

much larger, without any increase in the truncation error, when the logarithm rather than the natural value of the function is tabulated.

LUIGI JACCHIA

Harvard College Observatory
Cambridge, Massachusetts

* Work performed under Contract DA 19-020-ORD-2556 with the Office of Ordnance Research, U. S. Army.

Note on a Logarithm Algorithm

In a recent note,¹ D. SHANKS developed a well-known algorithm for the computation of logarithms,² in a way particularly suitable for use on automatic computing machines. In what follows I should like to point out that, if we use the value of $\mu = \ln a_0$ (the natural logarithm of a_0), the number of operations necessary for this computation can be cut down considerably in replacing about a third of the single steps indicated by Shanks by one division and one addition.

We assume as in the paper quoted that $a_0 > a_1 > 1$. To compute $\lambda = \log_{a_0} a_1$, we determine a sequence of numbers a_2, a_3, a_4, \dots , ($a_i > 1$) and a sequence of positive integers n_1, n_2, \dots , by the relations

$$a_i^{n_i} \leq a_{i-1} < a_i^{n_i+1}, \quad a_{i+1} = a_{i-1}/a_i^{n_i} \quad (i = 1, 2, \dots).$$

We then have

$$\lambda = \frac{1}{n_1 +} \frac{1}{n_2 +} \dots;$$

if we stop at the calculation of n_i we have an approximate value of λ by taking the i th convergent:

$$\mu_i = \frac{P_i}{Q_i} = \frac{1}{n_1 +} \frac{1}{n_2 +} \dots \frac{1}{n_i} = \lambda - \eta_i.$$

We will show that we have for η_i the formula

$$(I) \quad \eta_i = \eta_i^* + \rho_i, \quad \eta_i^* = (-1)^i \frac{a_{i+1} - 1}{\mu Q_i},$$

where the *error term* ρ_i can be estimated by

$$(II) \quad |\rho_i| \leq \mu Q_i \eta_i^2 \leq \mu |\eta_i|^{3/2} \leq \mu/Q_i^3$$

as soon as we have

$$(III) \quad \mu/Q_i \leq 1.7933 \dots$$

For $a_0 = 10$ we have $\mu = \mu_{10} \doteq 2.3026$ and (III) is certainly satisfied from $i = 2$ on. For $a_0 = 2$ we have

$$\mu = \mu_2 \doteq 0.6931.$$

Thus if we want, e.g., to compute $\log_2 a_1$ with an error less than 10^{-9} it is sufficient to stop with the computation of n_i as soon as Q_i exceeds 10^3 . In determining n_i by successive division we will already have the value of $a_{i+1} = a_{i-1}/a_i^{n_i}$ and then apply (I).³

In order to prove (I)–(III), observe that we have for all positive integers

$$(1) \quad a_{2\nu+1} = \frac{a_1^{Q_{2\nu}}}{a_0^{P_{2\nu}}}, \quad a_{2\nu} = \frac{a_0^{P_{2\nu}-1}}{a_1^{Q_{2\nu}-1}}.$$

Indeed, since $P_1 = 1, Q_1 = n_1, P_2 = n_2, Q_2 = n_1 n_2 + 1$, our formulae are verified immediately for

$$a_2 = \frac{a_0}{a_1^{n_1}}, \quad a_3 = \frac{a_1}{a_2^{n_2}} = \frac{a_1^{n_1 n_2 + 1}}{a_0^{n_2}}.$$

From there on we proceed by induction, using the well-known relations $P_i = n_i P_{i-1} + P_{i-2}, Q_i = n_i Q_{i-1} + Q_{i-2}$.

If we write now $\epsilon = (-1)^i$ we have from (1), for an even $i = 2\nu$:

$$a_{i+1} = e^{\mu \lambda Q_i - \mu P_i} = e^{\mu Q_i (\lambda - \mu \epsilon)},$$

that is,

$$(2) \quad a_{i+1} = e^{\epsilon \mu Q_i \eta_i};$$

this formula also holds in the case of an odd i , where we have $\epsilon = -1$.

