

APPROXIMATIONS

FOR THE
CONTROL DATA
1 6 0 4

by H a n s J.
M A E H L Y

APPROXIMATIONS

FOR THE

CONTROL DATA 1604

by
HANS J. MAEHL*^{*}

* Presently Princeton University mailing address: Institute
for Advanced Study--E. C. P., Princeton, New Jersey.
After 1 February 1960: Mathematics Department,
Syracuse University, Syracuse 10, New York.

The approximations reported here have been selected
from our work under Contract NONR 2406(00), supported
by the Bureau of Ships and its Applied Mathematics
Laboratory, David Taylor Model Basin.

PREFACE

The CONTROL DATA CORPORATION extends its sincerest appreciation to Dr. Hans J. Maehly for the task he has performed in preparing this report. The contents represent, we believe, a significant step forward in the development of numerical techniques.

Presented here is a (slightly) edited reproduction of Dr. Maehly's original text. An attempt was made to retain as much of the flavor of the initial report as possible.

K. H. Olson
Supervisor of Applications and Analysis
March, 1960

APPROXIMATIONS FOR THE CONTROL DATA 1604

<u>CONTENTS</u>	Page
INTRODUCTION AND MACHINE CHARACTERISTICS	1
1. APPROXIMATIONS FOR THE SQUARE ROOT	3
(1.1) Range	3
(1.2) The Newton Iteration Formula	3
(1.3) Best Linear Approximation	5
(1.4) A Simple Fractional Approximation	7
(1.5) A Fractional Approximation for Floating Point	10
2. APPROXIMATIONS FOR EXP (X)	12
3. APPROXIMATIONS FOR LOG X	16
(3.1) Reduction of the Range	16
(3.2) Rational Approximation for $\text{Log } \frac{1+t}{1-t}$	20
4. APPROXIMATIONS FOR ARCTAN (X)	25
(4.1) Reduction of the Range	25
(4.2) Approximations for ARCTAN t, $ t \leq \sqrt{2} - 1$	27
5. APPROXIMATIONS FOR TAN X	
(5.1) Reduction of the Range	29
(5.2) Basic Form of Approximations to TAN X	30
(5.3) Rational Approximations for $S(W^2)$	31
6. APPROXIMATIONS FOR SIN X AND COS X	34
(6.1) Reduction of the Range	34
(6.2) Matched Approximations for SIN X and COS X	37
(6.3) Rational Approximations for SIN X, $ X \leq \pi/2$	40
(6.4) Summary and Note	42
"FINALE PRESTO"	43
ACKNOWLEDGEMENTS	44

INTRODUCTION AND MACHINE CHARACTERISTICS

The purpose of this report is to describe and justify several approximations for the elementary non-rational functions which are, in our opinion, particularly suited for the Control Data 1604 computer. The manual for the computer has been carefully consulted since some of the machine characteristics have a decisive influence on the selection of best approximations. The most important of these characteristics are:

(i) Control Data 1604 is a BINARY COMPUTER

(ii) 1 word = 48 bits: sign + 47 bits for fixed point

= exponent (11) + (sign + 36) bits for floating point

Thus basic round-off $\leq \epsilon_0 = 2^{-48} = 3.55 \times 10^{-15}$ (fixed)

$\leq \epsilon_1 = 2^{-37} = 7.28 \times 10^{-12}$ (floating)

(iii) Average execution times are about: (in microseconds)

Fixed point	Add 7	multiply 45	divide 65
-------------	-------	-------------	-----------

Floating Point	Add 19	multiply 45	divide 56
----------------	--------	-------------	-----------

(iv) Size of memory: 32768 words of core, all equally accessible.

(Most customers will also use tape units)

Conclusions:

These machine characteristics will have the following effects on subroutines for special functions:

- (i) The basic ranges for such functions as $\exp(X)$, $\log X$, \sqrt{X} and $X^{1/3}$ are quite small; further reduction will not be necessary.
- (ii) For fixed point subroutines, the truncation error λ_0 should be smaller or about equal to ϵ_0 :

$$\lambda_0 \lesssim 3.55 \cdot 10^{-15}$$

though $\lambda_0 \approx 10^{-14}$ may be acceptable.

For floating point subroutines, the relative error λ_1 should be at most ϵ_1 :

$$\lambda_1 \leq 7.28 \cdot 10^{-12}$$

Internal round-off can be reduced by coding the subroutine internally in fixed point, using (some of) the 11 exponent bits.

- (iii) Division time $\approx 3/2$ multiplication time; therefore, fractional approximations can be used to great advantage.
- (iv) Though it is always desirable to make subroutines short, this restriction is not quite so serious with a 32,768 word memory as with 2000 or 4000 words. Even a short table of key values may be considered if this helps to save time. Extensive tables, however, should in general be avoided.

1. APPROXIMATIONS FOR THE SQUARE ROOT

(1.1) Range:

For a floating point subroutine, the exponent of X will be separated from the mantissa and the two cases, "exponent even" and "exponent odd", will be treated separately. The latter case is equivalent to " $\frac{1}{4} \leq \text{mantissa} \leq \frac{1}{2}$ ".

For a fixed point subroutine, the number will be "half-normalized" by an even number $2n$ of left-shifts and the two cases are then $\frac{1}{2} \leq X \cdot 2^{2n} < 1$ and $\frac{1}{4} \leq X \cdot 2^{2n} < \frac{1}{2}$.

These two ranges can be treated jointly or separately. Separation means better initial approximation and may save one iteration, depending on the type of initial approximation used and on the accuracy required.

(1.2) The Newton Iteration Formula

The theory of rational approximations to the square root can be understood best on the basis of the Newton Iteration Formula:

If Y_i is an approximation to $Y = \sqrt{X}$,

then

$$Y_{i+1} := \frac{1}{2} \left(Y_i + \frac{X}{Y_i} \right) * \tag{1.2.1}$$

will be a better approximation. Let δ_i be the "relative"

*) Throughout this report, the notation $(:=)$ stands for "is defined by" (cf. ALGOL).

or logarithmic error of Y_i , viz.

$$\delta_i := \ln (Y_i/Y) \quad (1.2.2)$$

The logarithmic error of Y_{i+1} will then be

$$\delta_{i+1} := \ln (Y_{i+1}) = \ln (\cosh \delta_i) \quad (1.2.3)$$

whence

$$\delta_{i+1} \approx \frac{1}{2} \delta_i^2 \quad \text{if } \delta_i \ll 1. \quad (1.2.4)$$

This is the Newton Iteration in its standard form.

It can be improved as follows: If the maximum of

$|\delta_i|$ is known (for a given interval $[X_1, X_2]$),

$$\lambda_i = \max_{[X_1, X_2]} |\delta_i| \quad (1.2.5)$$

Then the maximum of $|\delta_{i+1}|$ can be halved by

$$\text{redefining } \bar{Y}_{i+1} := \frac{Y_i + (X/Y_i)}{2 \sqrt{\cosh \lambda_i}} \quad (1.2.6)$$

so that

$$\bar{\lambda}_{i+1} := \max_{[X_1, X_2]} |\bar{\delta}_{i+1}| = \frac{1}{2} \ln (\cosh \lambda_i) \quad (1.2.7)$$

It will be noted, however, that this improvement by merely a factor 2 requires a true multiplication, while the original iteration formula does not, since division by 2 can be done by a right shift. Therefore, we shall use the original Newton formula for iteration. The improved formula, however, immediately leads to the best linear approximation.

(1.3) Best Linear Approximation

In order to find the best linear approximation, we start out with the best constant Y_0 and apply one improved Newton Iteration. Obviously,

$$Y_0 := (X_1 X_2)^{\frac{1}{4}} \quad (1.3.1)$$

is the "best" constant approximation to \sqrt{X} for the interval $[X_1, X_2]$, yielding the maximum error

$$\lambda_0 = \max_{[X_1, X_2]} |\ln (Y_0/Y)| = \frac{1}{4} \ln \frac{X_2}{X_1} \quad (1.3.2)$$

One "improved Newton step" yields

$$\bar{Y}_1 = \frac{Y_0 + X/Y_0}{2 \sqrt{\cosh \lambda_0}} = a + bX$$

with

$$a := \frac{(X_1 X_2)^{\frac{1}{4}}}{\sqrt{2 \left[(X_2/X_1)^{\frac{1}{4}} + (X_1/X_2)^{\frac{1}{4}} \right]}} \quad (1.3.3)$$

$$b := \frac{(X_1 X_2)^{-\frac{1}{4}}}{\sqrt{2 \left[(X_2/X_1)^{\frac{1}{4}} + (X_1/X_2)^{\frac{1}{4}} \right]}}$$

The relative error of this approximation is:

$$\bar{\lambda}_1 = \max_{[X_1, X_2]} |\ln (\bar{Y}_1/Y)| = \frac{1}{2} \ln \left[\frac{(X_2/X_1)^{\frac{1}{4}} + (X_1/X_2)^{\frac{1}{4}}}{2} \right] \quad (1.3.4)$$

or approximately

$$\bar{\lambda}_1 \approx \left[\frac{\ln (X_2/X_1)}{8} \right]^2 \quad (1.3.5)$$

It may be desirable (e.g. for scaling reasons) to have no error at the ends of the interval, i.e., for $X = X_1$ and $X = X_2$. The solution is:

$$\hat{Y}_1 = \hat{a} + \hat{b}X \quad \left[\begin{array}{l} \hat{a} := \frac{\sqrt{X_1 X_2}}{\sqrt{X_1} + \sqrt{X_2}} \\ \hat{b} := \frac{1}{\sqrt{X_1} + \sqrt{X_2}} \end{array} \right] \quad (1.36)$$

and the maximum error is exactly doubled:

$$\hat{\lambda}_1 = \ln \left[\frac{\left(\frac{X_2}{X_1}\right)^{\frac{1}{4}} + \left(\frac{X_1}{X_2}\right)^{\frac{1}{4}}}{2} \right] \approx 2 \left[\frac{\ln(X_2/X_1)}{8} \right]^2 \quad (1.3.7)$$

