27
7-27-76
DNTS

IS-3937

2-

THE ALECS NUMERICAL
SUBROUTINE LIBRARY

Kenneth R. Crouse
and Dennis V. Jensen

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: June 1976

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405-eng-82

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

THE ALECS NUMERICAL SUBROUTINE LIBRARY

By

Kenneth R. Crouse and Dennis V. Jensen

Ames Laboratory, ERDA
Iowa State University
Ames, Iowa 50011

Date Transmitted: June 1976

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PREPARED FOR THE U.S. ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION UNDER CONTRACT NO. W-7405-eng-82

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

—NOTICE—

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche \$2.25
Printed \$4.50

TABLE OF CONTENTS

	PAGE
ABSTRACT	iv
INTRODUCTION	1
SQRT	3
ISQRT	7
SIN	9
COS	13
ASIN	15
ACOS	20
TAN	22
ATAN	31
ATANR	37
LOG	41
EXP	46
REFERENCES	50

ABSTRACT

The optimized hardware floating-point versions of the built-in numerical subroutines used in the ALECS language are described. Information as to the algorithm used, the implementation conventions, and the accuracy of the results is provided for each subroutine. Accompanying each subroutine description is a flowchart of the implementation and a set of sample results. Possible error conditions are noted and the corresponding error codes are given.

Distribution List

Chicago Patent Office	1
USERDA-TIC	27
Ames Laboratory Library	5
Kenneth R. Crouse	2
Dennis V. Jensen	15
	—
Total	50

Introduction

The Ames Laboratory Experimental Control System, ALECS, is a PL/I based operating system which executes on a DEC PDP-15 computer. The numerical built-in functions of the ALECS language are implemented as system subroutines resident on disk and executed within the system transient area. (See reference 3.) All of the current hardware floating-point versions are single modules with no overlaying required and with no system calls made internal to the routines, except for the exit or a conditional return.

The floating-point arithmetic of the subroutines is carried out on the DEC FP15 floating-point processor. (See reference 4.) During its operation, this processor maintains 35 binary digits of mantissa, the sign of the mantissa, and 18 binary digits of 2's complement exponent. By software convention, though, all temporary storage locations and constants are single-precision floating-point numbers with 26 binary digits of mantissa, the sign of the mantissa, and a 9 binary digit 2's complement exponent. All floating-point numbers are maintained in a normalized form with rounding in the last binary digit.

Due to the speed and versatility of the FP15, the ALECS numerical subroutines were able to be optimized considerably in the conversion from the logical instruction sequence used with software floating-point to their present form, not only in terms of the execution time per operation, but also in terms of the total number of operations necessary.

The module descriptions contained in this document represent the optimized hardware floating-point form of the ALECS numerical subroutine library. In some cases the algorithms deviate from the previous implementations, but in most cases the logic remains the same. An effort was made to achieve maximum accuracy without long iterations or excessive special cases. The results of this library review and optimization are dealt with per each subroutine.

Many and diversified are the programmers that through the years have accumulated the algorithms contained in the ALECS numerical library. It is the authors' intent to make these proven algorithms and their specific accuracies known to the laboratory community. It is the authors' hope that future programmers will find this document a ready reference for their basic mathematical library needs.

SQRT

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given a nonnegative real argument X , the square root of X is computed. The algorithm used is Newton's method. If X_0 is an initial approximation to $\sqrt{X}$, then $X_1, X_2, \dots$ are calculated iteratively from

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{X}{X_n} \right).$$

II. Description of Implementation

1. The iteration is carried out until the absolute value of the difference in successive approximations is less than 2×10^{-6} .
2. The result is left in the floating-point A register.
3. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. All results are accurate to seven decimal places.
2. (See appendix 2 for sample results.)

IV. Error Codes

Error code 040033 is issued for a negative argument.

Appendix 1. Flow Chart of SQRT Implementation

X - initial argument
 EXP - original exponent on X
 XEXP - imposed exponent on X
 APP - approximation to X
 ANS - answer calculated at each iteration
 ϵ - error tolerance

calculate new answer

quit if change
less than ϵ

adjust exponent of result
according to original
exponent; $\lfloor \frac{EXP}{2} \rfloor$ means
largest integer $\leq \frac{EXP}{2}$

Appendix 2. Sample Results of SQR

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	0.0	0.0
.0000001	.3162277 E-03	.3162278 E-03
.002	.4472135 E-01	.4472136 E-01
.005	.7071067 E-01	.7071068 E-01
.008	.8944271 E-01	.8944272 E-01
.03	.1732050 E00	.1732051 E00
.05	.2236067 E00	.2236068 E00
.08	.2828427 E00	.2828427 E00
.10	.3162277 E00	.3162278 E00
.30	.5477225 E00	.5477226 E00
.45	.6708204 E00	.6708204 E00
.9999	.9999499 E00	.9999500 E00
2.0	.1414213 E01	.1414214 E01
7.0	.2645751 E01	.2645751 E01
15.0	.3872983 E01	.3872983 E01
245.0	.1565247 E02	.1565248 E02
270000.	.5196152 E03	.5196152 E03
6270000.	.2503996 E04	.2503997 E04
-2.0	Error code 040033	

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal digits.