From (2) we have now

$$(3) \quad a_{i+1} - 1 = \epsilon \mu Q_i \eta_i + \theta (\mu Q_i \eta_i)^2,$$

where θ is the value of the series

$$\frac{e^\gamma - 1 - \gamma}{\gamma^2} = \sum_{\nu=0}^{\infty} \frac{\gamma^\nu}{(\nu+2)!}, \quad \gamma = \epsilon \mu Q_i \eta_i.$$

Therefore we have certainly $|\theta| \leq 1$, if $|\gamma|$ does not exceed the positive root γ_0 of the equation

$$e^\gamma = 1 + \gamma + \gamma^2;$$

we find

$$\gamma_0 \doteq 1.7933.$$

Assuming now that we have

$$(4) \quad \mu Q_i |\eta_i| \leq \gamma_0,$$

we obtain (I), with $\rho_i = -\theta (\mu Q_i \eta_i)^2$, on dividing (3) by $\epsilon \mu Q_i$. The estimates (II) follow then immediately as we have $|\eta_i| \leq 1/Q_i^2, |\theta| \leq 1$. On the other hand, if (III) is satisfied, so is (4), since $Q_i |\eta_i| \leq 1/Q_i$.

We give for $a_0 = 10, a_1 = 2$ the exact values of the η_i and the values η_i^* computed by (I), in the last two columns of the following table for $i = 1, \dots, 7$. The three first columns of this table contain the corresponding values of Q_i, a_i and μ_i . Although the values of the a_i are given in the same example treated by

Shanks with 9 decimals for $i = 1, \dots, 6$, we have recomputed them and computed further a_7 and a_8 on the 10×10 Friden Desk Calculator without double precision or any similar artifice.

In computing $P_i/Q_i + \eta_i^*$ it is better not to compute the second term separately from (I) but to use the formula

$$P_i/Q_i + \eta_i^* = \left[P_i + (-1)^i \frac{a_{i+1} - 1}{\mu} \right] / Q_i,$$

in replacing in this way two divisions by one only.

i	Q_i	a_i	μ_i	η_i^*	η_i
1	3	2	.33333 33333 333	- 3.619 $\times 10^{-2}$	- 3.2 $\times 10^{-2}$
2	10	1.25	.30000 00000 000	1.042 $\times 10^{-3}$	1.03 $\times 10^{-3}$
3	93	1.024	.30107 52688 1720	- 4.549 $\times 10^{-5}$	- 4.527 $\times 10^{-5}$
4	196	1.00974 1958	.30102 04081 6327	9.6083 $\times 10^{-6}$	9.5875 $\times 10^{-6}$
5	485	1.00433 6278	.30103 09278 3505	- 9.32654 $\times 10^{-7}$	- 9.32171 $\times 10^{-7}$
6	2136	1.00104 1546	.30102 99625 468	3.312105 $\times 10^{-8}$	3.311717 $\times 10^{-8}$
7	13301	1.00016 2900	.30102 99977 4453	- 2.081453 $\times 10^{-9}$	- 2.0805493 $\times 10^{-9}$
8		1.00006 3748			

We see from this table that, while the error of μ_6 is 3.3×10^{-8} , the improved method gives already $\mu_4 + \eta_4^*$ with the error 2.1×10^{-8} . On the other hand in using all computations which give μ_6 we obtain at once a_7 and from there on obtain $\mu_6 + \eta_6^*$ by (I); this is in error by about 3.2×10^{-12} . If we go on to $i = 7$ the error of μ_7 is 2.2×10^{-9} , while since we obtain at once a_8 , we get from (I) the approximation $\mu_7 + \eta_7^*$ the error of which is about 9.0×10^{-14} . Here, however, the result is already influenced by the rounding off error in our value of a_8 . If we use instead the value of a_8 obtained by double precision: 1.00006 37223 565, we obtain for η_7^* the improved value $\bar{\eta}_7^* = - 2.0806156 \times 10^{-9}$ and the error of $\mu_7 + \bar{\eta}_7^*$ is then 6.6×10^{-14} . We see that it is hardly worth-while to go over to double precision unless we need essentially more than 10 decimals.

As the essential steps in the original method consist in the successive determination of *integers*, it is clear that this method is not very sensitive with respect to the rounding off. It appears a little unexpected that the same is true to a very great extent for the improvement of this method proposed here. It turns out that the denominators μQ_i in the formula (I) are sufficiently great to counterbalance to a certain extent the rounding off errors in the a_i .