A few numerical values are given below, including the errors after 1, 2 and 3 iterations (Standard Newton Iterations):

	$X_2/X_1 = 4$		$X_2/X_1 = 2$			
	$\hat{\lambda}_i$	$\bar{\lambda}_i$	$\hat{\lambda}_i$	$\bar{\lambda}_i$		
Initial error	λ_1	$5.89 \cdot 10^{-2}$	$2.95 \cdot 10^{-2}$	$1.49 \cdot 10^{-2}$	$7.47 \cdot 10^{-3}$	(1.3.8)
1 Iteration	λ_2	$1.74 \cdot 10^{-3}$	$4.34 \cdot 10^{-4}$	$1.11 \cdot 10^{-4}$	$2.79 \cdot 10^{-5}$	
2 Iterations	λ_3	$1.51 \cdot 10^{-6}$	$9.40 \cdot 10^{-8}$	$6.16 \cdot 10^{-9}$	$3.89 \cdot 10^{-10}$	
3 Iterations	λ_4	$1.14 \cdot 10^{-12}$	$4.42 \cdot 10^{-15}$	$1.89 \cdot 10^{-17}$	$7.57 \cdot 10^{-20}$	

Conclusions: Three iterations are needed after a linear initial approximation but the range $|\frac{1}{4}, 1|$ need not be separated into two smaller ones. $\hat{\lambda}_4$ is good enough for floating point, $\bar{\lambda}_4$ is just fine for fixed point.

(1.4) A Simple Fractional Approximation

The linear approximation $Y_1 = a + bX$ requires 1 addition + 1 multiplication, a fractional approximation of the form

$$Y_1 := a + \frac{b}{c + X} \quad (1.4.1)$$

requires little more, viz., 2 additions and 1 division. This investment will pay off if we can thereby save one iteration. This is indeed the case.

It can easily be shown from the general theory of approximation that we can expect our new $Y_1(X)$ to equal \sqrt{X} at three points rather than two (for the linear case). For reasons of scaling, viz., to avoid spill when computing the square root of $1 - 2^{-47}$, we shall choose 1 as one of these points. It can also be shown that the third value of X , say X_3 , where $Y_1(X) = \sqrt{X}$, must be the square of the second, if the relative error is to be minimized.

Therefore, we choose:

$$a + \frac{b}{c + X} = \sqrt{X} \quad \text{for} \quad \left[\begin{array}{l} X_1 = 1 \\ X_2 = \alpha^2 \\ X_3 = \alpha^4 \end{array} \right] \quad (1.4.2)$$

The solution is relatively simple:

$$\left. \begin{aligned} a &= 1 + \alpha + \alpha^2 \\ b &= -[\alpha + 2\alpha^2 + 2\alpha^3 + 2\alpha^4 + \alpha^5] \\ &= -(1 + \alpha)^2 (1 + \alpha^2) \cdot \alpha \\ c &= \alpha + \alpha^2 + \alpha^3 = \alpha \cdot a \end{aligned} \right] \quad (1.4.3)$$

For a fixed point square root routine, I recommend the following initial approximations:

(i) UPPER RANGE: $\frac{1}{2} \leq X < 1$

Fitting points: $X_1 = 1$

$$X_2 = \alpha^2$$

$$X_3 = \alpha^4$$

$$\alpha = \frac{109}{128}$$

In order to avoid binary round-off of the constants

$$\text{error: } \lambda_{\text{rel}} = 4.0 \times 10^{-4}$$

$$\left. \begin{array}{ll} a_1 = + \frac{42217}{16384} & \frac{a_1}{4} = +.511644 \text{ (octal)} \\ b_1 = - \frac{17 \ 30502 \ 29565}{3 \ 43597 \ 38368} & \frac{b_1}{16} = -.2411246171475 \text{ (octal)} \\ c_1 = + \frac{46 \ 01653}{20 \ 97152} & \frac{c_1}{4} = +.43067152 \text{ (octal)} \end{array} \right\} (1.4.4)$$

(ii) LOWER RANGE: $\frac{1}{4} \leq X < \frac{1}{2}$

Fitting Points: $X_1 = \frac{1}{4}$

$$X_2 = \frac{\alpha^2}{4}$$

$$X_3 = \frac{\alpha^4}{4}$$

$$\alpha = \frac{151}{128}$$

$$\text{error: } \lambda_{\text{rel.}} = 4.3 \times 10^{-4}$$

After an easy transformation we obtain:

$$\left. \begin{array}{ll} a_2 = + \frac{58513}{32768} & \frac{a_2}{2} = + .711104 \text{ (octal)} \\ b_2 = - \frac{46 \ 05801 \ 37335}{27 \ 48779 \ 06944} & \frac{b_2}{4} = - .32636267 \ 122734 \text{ (octal)} \\ c_2 = + \frac{59 \ 16935}{83 \ 88608} & \frac{c_2}{2} = + .26444407 \text{ (octal)} \end{array} \right\} (1.4.5)$$

The octal forms have already been scaled for the following recommended algorithms:

$$\begin{array}{ccc}
 \text{UPPER RANGE} & & \text{LOWER RANGE} \\
 \frac{(Y_1)}{2} := \left[\frac{a_1}{4} + \frac{(b_1/16)}{(c_1) + X/4} \right] 2 & \parallel \parallel \parallel & \frac{Y_1}{2} := \frac{(a_2)}{2} + \frac{(b_2/4)}{(c_2/2) + X/2} \\
 \swarrow & & \searrow \\
 Y_2 := \frac{X/4}{Y_1/2} + Y_1/2 & & \\
 & & \\
 Y_3 := \frac{X/2}{Y_2} + Y_2/2 & &
 \end{array} \tag{1.4.5}$$

All division by 2 and 4 should be executed as unrounded right-shifts. No spill should occur, I think, even if X is very close to 1, such as $X = 1 - 2^{-47}$ or $X = 1 - 2^{-46}$.
 Accuracy: The relative errors for Y_1, Y_2, Y_3 are:

UPPER RANGE	LOWER RANGE	
$\lambda_1 = 4.0 \cdot 10^{-4}$	$\lambda_1 = 4.3 \cdot 10^{-4}$	(1.4.6)
$\lambda_2 = 8.0 \cdot 10^{-8}$	$\lambda_2 = 9.4 \cdot 10^{-8}$	
$\lambda_3 = 3.2 \cdot 10^{-15}$	$\lambda_3 = 4.4 \cdot 10^{-15}$	
maximum <u>absolute error</u>	$\lambda_{\text{abs.}} = 3.2 \cdot 10^{-15}$	
Compare:	$2^{-48} = 3.5 \cdot 10^{-15}$	

Note: I assume that these data are correct and reasonably accurate, but I did not have a chance to check them as carefully as I should like to.

(1.5) Fractional Approximation for Floating Point

Since somewhat less accuracy is required for floating point, a simple fractional initial approximation of the form (1.4.1) may be used for the "full range", as e.g. $[\frac{1}{4}, 1]$. It is more convenient for deriving the formulae below, to treat the range

$$\frac{1}{2} \leq X \leq 2 \quad (1.5.1)$$

Since there are no scaling difficulties in floating point, and since the point $X = 1$, being in the logarithmic center of the interval will be one in which our initial approximation will be exact, we can use an exact best-fit approximation for this interval, minimizing the relative error, i.e. minimizing

$$\lambda_1 := \max_{[\frac{1}{2}, 2]} |\ln(Y_1 / \sqrt{X})| \quad (1.5.2)$$

where

$$Y_1 := a + \frac{b}{c + X} \quad (1.5.3)$$

It can be shown that the constants a , b , c can be computed as follows: *)

$$\left. \begin{aligned} v &:= (X_{\max} + 1)/(X_{\max} - 1) = 3 \\ w &:= [4v^2 (v^2 - 1)]^{1/3} \cong 6.60385 \ 4497 \\ z &:= \sqrt{w^2 - w + 1} = \sqrt{\frac{w^3 + 1}{w + 1}} \cong 6.16498 \ 4974 \\ m &:= \frac{1}{2} + \frac{3/4}{[\sqrt{w + 1} + \sqrt{2z - w + 2}][z + w - \frac{1}{2}]} = .51104 \ 01655 \\ a = c &= \frac{1 + m}{1 - m} \cong 3.09031 \ 5520 \\ b &= 1 - a^2 \cong -8.55005 \ 0013 \end{aligned} \right\} (1.5.4)$$

*) Derivation published in Math. Comp. 1960

Note: While the exact numerical value of m is not critical (it should rather be chosen a trifle too big, but not smaller than the exact value), the relation $b = 1 - a^2$ should be exactly fulfilled. I recommend, therefore, to truncate m to, say, 17 binaries (rounding up) and then to compute $b = 1 - a^2$ (by machine).

