[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 $g(x) = x - \dfrac{f(x)}{f'(x)}$ 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 $g_1(x) = 1/3\left(\dfrac{n}{x^2} + 2x\right)$, $g_2(x) = 1/3(2n^{1/2}x^{-1/2} + x)$ 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 $g(x) = \dfrac{8(nx)^{1/8} - x}{7}$, and the ninth root by using $g(x) = 1/9\left(8\left(\dfrac{n}{x}\right)^{1/8} + x\right)$, and so on.

Preston C. Hammer

University of Wisconsin
Madison, Wisconsin