Algorithms and Accuracy in the HP-35

A lot goes on in that little machine when it's computing a transcendental function.

By David S. Cochran

THE CHOICE OF ALGORITHMS FOR THE HP-35 received considerable thought. Power series, polynominal expansions, continued fractions, and Chebyshev polynominals were all considered for the transcendental functions. All were too slow because of the number of multiplications and divisions required. The generalized algorithm that best suited the requirements of speed and programming efficiency for the HP-35 was an iterative pseudo-division and pseudo-multiplication method first described in 1624 by Henry Briggs in 'Arithmetica Logarithmica' and later by Volder[1] and Meggitt[2]. This is the same type of algorithm that was used in previous HP calculators.

An estimate of program execution times was made, and it became apparent that, by using a bit-serial data word structure, circuit economies could be achieved without exceeding a one-second computation time for any function. Furthermore, the instruction address and instruction word could be bit-serial, too.

The complexity of the algorithms made multilevel programming a necessity. This meant the calculator had to have subroutine capability, as well as special flags to indicate the status and separations of various programs. In the HP-35, interrogation and branching on flag bits—or on arithmetic carry or borrow—are done by a separate instruction instead of having this capability contained as part of each instruction. This affords a great reduction in instruction word length with only a slight decrease in speed.

The arithmetic instruction set was designed specifically for a decimal transcendental-function calculator. The basic arithmetic operations are performed by a 10's complement adder-subtractor which has data paths to three of the registers that are used as working storage. Partial word designators (word select) are part of the instruction word to allow operating on only part of a number—for example, the mantissa or the exponent field.

Sine Algorithm

The sine routine illustrates the complexities of programming a sophisticated calculator. First, degrees are converted to radians by multiplying by $2\pi/360$. Then integer circles are removed by repeatedly subtracting 2π from the absolute value of the argument until the result is less than 2π. If the result is negative, 2π is added to make it positive. Further prescaling to the first quadrant isn't required. The resulting angle is resolved by repeatedly subtracting $\tan^{-1} 1$ and counting until overdraft, then restoring, repeatedly subtracting $\tan^{-1} 0.1$ and counting until overdraft, and so on. This is very similar to division with a changing divisor. Next the resulting pseudo-quotient is used as a multiplier. Beginning with an X vector of 1 and a Y vector of 0 a fraction of X is added to Y and a fraction of Y is subtracted from X for the number of times indicated by each multiplier digit. The fraction is a negative power of 10 corresponding to that digit position. The equations of the algorithm are:

pseudo-division $\left\{ \theta_{n+1} = \theta_n - \tan^{-1} k \right.$

pseudo-multiplication $\left\{ \begin{array}{l} X_{n+1} = X_n - Y_n k \\ Y_{n+1} = Y_n + X_n k \end{array} \right.$

$$X_0 = 1,\ Y_0 = 0$$
$$k = 10^{-j} \qquad j = 0,1,2 \ldots$$

The pseudo-multiplication algorithm is similar to multiplication except that product and multiplicand are interchanged within each iteration. It is equiva-

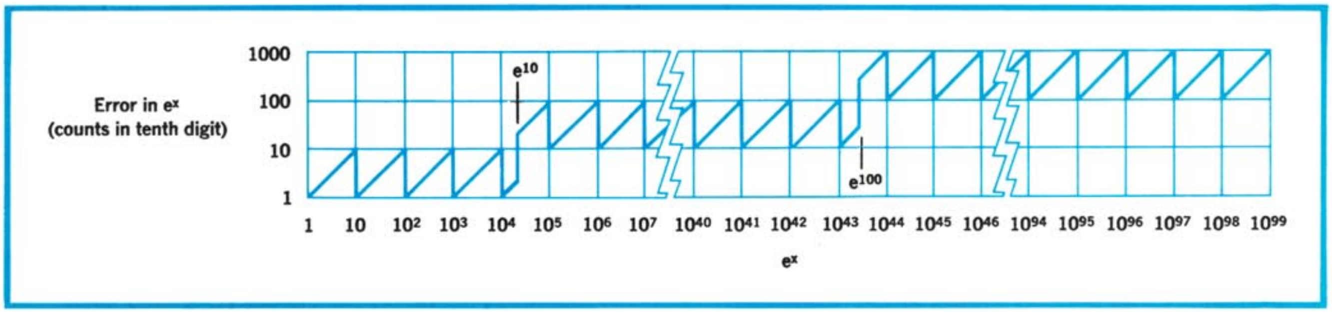


Fig. 1. *Accuracy of exponential function in HP-35 Calculator. Error bound is approximately δe^x, where δ is the error due to prescaling and the algorithm itself. δ is estimated to be equivalent to one count in the tenth significant digit of the argument x.*