Error bounds: The maximum relative error of the approximation (1.5.3) - (1.5.4) is

$$\left. \begin{array}{l} \text{whence} \\ \lambda_1 = 2.52614 \cdot 10^{-3} \\ \lambda_2 \cong 3.19 \cdot 10^{-6} \\ \lambda_3 \cong 5.09 \cdot 10^{-12} \\ \text{Compare: } 2^{-37} = 7.28 \cdot 10^{-12} \end{array} \right\} \quad (1.5.5)$$

To shift the range from $[\frac{1}{2}, 2]$ to $[\frac{x}{2}, 2x]$,

where x is an arbitrary number, take

$$\begin{aligned} a &= \frac{1+m}{1-m} \cdot x^{\frac{1}{2}} \\ c &= \frac{1+m}{1-m} \cdot x \quad \text{for } \frac{x}{2} \leq x \leq 2x \\ b &= \left[1 - \left(\frac{1+m}{1-m} \right)^2 \right] x^{3/2} \end{aligned} \quad (1.5.6)$$

In particular, if

$$a_1 := \frac{1+m}{1-m} \quad \text{and} \quad b_1 := -a_1^2 + 1 \quad (1.5.7)$$

(with $m = .51104 \ 01655$)

then

$$\left. \begin{array}{l} a = a_1 \cdot 2^n \\ c = a_1 \cdot 2^{2n} \\ b = b_1 \cdot 2^{3n} \end{array} \right\} \quad \text{for } 2^{2n-1} \leq x \leq 2^{2n+1} \quad (1.5.7)$$

2. APPROXIMATIONS FOR EXP (X)

All approximations in this section are based on the well-known continued fraction

$$e^X = 1 + \frac{2X}{2-X} + \frac{X^2}{6} + \frac{X^2}{10} + \frac{X^2}{14} + \frac{X^2}{18} + \frac{X^2}{22} + \dots \quad (2.1)$$

which can also be written in the form

$$e^X = 1 + \frac{2X}{S(X^2) - X} = \frac{S(X^2) + X}{S(X^2) - X} \quad (2.2)$$

where

$$S(X^2) = X \operatorname{cth} \frac{X}{2} = 2 + \frac{X^2}{6} + \frac{X^2}{10} + \frac{X^2}{14} + \dots \quad (2.3)$$

The first four approximants to $S(X^2)$ are:

$$S_0 := 2 \quad (2.4.0)$$

$$S_1 := 2 + \frac{X^2}{6} \quad (2.4.1)$$

$$S_2 := 2 + \frac{X^2}{6 + \frac{X^2}{10}} \equiv 12 - \frac{600}{60 + X^2} \quad (2.4.2)$$

$$S_3 := 2 + \frac{X^2}{6 + \frac{X^2}{10 + \frac{X^2}{14}}} \equiv 2 + X^2 \left(.05 + \frac{4.9}{42 + X^2} \right) \quad (2.4.3)$$

The last expressions in (2.4.2) and (2.4.3) are the forms which can be evaluated most quickly.

Range: It is well known that for a binary machine in floating point operation, the range of X can easily be reduced to

$$|X| \leq \frac{1}{2} \ln 2 \quad (2.5)$$

(cf. e.g. (2.10) below)

If we approximate e^X by

$$R_3(X) := \frac{S_3(X^2) + X}{S_3(X^2) - X} \quad (2.6)$$

then the maximum relative error will be

$$\lambda_3 := \left| \ln [e^{-X} \cdot R_3(X)] \right| \cong 2.8 \times 10^{-12} \quad (2.7)$$

for $|X| = \frac{1}{2} \ln 2$

While this is just good enough for a floating point exponential subroutine it may be worth while to note that for the same range the best-fit approximation R_3^* , viz.,

$$R_3^*(X) := \frac{S_3^*(X^2) + X}{S_3^*(X^2) - X} = 1 + \frac{2X}{S_3^*(X^2) - X}$$

with

$$S_3^*(X^2) := a + X^2 \left(b + \frac{c}{d + X^2} \right) \quad (2.8)$$

and

$$\begin{aligned} a &= 2.00000 \ 00000 \ 00575 \ 924 \\ b &= .04996 \ 24891 \ 36450 \ 764 \\ c &= 4.90315 \ 47989 \ 68682 \ 648 \\ d &= 42.01353 \ 28950 \ 41661 \ 680 \end{aligned}$$

reduces the error by a factor close to 256, thus:

$$\lambda_3^* := \max \left| \ln [e^{-X} R_3^*(X)] \right| \cong 1.11 \times 10^{-14} \quad (2.9)$$

for $|X| \leq \frac{1}{2} \ln 2$

Note: The above-mentioned reduction of the range to

$|X| \leq \frac{1}{2} \ln 2$ is achieved as follows: To compute e^u , find the integer n so that

$$u = n \ln 2 + X, \quad |X| \leq \frac{1}{2} \ln 2$$

thus

$$e^u = 2^n e^X$$

(2.10)

This n is simply added to the exponent of the result.

The only practical way to find n (for a general purpose subroutine) is to multiply u by $(1/\ln 2)$ and then to determine the nearest integer. If $[]$ denotes "integer part of" this can be written as follows:

$$z := u \cdot \left(\frac{1}{\ln 2} \right) \quad (2.11)$$

$$n := \left[z + \frac{1}{2} \right] \quad (2.12)$$

$$w := z - n \quad (2.13)$$

$$X := w \cdot \ln 2 \quad (2.14)$$

Comments: The multiplication (2.11) is due to the fact that we want e^X , while the machine has base 2. For many applications base 2 is just as good; for example, if the logarithmic and exponential subroutines are used to compute odd (or high) powers such as $X^{7/3}$ or X^{15} , i.e. in all those cases where logarithm and "antilogarithm" are used as auxiliary functions, just like \lg_{10} and 10^X are often used for numerical computations without an automatic computer.

I therefore, recommend that the basic subroutine computes 2^X and that the division by $\ln 2$ (multiplication by $1/\ln 2$) is executed outside, if necessary, or is done automatically as an option (Separate entry to basically the same subroutine).

The multiplication (2.14) can also be avoided since

$$\begin{array}{l}
 2^w = e^X = e^{w \ln 2} \approx \bar{R}(w) \\
 \text{if } R(w) := \frac{\bar{S}(w^2) + w}{\bar{S}(w^2) - w} = \frac{2w}{\bar{S}(w^2) - w} + 1 \\
 \text{with } \bar{S}(w^2) := \bar{a} + w^2 \left(\bar{b} + \frac{\bar{c}}{\bar{d} + w^2} \right)
 \end{array} \quad (2.15)$$

$$\begin{array}{l}
 \bar{a} := a/\ln 2 \\
 \bar{b} := b \cdot \ln 2 \\
 \bar{c} := c/\ln 2 \\
 \bar{d} := d/(\ln 2)^2
 \end{array} \quad (2.16)$$

The numerical values of a, b, c, d are the same as in (2.8), but those of $\bar{a}, \bar{b}, \bar{c}, \bar{d}$ have not yet been computed (on a normal desk computer, double precision is mandatory; on the CDC 1604 full fixed point precision will just be sufficient, at least for $\bar{b}, \bar{c}, \bar{d}$.) This reduces the number of multiplications (M) and divisions (D) to $2M + 2D$ for 2^Z and $3M + 2D$ for e^u . The entire subroutine will take around 400 μ sec.

3. APPROXIMATIONS FOR THE LOGARITHMIC FUNCTION

Note: Just as a subroutine for 2^X is somewhat shorter and simpler than one for e^X , the same will be true for $\log_2 X$ as compared to the natural logarithm $\ln X$. Therefore, we shall assume here that the subroutine proper will compute $\log_2 X$. The final multiplication

$$\ln X = (\ln 2) \log_2 X \quad (3)$$

can be done outside the subroutine or it will be "an optional extra at additional cost".

(3.1) Reduction of the Range

If X is represented in the machine as a normalized floating-point number, then the integer part of the logarithm will be the exponent of X minus one; or if

$$X = \xi \cdot 2^n, \quad \frac{1}{2} \leq \xi < 1 \quad (3.1.1)$$

then

$$Y := \log_2 X = n + m, \quad m := \log_2 \xi \quad (3.1.2)$$

This reduces the range for which $\log_2 \xi$ must be computed to the interval $[\frac{1}{2}, 1]$.

It is well-known that, for any given range of the argument of the logarithmic function, the series

$$\ln \xi = \ln \frac{1+t}{1-t} = 2t \left(1 + \frac{t^2}{3} + \frac{t^4}{5} + \frac{t^6}{7} + \dots \right) \quad (3.1.3)$$

converges much better than

$$\ln \xi = \ln(1+u) = u - \frac{u^2}{2} + \frac{u^3}{3} - \frac{u^4}{4} + \dots \quad (3.1.4)$$

The same is true for the corresponding continued fractions and for best-fit approximations of polynomial or fractional form. The maximum absolute value of t can further be reduced for the interval

$$\xi_{\min.} \leq \xi \leq \xi_{\max.} \quad \left. \begin{array}{l} \text{if we put} \\ t := \frac{\xi - \xi_0}{\xi + \xi_0} \\ \xi_0 = \sqrt{\xi_{\min.} \xi_{\max.}} \end{array} \right\} \quad (3.1.5)$$

$$\text{so that } \log \xi = \log \xi_0 + \log \frac{1+t}{1-t} \quad (3.1.6)$$

The range of t is then given by

$$|t| \leq t_{\max} := \frac{\xi_{\max} - \xi_0}{\xi_{\max} + \xi_0} \quad (3.1.7)$$

If we apply this method to the range $\frac{1}{2} \leq \xi \leq 1$

(cf. (3.1.1) above), we obtain

$$\begin{aligned} \xi_0 &= \sqrt{\frac{1}{2}} = .70710\ 67811\ 86547\ 5244 \\ \log_2 \xi &= 0.5 + \log \frac{1+t}{1-t} \\ t_{\max} &= \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx .17157\ 28752\ 5381 \end{aligned} \quad \left. \right\} \quad (3.1.8)$$

Thus t_{\max}^2 is just a little less than $3 \cdot 10^{-2}$ and each term of the power series (3.1.3) will add almost 2 more decimals; or a little more than 2 if the corresponding best-fit polynomial is used.