ISQRT

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given an integer argument X, the square root of X is computed using Newton's method, the iteration being given by

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{X}{X_n} \right).$$

II. Description of Implementation

1. The square root is calculated as in SQRT except that the original fixed-point argument is converted to floating-point before the computations, and the final result is converted from floating-point to fixed-point.
(See section II of SQRT for details of the algorithm's implementation.)
2. Iteration is continued until the absolute value of the difference in successive approximations is less than 2×10^{-6} .
3. The result is left in the accumulator upon completion.

III. Accuracy of Results

1. Correct results were given in all cases.
2. When the true answer is not an exact integer, the integer result is arrived at by truncating the fractional portion rather than rounding.
3. (See appendix 1 for sample results.)

IV. Error Codes

Error code 040033 is issued for a negative argument.

Appendix 1. Sample Results of ISQRT

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0	0	0
1	1	1
2	1	1
5	2	2
7	2	2
10	3	3
15	3	3
23	4	4
75	8	8
163	12	12
1500	38	38
80000	282	282
-1	Error code 040033	

*Underlined digits indicate disagreement with the correct value.

**Values are determined by truncating the correct real-valued result to an integer value.

SIN

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

For a real argument X, expressed in radians, the sine of X is calculated based on the following approximation.

$$\sin(Y) = Y * (\frac{\pi}{2} + c_5 Y^8 + c_4 Y^6 + c_3 Y^4 + c_2 Y^2),$$

where Y is in the range $[-1, +1]$.

The coefficients are

$$\begin{array}{ll} c_2 = -0.6459637 & c_4 = -0.0046738 \\ c_3 = 0.0796897 & c_5 = 0.0001515. \end{array}$$

II. Description of Implementation

1. All constants are expressed in single-precision floating-point format with the mantissa accurate to within 10^{-7} .
2. The initial argument must be a single-precision floating-point number.
3. The result is left in the floating-point A register.
4. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. In most cases all seven decimal digits are correct.
2. Near π , as the result goes to zero, the number of correct significant digits reduces ultimately to zero, but the magnitude of the error is always less than 10^{-7} .
3. For arguments whose magnitude is 10π or larger, the number of correct significant digits depends on how close the argument is to an exact multiple of π . In the cases tested (up to 1000 radians) the magnitude of the error was always less than 10^{-5} .
4. (See appendix 2 for sample results.)

IV. Error Codes

No error codes specific to this routine are generated.

Appendix 1. Flow Chart of SIN Implementation

ARG - initial argument
 X - fractional part of $\frac{|ARG|}{2\pi}$
 Y - $4 * X$
 FNEG - flag for negative argument

Appendix 2. Sample Results of SIN

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	0.0	0.0
.00001	.0999999 E-04	.1000000 E-04
.001	.9999998 E-03	.9999998 E-03
.002	.1999998 E-02	.1999999 E-02
.003	.2999995 E-02	.2999996 E-02
.004	.3999989 E-02	.3999989 E-02
.005	.4999979 E-02	.4999979 E-02
.006	.5999963 E-02	.5999964 E-02
.007	.6999942 E-02	.6999943 E-02
.008	.7999914 E-02	.7999915 E-02
.009	.8999878 E-02	.8999879 E-02
.01	.9999833 E-02	.9999833 E-02
.10	.9983341 E-01	.9983342 E-01
.20	.1986693 E00	.1986693 E00
.50	.4794255 E00	.4794255 E00
1.5	.9974950 E00	.9974950 E00
2.0	.9092974 E00	.9092974 E00
4.0	- .7568024 E00	- .7568025 E00
6.0	- .2794155 E00	- .2794155 E00
1.57	.9999997 E00	.9999997 E00
1.5707963	.1000000 E01	.1000000 E01
1.57159265	.9999997 E00	.9999997 E00
4.7	- .9999232 E00	- .9999233 E00
4.71159265	- .9999997 E00	- .9999997 E00
4.7131853	- .9999997 E00	- .9999997 E00
- .01	- .9999833 E-02	- .9999833 E-02
- 1.5	- .9974950 E00	- .9974950 E00
- 3.0	- .1411200 E00	- .1411200 E00
3.14	.1592684 E-02	.1492653 E-02
3.1415926	.4681337 E-07	.5358979 E-07
3.15159265	- .9999826 E-02	- .9999830 E-02
31.42	.4073127 E-02	.4073453 E-02
50.0	- .2623751 E00	- .2623749 E00
176.0	.7074963 E-01	.7075224 E-01
300.0	- .9997558 E00	- .9997558 E00
700.0	.5439781 E00	.5439705 E00
1000.0	.8268741 E00	.8268795 E00

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal places.

COS

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given a real argument X in radians, the cosine of X is calculated according to

$$\cos(X) = \sin\left(\frac{\pi}{2} + X\right),$$

where $\sin(\frac{\pi}{2} + X)$ is computed by the SIN routine. (See the description of SIN for details of the algorithm.)

II. Description of Implementation

(See part II of SIN.)

III. Accuracy of Results

1. In most cases results are accurate to seven decimal digits.
2. Near odd multiples of $\frac{\pi}{2}$, as the answer approaches zero, the number of correct significant digits reduces ultimately to zero, but the magnitude of the error is on the order of 10^{-7} at most.
3. For arguments greater than 2π in magnitude, there is some loss of accuracy, the worst results being correct to four decimal places after rounding. The magnitude of the error was less than 2×10^{-5} for all cases tested.
4. (See appendix 1 for sample results.)

