TWO ROUTINES TO CONVERT DATA TO UNITS IN DB FOR THE UNIVAC 1230—ETC(U)
AUG 70 C J BECKER, D M POTTER
UNCLASSIFIED
NUSC/NL-TM-2211-88-70
END
INTRODUCTION

During project PARKA, data were received, digitized and processed in real time by the UNIVAC 1230 computer aboard the prime receiving vessel SANDS. The final form of the processed data was displayed as listings and graphs with the values being expressed in db. To convert the data to db, two generalized programs, "TENLOGTEN" and "CONLOGIT" were written to calculate \( P = 10 \log_{10} X \), where \( X \) is a fixed point number within a specified set of limits. The major differences between these two routines are the methods used to obtain the natural logarithm and the scaling necessary for the input values.

ADMINISTRATIVE INFORMATION

This memorandum was prepared under NUSC Project Title: Long-Range Acoustic Transmission Experiments for Surveillance Systems Development; R. Hasse and R. Martin, NUSC/NL Principal Investigators. The sponsoring activity was ONR, Code 102-08; Dr. J. B. Hersey, Program Manager.

"TENLOGTEN"

Routine "TENLOGTEN" calculates \( 10 \log_{10} X \) by evaluating a 5th degree polynomial. Expressing \( 10 \log_{10} X \) as...
\[ b_5 = (10 \log_{10} e)a_5 = 0.13966237. \]

Routine "TENLOGTEN" calculates \( N \) and evaluates the above equation. A flow chart of "TENLOGTEN" is shown in Figure 1.

**Program Usage**

To call on routine "TENLOGTEN", enter the value \( X \) in the A register, scaled 22 bits, and RJP\(^*\)TENLOGTEN. The answer appears in the Q register scaled 3 bits.

If the inputted number \( X \) is less than or equal to zero, the routine returns a value of \( 4000000000 \). "CONLOGIT" also calculates \( P = 10 \log_{10}(X) \), by obtaining the natural logarithm using a subroutine called "NATLOG"\(^*\) and converting to \( P \) by equation (3). To use "NATLOG", the number \( X \) must be expressed in the form \( X = N \cdot 2^S \), where \( 1 \leq N \leq 37777776 \). "CONLOGIT", assuming the scale factor, \( S \), is 15, checks \( N \) to see that it is within limits. If it is not, "CONLOGIT" shifts \( N \) down so that the most significant bit is in the proper position and then adjusts \( S \) accordingly.

"NATLOG" uses a method attributed to Dr. H. Maehly\(^**\) to calculate the natural logarithm. Rewriting \( X \) as

\[ X = 2^m F, \]

where \( m \) is an integer and \( 1 \leq F \leq 2 \),

\[ \ln(X) = (m+\frac{1}{2}) \ln2 + P_0 + \sum_{k=1}^{3} \frac{-P_k}{1/F + T_k}, \]

where the vertical bars imply 'continued fractions,' that is

\[ \sum_{k=1}^{3} \frac{-P_k}{1/F + T_k}, \]

\[ \frac{-P_k}{1/F + T_k}, \frac{-P_k}{1/F + T_k}, \frac{-P_k}{1/F + T_k}, \]

\[ \frac{-P_k}{1/F + T_k}, \]

\[ * \text{ NATLOG was supplied to NUSC by UNIVAX as part of the software package delivered with the 1230 computer.} \]

\[ ** \text{ H. Maehly, Monthly Report, Institute for Advanced Study, Princeton, Oct 1956.} \]
and \( P_0 = 3.681656603 \),
\[ P_1 = 34.41069291, \]
\[ P_2 = 8.126503834, \]
\[ P_3 = 0.2665195666, \]
\[ T_1 = 10.37967214, \]
\[ T_2 = 2.05122813, \]
\[ T_3 = 0.4249952497. \]

Figure 2 is a flow diagram of "CONLOGIT" and Figure 3 is the flow chart of "NATLOG".

**Program Usage**

To use "CONLOGIT", the number \( X \) is put in the \( Q \) register, scaled to 15 bits. If \( X \) is negative, "CONLOGIT" sets it positive and proceeds, or if \( X \) is zero, an answer of -50.0 is given. If an error exit occurs in "NATLOG" caused by the scale factor \( S \) being outside of the limits, an answer of -100.0 is given. The answer, \( 10 \log_{10}(X) \) will be in the \( Q \) register scaled 3 bits.