While the reduction (3.1.8) will be sufficient to allow for fairly short rational approximations, it is worthwhile to note that a further reduction is possible without introducing any new explicit operations (i.e. other than those for determining the proper range). For if we divide the interval $[\xi_{\min}, \xi_{\max}]$ into n subintervals:

$$k^{\text{th}} \text{ interval} = [\xi_{k-1,k}, \xi_{k,k+1}]$$

$$\xi_{\min} = \xi_{0,1} < \xi_{1,2} < \dots < \xi_{n-1,n} < \xi_{n,n+1} = \xi_{\max} \quad (3.1.9)$$

and define
$$\xi_k := \sqrt{\xi_{k-1,k} \cdot \xi_{k,k+1}}$$

then for each
$$\xi \in [\xi_{k-1,k}, \xi_{k,k+1}]$$

we take
$$t := \frac{\xi - \xi_k}{\xi + \xi_k} \quad (3.1.10)$$

and hence have
$$|t| \leq t_k := \frac{\xi_k - \xi_{k,k-1}}{\xi_k + \xi_{k,k-1}}$$

or
$$t_k = \frac{r_k - 1}{r_k + 1}, \quad r_k := \sqrt{\xi_{k+1,k} / \xi_{k,k-1}}$$

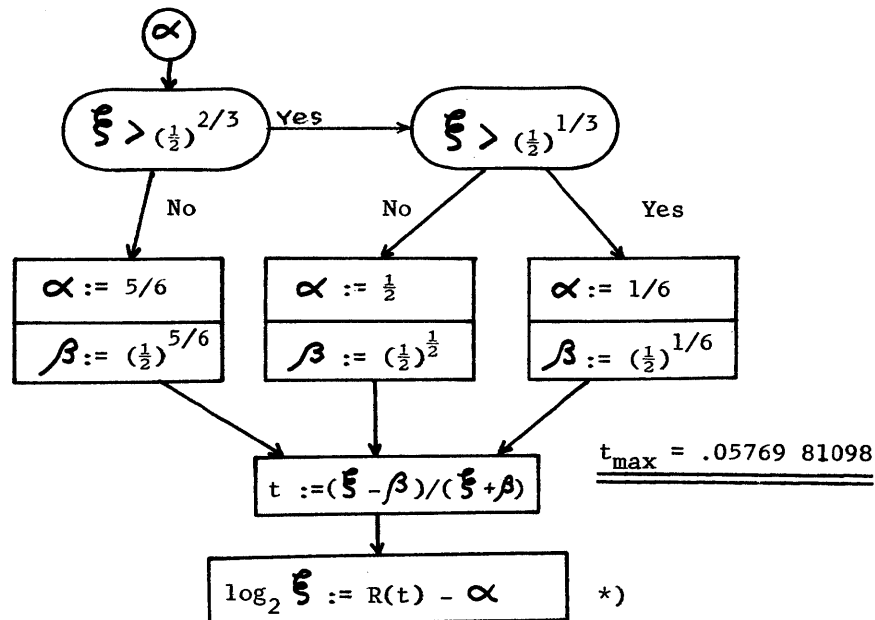
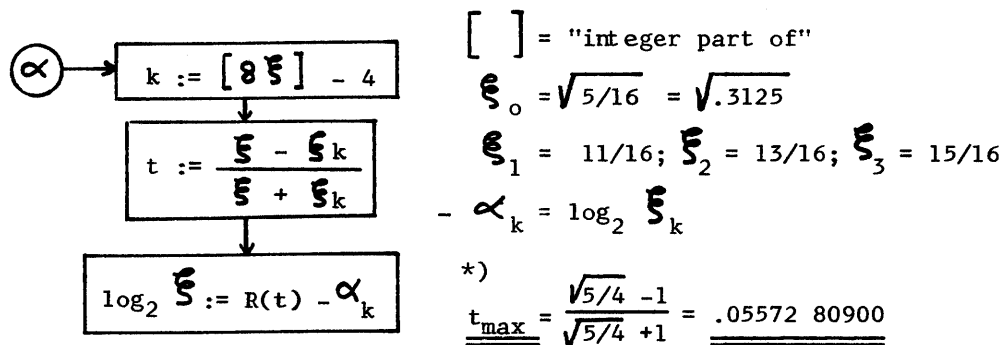
If the number of subintervals, n , is given then maximum (t_1, t_2, \dots, t_n) is minimized by logarithmic subdivision, viz.

$$r_1 = r_2 = \dots = r_n \quad (3.1.11)$$

but linear subdivision, viz.

$$\xi_{k,k+1} = \xi_{\min} + \frac{k}{n} (\xi_{\max} - \xi_{\min}) \quad (3.1.12)$$

may save enough time and/or storage to offset its lesser theoretical efficiency.

EXAMPLES:(i) Logarithmic Subdivision, min = $\frac{1}{2}$, max = 1, n = 3(ii) Linear Subdivision, min = $\frac{1}{2}$, max = 1, n = 4

In these two examples, t_{\max} is not much different, though

(ii) has $n = 4$, (i) only $n = 3$.

*) $R(t) \approx \log_2 \frac{1+t}{1-t}$ is a suitable rational approximation.

(3.2) Rational Approximations for $\ln \frac{1+t}{1-t}$

(i) Approximations of the form

$$\ln \frac{1+t}{1-t} \approx R(t) := t(a^* + \frac{b^*}{c + t^2}) \quad *) \quad (3.2.1)$$

This simple approximation is not accurate enough unless $[\frac{1}{2}, 1]$ is further divided. Approximate maximum errors (absolute errors) are given below:

TYPE OF SUBDIV.	n	max $\frac{\sum_{k=1}^n \frac{1}{k^2}}$	λ
none	1	2	3.29×10^{-9}
logarithmic	2	2	2.61×10^{-11}
logarithmic	3	$2 \frac{1}{3}$	1.53×10^{-12}
linear	4	1.25	1.20×10^{-12}
logarithmic	4	$2 \frac{1}{4}$	2.04×10^{-13}
linear	8	1.125	1.13×10^{-14}
logarithmic	8	$2 \frac{1}{8}$	1.60×10^{-15}

TABLE
(3.2.2)

For a standard floating point subroutine the maximum error should be below $7 \cdot 10^{-12}$, thus "logarithmic, n = 3" and "linear, n = 4" are suitable. For many special purposes, a routine which gives the logarithm in fixed point will also be very useful. The argument may be in fixed or floating representation, and a somewhat greater accuracy may then be required. We presently have computed the following coefficients of approximation.

*) For $\log_2 \frac{1+t}{1-t}$, divide the constants marked with an asterisk by

$\ln 2 = .69314\ 71805\ 59945\ 30941\ 72321\ \dots$ (cf. Tables Nat. Log. Vol. II, N.B.S. Appl. Math. Series #53)

$\max \frac{(\xi_{k,k+1})}{(\xi_{k-1,k})}$	t_{\max}	a^*, b^*, c	λ
2	.17157 2875	$a^* = .89554 \ 02099 \ 560$ $b^* = -1.82984 \ 55434 \ 565$ $c = -1.65677 \ 85798 \ 852$	$3.29 \cdot 10^{-9}$
$2^{1/2}$.08642 7234	$a^* = .89055 \ 57990 \ 96268$ $b^* = 1.84630 \ 58864 \ 29456$ $c = 1.66417 \ 19172 \ 70150$	$2.61 \cdot 10^{-11}$
$2^{1/3}$.05769 8110 *)	$a^* = .88963 \ 00669 \ 363587$ $b^* = -1.84938 \ 33168 \ 136899$ $c = -1.66555 \ 60110 \ 514012$	$1.53 \cdot 10^{-12}$
$2^{1/8}$.02165 7463	$a^* = .88899 \ 31487 \ 3553390$ $b^* = -1.85150 \ 43597 \ 8820645$ $c = -1.66651 \ 02989 \ 0598803$	$1.6 \cdot 10^{-15}$

TABLE
(3.2.3)

For very small ranges, up to about $\xi_{\max} / \xi_{\min} = 1.2$,
i.e., $t_{\max} \approx .05$, the "telescoping procedure for continued
fractions"*) can be used to compute a^* , b^* , c with
sufficient accuracy. For this particular case we obtain,

with $\epsilon = t_{\max}$:

$$\ln \frac{1+t}{1-t} \approx t \frac{P_0^* + P_1^* t^2}{Q_0 + Q_1 t^2} \quad \left[\begin{array}{l} P_0^* = 30 + \frac{3}{40} t_{\max}^6; \quad P_1^* = -8 - \frac{18}{5} t_{\max}^2 \\ Q_0 = 15; \quad Q_1 = -9 - \frac{9}{5} t_{\max}^2 + \frac{3}{10} t_{\max}^4 \end{array} \right] \quad (3.2.4)$$

from which the corresponding expression of the form (3.2.1)
can easily be derived.

*) cf. copy of my paper on this subject: "Methods for Fitting Rational Approximations", J.A.C.M., 1960.

*) May also be used for $n = 4$, linear, as long as the exact values
(yielding $\lambda = 1.45 \cdot 10^{-12}$) are not known.

(ii) Approximations of the Form

$$\ln \frac{1+t}{1-t} \approx R(t) := t \left[a^* + t^2 \left(b^* + \frac{c^*}{d+t^2} \right) \right] \quad (*) \quad (3.2.5)$$

This approximation yields nearly three decimals more than (3.2.1) - for the ranges given below, but it also requires one additional constant, one more addition and one more multiplication. This may not be too high a price if the subdivision of the interval $\left[\frac{1}{2}, 1 \right]$ can thereby be avoided; however, the error for the full range is approximately 10^{-11} , which is slightly above the basic round-off of $7 \cdot 10^{-12}$.