IV. Error Codes

There are no error codes specific to this routine.

Appendix 1. Sample Results of COS

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	1.0	1.0
.00001	.1000000 E01	.1000000 E01
.001	.9999995 E00	.9999995 E00
.002	.9999980 E00	.9999980 E00
.003	.9999955 E00	.9999955 E00
.004	.9999920 E00	.9999920 E00
.005	.9999875 E00	.9999875 E00
.006	.9999820 E00	.9999820 E00
.007	.9999755 E00	.9999755 E00
.008	.9999680 E00	.9999680 E00
.009	.9999595 E00	.9999595 E00
.01	.9999500 E00	.9999500 E00
.10	.9950041 E00	.9950042 E00
.50	.8775825 E00	.8775826 E00
1.5	.7073721 E-01	.7073720 E-01
1.6	- .2919953 E-01	- .2919952 E-01
3.0	- .9899925 E00	- .9899925 E00
4.0	- .6536436 E00	- .6536436 E00
- .01	.9999500 E00	.9999500 E00
- 1.5	.7073721 E-01	.7073720 E-01
- 3.0	- .9899925 E00	- .9899925 E00
3.14	- .9999987 E00	- .9999987 E00
3.1415926	- .1000000 E01	- .1000000 E01
3.15159265	- .9999500 E00	- .9999500 E00
1.57	.7963423 E-03	.7963267 E-03
1.5707963	.4681337 E-07	.2679490 E-07
1.57159265	- .7962954 E-03	- .7963231 E-03
4.7	- .1238865 E-01	- .1238866 E-01
4.71159265	- .7963891 E-03	- .7963303 E-03
4.7131853	.7962018 E-03	.7963195 E-03
8.0	- .1454998 E00	- .1455000 E00
31.42	.9999917 E00	.9999917 E00
50.0	.9649657 E00	.9649660 E00
176.0	.9974941 E00	.9974939 E00
300.0	- .2209711 E-01	- .2209662 E-01
700.0	- .8390993 E00	- .8391043 E00
1000.0	.5623870 E00	.5623791 E00

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal places.

ASIN

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

ASIN takes a real argument X and returns the arcsin of X expressed in radians in the range $[-\frac{\pi}{2}, +\frac{\pi}{2}]$. The following polynomial approximation is used.

$$\begin{aligned} \arcsin(X) = \frac{\pi}{2} - (1-X)^{\frac{1}{2}} & (a_0 + a_1X + a_2X^2 + a_3X^3 + a_4X^4 + a_5X^5 \\ & + a_6X^6 + a_7X^7) + \epsilon(X), \end{aligned}$$

where X must be in the range $0 \leq X \leq 1$,

and $|\epsilon(X)| \leq 2 \times 10^{-8}$.

The constants are taken from the National Bureau of Standards' Handbook of Mathematical Functions. They are

$a_0 = 1.5707963050$	$a_4 = .0308918810$
$a_1 = -.2145988016$	$a_5 = -.0170881256$
$a_2 = .0889789874$	$a_6 = .0066700901$
$a_3 = -.0501743046$	$a_7 = -.0012624911$

(See reference 1, page 81, in the reference list following the ALECS subprogram descriptions.)

II. Description of Implementation

1. The square root of $(1-X)$ is calculated internal to the ASIN routine with an error less than 2×10^{-7} .
(See the description of SQRT for details of the method.)
2. All constants are expressed in single-precision floating-point format, the mantissa being accurate to within 10^{-7} .
3. The argument must be a single-precision floating-point number.

4. Upon completion of the routine, the result is left in the floating-point A register.
5. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. Results for negative arguments are the same as for the corresponding positive arguments with the sign changed.
2. Results are generally accurate to six or seven decimal digits.
3. For arguments less than .01 in magnitude, results are accurate to five significant digits in the worst cases, but the magnitude of the error is always less than 10^{-7} .
4. Very close to 1 ($X = .9999999$) the magnitude of the error is 10^{-5} .
5. (See appendix 2 for sample results.)

IV. Error Codes

Error code 040042 is returned for an argument X such that $|X| > 1$.

Appendix 1. Flow Chart of ASIN Implementation

ARG - initial argument
 X - absolute value of argument
 ZFLAG - flag to be set if argument is negative
 FY, Y - intermediate results
 ANS - final answer

Appendix 2. Sample Results of ASIN

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	0.0	0.0
.02	.200013 <u>2</u> E-01	.2000133 E-01
.06	.600360 <u>5</u> E-01	.6003606 E-01
.10	.1001674 E00	.1001674 E00
.30	.3046926 E00	.3046927 E00
.70	.7753974 E00	.7753975 E00
.90	.111976 <u>9</u> E01	.1119770 E01
.99	.1429256 E01	.1429257 E01
1.0	.1570796 E01	.1570796 E01
- .10	- .1001674 E00	- .1001674 E00
- .90	- .111976 <u>9</u> E01	- .1119770 E01
-1.0	- .1570796 E01	- .1570796 E01
.0000001	.9999999 E-07	.1000000 E-06
.001	.9999999 E-03	.1000000 E-02
.002	.199999 <u>9</u> E-02	.2000001 E-02
.003	.300000 <u>0</u> E-02	.3000005 E-02
.004	.399999 <u>9</u> E-02	.4000011 E-02
.005	.500000 <u>0</u> E-02	.5000021 E-02
.006	.60000 <u>2</u> 8 E-02	.6000036 E-02
.007	.70000 <u>4</u> 3 E-02	.7000057 E-02
.008	.80001 <u>1</u> 5 E-02	.8000085 E-02
.009	.90001 <u>4</u> 0 E-02	.9000122 E-02
.01	.10000 <u>2</u> 0 E-01	.1000017 E-01
.9999999	.15703 <u>3</u> 9 E01	.1570349 E01
1.00001	Error 040042	
-1.00001	Error 040042	

