AD-A093 565 WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
EVALUATION OF COMPLEX LOGARITHMS AND RELATED FUNCTIONS.(U)
AUG 80 G J MIEL DAAG29-80-C-0041
UNCLASSIFIED MRC-TSR-2104 NL

1 of 1
AD
A093565

END
DATE
FILMED
2-81
DTIC

MRC Technical Summary Report #2104

EVALUATION OF COMPLEX LOGARITHMS AND RELATED FUNCTIONS

George J. Miel

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53706

August 1980

(Received July 25, 1980)

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

EVALUATION OF COMPLEX LOGARITHMS AND RELATED FUNCTIONS

George J. Miel[†]

Technical Summary Report #2104
August 1980

ABSTRACT

An algorithm is presented for computing ln z with complex arithmetic, by extending to the complex plane Carlson's treatment of a classical iteration using arithmetic and geometric means. Although not competitive with current techniques which handle the real and imaginary parts separately, the algorithm may be useful in special purpose applications. A detailed analysis of convergence, scaling, and roundoff is given. Standard identities and some minor bookkeeping allow the evaluation of inverse circular and inverse hyperbolic functions. It is also shown that the basic procedure is related to certain real algorithms.

AMS (MOS) Subject Classifications: 65D20, 30-04, 65B05

Key Words: Complex logarithms, Inverse circular functions, Inverse hyperbolic functions, Complex arithmetic, Acceleration of convergence

Work Unit Number 3 (Numerical Analysis and Computer Science)

[†] Department of Mathematical Sciences, University of Nevada, Las Vegas, NV [illegible].

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

SIGNIFICANCE AND EXPLANATION

Complex logarithms are usually computed with real arithmetic,

$$\ln(x + iy) = \frac{1}{2} \ln(x^2 + y^2) + i \tan^{-1}(y/x) ,$$

using the real $\ln$ and real $\tan^{-1}$ functions available in high level programming languages. This report studies a novel approach, based on complex arithmetic, obtained by extending to the complex plane a known real algorithm. The strategy consists of a basic iteration, which uses one complex square root per cycle, accelerated by Richardson extrapolation, a speeding-up process well-known to numerical analysts. With proper scaling of the argument, the accelerated procedure requires four complex square roots for accuracy to 10 decimals (compared to eighteen complex square roots in the non-accelerated case). Standard identities such as

$$\cosh^{-1} z = \ln(z + \sqrt{z^2 - 1}) ,$$

$$\cos^{-1} z = i \cosh^{-1} z ,$$

etc.

and some minor bookkeeping to account for principal branches, allow the evaluation of complex inverse trigonometric functions. The report analyzes the algorithm with respect to scaling, convergence, and stability, and it relates the algorithm to other procedures, including classical methods for calculating the constant π. Since complex square roots are done at the software level, and thus costly in machine time, the algorithm is not competitive with standard methods handling the real and imaginary parts separately with real arithmetic. However, its simplicity and stability could make it attractive for implementation in microcode or read only memory in special purpose applications requiring extensive usage of elementary complex functions.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

EVALUATION OF COMPLEX LOGARITHMS AND RELATED FUNCTIONS

George J. Miel[†]

1. Introduction. Logarithms of complex numbers are commonly computed using real arithmetic separately for the real and imaginary parts,

$$\ln(x + iy) = \frac{1}{2}\ln(x^2 + y^2) + i\tan^{-1}(y/x) , \qquad (1.1)$$

with suitable precautions to avoid numerical problems; see, e.g., [6, Algorithm 243]. We analyze an algorithm based on complex arithmetic, obtained by extending to the complex case Carlson's procedure [3]. The strategy consists of a basic iteration, which uses one complex square root per cycle, accelerated by Richardson extrapolation. The basic iteration generates one of the sequences of Borchardt's algorithm [2, p. 499], [4, p. 170]. For real arguments, this iteration is also related to Thacher's algorithm for inverse cosines [11], to Viete's infinite product for π [8, p. 26], and to the method of equal perimeters [8, p. 32]. As for the real case, the improvement due to extrapolation is substantial, the algorithm is reliable and stable, and storage needs are modest as there are no constants to be saved. Numerical experiments indicate no serious cancellation leading to loss of significant figures, as sometimes happens when a real algorithm is extended to the complex case. With an adequate reduction in the range of the independent variable, the accelerated procedure requires four complex square roots for 10D accuracy. Standard identities, and some bookkeeping to account for principal branches, allow the evaluation of inverse circular and inverse hyperbolic functions. The complex arithmetic