To call on "CONLOGIT", use the call "CONLOGIT" or use "RJP*CONLOGIT".

CLAIR J. BECKER
Mathematician

DAVID M. POTTER
Mathematician
APPENDIX A

LISTING OF "TENLOGTEN"

TENLOGTEN

ENTRY

ENT=B7#LOGCOUNT
ENT=Q=0#APPOS
JP=LOGERR
JP=LOGERR=AZERO
NORM=A#W(B7)
SUB=A=2000000000
STR=A=W (LOG)
ENT=B7#3
ENT=Q=W(LOGCONT4)

LOGLOOP

MUL=A=W(LOGARG)
ADD=A=W(LOGCONTB7)
STR=A#Q
B JP=B7#LOGLOOP
MUL=A=W(LOGARG)
STRVA=A#(LOGARG)
ENT=Q=W(LOGCOUNT)
SUB=Q#6
MUL=A=W(LOGCONT5)
ADD=Q=W(LOGARG)#QPOS
SUB=Q=100000#SK1P
ADD=Q=100000
RSH=Q=16D#SKIP

LOGERR

ENT=Q=4000000000
EXIT

LOGCON

DATA=4 . 3407541D,21D
DATA=-2 . 1363335D,23D
DATA=1 . 257173D,25D
DATA=-59091362D,27D
DATA=.13966237D,29D
DATA=-3 . 0102D,19D

LOGARG

0

LOGCOUNT

0

INPUT IN A SCALED 22D
N GOES TO W(LOGCOUNT) ON NORM INSTRUCTION
PREPARE Q FOR NORMALIZATION
NEGATIVE INPUT
INPUT WAS 0
1 . GE . X . LT . 2
MINUS 1

B4
EVALUATE POLYNOMIAL
ADD COEFFICIENTS
SCALING IS 27,25,23,21 AS
B7 IS 3,2,1,0
POLYNOMIAL
N
N-6
(N-6)(-3.0102)
PLUS POLYNOMIAL
ROUND OFF
SCALE TO 3 PLACES
ERROR VALUE
OUTPUT IN Q SCALED 3
PROCEDURE CONLOGIT
STR Q* A# APOS
CP*Q
JP* CL4 AZERO
STR B7* L(CL7)
ENT B7* 1
LSH Q 1
JP* CL3 QNEG
ESC B7* 13D
JP* CL2
JP* CL5
ENT B7* B7-13D
STR B7* CPL(167)
ENT Q* X(B7-15D)
CP*Q
REP B7
CL2
LSH~ Q 1
JP* CL3 QNEG
SCALE DOWN REQUIRED, JUMP TO CL3
JP* CL5
NUMBER WITHIN RANGE
VERY RETURN VALUE IN A
CL5
ENT Q* X(B7-15D)
CP*Q
MAKE POSITIVE
CL6
ADD Q* 400 APOS
SUB Q 4000
RSH Q* 9 # SKIP
CL7
ENT Q* X76337
ENT B7* 0
RETURN
CL4
ENT Q* X77157
JP* L(CONLOGIT)
END-PROC CONLOGIT
APPENDIX C
LISTING OF "NATLOG"

NATLOG

JP*O
STR*Q*O CPW(KITTY)
CLE*Q
ENT* A* A APOS
CP*A
RPT*29D
LSH*A*1 ANEG
JP*NAT2
STR*B7* L(NAT1)
LSH*A*W(KITTY+2)
ENT*Q*X(O)
ADD*Q*W(KITTY)
LSH*Q*3
ADD*Q*4
C1A
MUL*Q*W(POOL)
RSH*AQ*9D
ENT*A*A*AZERO
CP*A*ANOT
RPL*Y+L* (NATLOG)*SKIP
JP*L*(NATLOG)
STR*Q*W(KITTY+1)
ENT*Q*W(KITTY+2)
ENT*Y*Q*W(POOL3)
ENT*Y*Q*W(POOL3)
STR*A*W(KITTY+3)
ENT*Y*Q*W(POOL2)
STR*A*W(KITTY+4)
ENT*Y*Q*W(POOL1)
STR*A*W(KITTY+5)
ENT*Q*W(POOL3+1)
ENT* A*X77777
LSH*AQ*23D
DIV*W(KITTY+3)
RPL*Y*Q*W(KITTY+4)
ENT*Q*W(POOL2+1)
ENT* A*X77777
LSH*AQ*23D
DIV*W(KITTY+4)
RPL*Y*Q*W(KITTY+5)
ENT*Q*W(POOL1+1)
ENT* A*X77777
LSH*AQ*23D
DIV*W(KITTY+5)
ADD*Q*W(KITTY+1)
ENT*Y*Q*W(POOL+1)
LSH*A*1