*Underlined digits indicate disagreement with the correct value except for rounding in the last digit.

**Values are rounded to seven decimal digits.

ACOS

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

ACOS takes a real argument X and returns the arccosine of X in radians on the range $[0, \pi]$. The same algorithm is used as is used in ASIN, applying the identity

$$\arccos(X) = \frac{\pi}{2} - \arcsin(X).$$

(See the description of ASIN for details.)

II. Description of Implementation

(See section II of ASIN.)

III. Accuracy of Results

1. Results are generally accurate to a full seven decimal digits.
2. As the argument approaches 1.0, the number of correct significant digits diminishes until only one digit is correct for an argument of .9999999, where the magnitude of the error is less than 10^{-5} .
3. (See appendix 1 for sample results.)

IV. Error Codes

For an argument X such that $|X| > 1$, the error code 040041 is returned.

Appendix 1. Sample Results of ACOS

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
1.0	0.0	0.0
0.0	.1570796 E01	.1570796 E01
.001	.1569796 E01	.1569796 E01
.004	.1566796 E01	.1566796 E01
.008	.1562796 E01	.1562796 E01
.01	.1560796 E01	.1560796 E01
.02	.1550795 E01	.1550795 E01
.06	.1510760 E01	.1510760 E01
.08	.1490710 E01	.1490710 E01
.10	.1470628 E01	.1470629 E01
.50	.1047197 E01	.1047198 E01
.70	.7953988 E00	.7953988 E00
.90	.4510267 E00	.4510268 E00
- .00001	.1570806 E01	.1570806 E01
- .10	.1670963 E01	.1670964 E01
- .50	.2094395 E01	.2094395 E01
- .90	.2690565 E01	.2690566 E01
-1.0	.3141592 E01	.3141593 E01
.99	.1415395 E00	.1415395 E00
.999	.4472513 E-01	.4472509 E-01
.99999	.4471844 E-02	.4472140 E-02
.9999999	.4567452 E-03	.4472136 E-03
- .99999	.3137120 E01	.3137121 E01
1.00001	Error 040041	
-1.00001	Error 040041	

*Underlined digits indicate disagreement with the correct value except for rounding in the last digit.

**Values are rounded to seven decimal places.

TAN

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given a real argument X in radians, the tangent of X is computed.

The algorithm used comes from the FORTRAN mathematical subprogram library. (See reference 2, page 38, in the reference list following the ALECS subprogram descriptions.)

The algorithm consists of the following steps.

1. $|X|$ is divided by $\frac{\pi}{4}$ and the quotient and fractional part are separated: $\frac{|X|}{\pi/4} = q+r$.

2. Set $w = \begin{cases} r & \text{if } q \text{ is even, or} \\ 1-r & \text{if } q \text{ is odd.} \end{cases}$

Then $0 \leq w \leq 1$.

3. Set $q_0 = q \text{ modulo } 4$. Thus

$$\text{for } q_0 = 0, \quad \tan(|X|) = \tan\left(\frac{\pi}{4} \cdot w\right);$$

$$\text{for } q_0 = 1, \quad \tan(|X|) = \cot\left(\frac{\pi}{4} \cdot w\right);$$

$$\text{for } q_0 = 2, \quad \tan(|X|) = -\cot\left(\frac{\pi}{4} \cdot w\right);$$

$$\text{and for } q_0 = 3, \quad \tan(|X|) = -\tan\left(\frac{\pi}{4} \cdot w\right).$$

4. The approximations

$$\tan\left(\frac{\pi}{4} \cdot w\right) \doteq \frac{w \cdot P(u)}{Q(u)} \text{ and } \cot\left(\frac{\pi}{4} \cdot w\right) \doteq \frac{Q(u)}{w \cdot P(u)}$$

are used, where

$$P(u) = -8.460901 + u$$

$$Q(u) = -10.772754 + 5.703366u - 0.159321u^2$$

$$\text{and } u = \frac{1}{2} w^2.$$

5. If the initial argument X is negative, take

$$\tan(X) = -\tan(|X|).$$

The argument breakdown may be depicted on the unit circle:

$$q_o = 2; q = 2, 10, \dots$$

$$q_o = 1; q = 1, 9, \dots$$

$$q_o = 1; q = 5, 13, \dots$$

$$q_o = 2; q = 6, 14, \dots$$

$$\text{For } X_1: q = 0, q_o = 0, r = \frac{X_1}{\pi/4}, w = \frac{X_1}{\pi/4}, \tan(X_1) = \tan\left(\frac{\pi}{4} \cdot w\right)$$

$$\begin{aligned} \text{For } X_2: q = 1, q_o = 1, r &= \frac{X_2 - \pi/4}{\pi/4}, w = \frac{\frac{\pi}{2} - X_2}{\pi/4}, \tan(X_2) = \cot\left(\frac{\pi}{4} \cdot w\right) \\ &= \cot\left(\frac{\pi}{2} - X_2\right). \end{aligned}$$