Further error estimates are given below:

TYPE OF SUBDIV.	n	max $\frac{\epsilon_{k, k+1}}{\epsilon_{k-1, k}}$	λ (appr.)
none	1	2	10^{-11}
logarithmic	2	$2^{\frac{1}{2}}$	$2 \cdot 10^{-14}$
logarithmic	3	$2^{1/3}$	$5 \cdot 10^{-16}$
linear	4	1.25	$4 \cdot 10^{-16}$
logarithmic	4	$2^{\frac{1}{4}}$	$4 \cdot 10^{-17}$

TABLE
(3.2.6)

The constants a^* , b^* , c^* and d for the full range have been computed for this report:

$$\left. \begin{array}{l} \epsilon_{\max} / \epsilon_{\min} = 2 \\ t_{\max} = .17157288 \\ \text{maximum abs. error } \lambda = 1 \cdot 10^{-11} \end{array} \right\} \begin{array}{l} a^* = 1.99999\ 99994\ 91255 \\ b^* = .10907\ 88905\ 02997 \\ c^* = -.77731\ 40010\ 05492 \\ d = -1.39406\ 51451\ 76107 \end{array} \quad \text{TABLE (3.2.7)}$$

*) For $\log_2 \frac{1+t}{1-t}$ divide the constants marked with an asterisk by $\ln 2 = .69314\ 71805\ 59945\ 30941\ 72321 \dots$

(iii) Approximations of the Form

$$\ln \frac{1+t}{1-t} \approx R(t) = \frac{t}{a' + t^2 \left(b' + \frac{c'}{d + t^2} \right)} \quad *) \quad (3.2.8)$$

This approximation is only slightly more accurate, the error being about 64% of that of the previous approximation, (3.2.5). The number of constants is the same, but one multiplication has been replaced by a division. Furthermore, this form (3.2.8) is somewhat more susceptible to round-off errors (due to the finite word-length) during evaluation than the form (3.2.5). However, this need not bother us if the evaluation is done in fixed point (carefully scaled) with correct round-off to a floating point mantissa at the end.

For the full range $[\frac{1}{2}, 1]$ the maximum absolute error will be approximately $6.5 \cdot 10^{-12}$ (Note: The constants a' , b' , c' and d are not included in this report.)

For the half ranges, $[\frac{1}{2}, \sqrt{\frac{1}{2}}]$ and $[\sqrt{\frac{1}{2}}, 1]$, the maximum absolute error will be approx. $1.3 \cdot 10^{-14}$, but this approximation cannot be recommended for a full precision fixed-point routine since the round-off error will be bigger than in (3.2.5), and (3.2.9) will be faster.

*) For $\log_2 \frac{1+t}{1-t}$, multiply the constants a' , b' , c' by

$\ln 2 = .69314\ 71805\ 59945\ 30941\ 72321 \dots$

(iv) Approximation of the Form

$$\ln \frac{1+t}{1-t} \approx R(t) = t \left[a^* + \frac{b^*}{c + t^2 + \frac{d}{e + t^2}} \right] \quad (*) \quad (3.2.9)$$

This approximation is much more accurate than (3.2.8); it requires one more constant but only one more addition; the number of multiplications (1) and divisions (2) is the same. This approximation can be used for a full-precision fixed-point subroutine without subdividing the interval $[\frac{1}{2}, 1]$. The constants are:

for

$$\xi_{\max} / \xi_{\min} = 2, t_{\max} = .17157288, \lambda = 1.18 \cdot 10^{-14}$$

$$\left[\begin{array}{l} a^* = .57314 \ 62238 \ 34578 \\ b^* = -3.83907 \ 86035 \ 23797 \\ c = -3.08667 \ 66195 \ 74836 \\ d = - .61016 \ 03452 \ 67418 \\ e = -1.54047 \ 22733 \ 27729 \end{array} \right]$$

TABLE
(3.2.10)

Which of the approximations given above are most suitable for Control Data subroutines will depend not only on certain details of coding and machine characteristics, but also on the relative merits of saving time or storage space and on the range for which a fixed point logarithm may be used, i.e., on the number of bits available for the fractional part of a fixed point logarithm after suitable scaling.

*) For $\log_2 \frac{1+t}{1-t}$ divide the constants marked with an asterisk by $\ln 2 = .69314 \ 71805 \ 59945 \ 30941 \ 72321 \ \dots$

4. APPROXIMATIONS FOR THE ARCTAN Z(4.1) Reduction of the Range

The addition theorem for $\tan(\phi + \psi)$, viz.

$$\tan(\phi + \psi) = \frac{\tan \phi + \tan \psi}{1 - \tan \phi \tan \psi} \quad (4.1.1)$$

can be used to reduce the range for which $\arctan X$ or $\arctan(X/Y)$ must be computed, if we store, in the memory of the machine, a table of "key values" Z_k, ψ_k ,

$$\begin{aligned} Z_k &:= \tan \psi_k \\ \psi_k &= \arctan Z_k \end{aligned} \quad (4.1.2)$$

The subroutine will first find, for each argument Z , the table entry Z_k which is "nearest" to Z in the sense that $|\arctan Z - \psi_k|$ is minimized. After that the algorithm runs as follows:

$$\begin{aligned} t &:= \frac{Z - Z_k}{1 + ZZ_k} \quad *) \quad (= \tan \phi) \\ \phi &:= \arctan t \quad (\text{approximation}) \\ \arctan Z &:= \psi_k + \phi \end{aligned} \quad (4.1.3)$$

While this algorithm permits a very drastic reduction of the range its cost in time and storage is considerable. Time is lost for finding the best (ψ_k, Z_k) , and for computing t (requiring one each of the four basic operations).

*) To compute $\arctan X/Y$, use

$$t := \frac{X - YZ_k}{Y + XZ_k} \quad (4.1.4)$$

Storage space is needed at least for the Z_k , while the Ψ_k may be equidistant.

Conclusion: With a small table, too much time is lost and not enough is gained by the moderate reduction of the range; a large table will save some time but cost more memory space than desirable for a general purpose subroutine.

Special Values: The formula for t is, of course, very simple for $Z_k = 0$ ($t = Z$) and for $Z_k \rightarrow \infty$ ($t = \frac{1}{Z}$), but also for $Z_k = \pm 1$, ($t = (Z \mp 1)/(1 \pm Z)$). With these four values, we obtain the following short table:

RANGE OF Z	Z_k	Ψ_k	t
$Z < -(1 + \sqrt{2})$	$-\infty$	$-\pi/2$	$-\frac{1}{Z}$
$-(1 + \sqrt{2}) < Z < -(\sqrt{2} - 1)$	-1	$-\pi/4$	$\frac{1+Z}{1-Z}$
$ Z < \sqrt{2} - 1$	0	0	Z
$\sqrt{2} - 1 < Z < 1 + \sqrt{2}$	+1	$+\pi/4$	$\frac{Z-1}{Z+1}$
$Z > 1 + \sqrt{2}$	$+\infty$	$+\pi/2$	$-\frac{1}{Z}$

*) (4.1.5)

The first and last ranges can be joined if it is not required that

$$-\pi/2 < \arctan Z < +\pi/2$$

This method reduces the range to $|t| \leq \sqrt{2} - 1$

*) It is true that t can also be computed very easily for $\arctan X/Y$, since,

$$\text{if } Z = X/Y, \frac{1}{Z} = \frac{Y}{X}; \frac{1+Z}{1-Z} = \frac{Y+X}{Y-X} \text{ and } \frac{Z-1}{Z+1} = \frac{X-Y}{X+Y}$$

but how can we determine the range directly from X & Y , without computing Z ?

(4.2) APPROXIMATIONS FOR ARCTAN t , $|t| \leq \sqrt{2} - 1$

We have derived the following approximation from the continued fraction for arctangent of t

$$R_6(t) \approx \arctan t \quad \text{if}$$

$$R_6(t) := t \left[d_1 + \frac{e_1}{t^2 + d_2 + \frac{e_2}{t^2 + d_3 + \frac{e_3}{t^2 + d_4}}} \right] \quad (4.2.1)$$

with

$$\left. \begin{aligned} d_1 &= 0.20131 \ 20564 \ 40625 \ 303 \\ e_1 &= 3.11385 \ 00604 \ 57103 \ 14 \\ d_2 &= 5.40622 \ 85377 \ 62366 \ 96 \\ e_2 &= -3.92831 \ 57487 \ 32049 \ 88 \\ d_3 &= 2.71829 \ 04240 \ 10983 \ 87 \\ e_3 &= -0.15058 \ 39379 \ 13062 \ 15 \\ d_4 &= 1.33875 \ 95795 \ 46815 \ 11 \end{aligned} \right\} \quad (4.2.2)$$

The maximum relative error λ ,

$$\lambda_6 := \max_{|t| \leq \sqrt{2}-1} \left| \log \frac{R_6(t)}{\arctan t} \right| = 2.84 \cdot 10^{-14} \quad (4.2.3)$$

is much smaller than necessary for a good floating point routine; it is actually good enough for a fixed point routine, since the absolute error, $\lambda_{6 \text{ abs}} \approx 3 \cdot 2^{-48}$

$$\lambda_{6 \text{ abs}} = \max_{|t| \leq \sqrt{2}-1} |R_6(t) - \arctan t| = 1.11 \cdot 10^{-14} \quad (4.2.4)$$

For a floating point routine the somewhat simpler approximation $R_5(t)$ may be used:

$$R_5(t) := t \left[d_0 + t^2 \left(d_1 + \frac{e_1}{t^2 + d_2 + \frac{e_2}{t^2 + d_3}} \right) \right]$$

$$(\lambda = 3.9 \cdot 10^{-12})$$

where

$$\begin{aligned} d_0 &= 0.99999\ 99999\ 96107 \\ d_1 &= -0.01558\ 53710\ 18178 \\ e_1 &= -0.58531\ 51350\ 71831 \\ d_2 &= 2.10055\ 40871\ 65198 \\ e_2 &= -0.41900\ 30022\ 82544 \\ d_3 &= 1.62102\ 38336\ 34443 \end{aligned}$$

(4.2.5)

5. APPROXIMATIONS FOR TAN X(5.1) REDUCTION OF THE RANGE

The basic range for the tangent is given by the periodicity of $\tan X$.

$$\tan (X \pm \pi) \equiv \tan X \quad (5.1.1)$$

Therefore, the basic range is

$$\alpha - \pi/2 \leq X \leq \alpha + \pi/2 \quad (5.1.2)$$

where α may be chosen to be zero.