[†]Department of Mathematical Sciences, University of Nevada, Las Vegas, NV 89154.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

provides a unified design for a simple and modular software package. Unfortunately, the complex square roots preclude the algorithm from competition with the straightforward approach (1.1), which takes advantage of efficient real elementary functions provided in high level languages. However, the algorithm might be of interest in special purpose applications implemented in microcode or read only memory. Certain monotonicity properties of Borchardt's iteration can be exploited for computation in complex interval arithmetic [10].

2. The Basic Algorithm. The goal is to evaluate the single-valued function $\ln z$ whose range is $\{x + iy \mid -\pi \le y < \pi\}$. The rational operations of complex arithmetic and the principal complex square root are assumed available [6, Key A2]. The latter is the single-valued function whose range is

$$\mathbb{C}^{+} = \{x + iy \mid x > 0 \text{ or } x=0 \text{ with } y \ge 0\} .$$

Recall that

$$\sqrt{z^2} = z, \text{ if } z \in \mathbb{C}^{+} . \qquad (2.1)$$

We will use the function $F(z) = (z \coth z)/w$ and its expansion

$$F(z) = a_o + a_1 z^2 + a_2 z^4 + \dots + a_m z^{2m} + O(z^{2m+2}), \qquad (2.2)$$

where a_j depends on the Bernoulli number B_{2j},

$$a_j = \frac{2^{2j} B_{2j}}{(2j)!\, w} \quad ,$$

and where Landau's notation $f(z) = O(g(z))$ is adequately defined [7, p. 156]. In what follows, $D = \{z \mid |z| > 1\}$.

Theorem. Let $z \in D$ and

$$\xi_o = \frac{z+1}{z-1}, \quad \xi_{n+1} = \xi_n + \sqrt{\xi_n^2 - 1}, \quad n \ge 0. \qquad (2.3)$$

The numbers ξ_n satisfy

$$\xi_n = \coth 2^{-n-1} w, \qquad \xi_{n+1} = 2\xi_n + \varepsilon_n, \qquad (2.4)$$

where $w = \ln z$, $\varepsilon_n = -1/\xi_{n+1}$, $|\varepsilon_n| << |\xi_n|$, and the numbers $u_n = 2^{-n-1}\xi_n$ converge to $u = 1/w$ with a rate

$$u - u_{n+1} = 4^{-1}(u - u_n) + O(16^{-n-1}). \qquad (2.5)$$

Proof. For $n = 0$, $\xi_o = (e^w + 1)/(e^w - 1) = \coth w/2$. The relation $\zeta = a + ib$, $\operatorname{csch} \zeta = |\sinh \zeta|^{-2}(\sinh a \cos b - i \cosh a \sin b)$ implies that $\operatorname{csch} w/2 \in \mathbb{C}^+$ whenever $w \in \ln(D)$. Use the identities $\coth \zeta/2 = \coth \zeta + \operatorname{csch} \zeta$, $\operatorname{csch}^2\zeta = \coth^2\zeta - 1$ and (2.1) to get

$$\coth w/4 = \coth w/2 + \sqrt{\coth^2 w/2 - 1}, \qquad w \in \ln(D).$$

The relation (2.3) gives $\xi_{n+1} = 2\xi_n - 1/\xi_{n+1}$. The convergence follows from $u_n = F(2^{-n-1}w)$ and $\lim_{\zeta \to 0} F(\zeta) = 1/w$. Finally, use (2.2) with $m = 1$ to get (2.5).

The second expression in (2.4) shows that as n increases, ξ_{n+1} gets increasingly close to $2\xi_n$. This fact provides a simple variable precision scheme. A range reduction allows the evaluation of logarithms for machine representable arguments.

Basic Algorithm. Adequate precision is assumed available. Given $z' \neq 0$, proceed as follows to find $\ln z'$ correct to d decimals:

1. Factorize $|z'| = 2^r x$, $x \in [2,4)$. Let $z = 2^{-r}z'$.
2. Compute (2.3) with $n = 0, 1, \ldots, N$ where N is such that ξ_N and $2\xi_{N-1}$ agree to $d + 1$ decimals.
3. Let $\ln z' = (2^{-N-1}\xi_N)^{-1} + r \ln 2$.