II. Description of Implementation

1. The constants are expressed in single-precision floating-point format with the mantissa accurate to within 10^{-7} .
2. The argument X is assumed to be in single-precision floating-point format.
3. The result is left in the floating-point A register.
4. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. For most arguments the results are accurate to seven decimal places.
2. The magnitude of the results for negative arguments matches that of the corresponding results for positive arguments.
3. There is a loss of accuracy for arguments greater than 10π in magnitude. In general, at least four significant digits are correct for these cases.
4. Near π , as the answer goes to zero, the number of correct significant digits reduces ultimately to zero, but the magnitude of the error is always less than 10^{-7} .
5. Near odd multiples of $\frac{\pi}{2}$ there is an increasing loss of accuracy as the answer goes to infinity. The cut-off point ϵ ($=4*10^{-7}$) was chosen so that when $w \leq \epsilon$, the computation is not performed, and the error code is issued. As a result, the calculation is only carried out when at least two correct significant digits can be obtained.
6. (See appendix 2 for sample results.)

IV. Error Codes

Error code 040043 is given for an argument very close to an odd multiple of $\frac{\pi}{2}$. (See discussion of accuracy in section III.5 for exact values.)

Appendix 1. Flow Chart of TAN Implementation

X - initial argument

TEMP - temporary variable for $\frac{|X|}{\pi/4}$

q - integer part of $\frac{|X|}{\pi/4}$

r - fractional part of $\frac{|X|}{\pi/4}$

w - r or 1-r depending on q being even or odd respectively

q_0 - q modulo 4

$u = \frac{1}{2} 2^w$

S - $w \cdot P(u)$; $P(u)$ is defined in the algorithm description

T - $Q(u)$; $Q(u)$ is defined in the algorithm description

ANS - final result ($\tan(X)$)

FNEG - flag for $X < 0$

FCOT - flag for cotangent computation

FSIGN - flag for sign change when $q_0 = 2, 3$

ϵ - cut-off point for argument close to $\frac{\pi}{2}$

flag for cotangent
computation

sign change if
 $q_0 = 2$ or 3

compute $w \cdot P(u)$

compute $Q(u)$
(see part I.4 for values
that correspond to
 c_0 , c_1 , c_2 and c_3 .)

calculate $\tan(\frac{\pi}{4} \cdot w)$
or $\cot(\frac{\pi}{4} \cdot w)$

change sign if
 $q_0 = 2, 3$ or $X < 0$
and not both

Appendix 2. Sample Results of TAN

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	0.0	0.0
.00001	.9999999 E-05	.1000000 E-04
.001	.1000000 E-02	.1000000 E-02
.002	.2000002 E-02	.2000003 E-02
.003	.3000008 E-02	.3000009 E-02
.004	.4000021 E-02	.4000021 E-02
.005	.5000041 E-02	.5000042 E-02
.006	.6000071 E-02	.6000072 E-02
.007	.7000114 E-02	.7000114 E-02
.008	.8000170 E-02	.8000171 E-02
.009	.9000242 E-02	.9000243 E-02
.01	.1000033 E-01	.1000033 E-01
.10	.1003346 E00	.1003347 E00
.50	.5463024 E00	.5463025 E00
1.5	.1410141 E02	.1410142 E02
3.0	- .1425465 E00	- .1425465 E00
4.0	.1157821 E01	.1157821 E01
- .01	- .1000033 E-01	- .1000033 E-01
- .50	- .5463024 E00	- .5463025 E00
- 1.5	- .1410141 E02	- .1410142 E02
- 3.0	.1425465 E00	.1425465 E00
6.0	- .2910062 E00	- .2910062 E00
8.0	- .6799718 E01	- .6799712 E01
50.0	- .2719009 E00	- .2719006 E00
176.0	.7092737 E-01	.7092999 E-01
300.0	.4524373 E02	.4524474 E02
700.0	- .6482881 E00	- .6482752 E00
1000.0	.1470293 E01	.1470324 E01
31.42	.4073160 E-02	.4073487 E-02
3.14	- .1592686 E-02	- .1592655 E-02
3.1415926	- .4681337 E-07	- .5358979 E-07
3.14259265	.1000027 E-02	.9999967 E-02
3.14859265	.7000118 E-02	.7000111 E-02
3.15159265	.1000032 E-01	.1000033 E-01
4.7	.8071281 E02	.8071276 E02
4.71	.4185878 E03	.4185878 E03
4.71159265	.1255667 E04	.1255760 E04
4.7131853	- .1255814 E04	- .1255777 E04

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal places.

