References

[1] Licklider, J. C. R., Man-machine symbiosis, *IRE Trans. on Human Factors in Engineering*, vol HFE-1, Mar 1960, pp 4–11.
[2] Corbato, F. J., *et al.*, An experimental time-sharing system, *Proc. SJCC*, May 1962, pp 335–344.
[3] Munson, J. H., The scanning and measuring projector, Contract Rept UCRL-11154, Lawrence Radiation Lab., University of Calif., Dec, 1963.
[4] Hulsizer R. I., J. H. Munson, and J. H. Snyder, A system for the analysis of bubble chamber film based on the scanning and measuring projector, *Methods in Computational Physics*, vol 5, New York: Academic, to be published.
[5] Brown, R. M., *et al.*, "The CSX-1 computer," *IEEE Trans. on Electronic Computers*, vol EC-13, Jun 1964, pp 247–250.

A Class of Algorithms for Ln x, Exp x, Sin x, Cos x, Tan^{-1} x, and Cot^{-1} x

W. H. SPECKER

Introduction

Cantor, Estrin and Turn have described a special-purpose structure to implement sequential table look-up (STL) algorithms for the evaluation of ln x and exp x.[1] Tables of precomputed constants are used to transform the argument into a range where the function may be approximated by a simple polynomial. The transformation for ln x, originally proposed by Bemer,[2] is

$$\ln x = \ln \prod_{i=1}^{R} a_i x - \sum_{i=1}^{R} \ln a_i$$

where the set of constants a_i are precomputed. A power series expansion for ln x is

$$\ln x = (x-1) - \frac{1}{2!}(x-1)^2 + \frac{1}{3!}(x-1)^3 - \cdots$$

If

$$\prod_{i=1}^{R} a_i x \approx 1$$

then

$$\ln x \approx \prod_{i=1}^{R} a_i x - 1 - \sum_{i=1}^{R} \ln a_i$$

with an absolute error

$$< \frac{1}{2}\left(\prod_{i=1}^{R} a_i x - 1\right)^2$$

The intent of this note is to describe a new class of STL algorithms for the evaluation of ln x, exp x, sin x, cos x, tan^{-1} x and cot^{-1} x. One feature of these algorithms is that all of the real multiplications required reduce to one addition and one shift operation. (All of the complex multiplications required reduce to two additions and two shift operations.) Another feature is the relatively small number of stored constants required.

The algorithms for evaluating the trigonometric functions involve complex numbers. The notation used is as follows:

1) The imaginary number $\sqrt{-1}$ is denoted by j
2) "Re( )" means the real part of ( )
3) "Im( )" means the imaginary part of ( ).

Manuscript received September 8, 1964; revised October 9, 1964.
The author is with the Computer Div. of the Control Data Corp., Saint Paul, Minn.

[1] Cantor, D., G. Estrin, and R. Turn, Logarithmic and exponential function evaluation in a variable structure digital computer, *IRE Trans. on Electronic Computers*, vol EC-11, Apr 1962, pp 155–164.

[2] Bemer, R. W., A subroutine method for calculating logarithms, *Communication ACM*, vol 1, May 1958, pp 5–7.

Algorithm for ln x

1) Argument range considered:

$$0.5 \le x < 1$$

2) Transformation:

$$\ln x = \ln \prod_{i=1}^{R} a_i x - \sum_{i=1}^{R} \ln a_i \tag{1}$$

3) Definition of successive multipliers:

$$\begin{aligned} a_0 &= 1 \\ a_i &= 1 + 2^{-i} \quad \text{if} \quad (1+2^{-i}) \prod_{k=0}^{i-1} a_k x < 1 \\ a_i &= 1 \quad \text{otherwise} \end{aligned}$$

Note that multiplication of a real number by $1+2^{-i}$ need only require one shift operation of i places and one addition.