Further reductions can easily be achieved by the relation (4.1.1) of which

$$\tan (\phi \pm \pi/2) = -1/\tan \phi \quad (5.1.3)$$

is a special case. Another important special case is

$$\psi = \pm \pi/4, \quad \tan \psi = \pm 1, \text{ thus}$$

$$\tan (\phi \pm \pi/4) = \frac{\tan \phi \pm 1}{1 \mp \tan \phi} \quad (5.1.4)$$

(5.1.3) cuts the basic range to $|X| \leq \pi/4$; with

(5.1.4), we get down to $|X| \leq \pi/8$ *)

A code for the reduction of the range can be made efficient only if we introduce an auxiliary variable $W = \frac{X}{\pi} \cdot 2^k$.

We have arbitrarily chosen $k = 2$, thus

$$W := \frac{4}{\pi} X \quad (5.1.5)$$

$$\text{Let us write } t(W) := \tan \frac{\pi W}{4} = \tan X \quad (5.1.6)$$

then

$$t(W \pm 4) = t(W) \quad (5.1.7)$$

$$t(W \pm 2) = -1/t(W) \quad (5.1.8)$$

$$t(W \pm 1) = \frac{t(W) \pm 1}{1 \mp t(W)} \quad (5.1.9)$$

*) By help of a table of key values and using (4.1.1), the range can be further reduced, saving a little time at great cost in storage.

(5.2) BASIC FORM OF RATIONAL APPROXIMATION FOR TAN X

Since $\tan X$ and $t(W)$ are odd functions, we may write

$$t(W) = W \cdot T(W^2) = \frac{W}{S(W^2)} \quad (5.2.1)$$

The question is whether $T(W^2)$ or $S(W^2)$ should be used as an auxiliary function. Assuming that T and S could be computed equally fast (with comparable accuracy), the first form will be faster for the basic range since a multiplication takes less time than a division. However, if one of the relations (5.1.8) or (5.1.9) must be used, then the second form is much faster since

$$t(W \pm 2) = - \frac{1}{t(W)} = \frac{1}{W \cdot T} = \frac{S}{W} \quad (5.2.2)$$

and

$$t(W \pm 1) = \frac{t(W) \pm 1}{1 \mp t(W)} = \frac{WT \pm 1}{1 \mp WT} = \frac{W \pm S}{S \mp W} \quad (5.2.3)$$

Which method will be faster on the average? This depends on the frequency of arguments being in the basic range ($|X| \leq \pi/4$ or $\pi/8$) as opposed to those in the ranges requiring reduction by (5.2.2) or (5.2.3). If the distribution is uniform, then the second form is faster. *) The use of $S(W^2)$ also reduces the maximum time for the subroutine. Therefore, $S(W^2)$ will be recommended for a general purpose subroutine.

*) While this is true for both ranges ($|X| \leq \pi/4$, $|X| \leq \pi/8$), the difference is bigger if the smaller range, and thus (5.2.3), is used.

(5.3) RATIONAL APPROXIMATIONS FOR $S(W^2)$

Since

$$\tan X = \cfrac{X}{1} - \cfrac{X^2}{3} - \cfrac{X^2}{5} - \cfrac{X^2}{7} - \cfrac{X^2}{9} - \dots \quad (5.3.1)$$

$S(W^2)$ can be expressed by the continued fraction

$$\begin{aligned} S(W^2) &= \cfrac{W}{t(W)} = W / \tan \cfrac{\pi W}{4} \\ &= \cfrac{4}{\pi} - \cfrac{\cfrac{\pi}{4} W^2}{3} - \cfrac{\cfrac{\pi^2}{16} W^2}{5} - \cfrac{\cfrac{\pi^2}{16} W^2}{7} - \dots \end{aligned} \quad (5.3.2)$$

In the formulae given below, this expression has been modified in two ways:

- (i) The coefficients have been modified for best fit for the respective interval, i.e. so that

$$\lambda_{\text{rel.}} = \max \left| \ln \cfrac{S_n^*(W^2)}{S(W^2)} \right| \quad (5.3.3)$$

(= the relative error) is minimized.

- (ii) Simple algebraic transformations have been used to minimize the time required for evaluating the respective expression on the Control Data 1604.

If n is the "degree" of the approximation, viz.

$$S_n^*(W^2) = C_0 + \cfrac{C_1 W^2}{1} + \dots + \cfrac{C_n W^2}{1} \quad (5.3.4)$$

We obtain the following short table of maximum relative errors ($\lambda_{\text{rel.}}$):

n	$ x \leq \pi/4$	$ x \leq \pi/8$
3	$1.42 \cdot 10^{-8}$	$4.69 \cdot 10^{-11}$
4	$2.21 \cdot 10^{-11}$	$1.83 \cdot 10^{-14}$
5	$2.38 \cdot 10^{-14}$	$4.92 \cdot 10^{-18}$

TABLE
(5.3.5)

(i) APPROXIMATIONS FOR $|x| \leq \pi/4$

REDUCTION OF THE RANGE:

$$X \left(\frac{2}{\pi} \right) =: 2i + k + v$$

where i and k are integers and

$$\left[\begin{array}{l} |k + v| \leq 1 \\ |v| \leq \frac{1}{2} \end{array} \right]$$

ALGOR
(5.3.6)

for $k = \pm 1$; $\tan X := -S/W$

for $k = 0$; $\tan X := W/S$

where either

$$S := S_4^*(v^2) = a + \frac{b}{c + v^2 + \frac{d}{e + v^2}} \quad *)$$

$$(\underline{\lambda = 2.21 \cdot 10^{-11}})$$

a = 9.45815 57617 25496

b = 290.32031 00841 78635

c = -37.33612 85498 26952

d = -20.54475 60663 69045

e = -4.64212 22417 14098

(5.3.7)

$$\text{or } S := S_5^*(v^2) = a + v^2 \left[b + \frac{c}{d + v^2 + \frac{e}{f + v^2}} \right]$$

$$\underline{\lambda = 2.38 \times 10^{-14}}$$

a = .63661 97723 67596

b = -.07531 94869 91705

c = 3.88560 57227 68290

d = -14.87026 86251 97861

e = -57.81869 13873 68667

f = -9.32191 89536 46030

(5.3.8)

) About 4 to 5 bits ($1\frac{1}{2}$ decimals) may be lost by amplified round-off with this slightly unstable approximation. It is recommended to evaluate S_4^ in fixed point (48 bits) with proper scaling.

(ii) APPROXIMATIONS FOR $|x| \leq \pi/8$

REDUCTION OF THE RANGE:

$$X \left(\frac{4}{\pi} \right) =: 4i + k + W$$

Where i and k are integers and $\left[\begin{array}{l} |k + W| \leq 2 \\ |W| \leq \frac{1}{2} \end{array} \right]$

$$\text{for } k = -2; \quad \tan X := -S/W$$

$$\text{for } k = +1; \quad \tan X := \frac{W + S}{S - W}$$

$$\text{for } k = -1; \quad \tan X := \frac{W - S}{S + W}$$

$$\text{for } k = 0; \quad \tan X := W/S$$

ALGOR.
(5.3.9)

$$\text{Where either: } S := S_3^* (W^2) = a + W^2 \left(b + \frac{c}{d + W^2} \right)$$

With

$$a = 1.27323 \ 95447 \ 94842$$

$$b = - .07881 \ 15321 \ 78328$$

$$c = 3.11023 \ 62587 \ 99796$$

$$d = -16.99695 \ 38195 \ 49826$$

$$(\lambda = 4.69 \cdot 10^{-11})$$

(5.3.10)

Or

$$S := S_4^* (W^2) = a + \frac{b}{c + W^2 + \frac{d}{e + W^2}} \quad *)$$

With

$$a = 19.05374 \ 90250 \ 96548$$

$$b = 2368.80216 \ 75614 \ 4904$$

$$c = -151.08047 \ 87151 \ 32145$$

$$d = -331.56482 \ 92387 \ 31320$$

$$e = -18.56899 \ 78562 \ 14913$$

$$(\lambda = 1.83 \cdot 10^{-14})$$

(5.3.11)

) About 4 to 5 bits ($1\frac{1}{2}$ decimals) may be lost by amplified round-off with this slightly unstable approximation. It is recommended to evaluate S_4^ in fixed point (48 bits) with proper scaling.

6. APPROXIMATIONS FOR SIN X AND COS X(6.1) REDUCTION OF THE RANGE

The first and basic reduction of the range of the independent variable is brought about by the periodicity:

$$\begin{aligned} \sin X &= \sin (X + 2k \pi) \\ \cos X &= \cos (X + 2k \pi) \end{aligned} \quad k = \pm 1, \pm 2, \pm 3, \dots \quad (6.1.1)$$

and the basic relations

$$\left. \begin{aligned} \sin (X \pm \pi) &= -\sin X \\ \cos (X \pm \pi) &= -\cos X \end{aligned} \right\} \quad (6.1.2)$$

These reduce the range for which rational approximations for $\sin X$ and $\cos X$ must be found to

$$-\pi/2 \leq X \leq +\pi/2 \quad (6.1.3)$$

Further reductions are possible, but most of them do not pay:

- (i) Since $\sin (-X) = -\sin X$, $\cos (-X) = \cos X$, we could reduce the range to $0 \leq X \leq \pi/2$. But since this range is not symmetric, additional terms would appear in a best-fit approximation (viz. even powers for $\sin X$, odd powers for $\cos X$). Nothing would be gained, neither in speed nor in storage space.
- (ii) Formulae such as $\cos X = 2 \cos^2(X/2) - 1$ or $\sin X = \sin \frac{X}{3} (3 - 4 \sin^2 \frac{X}{3})$, etc. could also be used, but the evaluation of these formulae takes more time than can be saved by the respective reduction of the range, except for extremely high precision (double or triple precision).