It may be finally remarked that from a certain point of view the conclusions of D. Shanks concerning the linear character of convergence are perhaps slightly too optimistic. As a matter of fact the amount of work will not be measured by the number of "cycles," i.e., the number of partial quotients of the continued fraction used, but the total number of divisions; and this will be measured not by the number of the n_i but by their sum $n_1 + n_2 + \dots + n_i$. About this sum, however, KHINTCHINE has proved⁴ that its order "in the average" is not that of i but that of $i \lg i$. The "linearity of convergence" will probably fall off in the long run in the "general case."⁵

A. M. OSTROWSKI

Mathematical Institute of the University
Basle, Switzerland

¹ DANIEL SHANKS, "A Logarithm Algorithm," *MTAC*, v. 8, 1954, p. 60-64.

² See, e.g., J. TROPFKE, *Geschichte der Elementar-Mathematik*. Bd. II: 3. Aufl., 1933, p. 241-242.

³ In applying (I) we can of course replace the division by μ by the multiplication with $M = 1/\mu = \log_{a_0} e$, i.e., with the module of the logarithms to the base a_0 . In this way we replace two divisions indicated in (I) by one division and one multiplication.

⁴ A. KHINTCHINE, *Metrische Kettenbruchprobleme*. *Comp. Math.*, v. 1, 1935, p. 360-382, especially p. 376, 377.

⁵ Dr. D. SHANKS to whom this note was submitted in manuscript made the very interesting remark that the use of $\mu = \ln a_0$ can be eliminated from my formula (I) in replacing μ by

$$P_i(a_i - 1) + P_{i-1}(a_{i+1} - 1) = \mu + O\left(\frac{1}{Q_{i-1}}\right).$$

However, in this case, the exact error estimate in (II) has to be slightly changed.

Iterative Procedures for Taking Roots Based on Square Roots

Introduction. Several years ago, on the Model 1 C.P.C. at the Los Alamos Scientific Laboratory, there was incorporated by Dr. R. H. Stark a routine for taking square roots which proceeded at the same rate as the basic card feed operation. In connection with such an operation it is natural to ask if certain basic calculations might not be simplified by adding the square root as an operation. Today the existence of the Friden desk calculator, which also takes square roots directly, suggests that the question continues to have some cogency.

In connection with the Los Alamos C.P.C., we devised a procedure for taking cube roots and other roots which we present here for its conceivable usefulness and interest.

The Newton-Raphson iteration procedure for approximating roots of an equation $f(x) = 0$ establishes a function $g(x) = x - \frac{f(x)}{f'(x)}$ such that $x_{u+1} = g(x_u)$ gives an iterative procedure for finding roots starting with a guess x_0 .

To take the cube root of a positive number n , then, we may consider equations $x^3 - n = 0$, $x^{3/2} - n^{1/2} = 0$, $x^{3/4} - n^{1/4} = 0$. The corresponding respective iteration functions are $g_1(x) = 1/3\left(\frac{n}{x^2} + 2x\right)$, $g_2(x) = 1/3(2n^{1/2}x^{-1/2} + x)$ and $g_3(x) = 1/3(4n^{1/4}x^{1/4} - x)$. Now either of the last two functions are not feasibly used unless one can readily take square roots or fourth roots. In general, they are more costly than the usual iteration $g_1(x)$. However, if one has no preference between, say, square roots and other arithmetic operations, then either $g_2(x)$ or $g_3(x)$ will give an iteration which tends to converge in fewer steps than the standard one. For example, we have found cases in which the second routine was adequate in four iterations, whereas the g_1 routine would require five starting with the same guess.

To handle negative roots one may assign the sign of n to the first guess. We found it convenient to guess \sqrt{n} in the C.P.C. which found \sqrt{n} automatically. In general, one may take the seventh root of n , say, by using $g(x) = \frac{8(nx)^{1/8} - x}{7}$,

and the ninth root by using $g(x) = 1/9\left(8\left(\frac{n}{x}\right)^{1/8} + x\right)$, and so on.

PRESTON C. HAMMER

University of Wisconsin
Madison, Wisconsin