Various range reductions are possible. Instead of the modulus, one can use $|a + ib| = |a| + |b|$ or $|a + ib| = \max(|a|, |b|)$. The code should take advantage of the multiplications by powers of 2.

Roundoff Propagation. Numerical experiments indicate that the algorithm is remarkably stable. A simplified analysis shows why. Consider

$$\hat{\xi}_o = \xi_o - \varepsilon_o, \qquad \hat{\xi}_{n+1} = \phi(\hat{\xi}_n) - \delta_n, \qquad \hat{u}_n = 2^{-n-1}\hat{\xi}_n,$$

where $\phi(z) = z + \sqrt{z^2 - 1}$ and δ_n reflects the accuracy of the complex square root routine. Letting $\varepsilon_n = \xi_n - \hat{\xi}_n$,

$$\varepsilon_{n+1} \approx \phi'(\xi_n)\varepsilon_n + \delta_n, \qquad \phi'(\xi_n) = \xi_{n+1}/(\xi_{n+1} - \xi_n).$$

Since $\xi_{n+1} \approx 2\xi_n$, assume for simplicity that $\delta_n \approx 2\delta_{n-1}$, $\phi'(\xi_n) \approx 2$. Then $\varepsilon_N \approx N2^{N-1}\delta_o + 2^N\varepsilon_o$, and assuming no error in the multiplication by 2^{-N-1}, we get

$$u_N - \hat{u}_N \approx \frac{N}{4}\delta_o + \frac{\varepsilon_o}{2}. \tag{2.6}$$

The accumulated roundoff is acceptable if the square root routine is accurate to at least d + 2 decimals. Practice shows that cancellation causes actual roundoff to be smaller than (2.6).

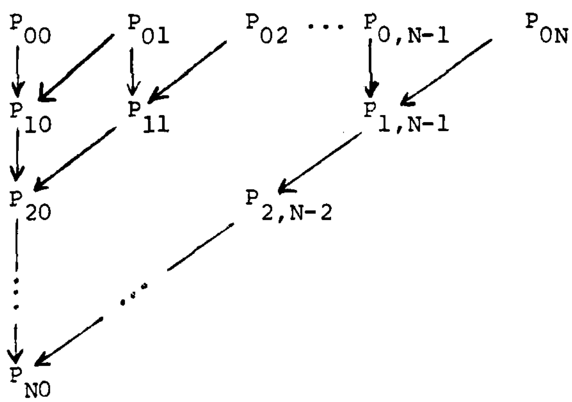
3. Richardson Extrapolation. Section 5 shows that $\{u_n\}$ is one of the sequences generated by Borchardt's algorithm [2], [3], [4, p. 170]. As for the real case, (2.5) indicates that successive errors in u_n are ultimately reduced by a factor 1/4. In order to speed-up convergence, we extend to the complex case the treatment of Carlson [3].

The procedure is given by

$$P_{on} = u_n, \qquad 0 \le n \le N, \tag{3.1}$$

$$P_{kn} = P_{k-1,n+1} + \frac{P_{k-1,n+1} - P_{k-1,n}}{4^k - 1}, \quad \begin{matrix} 1 \le k \le N \\ 0 \le n \le N-k \end{matrix} \quad . \tag{3.2}$$

The scheme generates a triangular array,



(3.3)

in which the arrows illustrate the dependence of P_{kn} on $P_{k-1,n}$ and $P_{k-1,n+1}$.

Recall that for $z \in D$, $\quad u_n = F(\zeta_n) \to u = 1/w, \quad \zeta_n = w/2^{n+1}$.

Let

$$F_0(\zeta) = F(\zeta), \qquad F_k(\zeta) = \frac{4^k F_{k-1}(\zeta) - F_{k-1}(2\zeta)}{4^k - 1} \quad .$$

Then (2.2) implies that

$$F_k(\zeta) = u + b_k \zeta^{2k+2} + O(\zeta^{2k+4}),$$

where b_k is a constant. Since $\zeta_n = 2\,\zeta_{n+1}$, we have $P_{kn} = F_k(\zeta_{n+k})$. Consequently,

$$P_{kn} - u = 4^{-(k+1)(k+n+1)} b_k w^{2k+2} + O(4^{-(k+2)(k+n+1)}).$$

Thus, the errors in successive elements of the k-th row of (3.3) are each time roughly reduced by a factor $4^{-(k+1)}$.