- (iii) Since the continued fraction and rational approximations for $\tan X$ are so good, we may be tempted to compute $\sin X$ and $\cos X$ from

$$\left. \begin{aligned} \sin X &= \frac{2t}{1+t^2} \\ \cos X &= \frac{1-t^2}{1+t^2} \end{aligned} \right\} \quad (6.1.4)$$

$t := \tan \frac{X}{2}$

Here again, the evaluation of $2t/(1+t^2)$ or $(1-t^2)/(1+t^2)$ takes more time than can be saved.

- (iv) If we store a table of key values $S_k = \sin X_k$ and $C_k = \cos X_k$, then $\sin X$ and $\cos X$ may be computed from

$$\left. \begin{aligned} \sin X &= S_k \cos \xi + C_k \sin \xi \\ \cos X &= C_k \cos \xi - S_k \sin \xi \end{aligned} \right\} \quad (6.1.5)$$

$X = X_k + \xi$

This requires two additional multiplications and a double table look up; in addition, both $\cos \xi$ and $\sin \xi$ must be approximated. This method does not pay off unless the range of ξ is made extremely small -- but this means a long table of many key values (perhaps 1000 or so).

- (v) So far, we have not yet made use of the relation

$$\sin X = \cos (\pi/2 - X) \quad (6.1.6)$$

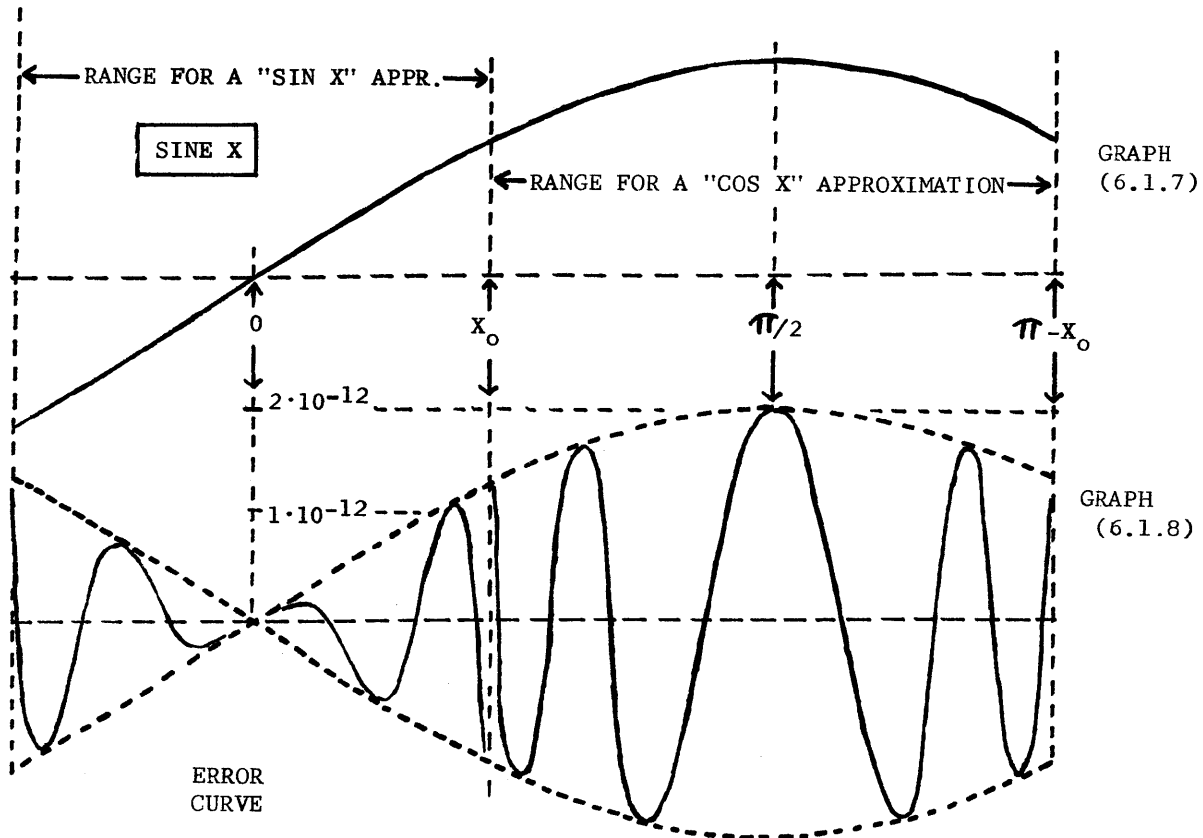
This can be used to obviate the need for one of the subroutines, *) but

*) Since $\sin X$ should come out as zero for $X = 0$, and good relative accuracy is desirable even if $|X| \ll 1$, for $\sin X$, the $\cos X$ subroutine may be dropped, but the $\sin X$ - subroutine should be retained.

it may also be used to reduce the range by taking "the other function" if $|x| > x_0$

where $0 < x_0 < \pi/2$

may be determined in such a way that the maximum error of two given forms of best-fit approximations for $\sin X$ & $\cos X$ are just equally big.



The sketch above illustrates this situation, assuming that the relative error has been minimized.

This is the only reduction as far as I can see, which is worth while.

(6.2) MATCHED RATIONAL APPROXIMATIONS FOR
SIN X, |X| < X₀, & COS X, |X| < π/2 - X₀

(i) A pair of approximations of the form

$$\left. \begin{aligned} \sin X &:= X \left(S_1 + \frac{S_2}{X^2 + S_3 + \frac{S_4}{X^2 + C_5}} \right) \\ \cos X &:= C_1 + \frac{C_2}{X^2 + C_3 + \frac{C_4}{X^2 + C_5}} \end{aligned} \right\} \quad (6.2.1)$$

will yield an error of approximately 2×10^{-11} *)

The matching point is approximately $X_0 \approx .90$. Since
 2×10^{-11} is too big even for (full precision) floating
point, the coefficients S_1 through C_5 have not been
 computed yet. They can be furnished on request.

(ii) A slightly better pair of approximations is

$$\left. \begin{aligned} \sin X &:= X \left(S_1 + X^2 \left(S_2 + X^2 \left(S_3 + \frac{S_4}{X^2 + S_5} \right) \right) \right) \\ \cos X &:= C_1 + X^2 \left(C_2 + X^2 \left(C_3 + \frac{C_4}{X^2 + C_5} \right) \right) \end{aligned} \right\} \quad (6.2.2)$$

The relative error is about $\lambda = 8 \times 10^{-12} \approx 1.1 \times 2^{-37}$
 which may be just acceptable for a 1604 floating point
 subroutine. The matching point is between .88 and .89. The
 coefficients can be computed fairly easily by a relatively
 simple iteration and will be furnished if requested.

NOTE: The numerical stability of (6.2.2) is much better
 than that of (6.2.1). (6.2.2) can be evaluated in floating
 point.

*) All errors quoted are relative, i.e.

$$\lambda = \max \left| \ln \frac{\text{APPROX.}}{\text{FUNCTION}} \right|$$

(iii) The following pair of approximations will yield 12 significant digits:

$$\left. \begin{aligned} \sin X &:= X \left(S_1 + \frac{S_2}{X^2 + S_3 + \frac{S_4}{X^2 + S_5}} \right) \\ \cos X &:= C_1 + X^2 \left(C_2 + \frac{C_3}{X^2 + C_4 + \frac{C_5}{X^2 + C_6}} \right) \end{aligned} \right\} \quad (6.2.3)$$

relative error: $\lambda \approx 4.56 \cdot 10^{-13}$

matching point: $X_0 \approx .6271$

COEFFI-
CIENTS:
*)

$S_1 =$	7.23084 68962 44279	$C_1 =$.99999 99999 99545
$S_2 =$	814.80758 58531 22316	$C_2 =$	- 1.67714 58152 33633
$S_3 =$	55.40962 23983 32114	$C_3 =$	271.20667 68780 46237
$S_4 =$	1262.62414 34758 4584	$C_4 =$	80.85518 72908 64723
$S_5 =$	16.75449 20850 08428	$C_5 =$	2442.54254 69501 6347
		$C_6 =$	16.33389 75777 12791

(iv) Since $|\sin X| \leq 1$, $|\cos X| \leq 1$, a fixed point $\sin X$ $\cos X$ subroutine is also feasible, approximately 14 digits will be needed for full fixed point accuracy. The following pair of approximations may be used:

$$\left. \begin{aligned} \sin X &:= X \left[S_1 + X^2 \left(S_2 + \frac{S_3}{X^2 + S_4 + \frac{S_5}{X^2 + S_6}} \right) \right] \\ \cos X &:= C_1 + X^2 \left(C_2 + \frac{C_3}{X^2 + C_4 + \frac{C_5}{X^2 + C_6}} \right) \end{aligned} \right\} \quad (6.2.4)$$

$$\lambda = 9.5 \cdot 10^{-15}, \quad X_0 = .8798$$

*) cf. "NOTE" on p. 42

The coefficients for (6.2.4) are:

$$\begin{array}{ll}
 S_1 = & .99999\ 99999\ 99990\ 520 & C_1 = & .99999\ 99999\ 99990\ 455 \\
 S_2 = & - .32590\ 23686\ 32526\ 89 & C_2 = & 1.70422\ 86567\ 42058\ 33 \\
 S_3 = & 71.63509\ 21318\ 01290\ 7 & C_3 = & 276.53438\ 35490\ 61398 \\
 S_4 = & 113.84748\ 44349\ 29748 & C_4 = & 81.38352\ 57153\ 53261\ 8 \\
 S_5 = & 4600.47709\ 02433\ 9799 & C_5 = & 2456.97667\ 28922\ 5259 \\
 S_6 = & 13.69104\ 85436\ 09095\ 3 & C_6 = & 16.57290\ 96384\ 87355\ 5
 \end{array}$$

The numerical stability of the formulae (6.2.4) is fair. Even with careful scaling the round-off error may be bigger than the truncation error. The next approximation will reduce both at a moderate increase in computing time.