4) Diminishing factor:

$$1 - \prod_{i=1}^{R} a_i x < 2^{-R}$$

5) Function approximation:

$$\ln x \approx \prod_{i=1}^{R} a_i x - 1 - \sum_{i=1}^{R} \ln a_i$$

6) Maximum absolute error:

$$2^{-2R-1}$$

Algorithm for Exp x

1) Argument range considered:

$$0.5 \le x < 1$$

2) Transformation:

$$\exp x = \left(\prod_{i=1}^{R} b_i\right)\left[\exp\left(x - \sum_{i=1}^{R} \ln b_i\right)\right] \tag{2}$$

3) Definition of successive multipliers:

$$\begin{aligned} b_0 &= 1 \\ b_i &= 1 + 2^{-i} \quad \text{if} \quad x - \ln(1+2^{-i}) - \sum_{k=0}^{i-1} \ln b_k \ge 0 \\ b_i &= 1 \quad \text{otherwise} \end{aligned}$$

4) Diminishing factor:

$$x - \sum_{i=1}^{R} \ln b_i < 2^{-R}$$

5) Function approximation:

$$\exp x \approx \prod_{i=1}^{R} b_i$$

6) Maximum absolute error:

$$2^{-R} \exp(1 + 2^{-R})$$

Algorithms for Tan^{-1} x, Cot^{-1} x

1) Argument range considered:

$$0 \le x$$

2) Transformation. The desired transformation involves complex numbers. It is similar to the transformation for ln x, as is evident in the following derivation:

$$\ln(1+jx) = \ln\left[(1+jx)\prod_{i=1}^{R}(1-jc_i)\right] - \sum_{i=1}^{R} \ln(1-jc_i)$$

$$\operatorname{Im}\left\{\ln(1+jx)\right\} = \operatorname{Im}\left\{\ln\left[(1+jx)\prod_{i=1}^{R}(1-jc_i)\right] - \sum_{i=1}^{R}\ln(1-jc_i)\right\}$$

$$\tan^{-1} x = \tan^{-1}\left\{\operatorname{Im}\left[(1+jx)\prod_{i=1}^{R}(1-jc_i)\right]\right\} + \sum_{i=1}^{R} \tan^{-1} c_i \tag{3}$$

Also

$$\cot^{-1} x = \tan^{-1}\left\{ \mathrm{Im}\left[(x+j1)\prod_{i=1}^{R}(1-jc_i)\right]\right\} + \sum_{i=1}^{R}\tan^{-1} c_i \qquad (4)$$

3) Definition of successive multipliers:

$$1 - jc_0 = 1$$

$$1 - jc_i = 1 - j2^{-i}$$

$$\text{if } 0 \le x \le 1 \text{ and } \mathrm{Im}\left[(1+jx)\prod_{k=1}^{i-1}(1-jc_k)\right] \ge 0$$

$$\text{or if } x > 1 \text{ and } \mathrm{Im}\left[(x+j1)\prod_{k=1}^{i-1}(1-jc_k)\right] \ge 0$$

$$1 - jc_i = 1 + j2^{-i} \quad \text{otherwise}$$

Note that multiplication of a complex number by $1 \pm j2^{-i}$ need require only two shift operations of i places and two additions. Example:

$$(v + jw)(1 - j2^{-i}) = (v + 2^{-i}w) + j(w - 2^{-i}v)$$

4) Diminishing factor:

$$\left|\mathrm{Im}\left[(1+jx)\prod_{i=1}^{R}(1-jc_i)\right]\right| < 2^{-R} \qquad 0 \le x \le 1$$

$$\left|\mathrm{Im}\left[(x+j1)\prod_{i=1}^{R}(1-jc_i)\right]\right| < 2^{-R} \qquad x > 1$$

5) Function approximations:

$$\left.\begin{aligned} \tan^{-1} x &\approx \sum_{i=1}^{R}\tan^{-1} c_i \\ \cot^{-1} x &\approx \frac{\pi}{2} - \sum_{i=1}^{R}\tan^{-1} c_i \end{aligned}\right\} 0 \le x \le 1$$

$$\left.\begin{aligned} \cot^{-1} x &\approx \sum_{i=1}^{R}\tan^{-1} c_i \\ \tan^{-1} x &\approx \frac{\pi}{2} - \sum_{i=1}^{R}\tan^{-1} c_i \end{aligned}\right\} x > 1$$