Error Bounds via Interpolation Theory. Let $f(z) = F(\sqrt{z}) = (\sqrt{z}\coth\sqrt{z})/w$ and $z_n = \zeta_n^2 = w^2/4^{n+1}$; (3.2) is the Neville scheme for evaluating polynomials with

$$P_{kn} = \text{value at } z{=}0 \text{ of the polynomial of degree} \le k$$
$$\text{which interpolates } f(z) \text{ at } z_n,\ z_{n+1},\ \ldots,\ z_{n+k};$$

see Brezinski [1, p.26]. The function f(z) is analytic for $|z| < \pi^2$.

Theorem. If $u = 1/w$, $w = \ln z$, and

$$1 < |z| \le e^{\sqrt{7\pi^2 - 18}/3} \tag{3.4}$$

then

$$|u - P_{kn}| < 2^{-(k+1)(k+2n+1)} |w|^{2k+1}\left(\frac{|w|^2}{2} + \pi\right). \tag{3.5}$$

Proof. Let $a = \frac{2\pi}{3}$ and let C" denote the square with vertices $a(\pm 1 \pm i)$. We first show that

$$\max_{z \in C''} |\coth z| \le \coth a. \tag{3.6}$$

For the side $z = a + iy$, $-a \le y \le a$, we get

$$|\coth z| \le \frac{|e^{a+iy}| + |e^{-a-iy}|}{\big||e^{a+iy}| - |e^{-a-iy}|\big|} = \frac{e^a + e^{-a}}{e^a - e^{-a}} = \coth a.$$

For $z = x + ia$, $-a \le x \le a$, use $e^{ia} = \frac{1}{2}(-1 + i\sqrt{3})$ to obtain