Appendix 2. Sample Results of TAN Continued

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
1.57	.1255 <u>741</u> E04	.1255766 E04
1.5707	.1037 <u>969</u> E05	.1038133 E05
1.57079	.15764 <u>88</u> E06	.1580579 E06
1.570794	.42299 <u>83</u> E06	.4297757 E06
1.570796	.3051 <u>630</u> E07	.3060023 E07
1.5707961	Error Code 040043	.4409270 E07
1.5707963	Error Code 040043	.3732054 E08
1.5707966	Error Code 040043	- .3660254 E07
1.5707967	- .26701 <u>76</u> E07	- .2679492 E07
1.5707968	- .21361 <u>41</u> E07	- .2113249 E07
1.5707969	- .17801 <u>17</u> E07	- .1744576 E07
1.570797	- .15258 <u>15</u> E07	- .1485431 E07
1.570798	- .59337 <u>26</u> E06	- .5976553 E06
1.5708	- .27386 <u>42</u> E06	- .2722418 E06
1.571	- .49095 <u>41</u> E04	- .4909825 E04

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal places.

ATAN

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given a real argument X, the arctangent of X is calculated in radians on the range $(-\frac{\pi}{2}, +\frac{\pi}{2})$.

The algorithm, taken from the FORTRAN mathematical subprogram library, proceeds in the following steps. (See reference 2, page 22, in the reference list following the ALECS subprogram descriptions.)

1. If $|X| > 1$, take $Y = \frac{1}{|X|}$, else take $Y = |X|$.
2. If $Y > \tan(15^\circ)$, reduce it to the range $[-\tan(15^\circ), \tan(15^\circ)]$ by taking $Y = \frac{\sqrt{3*Y}-1}{Y+\sqrt{3}}$.

3. Use the approximation

$$\frac{\arctan(Y)}{Y} \approx 0.60310579 - 0.05160454Y^2 + \frac{0.55913709}{Y^2+1.4087812}.$$

4. For arguments at step 2 such that $Y > \tan(15^\circ)$, use

$$\arctan(Y) = 30^\circ + \arctan\left(\frac{\sqrt{3*Y}-1}{Y+\sqrt{3}}\right).$$

5. For initial arguments X such that $|X| > 1$ use

$$\arctan(|X|) = \frac{\pi}{2} - \arctan\left(\frac{1}{|X|}\right).$$

6. For initial arguments $X < 0$, use

$$\arctan(X) = -\arctan(-X).$$

II. Description of Implementation

1. All constants are expressed in single-precision floating-point format with the mantissa accurate to within 10^{-7} .

2. The initial argument must be a single-precision floating-point number.
3. Upon completion the result is left in the floating-point A register.
4. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. The magnitude of the results for negative arguments matches that of the results for the corresponding positive arguments.
2. The results are generally accurate to the full seven decimal places. No problem areas arose for the arguments tested.
3. (See appendix 2 for sample results.)

IV. Error Codes

No error codes are specific to this routine.

Appendix 1. Flow Chart of ATAN Implementation

X - initial argument
 Y - absolute value of X
 FNEG - flag for negative X
 FONE - flag for $|Y| > 1$
 FTAN15 - flag for $Y > \tan(15^\circ)$

- a. $\sqrt{3} * Y - 1$ is calculated as $(\sqrt{3} - 1) * Y - 1 + Y$ to avoid losing significant digits.

$$b. \quad 0.60310579 - 0.05160454Y^2 + \frac{0.55913709}{Y^2 + 1.4087812} = \frac{c_1 Y^4 + c_2 Y^2 + c_3}{Y^2 + c_3}$$

where

$$\begin{aligned} c_1 &= -0.05160454 \\ c_2 &= 0.53040628 \\ c_3 &= 1.4087812 \end{aligned}$$

Appendix 2. Sample Results of ATAN

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	0.0	0.0
.001	.9999996 E-03	.9999997 E-03
.002	.1999997 E-02	.1999997 E-02
.003	.2999991 E-02	.2999991 E-02
.004	.3999978 E-02	.3999979 E-02
.005	.4999958 E-02	.4999958 E-02
.006	.5999927 E-02	.5999928 E-02
.007	.6999885 E-02	.6999886 E-02
.008	.7999829 E-02	.7999829 E-02
.009	.8999757 E-02	.8999757 E-02
.01	.9999666 E-02	.9999667 E-02
.02	.1999733 E-01	.1999733 E-01
.04	.3997868 E-01	.3997869 E-01
.08	.7982998 E-01	.7982999 E-01
.10	.9966865 E-01	.9966865 E-01
.20	.1973955 E00	.1973956 E00
.50	.4636476 E00	.4636476 E00
1.0	.7853981 E00	.7853982 E00
2.0	.1107148 E01	.1107149 E01
5.0	.1373400 E01	.1373401 E01
10.0	.1471127 E01	.1471128 E01
100.	.1560796 E01	.1560797 E01
100000.	.1570786 E01	.1570786 E01
.00001	.0999999 E-04	.1000000 E-04
- .1	- .9966865 E-01	- .9966865 E-01
- 1.0	- .7853981 E00	- .7853982 E00
-100000.	- .1570786 E01	- .1570786 E01

*Underlined digits indicate disagreement with the correct value except for rounding in the last digit.

**Values are rounded to seven decimal places.

ATANR

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given two real arguments X and Y, the arctangent of $\frac{X}{Y}$ is computed in radians on the range $(-\pi, \pi)$.

The algorithm used is the same as is used for ATAN. (See part I of the description of ATAN for details.)