6) Maximum absolute error:

$$2^{-R}$$

Algorithms for Sin x, Cos x

1) Argument range considered:

$$0 \le x \le \frac{\pi}{2}$$

2) Transformation: The desired transformation involves complex numbers. It is similar to the transformation for exp x, as is evident from the following derivation.

$$\exp(jx) = \left[\prod_{i=0}^{R}(1+jd_i)\right]\exp\left[jx - \sum_{i=0}^{R}\ln(1+jd_i)\right]$$

$$\exp(jx) = \left[\prod_{i=0}^{R}(1+jd_i)\right] \cdot \exp\left[j\left(x - \sum_{i=0}^{R}\tan^{-1} d_i\right) - \sum_{i=0}^{R}\ln(1+d_i^2)^{1/2}\right]$$

$$\exp(jx) = \left[\prod_{i=0}^{R}(1+d_i^2)^{-1/2}(1+jd_i)\right] \cdot \exp\left[j\left(x - \sum_{i=0}^{R}\tan^{-1} d_i\right)\right] \qquad (5)$$

3) Definition of successive multipliers:

$$1 + jd_0 = 1 + j1$$

$$1 + jd_i = 1 + j2^{-i} \quad \text{if} \quad x - \sum_{k=0}^{i-1}\tan^{-1} d_k \ge 0$$

$$1 + jd_i = 1 - j2^{-i} \quad \text{otherwise}$$

Also define

$$\begin{aligned} K &= (1+jd_0)\left[\prod_{i=0}^{R}(1+d_i^2)^{-1/2}\right] \\ &= (1+j1)\left[\prod_{i=0}^{R}(1+2^{-2i})^{-1/2}\right] \\ &= (1+j1)(0.60\cdots) \qquad R \ge 3 \end{aligned}$$

4) Diminishing factor:

$$\left|x - \sum_{i=0}^{R}\tan^{-1} d_i\right| < 2^{-R}$$

5) Function approximations:

$$\sin x \approx \mathrm{Im}\left[K\prod_{i=1}^{R}(1+jd_i)\right]$$

$$\cos x \approx \mathrm{Re}\left[K\prod_{i=1}^{R}(1+jd_i)\right]$$

(Note that K is effectively the first multiplier.)

6) Maximum absolute error:

$$2^{-R}\exp(2^{-R})$$

Implementation

To simplify analysis, assume that for each function,

1) the argument x is within the stated range for the appropriate algorithm
2) the maximum absolute error which can be tolerated is $2^{-R}\exp(1+2^{-R})$
3) round-off error may be neglected.

The constants which must be precomputed are then $\pi/2$, K, $\ln(1+2^{-1})$, $\ln(1+2^{-2})$, $\cdots$, $\ln(1+2^{-R})$, $\tan^{-1} 2^{-1}$, $\tan^{-1} 2^{-2}$, $\cdots$, $\tan^{-1} 2^{-R}$. Less than half of the constants need actually be stored. For example,

$$\tan^{-1} 2^{-R} \approx 2^{-R}$$

with an absolute error

$$< \tfrac{1}{3}2^{-3R}$$

and

$$\ln(1+2^{-R}) \approx 2^{-R}$$

with an absolute error

$$< 2^{-2R-1}$$

The algorithms may be either wired into a computer structure, or programmed. A basic hardware configuration would include one adder/subtractor, one multilength shift network (from 1 to R places) and a table of somewhat less than R constants. Approximately R additions and $R/2$ shift operations are required to evaluate ln x. Approximately $(3/2)R$ additions and R shift operations are required to evaluate exp x. Approximately $(7/3)R$ additions and $2R$ shift operations are required to evaluate sin x, cos x, $\tan^{-1} x$, or $\cot^{-1} x$. Sin x and cos x are obtained simultaneously.

A more elaborate hardware configuration might include three adders and two shift networks. All of the functions considered could then be evaluated in approximately the time to perform R additions and R shift operations.