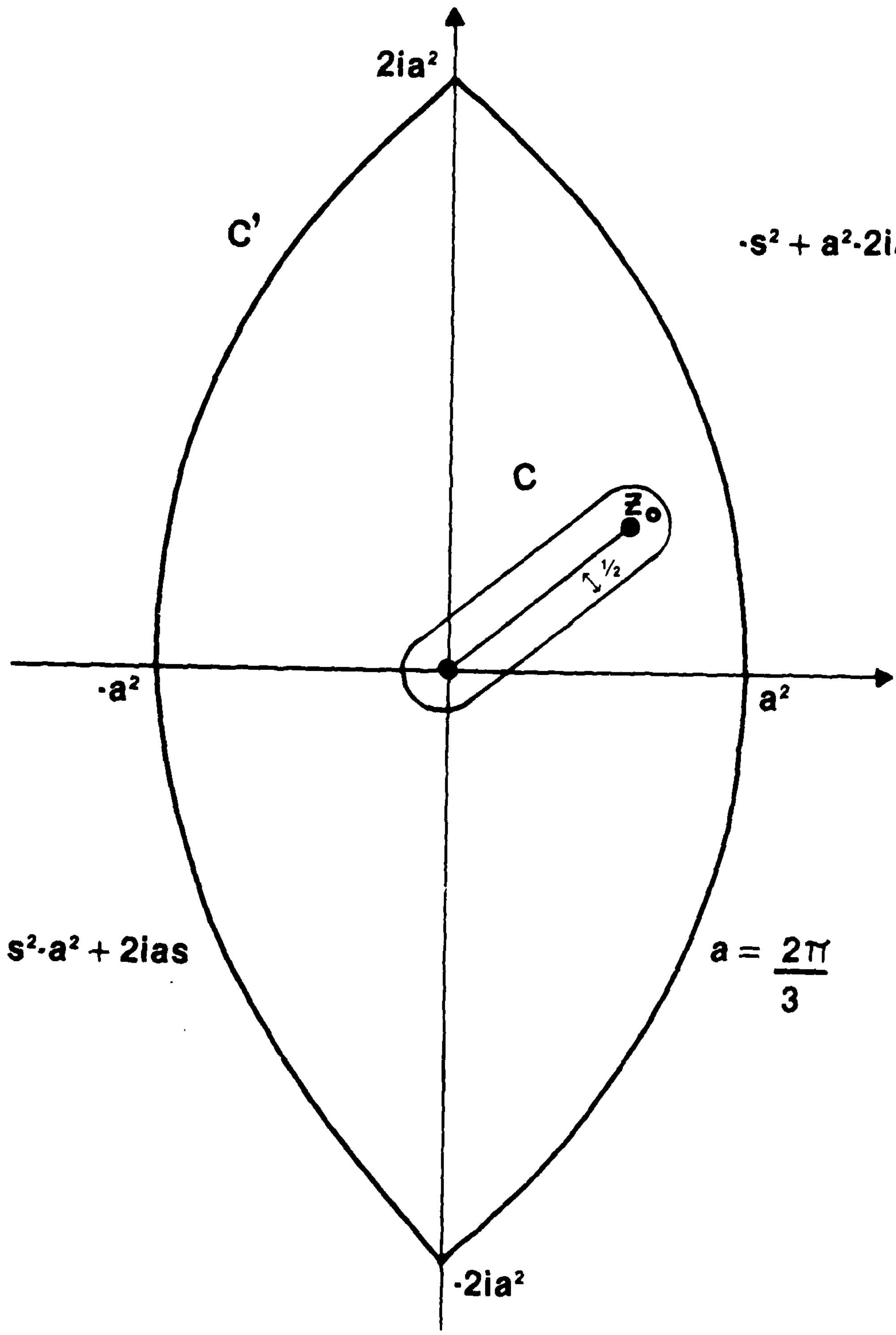


FIG. Estimate $\max|\sqrt{t}\coth\sqrt{t}|$ for $t \in C'$ rather than $t \in C$ using that $\sqrt{t}$ is then on the square C''.

$$|\coth z| = \left|\frac{1 - i\sqrt{3}\tanh x}{\tanh x - i\sqrt{3}}\right| = \left(\frac{3\tanh x + 1}{\tanh^2 x + 3}\right)^{1/2} \equiv \psi(x).$$

Similarly for the other two sides. The function $\psi(x)$ reaches its maximum at $x=\pm a$. Since $\psi(\pm a) < \coth a$, we get (3.6).

Now, let C denote the union of two semicircles of radius 1/2 and centers at 0 and $z_0 = \frac{w^2}{4}$ respectively, and two parallel segments as shown in the figure. Consider also the union of two partial parabolas,

$$C' = \{\pm(s^2 - a^2 + 2ias) \mid -a \le s \le a\}.$$

The curve C' is inside the circle of analyticity and (3.4) implies that C is not outside C'.

By a classic result,

$$u - P_{kn} = \frac{(-1)^{k+1}}{2\pi i} \cdot \frac{z_n z_{n+1} \cdots z_{n+k}}{w} \cdot I, \qquad (3.7)$$

where

$$I = \int_C \frac{g(t)\, dt}{t(t-z_n)\cdots(t-z_{n+k})}, \quad g(t) = \sqrt{t}\coth\sqrt{t}.$$

But

$$|I| \le 2^{k+2}(2|z_0| + \pi)\, M, \qquad (3.8)$$

where $M \ge \max_{t\in C} |g(t)|$. We have

$$\max_{t\in C} |g(t)| \le \max_{t\in C'} |g(t)| = \max_{z\in C''} |g(z^2)| \le \sqrt{2}\, a \coth a \approx 0.972\,\pi,$$

where we used (3.6). To complete the proof, take $M = \pi$ and use (3.7) and (3.8).

For the range reduction described earlier, (3.4) is satisfied and $|w| < |\ln 4 + i\pi|$. For accuracy to d decimals, set the bound in (3.5) no greater than $\frac{1}{2}10^{-d}$ and thus obtain:

$$\text{Non-accelerated case } (k=0, n=N) \qquad N \geq 1.66\, d + 2.48,$$

$$\text{Accelerated case } (k=n=N) \qquad N \geq \sqrt{1.11\, d + 1.66} - 0.07.$$

Actual computation shows that 10D accuracy requires $N = 18$ in the non-accelerated case and $N = 4$ in the extrapolated case. The acceleration defined by (3.1) and (3.2) is easily coded. The code should take advantage of the multiplications by powers of 4. Variable precision is possible by a priori specification of N or by on-line comparison of P_{N0} and $P_{N-1,1}$.