lent to a rotation of axes. The resultant Y and X vectors are proportional to the sine and cosine respectively. The constant of proportionality arises because the axis rotation is by large increments and therefore produces a stretching of the unit circle. Since this constant is the same for both sine and cosine their ratio is identically equal to the tangent. The signs of each are preserved. The sine is derived from the tangent by the relationship

$$\sin\theta = \frac{\tan\theta}{(1+\tan^2\theta)^{1/2}}.$$

Accuracy and Resolution

Determination of the accuracy of the HP-35 is as complex as its algorithms. The calculator has internal roundoff in the 11th place. In add, subtract, multiply, divide, and square root calculations the accuracy is $\pm 1/2$ count in the 10th digit. In calculating the transcendental functions many of these elementary calculations are performed with the roundoff error accumulating. In the sine computation there is a divide, a multiply, and a subtract in the prescale operation, and there are two divides, a multiply, an addition, and a square root in the post-computation. Roundoff errors in these calculations must be added to the error of the basic algorithm to get the total error.

Accuracy and resolution are sometimes in conflict; for example, the subtraction of .9999999999 from 1.0 yields only one digit of significance. This becomes very important, for example, in computations of the cosines of angles very close to 90°. The cosine of 89.9° would be determined more accurately by finding the sine of 0.1°. Similarly, the sine of 10^{10} wastes all ten digits of significance in specifying the input angle, because all integer circles will be discarded.

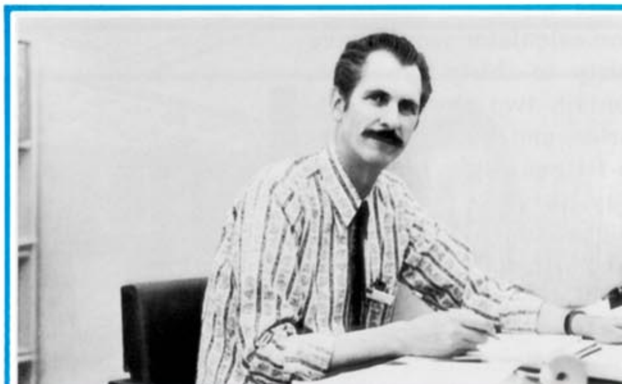
For many functions there is no simple exact expression for the error. The exponential function is a good example. Let δ be the accumulated prescaling error and computational error in the algorithm, referred to the input argument x. Then for $\delta << 1$,

$$e^{x+\delta} - e^x = e^\delta e^x - e^x = e^x(e^\delta - 1) \approx \delta e^x.$$

Fig. 1 shows the error bound for the exponential function for various arguments, assuming that δ is equivalent to one count in the tenth significant digit of x.

References

1. Jack E. Volder, 'The CORDIC Trigonometric Computing Technique.' IRE Transactions on Electronic Computers, September 1959.
2. J. E. Meggitt, 'Pseudo Division and Pseudo Multiplication Processes,' IBM Journal, April 1962.



David S. Cochran

Dave Cochran is HP Laboratories' top algorithm designer and microprogrammer, having now performed those functions for both the 9100A and HP-35 Calculators. He was project leader for the HP-35. Since 1956 when he came to HP, Dave has helped give birth to the 204B Audio Oscillator, the 3440A Digital Voltmeter, and the 9120A Printer, in addition to his work in calculator architecture. He holds nine patents on various types of circuits and has authored several papers. IEEE member Cochran is a graduate of Stanford University with BS and MS degrees in electrical engineering, received in 1958 and 1960. His ideal vacation is skiing in the mountains of Colorado.