of H that requires only one matrix inversion:

$$F_1 = 8 \sin H(5I + 3\cos H)^{-1}$$

Since H is symmetric the error in this approximation is bounded by the scalar case. Expanding in θ shows that

$$\frac{8\sin\theta}{5 + 3\cos\theta} = \theta + \frac{\theta^3}{48} + \cdots$$

We may improve this approximation near $\theta = 0$ by using instead

$$F_2 = 3\sin H(2I + \cos H)^{-1}$$

This is a 5th order approximation since

$$\frac{3\sin\theta}{2 + \cos\theta} = \theta - \frac{\theta^5}{180} + \cdots$$

Seventh order accuracy can be obtained at the cost of only one inversion and one multiplication by using

$$F_3 = 18\sin H(10 + 5\cos H)^{-1} - \sin H(4I - \cos H)/15$$

This has the scalar expansion

$$\frac{18\sin\theta}{10 + 5\cos\theta} - \frac{\sin\theta(4I - \cos\theta)}{15} = \theta - \frac{\theta^7}{630} + \cdots$$

Unfortunately these approximations are only accurate if the eigenvalues of H are near zero. This can be ensured by taking several square roots of e^{iH} but we will not pursue this here.

A second option might be to try to control the error over $\theta \in (-\pi, \pi)$ in a least squares sense using products of $\sin\theta$ and $\cos\theta$. This leads in a natural way to the Fourier expansion of θ in terms of $\sin\theta, \sin 2\theta, \ldots$

$$\theta = 2\sin\theta - \sin 2\theta + \cdots + \frac{(-1)^{n+1}2}{n}\ \sin n\theta + \cdots$$

However as might be expected from the harmonic nature of the Fourier coefficients this series is too slowly convergent to be of computational use.

4.2 Numerical Experiments

The first numerical experiments were designed to test whether the method of selecting the number k of square root stages was optimal. We used various values of k for a group of matrices of sizes ranging from 5 to 500. The unexpected result of this series of tests was that in every case $k = 2$ was the optimal value! That is, for the THA method the most computationally efficient approach was to use just one square root approximation to get T_2 and then use a Padé approximation of $\arctan T_2$. It was gratifying to see that this was also the value of k selected by our procedure for estimating the computational effort. Apparently the Padé arctangent approximations are much more efficient than repeated square roots, at least when the eigenvalues of the matrix are in the interval $(-\pi/4, \pi/4)$ which is the case for $k \geq 2$. Table 1 reports the results of a test for a 100×100 unitary matrix. In this table, the number of matrix operations and inversions are summed and referred to as the number of matrix operations.

k	Number of Matrix Operations	Requested Error Tolerance	Actual Error
2	2	10^{-1}	1.7×10^{-2}
3	4	10^{-1}	1.2×10^{-3}
4	6	10^{-1}	3.3×10^{-3}
2	6	10^{-3}	3.8×10^{-4}
3	10	10^{-3}	4.0×10^{-5}
4	14	10^{-3}	1.9×10^{-5}
2	9	10^{-5}	9.3×10^{-8}
3	12	10^{-5}	2.5×10^{-8}
4	16	10^{-5}	1.4×10^{-6}

Table 1: Computational efficiency for various values of k.

A second set of tests compared the THA method with the quick approximations F_2 and F_3. These quick approximations work very well for problems where the eigenvalues of H are near zero but not so well when the eigenvalues approach $\pm\pi$. Table 2 illustrates this for an example of the form $H = \rho H_{nor}$ where H_{nor} has been normalized to have eigenvalues in the interval $(-1, 1)$ and ρ is a scaling factor that puts the eigenvalues of H into the interval $(-\rho, \rho)$. For this example the requested error tolerance for the THA method was 10^{-3}.

Eigenvalue Range of H	Error THA Approx.	Error F_2	Error F_3
$-\frac{\pi}{8}$ to $\frac{\pi}{8}$	3.3×10^{-6}	5.3×10^{-5}	2.3×10^{-6}
$-\frac{\pi}{4}$ to $\frac{\pi}{4}$	1.1×10^{-4}	1.8×10^{-3}	2.9×10^{-4}
$-\frac{\pi}{2}$ to $\frac{\pi}{2}$	1.4×10^{-4}	7.1×10^{-2}	3.8×10^{-2}
$-\frac{3\pi}{4}$ to $\frac{3\pi}{4}$	2.7×10^{-4}	7.2×10^{-1}	6.1×10^{-1}

Table 2: Comparing THA with quick approximations for H with eigenvalues in $(-\rho, \rho)$.

5 Conclusion

We have shown that for the class of unitary matrices the approach of Cheng et al. [4] can be extended via a tangent half-angle iteration. This method avoids complex arithmetic and utilizes the incomplete square root approximations of the

Denman-Beavers iteration; we have presented an analysis of the error that this induces in the half-angle sequence. From this analysis we are able to compute approximations of prescribed accuracy for the logarithm of a unitary matrix. The tangent half-angle approach has the advantage of only requiring the matrix operations of multiplication and inversion; as such it is amenable to implementation in a parallel computing environment.

References

[1] George A. Baker, Jr., *Essentials of Padé Approximants*, Academic Press, New York, 1975.

[2] Richard J. Bagby, "A Convergence of Limits," *Mathematics Magazine*, 71(4), pp. 270–277, 1998.

[3] Åke Björck and Sven Hammarling, "A Schur Method for the Square Root of a Matrix," *Lin. Alg. Appl.*, 52/53 pp. 127–140, 1983.

[4] Sheung Hun Cheng, Nicholas J. Higham, Charles S. Kenney, and Alan J. Laub, "Approximating the Logarithm of a Matrix to Specified Accuracy," Numerical Analysis Report No. 353, Manchester Centre for Computational Mathematics, Manchester, England, November 1999, 14 pages.

[5] Eugene D. Denman and Alex N. Beavers, Jr., "The Matrix Sign Function and Computations in Systems," *Appl. Math. and Comput.*, 2, pp. 63–94, 1976.

[6] Roger A. Horn and Charles R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[7] Nicholas J. Higham, "Computing the Polar Decomposition – with Applications," *SIAM J. Sci. Statist. Comput.*, 7(4), pp. 1160–1174, 1986.

[8] Nicholas J. Higham, "Computing Real Square Roots of a Real Matrix," *Lin. Alg. Appl.*, 88/89, pp. 405–430, 1986.

[9] N.J. Higham and R.S. Schreiber, "Fast Polar Decomposition of an Arbitrary Matrix," *SIAM J. Sci. Statist. Comput.*, 11(4), pp. 648–655, 1990.

[10] Nicholas J. Higham, "The Matrix Sign Decomposition and Its Relation to the Polar Decomposition," *Lin. Alg. Appl.*, 212/213, pp. 3–20, 1994.

[11] N.J. Higham and P. Papadimitriou, "A Parallel Algorithm for Computing the Polar Decomposition," *Parallel Comput.*, 20(8), pp. 1161–1173, 1994.

[12] Nicholas J. Higham, "Stable Iterations for the Matrix Square Root," *Numerical Algorithms*, 15(2), pp. 227–242, 1997.

[13] Nicholas J. Higham, "Evaluating Padé Approximations of the Matrix Logarithm," unpublished manuscript, Feb., 2000.

[14] C.S. Kenney and A.J. Laub, "Padé Error Estimates for the Logarithm of a Matrix," *Int. J. Control*, 50(3), pp. 707–730, 1989.

[15] C.S. Kenney and A.J. Laub, "Condition Estimates for Matrix Functions," *SIAM J. Matrix Anal. Appl.*, 10(2), pp. 191–209, 1989.