Roundoff Propagation. If the values in the first row of (3.3) are contaminated with errors whose magnitudes are less than ε, then the errors later in the extrapolation have magnitude which nowhere exceed 2ε. Combining this with (2.6), we get

$$|P_{N0} - \hat{P}_{N0}| \lesssim \left|\frac{N}{2}\delta_0 + \varepsilon_0\right| + |\zeta|,$$

where $\hat{P}_{N0}$ is the computed value of P_{N0} and ζ is the accumulated roundoff in the computation of (3.2). The major term is $\frac{N}{2}\delta_0$ and the extrapolation is well-conditioned provided that the square root routine is of good quality.

4. Inverse Trigonometric Functions. The table below specifies for each function a bijection between the domain and given range.

FUNCTION	RANGE
ln	$R_1 = \{a+ib \mid -\pi \le b < \pi\}$
$\cosh^{-1}$	$R_2 = \{a+ib \mid -\pi < b < 0 \text{ or } b = -\pi, 0 \text{ with } a \ge 0\}$
$\sinh^{-1}$	$R_3 = \{a+ib \mid -\pi/2 < b < \pi/2 \text{ or } b = \pm\pi/2 \text{ with } a \ge 0\}$
$\tanh^{-1}$	$R_4 = \frac{1}{2} R_1 - \{-i\pi/2\}$
$\cos^{-1}$	iR_2
$\sin^{-1}$	iR_3
$\tan^{-1}$	$-iR_4$

The following procedures use standard identities to evaluate the functions.

Algorithm for $w = \cosh^{-1} z$.

$w' = \ln(z+\sqrt{z^2-1}) = a+ib$.

If $b>0$ then $w=-w'$.

If $b=-\pi$, 0 and $a<0$ then $w=-a+ib$.

Otherwise, $w=w'$.

Algorithm for $w = \sinh^{-1} z$.

$w' = \ln(z+\sqrt{z^2+1}) = a+ib$.

If $b \in [-\pi,-\pi/2)$ then $w=-i\pi-w'$.

If $b \in (\pi/2,\pi)$ then $w=i\pi-w'$.

If $b=\pm\pi/2$ and $a<0$ then $w=-a+ib$.

Otherwise, $w=w'$.

Algorithm for $w = \cos^{-1} z$.

$w = i \cosh^{-1} z$.

Algorithm for $w = \sin^{-1} z$.

$w = i \sinh^{-1}(-iz)$.

Algorithm for $w = \tanh^{-1} z$.

$z \ne \pm 1$, $w=\frac{1}{2} \ln(\frac{1+z}{1-z})$.

Algorithm for $w = \tan^{-1} z$.

$z \ne \pm i$, $w=-i \tanh^{-1} iz$.

In the case of $\cosh^{-1}$ and $\sinh^{-1}$, some logic is needed in order to choose the value in the specified range, since these functions are double-valued in the range of ln, [9, p. 417].

5. Connection with Real Algorithms. For real arguments, our accelerated procedure is exactly equivalent to Carlson's treatment of Borchardt's algorithm [3]. The non-accelerated procedure is related to Thacher's algorithm [11] for real inverse cosines and to classical methods for calculating π.

Borchardt's Algorithm. If $u_{-1}, v_{-1} > 0$ then the sequences

$$u_{n+1} = \frac{1}{2}(u_n+v_n), \qquad v_{n+1} = \sqrt{u_{n+1}v_n}, \qquad n \geq -1, \tag{5.1}$$

converge monotonically to a common limit $B(u_{-1},v_{-1})$, [2]. We show that

$$u_n = 2^{-n-1}\xi_n, \qquad \xi_{n+1} = \xi_n + \sqrt{\xi_n^2 - u_{-1}^2 + v_{-1}^2} \ . \tag{5.2}$$

Get the invariant $4^{n+1}(u_{n+1}^2 - v_{n+1}^2) = c$ and then substitute $v_n = (u_n^2 - 4^{-n}c)^{1/2}$ in the first relation of (5.1). Our basic algorithm generates (5.2) corresponding to $B\left(\frac{z^2+1}{z^2-1}, \frac{2z}{z^2-1}\right) = \frac{1}{\ln z}$.

Thacher's Algorithm. If $R_1 = \sqrt{2z+2}$, $R_{n+1} = \sqrt{R_n+2}$ then $t_n = 2^n\sqrt{|R_n-2|}$ converge to $\cos^{-1}z$ if $|z| < 1$ and to $\cosh^{-1}z$ if $|z| \geq 1$, [11]. It turns out that t_n is the reciprocal of v_n generated by (5.1) with proper u_{-1} and v_{-1}.