(v) The following pair looks fine for a high-precision fixed-point subroutine:

$$\left. \begin{array}{l}
 \sin X := X \left(S_1 + X^2 \left(S_2 + X^2 \left(S_3 + X^2 \left(S_4 + \frac{S_5}{X^2 + S_6} \right) \right) \right) \right) \\
 \cos X := C_1 + X^2 \left(C_2 + X^2 \left(C_3 + X^2 \left(C_4 + \frac{C_5}{X^2 + C_6} \right) \right) \right)
 \end{array} \right\} (6.2.5)$$

$$\lambda \approx 8.1 \times 10^{-15}, \quad X_0 \approx .885$$

Since $\sin X$ and $\cos X$ must obviously be scaled by a factor of 1/2 in order to avoid overflow, $\lambda_{\text{rel}} = 8.1 \times 10^{-15}$ means that the truncation error is only slightly in excess of 1 unit of the last binary. Since (6.2.5) is very stable, the sum of round-off plus truncation error should be less than 2 or 3 units of the last binary for most values of X .

(6.3) RATIONAL APPROXIMATIONS FOR SIN X, $|X| \leq \pi/2$

Approximations to $\sin X$ for the full range, $-\pi/2 \leq X \leq +\pi/2$ require one division or multiplication more than those in (6.2).

The coefficients have not yet been computed, except for the first approximation, the accuracy of which is not sufficient for a full-precision floating point subroutine. The error estimates for the other approximations are believed to be correct within about $\pm 10\%$ of the values given below. (Please ask for coefficients if interested.)

(i) APPROXIMATION OF THE FORM $\sin X \approx X \cdot P_5(X^2)$

$$\sin X := X (S_1 + X^2 (S_2 + X^2 (S_3 + X^2 (S_4 + X^2 (S_5 + X^2 S_6))))))$$

$$\lambda_{\text{rel}} = 2.1 \times 10^{-11}$$

$$S_1 = +.99999 99999 79082 \quad *)$$

$$S_2 = -.16666 66660 92171 \quad (6.3.1)$$

$$S_3 = +.00833 33307 30723$$

$$S_4 = -.00019 84083 38222$$

$$S_5 = +.00000 27524 01177$$

$$S_6 = -.00000 00238 68930$$

(ii) APPROXIMATION OF THE FORM $\sin X = X P_4(X^2)/Q_1(X^2)$

$$\sin X := X(S_1 + X^2 (S_2 + X^2 (S_3 + X^2 (S_4 + \frac{S_5}{X^2 + S_6})))) \quad (6.3.2)$$

$$\lambda_{\text{rel}} = 1.0 \times 10^{-11}$$

This approximation is at least twice as accurate as (6.3.1) and can be computed faster, since 2 multiplications have been replaced by 1 division. Numerical stability is very good.

*) cf. "NOTE" p. 42

(iii) APPROXIMATION OF THE FORM $\text{SIN } X \approx X \cdot P_3(X^2)/Q_2(X^2)$

$$\text{sin } X := X (S_1 + X^2 (S_2 + \frac{S_3}{X^2 + S_4 + \frac{S_5}{X^2 + S_6}})) \quad (6.3.3)$$

$$\lambda_{\text{rel}} \approx 1.1 \times 10^{-11}$$

Two more multiplications have been replaced by one division, saving time but losing about 10% in accuracy. Numerical stability is not as good as in (6.3.2); therefore, a well-scaled fixed-point evaluation is mandatory!

We now proceed to the next degree of approximation which will yield about 8 additional bits:

(iv) APPROXIMATION OF THE FORM $\text{SIN } X = X P_6(X^2)$

$$\text{sin } X := X (S_1 + X^2 (S_2 + X^2 (S_3 + X^2 (S_4 + X^2 (S_5 + X^2 (S_6 + S_7 X^2)))))) \quad (6.3.4)$$

$$\lambda \approx 6.2 \cdot 10^{-14} \quad *)$$

(v) APPROXIMATION OF THE FORM $\text{SIN } X = X \cdot P_5(X^2)/Q_1(X^2)$

$$\text{sin } X := X (S_1 + X^2 (S_2 + X^2 (S_3 + X^2 (S_4 + X^2 (S_5 + \frac{S_6}{X^2 + S_7})))))) \quad (6.3.5)$$

$$\lambda \approx 2.3 \cdot 10^{-14} \quad *)$$

(vi) APPROXIMATION OF THE FORM $\text{SIN } X = X \cdot P_4(X^2)/Q_2(X^2)$

$$\text{sin } X := X (S_1 + X^2 (S_2 + X^2 (S_3 + \frac{S_4}{X^2 + S_5 + \frac{S_6}{X^2 + S_7}}))) \quad (6.3.6)$$

$$\lambda \approx 2.3 \cdot 10^{-14} \quad *)$$

*) Numerical stability: Good for (6.3.4) through (6.3.6)

(6.4) SUMMARY AND NOTE

In my opinion, the following approximations deserve prime consideration:

FLOATING POINT:	Almost full accuracy	(6.2.2)
	Full accuracy	(6.2.3)
FIXED POINT:	Full accuracy	(6.2.5)

The approximations in (6.3) take more time and are a little less accurate.

NOTE:

It may be convenient to introduce, in the beginning of a sin & cos subroutine, an auxiliary variable

$$v := X \cdot \frac{2}{\pi}$$

(cf. ALGOR. 5.3.6), so that rational approximations not for $\sin X$, but for $\sin \frac{v\pi}{2}$ are needed. The coefficients of such approximations can easily be found from those for $\sin X$ or $\cos X$. An example is given below:

Substitute $\frac{v\pi}{2}$ for X in, say, (6.2.3)

$$\begin{aligned} \sin \frac{v\pi}{2} &= \frac{v\pi}{2} \left(S_1 + \frac{S_2}{(v\pi/2)^2 + S_3} + \frac{S_4}{(v\pi/2)^2 + S_5} \right) \\ &= v \left(\sigma_1 + \frac{\sigma_2}{v^2 + \sigma_3} + \frac{\sigma_4}{v^2 + \sigma_5} \right) \end{aligned}$$

$$\sigma_1 = \frac{\pi}{2} S_1 \quad \sigma_3 = \left(\frac{2}{\pi}\right)^2 S_3 \quad \sigma_5 = \left(\frac{2}{\pi}\right)^2 S_5$$

$$\sigma_2 = \frac{2}{\pi} S_2 \quad \sigma_4 = \left(\frac{2}{\pi}\right)^4 S_4$$

FINALE PRESTO

I should have liked to discuss more functions in this report but serious time limitations have prevented me from doing so.

We have some theoretical and numerical results for

$$\int_0^X e^{-t^2} dt \quad \text{and} \quad \int_0^X \frac{e^{-t}}{t} dt$$

in particular for $X \rightarrow \infty$ (whence $\int_X^{\infty} e^{-t^2} dt$ and $\int_X^{\infty} \frac{e^{-t}}{t} dt$).

A few simple functions, such as $\tanh X$, $\ln \cosh X$, $\frac{e^X - 1}{X}$ and other functions for which either a well convergent power series or a well convergent continued fraction is known, can readily be treated, i.e., coefficients of suitable best-fit approximations can be obtained with our codes as soon as we find time for punching and running.

Some other functions, such as $\Gamma(X)$ for real values of X , have been studied and more work will be needed to make them ready for machine computation of the coefficients for best-fit approximations.

This semi-formal report has not been checked (for style, spelling and mathematical and numerical accuracy) as carefully as we would have checked a formal publication. Every comment and correction, and in particular reports on numerical checking of approximations given herein will be greatly appreciated.

ACKNOWLEDGEMENTS

Mr. Charles MESZTENYI has greatly contributed to all phases of the work reported here, viz., the analysis of special functions, planning and coding for the IBM 650 and running these codes for a great number of approximations (many more than listed in this report).

Dr. Christoph WITZGALL has also contributed to all phases, though he has joined our Project only one year ago.

Mrs. Birgitta NORELIUS-ASPLUND also joined us just a year ago but has taken a great burden of routine coding and running. Her accurate and reliable work has been most valuable.

THIS PROJECT AT PRINCETON UNIVERSITY

Was supported by the Bureau of Ships and its Applied Math. Lab., David Taylor Model Basin, under Contract Nonv 2406(00). It will terminate 31 JAN 1960.

Dr. WITZGALL will stay with Princeton University's Mathematics Dept. working on another project (Linear Programming) with Professor Tucker.

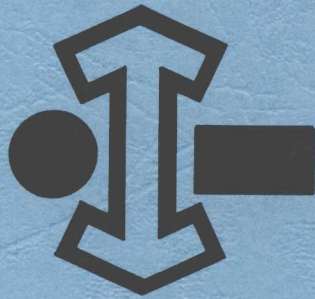
Mrs. NORELIUS will eventually return to Sweden, Mr. MESZTENYI and myself will go to Syracuse University where I hope to continue similar work.

PRINCETON
15 JAN 1960

Address all future
correspondence to:

Hans J. Maehly

HANS J. MAEHLY
MATHEMATICS DEPT.
SYRACUSE UNIVERSITY
SYRACUSE 10, NEW YORK



CONTROL DATA CORPORATION

501 Park Avenue, Minneapolis 15, Minnesota