II. Description of Implementation

1. Both X and Y are assumed to be single-precision floating-point numbers.
2. The result is left in the floating-point A register.
3. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. The magnitude of the results for negative X arguments matches that of the results for the corresponding arguments with X positive.
2. For all cases tested, the results were accurate to seven decimal places.
3. (See appendix 2 for sample results.)

IV. Error Codes

No error codes are specific to this routine.

Appendix 1. Flow Chart of ATANR Implementation

X - argument of numerator
 Y - argument of denominator
 ARG - X/Y
 FNEG - flag for $X < 0$

flag negative X

$$ATAN(+\infty) = \frac{\pi}{2},$$

$$ATAN(-\infty) = -\frac{\pi}{2}$$

(see description of
ATAN implementation)

adjust answer
according to
signs of X,Y
as below

$X \geq 0, Y \geq 0$	$0 \leq \text{ATAN} \leq \frac{\pi}{2}$
$X \geq 0, Y < 0$	$\frac{\pi}{2} < \text{ATAN} \leq \pi$
$X < 0, Y < 0$	$-\pi < \text{ATAN} \leq -\frac{\pi}{2}$
$X < 0, Y \geq 0$	$-\frac{\pi}{2} < \text{ATAN} < 0$

Appendix 2. Sample Results of ATANR

<u>Arguments</u>				
<u>X</u>	<u>Y</u>	<u> X/Y </u>	<u>Result*</u>	<u>Correct Value**</u>
.001	1.	.001	.9999996 E-03	.9999997 E-03
.002	1.	.002	.1999997 E-02	.1999997 E-02
.005	1.	.005	.4999958 E-02	.4999958 E-02
.01	1.	.01	.9999666 E-02	.9999667 E-02
.04	1.	.04	.3997868 E-01	.3997869 E-01
.50	1.	.50	.4636476 E00	.4636476 E00
5.	1.	5.	.1373400 E01	.1373401 E01
50.	1.	50.	.1550799 E01	.1550799 E01
100000.	1.	100000.	.1470786 E01	.1570786 E01
1.0	2.0	.50	.4636476 E00	.4636476 E00
1000.	2000.	.50	.4636476 E00	.4636476 E00
.0001	.0002	.50	.4636476 E00	.4636476 E00
1.	1.	1.	.7853981 E00	.7853982 E00
-1.	1.	1.	- .7853981 E00	- .7853982 E00
1.	-1.	1.	.2356194 E01	.2356195 E01
-1.	-1.	1.	- .2356194 E01	- .2356195 E01
-1.	1000.	.001	- .9999996 E-03	- .9999997 E-03
-1.	-1000.	.001	- .3140592 E01	- .3140593 E01
-100.	1.	100.	- .1560796 E01	- .1560797 E01
100.	-1.	100.	.1580796 E01	.1580797 E01
.001	-1.	.001	.3140592 E01	.3140593 E01
.002	-1.	.002	.3139592 E01	.3139593 E01
.005	-1.	.005	.3136592 E01	.3136593 E01
100000.	-1.	100000.	.1570806 E01	.1570807 E01
1000.	-1.	1000.	.1571796 E01	.1571796 E01
.0001	-1.	.0001	.3141492 E01	.3141493 E01
0.	1.	0.	0.0	0.0
0.	-1.	0.	.3141592 E01	.3141593 E01
1.	0.	∞	.1570796 E01	.1570796 E01
-1.	0.	∞	- .1570796 E01	- .1570796 E01
0.	0.	∞	.1570796 E01	.1570796 E01

*Underlined digits indicate disagreement with the correct value except for rounding in the last place.

**Values are rounded to seven decimal digits.

LOG

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

Given a real argument X , the natural logarithm of X , $\ln(X)$, is computed, where $e^{\ln(X)} = X$.

The approximation used is

$$\ln(1+X) = a_1X + a_2X^2 + a_3X^3 + a_4X^4 + a_5X^5 + a_6X^6 + a_7X^7 + a_8X^8 \\ + \epsilon(X), \quad |\epsilon(X)| \leq 3 \cdot 10^{-8}.$$

X must be in the range $0 \leq X \leq 1$.

The constants are

$a_1 = .99999\ 64239$	$a_5 = .16765\ 40711$
$a_2 = -.49987\ 41238$	$a_6 = -.09532\ 93897$
$a_3 = .33179\ 90258$	$a_7 = .03608\ 84937$
$a_4 = -.24073\ 38084$	$a_8 = -.00645\ 35442$

and are taken from the National Bureau of Standards' Handbook of Mathematical Functions.

(See reference 1, page 69, in the reference list following the ALECS subprogram descriptions.)

II. Description of Implementation

1. The constants are expressed in single-precision floating-point format with the mantissa accurate to within 10^{-7} .
2. The argument must be expressed in single-precision floating-point format.
3. The result is left in the floating-point A register.
4. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. Most results are accurate to seven decimal places.

2. Very close to 1.0 ($1.0 \pm .00001$) only three digits are accurate, but the magnitude of the error is less than 10^{-7} .
3. As the argument progresses away from 1.0, successively more digits become correct until, for $1.0 \pm .2$, the results again become accurate to seven decimal places.
4. (See appendix 2 for sample results.)

IV. Error Codes

Error code 040044 is issued for a negative or zero argument.