Method of Equal Perimeters. If $\mu > 2$ then

$$B\left(\frac{1}{\mu}\cot\frac{\pi}{\mu}, \quad \frac{1}{\mu}\csc\frac{\pi}{\mu}\right) = \frac{1}{\pi},$$

$$u_n = \frac{1}{2^{n+1}\mu}\cot\frac{\pi}{2^{n+1}\mu} = \text{radius of inscribed circle in a regular } 2^{n+1}\mu\text{ - gon of perimeter 2.}$$

Descartes worked with $\mu = 4$, [8, p.32].

Viete's Infinite Product. If $g_{-1} = \sqrt{2}$, $g_0 = \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}$, $g_n = \sqrt{\frac{1}{2} + \frac{1}{2} g_{n-1}}$ then $\lim(g_{-1}g_0 \cdots g_n) = 4/\pi$, [8, p.26]. We have $g_{-1}g_0 \cdots g_n = v_n$, where v_n corresponds to $B(1, \sqrt{2}) = 1/\sin^{-1}\sqrt{\frac{1}{2}}$.

REFERENCES

[1] C. BREZINSKI, Accélération de la Convergence en Analyse Numérique, Lecture Notes in Mathematics, Vol. 584, Springer-Verlag, New York, 1977.

[2] B.C. CARLSON, Algorithms involving arithmetic and geometric means, Amer. Math. Monthly, 78(1971), pp. 496-505.

[3] B.C. CARLSON, An algorithm for computing logarithms and arctangents, Math. Comp., 26(1972), pp. 543-549.

[4] B.C. CARLSON, Special Functions of Applied Mathematics, Academic Press, New York, 1977.

[5] B.C. CARLSON, Computing elliptic integrals by duplication, Numer. Math., 33(1979), pp. 1-16.

[6] Collected Algorithms from ACM, ACM Algorithms Distribution Service, % IMSL, Inc., 7500 Bellaire Blvd., Houston, TX 77036.

[7] J. DIEUDONNÉ, Infinitesimal Calculus, Houghton Mifflin, Boston, 1971.

[8] E.W. HOBSON, Squaring the Circle, a History of the Problem, Chelsea, New York, 1953.

[9] K. KNOPP, Theory and Application of Infinite Series, second English edition, Blackie & Son, London and Glasgow, 1951.

[10] G.J. MIEL, Evaluation of complex logarithms and related functions with interval arithmetic, Proceedings of the International Symposium on Interval Mathematics, held May 27-31, 1980, in Freiburg, W. Germany, Academic Press, New York, (to appear).

[11] H.C. THACHER, JR., Iterated square root expansions for the inverse cosine and inverse hyperbolic cosine, Math. Comp., 15(1961), pp. 399-403.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2104
2. GOVT ACCESSION NO.	AD-A093565
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	EVALUATION OF COMPLEX LOGARITHMS AND RELATED FUNCTIONS.
5. TYPE OF REPORT & PERIOD COVERED	Summary Report - no specific reporting period
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	George J. Miel
8. CONTRACT OR GRANT NUMBER(s)	DAAG29-80-C-0041
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Mathematics Research Center, University of Wisconsin, 610 Walnut Street, Madison, Wisconsin 53706
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	Work Unit Number 3 (Numerical Analysis & Computer Science)
11. CONTROLLING OFFICE NAME AND ADDRESS	U. S. Army Research Office, P.O. Box 12211, Research Triangle Park, North Carolina 27709
12. REPORT DATE	August 1980
13. NUMBER OF PAGES	14
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Complex logarithms
Inverse circular functions
Inverse hyperbolic functions
Complex arithmetic
Acceleration of convergence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm is presented for computing $\ln z$ with complex arithmetic, by extending to the complex plane Carlson's treatment of a classical iteration using arithmetic and geometric means. Although not competitive with current techniques which handle the real and imaginary parts separately, the algorithm may be useful in special purpose applications. A detailed analysis of convergence, scaling, and roundoff is given. Standard identities and some minor bookkeeping allow the evaluation of inverse circular and inverse hyperbolic functions. It is also shown that the basic procedure is related to certain real algorithms.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)