ENTRANCE
COMPLEMENT, SAVE SCALE FACTOR
CLEAR REGISTER
IS NUMBER POSITIVE
NO, COMPLEMENT
AND COUNT SHIFT
NUMBER IS ZERO
SAVE COUNT
ORIGINAL INPUT
M SC 3 IN Q
M+½ SC=3
(M+½)LN2 SC=3+29D
RIGHT TO SC-23D
IS A 0 or all 7's
YES
ERROR EXIT
F
F+T3 SC=23D
F+T3 SC=23D
F+T2 SC=23D
F+T1 SC=23D
-P3 SC=23D
SET SIGN
ADJUST SC TO 46D
SAVE FOR NEXT CONTINUED FRACTION
-P2 SC=23D
SET SIGN
ADJUST SC TO 46D
SAVE FOR NEXT CONTINUED FRACTION
-P1 SC=23D
SET SIGN
ADJUST SC TO 46D
ANSWER
SCALE ANSWER TO 24D BITS
<table>
<thead>
<tr>
<th>POOL</th>
<th>JP•L (NATLOG)</th>
<th>NORMAL EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL</td>
<td>26134,41377</td>
<td>LN(2) SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>0101009090</td>
<td>P0 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>0015463030</td>
<td>T1 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>7767361257</td>
<td>P3 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>0101507044</td>
<td>T2 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>7377477770</td>
<td>-P2 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>0511414431</td>
<td>T1 SC=23D</td>
</tr>
<tr>
<td>POOL</td>
<td>5662667151</td>
<td>-F1 SC=23D</td>
</tr>
</tbody>
</table>

**KITTY**

**RESERVE 6**

**BEST AVAILABLE COPY**
Fig. 1 - Flow Chart of "TENLOGTEN"
**Fig. 2 - Flow Diagram of CONLOGIT**

**CONLOGIT**
Input in Q register

- **N = 0?**
  - YES: SET \( \log_{10}(N) \) TO -50.0 → RETURN
  - NO: **N > 377777?**
    - YES: SCALE DOWN
    - NO: SET SCALE FACTOR TO 15

- **NATLOG**
  \( \ln(N) \)

- **ERROR?**
  - YES: SET \( \log_{10}(N) \) TO -100.0 → RETURN
  - NO: CONVERT \( \ln(N) \) TO \( 10/\log_{10}(N) \) AND ROUND

- SCALE TO 3 bits
- RETURN

**BEST AVAILABLE COPY**
**Fig. 3 - Flow Diagram of NALOG**

1. **INPUT IN A REGISTER**
   - **SAVE SCALE FACTOR, S**

2. **N = -N**
   - **N ≠ 0?**
     - **YES**
       - **NORMALIZE N AND SAVE COUNT (CT)**
       - **ERROR EXIT**
       - **RETURN ADDRESS NOT UPDATED**
     - **NO**
   - **YES**

3. **N = 0?**
   - **NO**
     - **SCALE N TO 23 BITS**
     - **CALCULATE m BASED ON CT AND S**
     - **B**
   - **YES**

4. **(m + ½) ln 2 SCALING 23 BITS**

5. **S WITHIN LIMITS?**
   - **YES**
     - **UPDATE RETURN ADDRESS TO PASS ERROR EXIT**
   - **NO**

6. **PARTIAL 1 = -P1/P2 + N + T1**
   - **SCALING 23 BITS**

7. **PARTIAL 2 = P2/PARTIAL 1 + N + T1**
   - **SCALING 27 BITS**

8. **ln(N) = -P1/PARTIAL 2 + P0 + (m + ½) ln 2**
   - **SCALE ln(N) TO 24 BITS**

9. **EXIT**
   - **ERROR HANDLED**