Appendix 1. Flow Chart of LOG Implementation

ARG - initial argument
 X - transformed value of ARG
 Y - X-1
 EXP - exponent on X

error if $ARG \leq 0$

adjust exponent so X,
 the mantissa, is
 between 1 and 2

save adjusted
 exponent

$0 \leq Y \leq 1$

$$\text{LN}(X) = \text{LN}(1+Y)$$

$$\begin{aligned} \text{LN}(\text{ARG}) &= \text{LN}(X * 2^{\text{EXP}}) = \\ &\text{LN}(X) + \text{LN}(2^{\text{EXP}}) \end{aligned}$$

Appendix 2. Sample Results of LOG

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
.00001	.1151292 E02	.1151293 E02
.001	- .6907755 E01	- .6907756 E01
.01	- .4605170 E01	- .4605170 E01
.10	- .2302585 E01	- .2302585 E01
.20	- .1609437 E01	- .1609438 E01
.50	- .6931471 E00	- .6931472 E00
.90	- .1053605 E00	- .1053605 E00
.99999	- .1002848 E-04	- .1000005 E-04
1.0	0.0	0.0
1.00001	.1001349 E-04	.9999950 E-05
1.01	.9950296 E-02	.9950331 E-02
1.02	.1980260 E-01	.1980263 E-01
1.03	.2955877 E-01	.2955880 E-01
1.04	.3922068 E-01	.3922071 E-01
1.1	.9531020 E-01	.9531018 E-01
1.2	.1823215 E00	.1823216 E00
2.0	.6931471 E00	.6931472 E00
5.0	.1609437 E01	.1609438 E01
10.0	.2302585 E01	.2302585 E01
100.	.4605170 E01	.4605170 E01
100000.	.1151292 E02	.1151293 E02
1.0 E50	.1151292 E03	.1151293 E03
0.0	Error 040044	
1.0	Error 040044	

*Underlined digits indicate disagreement with the correct value except for rounding in the last digit.

**Values are rounded to seven decimal places.

EXP

ALECS NUMERICAL SUBROUTINE

I. Description of Algorithm

For a real argument X , EXP calculates e^X using the fact that

$$e^X = \sum_{i=0}^{\infty} \frac{X^i}{i!}.$$

The terms in the series are calculated iteratively from

$$\frac{X^{n+1}}{(n+1)!} = \frac{X}{n+1} \cdot \frac{X^n}{n!}$$

until the magnitude of the last term calculated is less than or equal to 10^{-8} .

II. Description of Implementation

1. The argument must be a single-precision floating-point number.
2. If the argument is larger than 176 in absolute value, an overflow would occur, so the calculation is not performed, and an error is signaled.
3. Upon normal completion the result is left in the floating-point A register.
4. (See appendix 1 for flow chart.)

III. Accuracy of Results

1. The results showed no inaccuracies except for a slight error in the seventh decimal digit for large negative arguments (i.e., -30).
2. (See appendix 2 for sample results.)

IV. Error Codes

If the absolute value of the argument exceeds 176, the routine terminates with the error code 040045.

Appendix 1. Flow Chart of EXP Implementation

ARG - initial argument
 ZFLAG - flag for negative argument
 X - absolute value of argument
 AD - increment in series expansion
 ANS - calculated result
 N - index used to compute the increment

calculate increment

increment result

quit if increment small

$$e^{-X} = \frac{1}{e^X}$$

Appendix 2. Sample Results of EXP

<u>Argument</u>	<u>Result*</u>	<u>Correct Value**</u>
0.0	.1000000 E01	.1000000 E01
.01	.1010050 E01	.1010050 E01
.03	.1030454 E01	.1030455 E01
.07	.1072508 E01	.1072508 E01
.10	.1105170 E01	.1105171 E01
.50	.1648721 E01	.1648721 E01
.90	.2459603 E01	.2459603 E01
1.0	.2718281 E01	.2718282 E01
2.0	.7389056 E01	.7389056 E01
5.0	.1484131 E03	.1484132 E03
10.0	.2202646 E05	.2202647 E05
20.0	.4851651 E09	.4851652 E09
30.0	.1068647 E14	.1068647 E14
- .01	.9900498 E00	.9900498 E00
- 1.0	.3678794 E00	.3678794 E00
- 10.0	.4539993 E-04	.4539993 E-04
- 30.0	.9357621 E-13	.9357623 E-13
177.	Error 040045	
-177.	Error 040045	

*Underlined digits indicate a disagreement with the correct value except for rounding in the last place.

**Values given are rounded to seven decimal places.

References for ALECS Numerical Subroutines

1. Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables, Edited by M. Abramowitz and I. Stegun, National Bureau of Standards Applied Mathematics Series 55, 1964.
2. IBM System/360 Operating System FORTRAN IV Library-Mathematical Service Subprograms, Program Number 360S-LM-501, IBM Corporation, Programming Publications, New York, New York, September, 1972.
3. ALECS Language Reference Manual, J. H. Campbell, G. F. Covert, and B. J. Helland, Ames Laboratory, ERDA, Iowa State University, Ames, Iowa, January, 1976.
4. FP15 Floating Point Processor Programmer's Reference Manual, Document Number DEC-15-HQEA-D, Digital Equipment Corporation, Maynard, Massachusetts, April, 1971.